AD NUMBER

AD879465

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

**Distribution authorized to U.S. Gov't. agencies and their contractors;
Administrative/Operational Use; 13 OCT 1970.
Other requests shall be referred to Army Materiel Command, Alexandria, VA.**

AUTHORITY

USAMC ltr, 14 Jan 1972

AMC PAMPHLET **AMCP 706-329**

ENGINEERING DESIGN HANDBOOK

FIRE CONTROL SERIES

SECTION 3

72-19

FIRE CONTROL

COMPUTING SYSTEMS

HEADQUARTERS, U.S. ARMY MATERIEL COMMAND **OCTOBER 1970**

HEADQUARTERS
UNITED STATES ARMY MATERIEL COMMAND
WASHINGTON, D. C. 20315

AMC PAMPHLET
No. 706-329

13 October 1970

ENGINEERING DESIGN HANDBOOK

SECTION 3, FIRE CONTROL COMPUTING SYSTEMS

Paragraph Page

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

CHAPTER 7. ANALOG-DIGITAL CONVERSION TECHNIQUES

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS(cont)

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont)

LIST OF ILLUSTRATIONS (cont)

LIST OF ILLUSTRATIONS (cont)

LIST OF ILLUSTRATIONS (cont)

LIST OF ILLUSTRATIONS (cont)

LIST OF ILLUSTRATIONS (cont)

LIST OF TABLES

LIST OF EXAMPLES

LIST OF INFORMATION SUMMARIES

LIST OF DERIVATIONS

ACKNOWLEDGMENTS

The following figures and tables of this Handbook are based wholly or in part on copyrighted material and have been used with the permission of the publishers concerned.

Tables 1-1 and 1-2 Figs. 6-4, 6-6, and 6-7	From J. Eugene Alexander and J. Milton Bailey, SYSTEMS ENGINEERING MATHEMATICS, Copyright © 1962. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
Table 3-3	From John M. McCormick & Mario G. Salvadori, NUMERICAL METHODS IN FORTRAN, Copyright © 1964. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
Table 4-4	From SWITCHING CIRCUITS AND LOGICAL DESIGN by Samuel H. Caldwell, Copyright © 1958 by John Wiley & Sons, Inc. Used by permission of John Wiley and Sons, Inc.
Figs. 4-1, 4-2, and 4-3	From SYSTEM ENGINEERING by Harry H. Goode and Robert E. Machol, Copyright © 1957 by McGraw-Hill, Inc. Used by permission of McGraw-Hill Book Company.
Figs. 4-5, 4-6, 4-32, 4-33, 4-34, 4-42, 4-44, and 7-4	From Ivan Flores, COMPUTER LOGIC The Functional Design of Digital Computers, Copyright © 1960. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
Figs. 4-8, 4-36, 4-37, and 4-38	From LOGICAL DESIGN OF DIGITAL COMPUTERS by Montgomery Phister, Jr., Copyright © 1958 by John Wiley & Sons, Inc. Used by permission of John Wiley & Sons, Inc.
Figs. 4-9, 4-10, 4-11, 4-12, 4-13, 4-14, 4-15, 4-16, 4-17, 4-18, 4-27, 5-1, 5-3, 5-4, 5-5, and 7-6	From DIGITAL COMPUTER AND CONTROL ENGINEERING by Robert S. Ledley, Copyright © 1960 by McGraw-Hill, Inc. Used by permission of McGraw-Hill Book Company.
Figs. 4-23 and 5-8	From ARITHMETIC OPERATIONS IN DIGITAL COMPUTERS by R. K. Richards, Copyright © 1955, by Litton Educational Publishing, Inc. Used by permission of Van Nostrand Reinhold Company.
Fig. 4-39	From MATHEMATICS AND COMPUTERS by George R. Stibitz and Jules A. Larrivee, Copyright © 1957 by McGraw-Hill, Inc. Used by permission of McGraw-Hill Book Company.
Fig. 6-20	From an article by William W. Seifert titled "Differential Analyzer" in Vol. 4 of McGRAW-HILL ENCYCLOPEDIA OF SCIENCE AND TECHNOLOGY, Copyright © 1960 by McGraw-Hill, Inc. Used by permission of McGraw-Hill Book Company.
Figs. 6-24, 6-25, 6-26, 6-27, 6-39, 6-40, 6-41, 6-42, 6-43, 6-44, 6-59, 6-60, 6-61, and 6-62	From CONTROL SYSTEMS ENGINEERING, edited by William W. Seifert and Carl W. Steeg, Jr., Copyright © 1960 by McGraw-Hill, Inc. Used by permission of McGraw-Hill Book Company.
Figs. 6-28, 6-58, 6-63, 6-64, 6-65, 6-66, 6-67, 6-68, 6-69, 6-70, 6-71, 6-72, 6-74, 6-81, 6-82, 6-83, 6-84, 6-85, 6-88, and 6-89	From an article by William W. Seifert titled "Analog Computer" in Vol. 1 of McGRAW-HILL ENCYCLOPEDIA OF SCIENCE AND TECHNOLOGY, Copyright © 1960 by McGraw-Hill, Inc. Used by permission of McGraw-Hill Book Company.
Fig. 7-9	From Ivan Flores, "Digital, Analog, Hybrid Shaft-Angle Indicator with Frictionless 'Optical Gearing' ", Electromechanical Design, May 1958, Copyright @ 1958. Used by permission of Benwill Publishing Corp.
Fig. 7-12	From NOTES ON ANALOG-DIGITAL CONVERSION TECHNIQUES, edited by Alfred K. Susskind, by permission of the M.I.T. Press, Cambridge, Massachusetts. Copyright © 1957 by The Massachusetts Institute of Technology.

FOREWORD

INTRODUCTION

The Fire Control Series fornis part of the Engineering Design Handbook Series which presents engineering information antl quantitative data for the design and construction of Army equipment. In particular, the handbooks of the Fire Control Series have been prepared to aid the designers of Army fire control equipment and systems, and to serve as a reference guide for all military and civilian personnel who may be interested in the design aspects of such material.

The handbooks of the Fire Control Series are based on the fundamental parameters of the fire control problem antl its solution. In all problems of control over the accuracy of weapon fire, some method or system of firc control is employed that derives its intelligence from the acquisition and tracking of a target; evaluates this system-input intelligence by computation; and, finally, applies the output information to the positioning of a weapon along the line of fire. Primary emphasis is laid on the systematic approach required in the design of present-day fire control equipment and systems. This approach involves (1)thorough analysis of the particular fire control problem at hand, (2)establishment of the most suitable mathematical model, and (3) mechanization of tliis mathematical model.

ORGANIZATIONAL BREAKDOWN

To accomplish the aforenoted objectives, the Fire Control Series will'consist primarily of the following four main sections, each published as a separate handbook:

a. Section 1, Fire Control Systems - General (AMCP 706-327)
b. Section 2, Target Acquisition, Location and Tracking Systems (AMCP 706-328)
c. Section 3, Fire Control Computing Systems (AMCP 706-329)
d. Section 4, Weapon Pointing Systems (AMCP 706-330)

An additional handbook of the Fire Control Scries is AMC Pamphlet AMCP 706-331, Compensating Elements. The following paragraphs summarize the content of each of these five handbooks.

Scction 1 introduces the subject of fire control systems, discloses the basic fire control problem and its solution (in functional terms), delineates system-design philosophy, and discusses the application of maintcnance antl human engineering principles and standard design practices to fire control system design.

Section2 is devoted tothe first aspect of firc control, i.e., gathering intelligence on targct position and motion.

Scction 3, because of the complexity of thc subject of computing systems, is divided into three parts that are preceded by an introductory discussion of the roles of computing systems in Army fire control and by a dcscription of specific roles played in particular firc-control applications. Part I discusses the first step in system design, i.e., thc cstablishment of a mathematical model for thc solution of a fire control problem. Emphasis is given to the basis, derivation, and manipulation of mathematical models. Part II discusses the various computing devicc that perform useful functions in fire control computing systems. The discussion ranges from simple mechanical linkages to complex digital computers. Types of devices in each classification are briefly described; external sources are referenced for detailed information where practical. Part III discusses the various ways in which the computing devices described in Part II can be applied to thc mechanization of the mathematical modcls described in Part I. It stresses that a fire control computing system designer needs to apply his talents in three special ways: (1)to improvise and innovate as needed to mcct particular problems that may arise, (2) to use ingenuity in obtaining the simplest and most economical devices for the particular rcquirement at hand, and (3) to master thc many problems that result from intrasystem interactions when individually satis-

factory components are combined in complex computing systems. Examples culled from actual fire-control-system design work illustrate the concepts given.

Section 4 of the Fire Control Series discusses weapon-pointing systems with respect to (1) input intelligence and its derivation, (2) the means of implementing weapon-pointing for the two basic types of weapon-pointing systems from the standpoint of system stability, (3) general design considerations, and (4) the integration of components that form a complete fire control system.

AMCP 706-331 presents information on: (1) the effects of out-of-level conditions anti a displacement between a weapon and its aiming device, and (2) the instrumentation necessary to correct the resulting errors. It also presents general reference information on compensating elements that pertains to accuracy considerations, standard design practices; and considerations of general design, manufacture, field use, maintenance, and storage.

PREPARATION

The handbooks of the Fire Control Series have been prepared under the direction of the Engineering Handbook Office, Duke University, under contract to the Army Research Office-Durham. With the exception of the handbook titled Compensating Elements, the material for the Fire Control Series -- Sections 1 and 3 -- was prepared by the Jackson & Moreland Division of United Engineers and Constructors Inc., Boston, Massachusetts, under subcontract to the Engineering Handbook Office. The Jackson & Moreland Division was assisted in its work by consultants who are recognized authorities in various aspects of fire control. Specific authorship is indicated where appropriate. Overall technical guidance and assistance were rendered by Frankford Arsenal; coordination and direction of this effort were provided by Mr. Leon G. Pancoast of the Fire Control Development & Engineering Laboratories at Frankford Arsenal.

PREFACE

The Engineering Design Handbook Series of the Army Materiel Command is a coordinated series of handbooks containing basic information and fundamental data useful in the design and development of Army materiel and systems. The handbooks are authoritative reference books of practical information and quantitative facts helpful in the design and development of Army materiel *so* that it will meet the tactical and the technical needs of the Armed Forces.

The Handbooks are readily available to all elements of AMC, including personnel and contractors having a need and/or requirement. The Army Materiel Command policy is to release these Engineering Design Handbooks to other DOD activities and their contractors and to other Government agencies in accordance with current Army Regulation 70-31, dated 9 September 1966. Procedures for acquiring these Handbooks follow:

a. Activities within AMC and other DOD agencies order direct on an official form from:

Commanding Officer
Letterkenny Army Depot
ATTN: AMXLE-ATD
Chambersburg, Pennsylvania 17201

b. Contractors who have Department of Defense contracts should submit their requests through their contracting officer with proper justification to the address listed in par. a.

c. Government agencies other than DOD having need for the Handbooks may submit their request directly to the address listed in par. a or to:

Commanding General
U. S. Army Materiel Command
ATTN: AMCAM-ABS
Washington, D. C. 20315

d. Industries not having Government contracts (this includes colleges and universities) must forward their requests to:

Commanding General
U. S. Army Materiel Command
ATTN: AMCRD-TV
Washington, D. C. 20315

e. All foreign requests must be submitted through the Washington, D. C. Embassy to:

Assistant Chief of Staff for
Intelligence
ATTN: Foreign Liaison Office
Department of the Army
Washington, D. C. 20310

All requests, other than those originating within DOD, must be accompanied by a valid justification.

Comments and suggestions on this handbook are welcome and should be addressed to Army Research Office-Durham, Box CM, Duke Station, Durham, North Carolina 27706.

INTRODUCTION*

As pointed out in Section 1† of the Fire Control Series, computers play a very significant role during the design phase for a fire control system, and a computer is an integral part of every complete modern fire control system. The function of the computer in a fire control system can be illustrated by considering for a moment the case of an individual attempting to hit a moving target with a rifle. If he is to be successful, he must estimate tlie distance to the target and the rate at which the line-of-sight to the target is rotating and must have a knowledge of the projectile characteristics, such as velocity and gravity drop. He must then compute the direction in which to point the weapon to achieve a hit, and so point the weapon. If a strong wind is blowing, he must also take this into account for long-range shots. Obviously, if the individual attempted to carry out detailed conscious calculations, his target would have disappeared before he was ready to pull the trigger. The expert marksman has, through considerable experience, learned to include each of these factors in a rapid mental appraisal of the situation at hand. As the target velocity is increased and the range extended, however, the ability of the individual to apply the required correction factors is exceeded and successful shots can be achieved only if rapid, accurate assistance is provided for gathering the required data, carrying out the necessary computations, and pointing the weapon as required. In the provision of this assistance, modern fire control systems have evolved (see Chapter 1 of Section 1 of the Fire Control Series). In each of these systems, the computer serves as a vital element.

Until approximately 1950 to 1955, analog computers were used almost exclusively in fire control systems because the digital-computer art had not yet progressed to the stage where tlie required operating speeds could be achieved. Now, the demands of many fire control problems can be met by either an analog or a digital computer, with the choice frequently based upon such considerations as the desire to use the same computer design in several different systems or tlie background of the particular group of designers responsible for the fire control system. (Such basic factors as cost, size, weight, power requirements, complexity, reliability, solution speed, solution accuracy, and the nature of environmental effects must, of course, always continue to receive careful attention in relationship to the particular circumstances under which a given computer is destined for use.) Worthy of special note is the recognition during recent years of the promising potential for fire-control-system applications of the digital differential analyzer -- an incremental computer consisting of a collection of digital integrators interconnected in such a way as to solve integro-differential equations.

In addition to the use of computers in the design phase of a fire control system and as an integral part of every complete modern fire control system, computers have come to serve mankind increasingly in everyday technology. As a matter of fact, the development of high-speed electronic digital-computing equipment has created a revolution in technology. Because of the pioneer role played by the U.S. Army in the development of high-speed electronic digital computers, it is particularly appropriate to briefly discuss this development here.

Army activity in this field started after the outbreak of World War II, when the need for rapid computational equipment for use in connection with the massive computing problems involved in the preparation of firing tables and related ballistic data became increasingly apparent. At that time, some of the computations were being made by the Bush

---

* Prepared by W. W. Seifert, this Introduction incorporates information from various U.S. Army documents--in particular, "Historical Monograph, Electronic Computers Within the Ordnance Corps", by Karl Kempf, Historical Officer, Aberdeen Proving Ground, Maryland; published by *APG* in November 1961.

† Fire Control Systems—General (AMCP 706-327).

Differential Analyzer.* With considerable improvements in performance resulting from design modifications provided during the early 1940's by the Moore School of Electrical Engineering at the University of Pennsylvania, this machine proved to be of tremendous value during World War II. Used primarily to compute trajectories for firing tables and to prepare trajectory charts for use with VT fuzes, this machine could compute a 60-second trajectory in about 15 minutes. In contrast, a human operator using a desk calculator required about 20 hours to perform the same computation.

As a result of the urgent need for some means to provide accurate computation at considerably higher speeds than those obtainable with the Bush Differential Analyzer, much thought went into the solution of this problem. It became apparent at the University of Pennsylvania that use could be made of the fast reaction time of electron tubes in an extensive array to add or subtract impulses, and thus make possible the design of a machine that would deal with numbers in a manner that would far surpass the speed and accuracy of the Bush machine. Accordingly, in 1943 the U.S. Army awarded a research and development contract to the University of Pennsylvania for the design and construction of ENIAC (for Electronic Numerical integrator And Computer). This contract was based specifically on technical concepts underlying the design of an electronic computer that were contained in an outline prepared by Dr. John Mauchly and Dr. J. Presper Eckert, Jr. of the Moore School of Electrical Engineering.

Completed in 1945, ENIAC was the world's first electronic automatic computer.† Its subsequent installation in the Ballistic Research Laboratories (BRL) at Aberdeen Proving Ground marked the beginning of the widespread use of electronic computing machines. ENIAC was a decimal machine in which ten decade ring counters -- one per decimal place -- and one P M (plus or minus) counter formed the basic arithmetic and storage unit. It utilized 19,000 vacuum tubes (of 16 different types), 1500 relays, and hundreds of thousands of resistors, capacitors, and inductors. It consumed nearly 200 kilowatts of power. Its thirty separate units weighed more than 30 tons. This huge collection of circuits could calculate a 60-second trajectory in less than the actual time of flight of the projectile from the gun to the target.

Even before the development of ENIAC had been completed, however, it was realized that a serial binary machine with delay-line storage (an early type of memory device) would have additional advantages. A binary machine would utilize numbers to the base two instead of the traditional base ten. Numbers would be translated into a series of ONES and ZEROS, values that could be easily handled by electron tubes arranged either to conduct a signal or block it -- a switching function that could be handled at high speed. Nonetheless, ENIAC remained a solid computational workhorse for the ten-year period of 1946-55, during which it was in constant operation. It was the major instrument for computation for all ballistic tables for the U.S. Army and the U.S. Air Force -- dominating the computer field during the period 1949-52. It was also used for calculations relevant to other fields -- weather prediction, atomic energy, cosmic-ray studies, thermal ignition, random-number studies, and wind-tunnel design problems, to mention a few. (Electronic computers were not yet available from commercial sources.)

ENIAC was the prototype from which most other modern computers have evolved (see the computer tree of Fig. I-1). It embodied almost all of the components and concepts of later high-speed storage and control devices. Although built primarily for integration of the equations of external ballistics by a step-by-step process, it was sufficiently

---

* This was an electromechanical analog device utilizing mechanical integrators of the wheel-and-disc type that was developed by Dr. Vannevar Bush and his associates at Massachusetts Institute of Technology in the late 1920's. Incorporating improvements made in the early 1930's, a Bush Differential Analyzer was installed at Aberdeen Proving Ground in 1935.

† It should be noted that the Mark I Relay Computer (also called the Automatic Sequence-Controlled Calculator), completed in 1944 at Harvard University by Howard Aiken in cooperation with IBM engineers and Harvard graduate students, was the first automatic computer ever completed. The operation of this machine was based on electromechanical principles. Although the machine was efficient, fast, and capable of solving a wide variety of problems, its speed could not approach that of the electronic type of automatic computer.

flexible to be applied to a wide range of large-scale computations other than numerical integration of differential equations.

The urgent need for an operational computer had made it imperative to freeze the engineering design of ENLAC during the early stages of development. As work on ENIAC permitted, however, the design and construction of an improved computer for RRL having much smaller size, greater flexibility, and better mathematical performance were pushed forward under U.S. Army sponsorship at the Moore School of Electrical Engineering, University of Pennsylvania. The design for this computer, named EDVAC (for Electronic Discrete Variable Automatic Calculator), was proposed in 1945 by Dr. John von Neumann, one of the world's leading mathematicians, who had been attracted by the problems of computer design. The major features of this computer were the use of the binary system rather than the decimal system of numeration, a serial arithmetic mode, a four-address command structure, a total of 16 possible operations that could be performed by the computer, and duplicate circuitry for check purposes.

EDVAC was also the first computer with an internally-stored program and was thus a major improvement over ENLAC, which required considerable human effort to change the different programs. With ENIAC, the different sections of the computer were connected together via plug-in cables that had to be changed for each particular type of problem. If the computations had to be interrupted for a few days, to permit some other problem of high priority to be run on the computer, the complex tangle of plug-in cables had to be rearranged manually. Also, when the run was completed, the machine had to be "re-wired" for the first problem. With an internally-stored program device, the instructions are stored, each storage location is queried, and each instruction is interpreted and executed as a matter of formality until all the instructions comprising a given program are carried out.

Mork on EDVAC stimulated design and construction by other groups of a large family of similar computers, including SEAC, FLAC, DYSEAC, MIDAC,* and the later commercial types, such as the UNIVAC's (see Fig. I-1). Computer development was further encouraged by the Army via a research contract with the Institute for Advanced Study, Princeton, New Jersey (later supported also by the Air Force and Navy).

From this support of computer research came the ORDVAC (for Ordnance Variable Computer), the BRL's third electronic computing machine. This was a parallel binary computer that belongs to the group of computers whose basic logic was developed by the Institute for Advanced Study at Princeton, New Jersey. The ORDVAC family of computers includes such machines as the AVIDAC, MANIAC, ILLIAC, ORACLE, JOHNNIAC, and CYCLONE.†

These different designs constituted little if anything new in innate computer design, but carried out existing design principles using the fruits of the ever-advancing technology of electronics -- such things as improved memory techniques, smaller vacuum tubes, improved diodes, and the like. During the early 1950's, a major part of the scientific computational workload of the Western world was accomplished on these machines.

The rapid, competitive evolution of computers made it apparent at an early stage that prospective users and designers of computers in industry and in government would benefit from a comprehensive survey of designs in being. BRL accordingly made a nation-wide survey in 1955. This showed that at that time approximately 87 different types of commercial and scientific digital computers were operational in this country. A second

---

* SEAC - Standards Eastern Automatic Computer
FLAC - Florida Automatic Computer
DYSEAC - Second SEAC
MIDAC - Michigan Digital Automatic Computer

† AVIDAC - Argonne Version of the Institute's Digital Automatic Computer
MANIAC - Mathematical Analyzer Numerical Integrator and Computer
ILLIAC - Illinois Automatic Computer
ORACLE - Oak Ridge Automatic Computer and Logical Engine
JOHNNIAC - John (von Neumann) Integrator and Automatic Computer
CYCLONE - (an arbitrary name indicating high speed) Iowa State University

survey by BRL, macle in 1957, showed that this total had risen to 103. A third survey in 1961 indicated the existence of over 222 different types of electronic digital computing systems, involving tens of thousands of units throughout the United States. Fig. I-1 indicates there are approximately 500 different types in operation today.

These computers are committed to the solution of almost every conceivable type of computing and data-processing problem -- in defense, industry, science, commerce, service operation, and manufacturing. A vital element in almost every defense system, the computer has become even more significant in industry and commerce.

The overall discussion of electronic digital computers given thus far has covered the historical development of serial computers (represented by EDVAC) and of parallel computers (represented by ORDVAC). Both of these computers are shown in Fig. I-1 at the lower ends of two separate limbs of a computer tree whose trunk represents the development of ENLAC. As noted in Fig. I-1, this separation tends to distinguish the business computers on the left limb from the scientific computers on the right limb.

The electronic digital computers that have been developed specifically to meet military needs are identified on the center limb of the computer tree. Among those indicated is FADAC (for Field Artillery Digital Automatic Computer). This computer was developed under the direction of Frankford Arsenal in the late 1950's as a sequel to Field Artillery Fire Control System, M35, which employed an electromechanical computer whose accuracy was adequate for the shorter-range weapons -- such as the 105 mm and 155 mm howitzers -- but was not adequate for guns and free rockets. FADAC represents the latest development in connection with the ever-present need to solve field-artillery fire control problems with greater accuracy and speed.

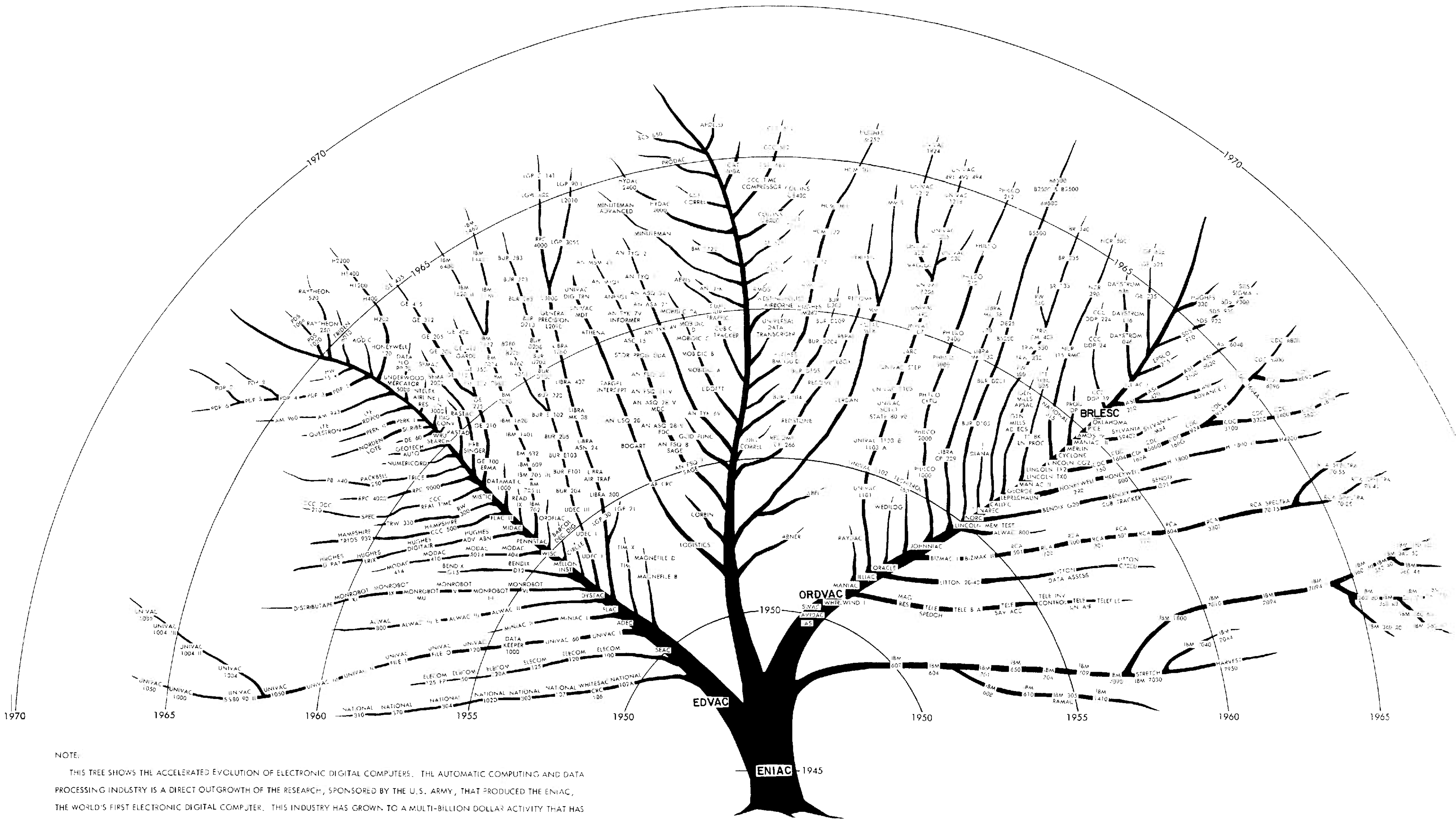
FADAC is a solid-state electronic digital computer whose background is discussed in Chapter 1 of Section 1 of the Fire Control Series and whose technical aspects are discussed in Chapter 4 of the present section. Its overall capabilities, however, merit summation here:

1. FADAC can provide firing data for a battery of weapons. On a one-battery-at-a-time basis, it can provide firing data for mortars, howitzers, guns, and free rockets -- with complete applicability to any kind of ammunition these weapons may be using. In emergencies, it can provide data for up to five similar-type batteries on a rotating basis. By using the FADAC's memory loading unit, authorized field personnel can make program changes that permit switching from the solution of one type of fire control problem to another within just a few minutes.

2. FADAC could be used with the PERSHING, SERGEANT, LACROSSE, and NIKE-HERCULES weapon systems.

3. In addition to use in fire control systems and missile systems, F'ADAC can also be employed in fire planning, survey computations, counter-battery computation, reduction of meteorological data, and as universal automatic check-out equipment

A universal computer capable of solving all field-artillery fire control problems has always seemed to lie in the future. However, continuous study at Frankford Arsenal on increasing the application of FADAC has yielded results that make this computer a candidate for the title "Universal Artillery Computer".



NOTE:

THIS TREE SHOWS THE ACCELERATED EVOLUTION OF ELECTRONIC DIGITAL COMPUTERS. THE AUTOMATIC COMPUTING AND DATA PROCESSING INDUSTRY IS A DIRECT OUTGROWTH OF THE RESEARCH, SPONSORED BY THE U.S. ARMY, THAT PRODUCED THE ENIAC, THE WORLD'S FIRST ELECTRONIC DIGITAL COMPUTER. THIS INDUSTRY HAS GROWN TO A MULTI-BILLION DOLLAR ACTIVITY THAT HAS PENETRATED EVERY PROFESSION AND TRADE IN GOVERNMENT, BUSINESS, INDUSTRY, AND EDUCATION. THE TRUNK RESTS ON THE ENIAC. THE SERIAL COMPUTERS, REPRESENTED BY THE EDVAC, AND THE PARALLEL COMPUTERS, REPRESENTED BY THE ORDVAC, ARE SHOWN AS SEPARATE LIMBS. THIS SEPARATION TENDS TO DISTINGUISH THE BUSINESS COMPUTERS ON THE LEFT LIMB FROM THE SCIENTIFIC COMPUTERS ON THE RIGHT LIMB. THE COMPUTERS THAT WERE DEVELOPED SPECIFICALLY TO MEET MILITARY NEEDS ARE SHOWN ON THE CENTER LIMB. MANUFACTURERS HAVE ENTERED THE ELECTRONIC COMPUTER FIELD AT DIFFERENT TIMES, AS SHOWN BY THE VARIOUS BRANCHES. ONLY UNIVERSITY AND GOVERNMENT SPONSORED COMPUTERS ARE SHOWN ALONG THE LIMBS. THE RADIAL DISTANCE FROM THE ENIAC IS AN APPROXIMATE INDICATION OF THE YEAR EACH COMPUTER WAS EITHER DEVELOPED, CONSTRUCTED, OR PLACED IN OPERATION.

Prepared by Department of the Army

Figure I-1. The computer tree for electronic digital computers.

PART I
MATHEMATICAL MODELS FOR FIRE CONTROL COMPUTING SYSTEMS

CHAPTER 1*
THE ROLE OF THE MATHEMATICAL, MODEL IN THE DESIGN PROCESS

1-1 DEFINITION AND IMPORTANCE OF A MATHEMATICAL MODEL

In Section 1† of the Fire Control Series, a mathematical model is defined as any scheme for the manipulation of ideas in a group wherein the individual ideas are identified by means of more or less abstract symbols and wherein manipulations are conducted in accordance with precise rules of logic. Mathematical models take on a variety of forms, depending upon the particular system they are being used to study. Such models provide the system designer with a powerful tool that enables him to develop a system not merely by intuition and trial and error with the physical system but by bringing to bear on his problem a considerable body of mathematical techniques, and thereby raises his design process from an art to a science.

The first requirement and advantage that the system designer faces in using mathematical models is that of deriving an accurate model for the physical system being considered. If the designer is to carry out this step in a satisfactory manner, he must understand the system and the interrelationships between its parts in considerably more detail than he might otherwise be forced to employ. Formulation of the model is thus of value in itself, but usually is taken as the first step in a mathematical study aimed at optimizing certain parameters in the system. This optimization may be carried out using purely analytical techniques, graphical techniques, or by studying the model on either an analog or a digital computer. Chapter 2 outlines a number of these techniques. As background for this discussion, par. 1-2 summarizes some of the more important mathematical expressions used for describing important natural laws that relate to physical systems, and par. 1-3 summarizes the characteristics and limitations of mathematical models.

1-2 MATHEMATICAL MODELS FOR PHYSICAL SYSTEMS

If one is to establish a mathematical model or description for a physical system, he must be able to express causes and effects in mathematical terms for each individual element of the system and be able to describe mathematically the manner in which these elements interact. Depending on the purpose of the specific analysis, the individual elements may be single components -- such as resistors, capacitors, and vacuum tubes -- or complete amplifiers or even a complete radar set. Instead of electric-circuit elements, the system may be composed of mechanical components -- such as springs, dampers and inertial elements -- or of fluid elements -- such as valves, orifices, and fluid pumps and motors. Some systems likewise contain magnetic, acoustic, or thermal elements. Frequently, a complex system includes a mixture of elements of several of these types.

Fortunately, the modern analyst is able to draw on the work of Newton: Kirchoff, d'Alembert, Coulomb, and many others who were able to formulate mathematical relationships to express their experimental observations on particular physical systems.

* By W. W. Seifert.

† Fire Control Systems--General (AMCP *706-327*).

Basic requirements for the analyst who desires to formulate a mathematical description for a system are (1) that he understand thoroughly the laws relating to the types of elements from which his system is composed and (2) that he understand the range of variables for which the elements of his physical system behave as ideal elements by obeying the ideal laws, and the manner in which their performance departs from the ideal outside this range. It is impossible in a single chapter to outline all the relationships that an analyst would require in analyzing the various systems with which he might be confronted. However, a brief discussion of several illustrative mathematical descriptions for physical systems is provided, and a number of other relationships are tabulated.

In order to develop and utilize mathematical descriptions for physical systems, it is first necessary to define the symbols that are to be used in writing these descriptions. Although agreement on symbols is far from unanimous, the discussion which follows uses symbols that have received wide usage.

As an illustration of a basic mathematical description of a physical phenomenon, consider one of the fundamental laws of electrostatics. Out of some of the earliest work on static electricity grew the concept of electric charge, which gradually has come to be represented symbolically by the letter q. Early experimenters found that if two point charges of electricity of opposite kind are in the neighborhood of each other, they exert attractive forces on each other. If they are of the same kind, however, they exert repulsive forces on each other. Furthermore, the force that one exerts on the other is determined by the distance between the charges and the magnitude of the charges. The work of Cavendish and Coulomb in the late 1780's established the inverse-square law of electrostatic force, which states that the force between two point charges of electricity is directly proportional to the product of the charges and inversely proportional to the square of the distance between them. Mathematically, this statement, which has come to be called Coulomb's law, takes the form

$$F = K \frac{q_A q_B}{r^2} \qquad (1-1)$$

where F represents the force between the two point charges, q_A and q_B represent the two charges, r represents the distance by which the charges are separated, and K is the proportionality constant. This constant depends upon the units used to measure the force, the distance, and the charges and also upon the medium in which the experiment is conducted. The force found when this experiment is performed in a high-quality insulating oil differs from that found when the experiment is performed in air. For such an experiment, the pertinent parameter of the medium is its dielectric constant k. In terms of this constant, Eq. 1-1 can be rewritten in the form

$$F = K_1 \frac{q_A q_B}{k r^2} \qquad (1-2)$$

where K_1 depends only on the units in which the quantities are measured.

As man's understanding of electricity grew, he discovered ways to produce steady flows of current I which he then associated with the rate at which charge was moving through a system, i.e.,

$$I = \frac{dq}{dt} \qquad (1-3)$$

He also discovered that when a battery (voltaic cell) was connected in a closed circuit the current that flowed was determined by the voltage E of the cell and a property of the circuit determined by the length, cross-sectional area, and composition of the conductors. This property of the circuit came to be known as its resistance R and Ohm deduced the following relationship which now bears his name:

$$IR = E \qquad (1-4)$$

Beginning with Oersted's discovery in 1820 that a magnetic needle tends to set itself at right angles to a wire through which an electric current is flowing, Faraday and others began to experiment with, and attempt to discover the laws that govern, phenomena of

electromagnetic induction. Their efforts led to the definition of such new quantities as inductance L and to new laws such as

$$e = L\frac{di}{dt} \qquad (1-5)$$

which relates the instantaneous voltage e across an inductance to the rate at which the instantaneous current i through the inductance is changing.

As knowledge of the behavior of electrical systems grew, so did the knowledge of other types of systems, such as mechanical, hydraulic, and thermal systems. Furthermore, certain similarities were found to exist between the ways in which entirely different types of systems performed. For example, the flow of current through a conductor was likened to the flow of water through a pipe. In each case, it was observed that the flow increased as the forcing function (voltage or pressure) increased.

Table 1-1 lists the principal elements and parameters used to describe physical systems, and gives symbols and units that are comnionly used in describing these systems. it should be understood, of course, that other systems of units also find wide usage. In particular, the MKS (meter, kilogram, second) system of units is rapidly becoming the standard for all educational systems and governments. Accordingly, pertinent information concerning physical constants and conversion factors in terms of the MKS system is presented in the appendix to this chapter.

Table 1-2 further develops the similarity between different physical systems by summarizing the expressions for power dissipation and energy storage, and giving the differential equation that describes a simple system containingone of each of the types of elements belonging to a particular family. It should be noted that two rows of entries appear for each system and that the associated differential equations are of the same form. The reason for this similarity can be illustrated by examination of the two equivalent electrical networks shown in Fig. 1-1. The top network represents a parallel combination of a conductance (reciprocal resistance), an inductance, arid a capacitance driven by a current generator. The lower network represents a series combination of these same elements (with resistance shown in place of conductance) driven by a voltage generator. In the first case, it is desired to set up an expression for the instantaneous voltage e(t) across the network, while in the second the instantaneous current i(t) flowing in the network is desired.

For the first case, the differential equation from which e(t) can be computed is found by summing the currents through the three elements, i.e.,

$$i(t) = i_C(t) + i_G(t) + i_L(t) \qquad (1-6)$$

Substitution of expressions for these element currents in terms of voltage shows that

$$i(t) = C\frac{de}{dt} + Ge + \frac{1}{L}\int e\,dt \qquad (1-7)$$

For the second case, the differential equation is formed by equating the applied voltage to the sum of the voltages across the individual elements, i.e.,

$$e(t) = e_L(t) + e_R(t) + e_C(t) \qquad (1-8)$$

When these element voltages are expressed in terms of the loop current i(t), the resultant equation becomes

$$e(t) = L\frac{di}{dt} + Ri + \frac{1}{C}\int i\,dt \qquad (1-9)$$

Comparison of Eqs. 1-7 and 1-9 shows that one could be derived from the other if the following substitutions were made:

$$i \rightarrow e$$

$$C \rightarrow L$$

$$G \rightarrow R$$

$$\frac{1}{L} \rightarrow \frac{1}{C}$$

TABLE 1-1. SYMBOLS AND UNITS.

System	Parameter or element	Symbol	Unit	Pictorial symbol
1. *Electrical*	Voltage	e	volt	
	Current	i	ampere	R
	Charge	q	coulomb	
	Power	power	watt	
	Angular velocity	ω	radians/second	L
	Energy	W	joule	
	Resistance	R	ohm	
	Conductance	G	mho	C
	Inductance	L	henry	
	Capacitance	C	farad	
2. *Mechanical rectilineal*	Force	f	pounds	
	Velocity	v	feet/second	
	Displacement	x	feet	R
	Acceleration	a	feet/second^2	
	Acceleration of gravity	g	32.2 feet/second^2	
	Power	power	foot-pound/second	m
	Energy	W	foot-pound	
	Viscous friction	R	pound-second/foot	k
	Mass	m	pound-second^2/foot	
	Spring constant	k	pound/foot	
3. *Mechanical rotational*	Torque	T	pounds-feet	
	Angular velocity	ω, $\dot{\theta}$	radians/second	B
	Angular displacement	θ	radians	
	Power	power	foot-pound/second	J
	Energy	W	foot-pound	
	Rotational friction	B	pound-foot-second	
	Inertia	J	pound-foot-second^2	c
	Rotational spring constant	c	pound-foot/radian	
4. *Hydraulic*	Pressure	p	pound/foot^2	
	Flow rate	q	foot^3/second	
	Volume	V	foot3	R R
	Power	power	foot-pound/second	
	Energy	W	foot-pound	
	Resistance	R	pound-second/foot^5	M
	Inertance	M	pound-second^2/foot^5	C
	Capacitance	C	foot^5/pound	
	Bulk modulus	B	pound/foot^2	
	Density	ρ	pound/foot^3	
5. *Pneumatic*	Pressure	p	pound/foot^2	
	Flow rate	q	foot^3/second	R R
	Volume	V	foot^3	
	Power	power	foot-pound/second	M
	Energy	W	foot-pound	
	Resistance	R	pound-second/foot^5	C
	Inertance	M	pound-second^2/foot^5	
	capacitance	C	foot^5/pound	
6. *Thermodynamic*	Temperature	θ	°F	R
	Heat flow	q	BTU/second	
	Heat	H	BTU	C
	Resistance	R	degree-second/BTU	
	Capacitance	C	BTU/degree	

TABLE 1-2. SUMMARY OF ANALOGIES.

System	Parameters: Forcing function	Parameters: Response function	Parameters: Alternate response	Elements: Dissipative	Elements: Energy Storage 1	Elements: Energy Storage 2	Relations between Parameters and Elements: Dissipative	Response Function: Energy Storage 1	Response Function: Energy Storage 2	Alternate Response: Dissipative	Alternate Response: Energy Storage 1	Alternate Response: Energy Storage 2	Equation of single-degree-of-freedom system containing all types of elements	Power dissipation	Energy Storage 1	Energy Storage 2	System
Electrical	voltage e	current i	charge q	resistance R	inductance L	capacitance C	$e = Ri$	$e = L\frac{di}{dt}$	$e = \frac{1}{C}\int i\,dt$	$e = R\frac{dq}{dt}$	$e = L\frac{d^2q}{dt^2}$	$e = \frac{1}{C}q$	$L\frac{di}{dt} + Ri + \frac{1}{C}\int i\,dt = e$	power $= i^2R$	$W = \frac{1}{2}Li^2$	$W = \frac{1}{2}Ce^2$	Electrical
	current i	voltage e		conductance G	capacitance C	inductance L	$i = Ge$	$i = C\frac{de}{dt}$	$i = \frac{1}{L}\int e\,dt$				$C\frac{de}{dt} + Ge + \frac{1}{L}\int e\,dt = i$	power $= e^2G$	$W = \frac{1}{2}Ce^2$	$W = \frac{1}{2}Li^2$	
Mechanical Rectilineal	force f	velocity v	displacement x	rectilineal resistance R	mass m	reciprocal of spring constant $1/k$	$f = Rv$	$f = m\frac{dv}{dt}$	$f = k\int v\,dt$	$f = R\frac{dx}{dt}$	$f = m\frac{d^2x}{dt^2}$	$f = kx$	$m\frac{dv}{dt} + Rv + k\int v\,dt = f$	power $= v^2R$	$W = \frac{1}{2}mv^2$	$W = \frac{1}{2}\frac{1}{k}f^2$	Mechanical Rectilineal
	velocity v	force f		reciprocal of rectilineal resistance $1/R$	reciprocal of spring constant $1/k$	mass m	$v = \frac{f}{R}$	$v = \frac{1}{k}\frac{df}{dt}$	$v = \frac{1}{m}\int f\,dt$				$\frac{1}{k}\frac{df}{dt} + \frac{f}{R} + \frac{1}{m}\int f\,dt = v$	power $= \frac{f^2}{R}$	$W = \frac{1}{2}\frac{1}{k}f^2$	$W = \frac{1}{2}mv^2$	
Mechanical Rotational	torque T	angular velocity ω	angular displacement θ	rotational resistance B	moment of inertia J	reciprocal of rotational spring constant $1/c$	$T = B\omega$	$T = J\frac{d\omega}{dt}$	$T = c\int \omega\,dt$	$T = B\frac{d\theta}{dt}$	$T = J\frac{d^2\theta}{dt^2}$	$T = c\theta$	$J\frac{d\omega}{dt} + B\omega + c\int \omega\,dt = T$	power $= \omega^2B$	$W = \frac{1}{2}J\omega^2$	$W = \frac{1}{2}\frac{1}{c}T^2$	Mechanical Rotational
	angular velocity w	torque T		reciprocal of rotational resistance $1/B$	reciprocal of rotational spring constant $1/c$	moment of inertia J	$\omega = \frac{T}{B}$	$\omega = \frac{1}{c}\frac{dT}{dt}$	$\omega = \frac{1}{J}\int T\,dt$				$\frac{1}{c}\frac{dT}{dt} + \frac{T}{B} + \frac{1}{J}\int T\,dt = \omega$	power $= \frac{T^2}{R}$	$W = \frac{1}{2}\frac{1}{c}T^2$	$W = \frac{1}{2}J\omega^2$	
Hydraulic	pressure p	volumetric flow q	volume V	hydraulic resistance R	inertance M	hydraulic capacitance C	$p = Rq$	$p = M\frac{dq}{dt}$	$p = \frac{1}{C}\int q\,dt$	$p = R\frac{dV}{dt}$	$p = M\frac{d^2V}{dt^2}$	V	$M\frac{dq}{dt} + Rq + \frac{1}{C}\int q\,dt = p$	power $= q^2R$	$W = \frac{1}{2}Mq^2$	$W = \frac{1}{2}Cp^2$	Hydraulic
	volumetric flow q	pressure p		reciprocal of hydraulic resistance $1/R$	hydraulic capacitance C	inertance M	$q = \frac{p}{R}$	$q = C\frac{dp}{dt}$	$q = \frac{1}{M}\int p\,dt$				$C\frac{dp}{dt} + \frac{p}{R} + \frac{1}{M}\int p\,dt = q$	power $= \frac{p^2}{R}$	$W = \frac{1}{2}Cp^2$	$W = \frac{1}{2}Mq^2$	
Pneumatic	pressure p	volumetric flow q	volume V	pneumatic resistance R	inertance M	pneumatic capacitance C	$p = Rq$	$p = M\frac{dq}{dt}$	$p = \frac{1}{C}\int q\,dt$	$p = R\frac{dV}{dt}$	$p = M\frac{d^2V}{dt^2}$	$p = \frac{V}{C}$	$M\frac{dq}{dt} + Rq + \frac{1}{C}\int q\,dt = p$	power $= q^2R$	$W = \frac{1}{2}Mq^2$	$W = \frac{1}{2}Cp^2$	Pneumatic
	volumetric flow q	pressure p		reciprocal of resistance $1/R$	pneumatic capacitance	inertance M	$q = \frac{p}{R}$	$q = C\frac{dp}{dt}$	$q = \frac{1}{M}\int p\,dt$				$C\frac{dp}{dt} + \frac{p}{R} + \frac{1}{M}\int p\,dt = q$	power $= \frac{p^2}{R}$	$W = \frac{1}{2}Cp^2$	$W = \frac{1}{2}Mq^2$	
Thermodynamic	temperature θ	heat flow q	heat H	thermal resistance R		thermal capacitance	$e = Rq$		$\theta = \frac{1}{C}\int q\,dt$	$\theta = R\frac{dH}{dt}$		$\theta = \frac{H}{C}$	$Rq + \frac{1}{C}\int q\,dt = \theta$				Thermodynamic
	heat flow q	temperature θ		thermal conductance G	thermal capacitance C		$q = Ge$	$q = C\frac{d\theta}{dt}$					$C\frac{d\theta}{dt} + G\theta = q$				

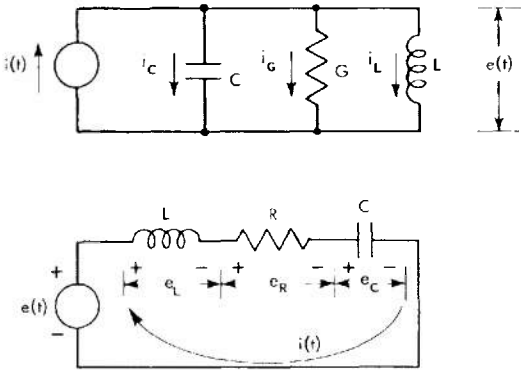


Figure 1-1. Equivalent, or dual, electrical networks.

The reciprocal quantities $\frac{1}{L}$ and $\frac{1}{C}$ are sometimes designated Γ and S ~respectively, whereupon the last correspondence above becomes

$$\Gamma \rightarrow S$$

When the differential equations describing two networks composed of the same class of physical elements (such as electrical or mechanical) correspond in this manner, the networks are called duals. For a large class of networks, such duals exist and frequently represent alternative means for realizing a given type of dynamic system performance.†

Techniques for formulating the integro-differential equations for electrical net?works[2,3]†† and for more general systems have now been developed to a high degree. A set of these equations sufficient to describe a given system that is under consideration represents a mathematical model for the system, and the development of such a model constitutes the first step toward determination of the performance characteristics of this system. The fact that a variety of different types of physical systems can be described by equations of the same form facilitates considerably the study of a variety of systems.

The equations shown in Table 1-2 describe the restricted but very important class of linear systems. While any physical system can be driven into regions of nonlinear operation, many systems do behave in an essentially linear fashion over a wide useful operating range. The reason why systems that operate in an essentially linear manner are so important is that the mathematical techniques for analyzing such systems are highly developed and relatively easy to apply. Consequently, although the analyst should always keep before him a clear picture of the ways in which the system he is studying departs from linearity, he should, as a first step in his analysis, determine whether or not useful results could be obtained from study of a linearized representation of the system. If, under normal use, the system operates in an essentially linear fashion, very useful preliminary estimates of system characteristics can be obtained at much less effort than if the nonlinearities were included. At a later stage in the analysis, it may be desirable to include nonlinear terms in the mathematical model, but their inclusion substantially increases the difficulty of obtaining analytic solutions and may force the analyst to resort to computer methods of solution.** While a computer solution can frequently serve in such a circumstance to provide a more faithful representation of a system than might otherwise be obtainable, a good general rule to observe is the following: If one can obtain a satisfactory solution without the use of a computer, he should do so since he will then be likely to better understand the problem.

1-3 CHARACTERISTICS AND LIMITATIONS OF MATHEMATICAL MODELS

A mathematical model is merely a convenient way in which to describe a physical system. If such a model is to be useful, it must (1) represent the physical system sufficiently well that solutions obtained by studying the model yield useful information about

---

* It should be noted that the symbol S used here has no relationship to the symbol s used to represent the Laplace transform variable.

† For a considerably more extensive treatment of this subject, the reader is referred *to* Chapter 3 of Reference 1.

** A general bibliography of references relating to the analysis of nonlinear systems appears at the end of Chapter 2.

†† Superscript numbers refer to References at the end of each chapter.

the performance of the actual system and (2) be amenable to analysis. Actually, neither of these requirements is absolute. A crude mathematical model may be easy to study and may provide very useful information, while a more sophisticated model might yield considerably more accurate results but might be very difficult to study. While misleading results may be obtained if the model used does not take into account all the significant system characteristics, there is little point in employing a model that is more complex than is required to obtain results that are of sufficient accuracy for the particular purpose at hand. The experienced analyst employs simple models during the early stages of his investigation as a means for examining a broad range of possible systems and establishing preliminary bounds on system parameters. As the design proceeds, the model may be elaborated upon so as to represent the system more accurately. Furthermore, in the latter stages of analysis, it may be desirable to determine how the system performs when subjected to inputs and disturbances that can be described only in a statistical manner or when certain system parameters deviate in some randomly described fashion from the design values. While such effects can be included in the mathematical model, the resulting equations frequently become so complex as to preclude analytic solution and require simulation on an analog or a digital computer.

With more complex systems, the analyst may initially be unable to formulate as precise a mathematical model as he may wish. In fact, if the phenomena involved in some portion of the system are not well understood, the analyst may be forced to collect experimental data on that portion of system and then attempt to develop a mathematical model that will correspond with the data. This may require considerable effort and involve a number of attempts at refining the model or developing completely different ones as the phenomena involved become better understood.

Possibly the greatest danger that the analyst faces in using a mathematical model lies in his placing too much reliance on the fact that he has been able by one means or another to formulate and obtain solutions from a mathematical model, and then being misled by the results obtained. The solutions may be 100 percent correct but the model may not represent the physical system, either as a result of an actual error introduced in formulating it or because intentional simplifications have been made for the purpose of reducing the mathematical complexity and subsequently these simplifications have been forgotten. This type of pitfall is best avoided by experience and by comparison, at appropriate steps in the design, of results obtained from the model or subportions of it with experimental results obtained directly, using corresponding portions of the actual system. At some stages in the development of a complex device or system, it is frequently appropriate to run simulation studies in which portions of the physical equipment from the actual system are employed, while the remainder is simulated on a computer or with special-purpose devices. In fact, this technique is frequently carried to the point where essentially the whole system is tested by supplying it with simulated inputs and possibly by substituting dummy loads or synthetic disturbing torques on the output. In this manner, the system can be exercised for extended periods under conditions much more favorable for the experimenter and frequently at very great savings in both time and money. For example, test of a fire control system against real targets is much more difficult and time consuming than determination of its performance when subjected to synthetic inputs. Model studies do not remove the necessity for performing a final evaluation of a system under actual field conditions but, if the model studies have been well thought out and carried through, the field tests should proceed very smoothly.

The analyst's normal wishes are (1) to refine his model so that results obtained from it correspond very closely to those obtained from tests on the actual system and (2) to study the model in sufficient detail to enable him to arrive at parameters that will give optimum performance of the system. However, the optimum-parameter settings for well-designed systems are usually rather broad. Furthermore, a mathematical model necessarily differs from the physical system it is designed to describe and discrepancies necessarily exist between the performance of the model and of the physical system. Determination of the time at which it is appropriate

to terminate model studies and freeze the design of the actual system is one of the major decisions facing a project engineer. Unfortunately, as with many decisions of this type, little of general value can be said. Each situation must be examined in the light of the applicable technical background for the design and the nontechnical pressures for completion of the project. Experience in the technical areas involved and basic good judgment are the most important factors in reaching an appropriate decision.

The chapter which follows outlines the principal mathematical tools used by the system designer and discusses the use of mathematical models to determine system accuracy and dynamic performance. This material represents information that is essential for the man engaged in the design of systems where dynamic effects are important.

APPENDIX TO CHAPTER 1*

PHYSICAL CONSTANTS AND CONVERSION FACTORS

A. G. McNish†

Contents

* The content of this appendix is that of Section 2 of the Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables issued in June 1964 as part of the Applied Mathematics Series of the National Bureau of Standards.

† National Bureau of Standards.

Physical Constants and Conversion Factors

The tables in this chapter supply some of tho more commonly needed physical constants and conversion factors.

All scientific measurements are based upon four internationnl arbitrarily adopted units, tlie magnitudes of which are fixed by four agreed on standards:

Length—the metor—fixed by tho vacuum wavelength of radiation corresponding to tho transition $2P_{10}-5D_5$ of krypton 86

(1 meter= 1650763.73λ)

Mass—the kilogram—fixed by the international kilogram at Sèvres, France.

Time—the second—fixed as, 1/31,556,925.9747 of the tropical year 1900 at 12^h ephemeris time.

Temperature—the degree—fixed on a thermodynamic basis by taking the temperature for the triple point of natural water as 273.16 °K. (The Celsius scale is obtained by adding −273.15 to the Kelvin scale.)

All other units are defined in terms of them by assigning the value unity to the proportionality constant in each defining equation, the system so derived being called the MKS system. Taking the 1/100 part of the meter as the unit of length and the 1/1000 part of tho kilogram as the unit of mass similarly gives rise to the CGS system, often used in physics and chemistry. The more common named units arid their conversion factors are given in Table A-I.

Table A-I. Common Units and Conversion Factors

Quantify	MKS *name*	CGS *name*	MKS *unit*/ CGS *unit*
Force, F	newton	dyne	10^5
Energy, W	joule	erg	107
Power, P	watt	--------	107

The practical, or MKSA, electrical units are defined by the force per unit length between two infinitely long parallel filamentary conductors carrying current when unit distance apart in a vacuum by the equation $\Gamma_m I_1 I_2/4\pi = 2F$. If F is in newtons and Γ_m has the numerical value $4\pi\times10^{-7}$ then I_1 and I_2 are measured in terms of the practical unit, the ampere. The customary equations of the rationalized MKSA system then define the other electric and magnetic units. The force between electric charges in a vacuum in this systeni is given by $Q_1Q_2/4\pi\Gamma_e r^2 = F$, Γ_e having the numerical vuluo $10^7/4\pi c^2$ where c is the speed of light in meters per second ($\Gamma_e = 8.854\times10^{-12}$).

The CGS unratiorialized system is obtained by deleting 4π in the denominators in these equations and expressing F in dynes nnd r in centimeters Setting Γ_m equal to unity defines the CGS unrationalized electromagnetic system (emu), Γ_e taking the numerical value of $1/c^2$. Setting Γ_e equal to unity defines the CGS unrationalized electrostatic system (esu), Γ_m taking the numerical value of $1/c^2$.

The Lorentz-Heaviside system involves a different process of rationalization.

Table A-11. Names and Conversion Factors for Electric and Magnetic Units

Quantity	MKS name	emu name	esu name	MKS unit/ emu unit	MKS unit/ esu unit
Current	ampere	abampere	statampere	10^{-1}	$\sim3\times10^9$
Charge	coulomb	abcoulomb	statcoulomb	10^{-1}	-3×10^9
Potential	volt	abvolt	stntvolt	10^8	$\sim(1/3)\times10^{-3}$
Resistance	ohm	abohm	statohm	10^9	$\sim(1/9)\times10^{-11}$
Inductance	henry	centimeter	----------	10^9	$\sim(1/91\times10^{-11}$
Capaçitance	farad	----------	centimeter	10^{-9}	$\sim9\times10^{11}$
Magnetizing force	amp. turns/ meter	oersted	----------	$4\pi\times10^{-3}$*	$\sim3\times10^9$*
Magnetomotive force	amp. turns	gilbert		$4a\times10^{-1}$*	$-3/10^{6}$*
Magnetic flux	weber	maxwell	----------	10^8	$\sim(1/3)\times10^{-3}$
Magnetic flux density	tesla	gauss	----------	10^4	$\sim(1/3)x\ 10^{-6}$
Electric displacement	----------	----------	----------	10^{-5}*	$\sim3\times10^{5}$*

Example: If the value assigned to a current is 100 amperes its value in abamperes is $100\times10^{-1}=10$.

*Divide this number by 4π if unrationalized MKS system is involved; other numbers nre unchanged.

The adjusted values of constants given in Table A-111 are those recommended by the National Academy of Sciences–National Research Council Committee on Fundamental Constants in 1963. The error limits are three times the standard errors estimated from the experimental data included in the adjustment. Values, where pertinent, are based on the unified scale of atomic masses in which the atomic mass unit (u) is defined at 1/12 of the mass of the atom of the ^{12}C nuclide.

Table A-111. Adjusted Values of Constants

Constant	Symbol	Value	Est. ‡ error limit	Unit: Système International (MKSA)		Unit: Centimeter-gram-second (CGS)	
Speed of light in vacuum	c	2.997925	3	$\times 10^{8}$	m s^{-1}	$\times 10^{10}$	cm s^{-1}
Elementary charge	e	1.00210	7	10^{-19}	C	10^{-20}	$cm^{1/2}g^{1/2}$ *
		4.80298	20	. . .	. . . . .	10^{-10}	$cm^{3/2}g^{1/2}s^{-1}$ †
Avogadro constant	N_A	6.02252	28	10^{23}	mol^{-1}	10^{23}	mol^{-1}
Electron rest mass	m_e	9.1091	4	10^{-31}	kg	10^{-28}	g
		5.48597	9	10^{-4}	u	10^{-4}	u
Proton rest mass	m_p	1.67252	8	10^{-27}	kg	10^{-24}	g
		1.00727663	24	10^{0}	u	10^{0}	u
Neutron rest mass	m_n	1.67182	8	10^{-27}	kg	10^{-24}	g
		1.0086654	13	loo	u	10^{0}	u
Faraday constant	F	9.61870	16	10^{4}	C mol^{-1}	10^{3}	$cm^{1/2}g^{1/2}mol^{-1}$ *
		2.89261	5	. . . . .	. . . . .	10^{14}	$cm^{3/2}g^{1/2}s^{-1}mol^{-1}$ †
Planck constant	h	6.6256	5	10^{-34}	J s	10^{-27}	erg s
	$\hbar$	1.05450	7	10^{-34}	J s	10^{-27}	erg s
Fine structure constant	α	7.29720	10	10^{-3}	. . . . .	10^{-3}	
	$1/\alpha$	1.370388	19	10	. . . . .	10^{2}	
	$\alpha/2\pi$	1.161385	16	10^{-3}	. . . . .	10^{-3}	
	α^2	5.32492	14	10^{-5}	. . . . .	10^{-5}	
Charge to mass ratio for electron	e/m_e	1.758796	19	10^{11}	C kg^{-1}	107	$cm^{1/2}g^{-1/2}$ *
		5.27274	6	. . . . .	. . . . .	10^{17}	$cm^{3/2}g^{-1/2}s^{-1}$ †
Quantum-charge ratio	h/e	4.13556	12	10^{-15}	J s C^{-1}	10^{-7}	$cm^{3/2}g^{1/2}s^{-1}$ *
		1.37947	4	. . . . .	. . . . .	10^{-17}	$cm^{1/2}g^{1/2}$ †
Compton wavelength of electron	λ_C	2.42621	6	10^{-12}	m	10^{-10}	cm
	$\lambda_C/2\pi$	3.86144	9	10^{-13}	m	10^{-11}	cm
Compton wavelength of proton	$\lambda_{C,p}$	1.32140	4	10^{-15}	m	10^{-13}	cm
	$\lambda_{C,p}/2\pi$	2.10307	6	10^{-16}	m	10^{-14}	cm
Rydberg constant	R_∞	1.0973731	3	107	m^{-1}	10^{5}	cm^{-1}
Bohr radius	a_0	5.29167	7	10^{-11}	m	10^{-9}	cm
Electron radius	r_e	2.81777	11	10^{-15}	m	10^{-13}	cm
	r_e^2	7.9398	6	10^{-30}	m^2	10^{-26}	cm^2
Thomson cross section	$8\pi r_e^2/3$	6.6516	5	10^{-29}	m^2	10^{-25}	cm^2
Gyromagnetic ratio of proton	γ	2.67519	2	10^{8}	rad $s^{-1}T^{-1}$	10^{4}	rad $s^{-1}G^{-1}$ *
	$\gamma/2\pi$	4.25770	3	107	Hz T^{-1}	10^{3}	$s^{-1}G^{-1}$ *
(uncorrected for diamagnetism. H_2O)	γ'	2.67512	2	10^{8}	rad $s^{-1}T^{-1}$	10^{4}	rad $s^{-1}G^{-1}$ *
	$\gamma'/2\pi$	4.25759	3	10^{7}	Hz T^{-1}	10^{3}	$s^{-1}G^{-1}$ *
Bohr magneton	μ_B	9.2732	6	10^{-24}	J T^{-1}	10^{-21}	erg G^{-1} *
Nuclear magneton	μ_N	5.0505	4	10^{-27}	J T^{-1}	10^{-24}	erg G^{-1} *
Proton moment	μ_p	1.41019	13	10^{-26}	J T^{-1}	10^{-23}	erg G^{-1} *
	μ_p/μ_N	2.79276	?	10^{0}	. . . . .	10^{0}	
(uncorrected for diamagnetism, H_2O)	μ'_p/μ_N	2.79368	7	10^{0}	. . . . .	10^{0}	
Anomalous electron moment corrn.	$(\mu_e/\mu_0)-1$	1.159615	15	10^{-3}	. . . . .	10^{-3}	
Zeeman splitting constant	μ_B/hc	4.66858	4	10^{1}	$m^{-1}T^{-1}$	10^{-5}	$cm^{-1}G^{-1}$ *
Gas constant	R	8.3143	12	10^{0}	J $°K^{-1}$ mol^{-1}	107	erg $°K^{-1}$ mol^{-1}
Normal volume perfect gas	V_0	2.21136	30	10^{-2}	m^3 mol^{-1}	10^{4}	cm^3 mol^{-1}
Boltzmann constant	k	1.38054	18	10^{-23}	J $°K^{-1}$	10^{-16}	erg $°K^{-1}$
First radiation constant ($2\pi hc^2$)	c_1	3.7405	3	10^{-16}	W m^2	10^{-5}	erg cm^2 s^{-1}
Second radiation constant	c_2	1.43879	19	10^{-2}	m °K	10^{0}	cm °K
Wien displacement constant	b	2.8978	4	10^{-3}	m °K	10^{-1}	cm °K
Stefan-Boltzmann constant	σ	5.6697	29	10^{-8}	W m^{-2} $°K^{-4}$	10^{-5}	erg cm^{-2} s^{-1} $°K^{-4}$
Gravitational constant	G	6.670	15	10^{-11}	N m^2 kg^{-2}	10^{-8}	dyn cm^2 g^{-2}

‡Based on 3 std. dev, applied to lost digits in preceding column. *Electromagnetic system. †Electrostatic system.
C—coulomb J—joule Hz—hertz W—watt N—newton T—tesla G—gauss

Table A-IV. Miscellaneous Conversion Factors

Standard gravity g_0	$=9.80665$ m sec^{-2}
Standard atmospheric pressure P_0	$=1.013250\times10^5$ newtons m^{-2} 10^6 dynes cm^{-2}
1 Thermodynamic calorie[2] cal_c	$=4.1840$ joules
1 *I T* calorie[3] cnl,	$=4.1868$ joules
1 liter 1	$=1.000028\times10^{-3}$ m^3
1 Angstrom unit Å	$=10^{-10}$ m
1 Bar	$=10^5$ newtons m^2 10^6 dynes cm^2
1 Gal	$=10^{-2}$ m sec^{-2} 1 cm sec^{-2}
1 Astronomical unit a.u.	$=1.495\times10^{11}$ m
1 Light year	$=9.46\times10^{15}$ m
1 Parsec	$=3.08\times10^{16}$ m $=3.26$ light years

1 **Curie, the** quantity **of** radioactive material undergoing 3.700×10^{10} disintegrations sec^{-1}.

1 Roentgen, the exposure of **x-** or gamma radiation which produces together with its secondaries 2.082×10^9 electron-ion **pairs** in 0.001293 **gin** dry air.

Formula for index of refraction of atmosphere for radio waves ($f<3\times10^{10}$) $(n-1)10^6=(77.6/T)(p+4810e/T)$, where n is refractive index; T temperature °K; **p** total pressure in millibars, e water vapor partial pressure in millibars.

Factors for converting the customary United Stntcs units to units of tlie metric system arc given in **Table A-V.**

Table A-V. Factors for Converting Customary U.S. Units to Metric Units

1 yard	**0.0144** meter
1 foot	0.3048 meter
1 inch	**0.0254** meter
1 statute mile	1609.344 meters
1 nautical mile (international)	1852 meters
1 pound (avtlp.)	**0.45359237** kilogram
1 o%.(avdp.)	0.0283495 kilogram
1 pound force	**4.44823** newtons
1 slug	14.5939 kilograms
1 poundal	**0.135255** newtons
1 foot pound	1.35582 joules·
Temperature (Fahrenheit)	32+ (9/5)(temperature Celsius)
1 British thermal unit[4]	1055 joules

Geodetic constants for the international (Hayford) spheroid are given in **Table A-VI.** The gravity values are on tlic basis of **tlic old** Potsdam value and have not been corrected for more recent determinations. They are probably about **13** parts per million too great. They are calculated for the surface of the geoid by the international formula.

Table A-VI. Geodetic Constants

$a=6{,}378{,}388$ m; $f=1/297$; $b=6{,}356{,}912$ m

Latitude	Length of 1′ of parallel	Length of 1′ of meridian	g msec^{-2}
	Meters	*Meters*	*Meters*
0°	1,855.398	**1,842.925**	9.780490
15	1,792.580	**1,544.170**	9.783940
30	1,605.174	1,847.580	9.793378
45	1,314.175	1,852.256	9.806294
60	930.047	1,856.951	9.819239
75	481.725	1,860.401	9.828734
90	0	1,861.666	9.832213

[2] Used principally by chemists.

[3] Used principally by engineers.

[4] Various definitions are given for the British thermal unit. This represents a rounded mean value differing from none of the more important definitions by more than 3 in 10^4.

REFERENCES

1. J. E. Alexander, and J. M. Bailey, Systems Engineering Mathematics, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1962.
2. E. A. Guillemin, Introductory Circuit Theory, John Wiley & Sons, Inc., New York, N. Y., 1955.
3. E. A. Guillemin, The Mathematics of Circuit Analysis, John Wiley & Sons, Inc., New York, N. Y., 1949 or 1951.
4. M. E. Gardner and J. L. Barnes, Transients in Linear Systems, Vol. 1, John Wiley & Sons, Inc., New York, N. Y., 1942.

CHAPTER 2*
DETERMINATION OF THE ACCURACY AND DYNAMIC RESPONSE OF A SYSTEM FROM STUDIES OF ITS MATHEMATICAL MODEL

2-1 INTRODUCTION

As discussed in par. 1-2, the first step the analyst faces in carrying out a theoretical study of the performance of a system is that of establishing a mathematical model for the system. He does this based upon a knowledge of the basic laws that describe mechanical, electrical, hydraulic, and other systems (including combinations of these systems) and upon a thorough and detailed understanding of the particular system with which he is concerned. The result of this step usually takes the form of a differential equation or, more generally, a set of differential equations that, in mathematical terms, describe the performance of the system.

The next step is to solve these equations by either analytic techniques or computer simulation techniques so as to obtain specific information showing how the system would respond to different types of inputs. This enables the designer to select the adjustable system parameters in such a way as to optimize system performance.

The first part of this chapter (see par. 2-2) surveys analytic techniques. Specifically, the application of such mathematical techniques as linear-differential-equation theory, frequency-domain analysis, frequency-response techniques, block diagrams and signal-flow graphs, statistical theory, and nonlinear analysis are described. The second part of the chapter (see par. 2-3) provides a brief discussion of the way in which analog and digital simulation techniques can be employed in studying mathematical models that are too complex for analysis by direct analytic techniques. The third part of this chapter (see par. 2-4) describes the application of digital computation to the branch of mathematics known as numerical analysis and summarizes the main aspects of the numerical techniques that can now be employed. Since a thorough discussion of these topics is beyond the scope of this handbook, a number of the more important references in each area are provided in order to enable the reader to obtain further information concerning those topics he finds of particular interest,

2-2 MATHEMATICAL TECHNIQUES

2-2.1 GENERAL

This summary of mathematical techniques deals with various methods of determining the dynamic response of physical systems from the differential equations that describe them. The type of response sought depends upon several factors: the specifications of the system; the design procedure adopted; and the limitations imposed by test conditions encountered when seeking experimental verification of the design performance.

Differential equations can be classified as follows:

(a) Linear differential equations with constant coefficients.
(b) Linear differential equations with time-varying coefficients.
(c) Nonlinear differential equations.

* By W. W. Seifert (par. 2-1, 2-2 and 2-3) and E. St. George, Jr. (par. 2-4).

Of these three classes, constant-coefficient linear differential equations are, by far, the most widely used and the best understood. The subject matter of par. 2-2.2 through par. 2-2.6 is focused exclusively on methods of solving equations in this class. For a discussion of nonlinear differential equations and some of the techniques employed for treating them, see par. 2-2.7 through par. 2-2.7.3.4. Linear differential equations with time-varying coefficients represent an intermediate case and are discussed in par. 2-2.7 in connection with nonlinear analysis.

2-2.2 LINEAR-DIFFERENTIAL-EQUATION THEORY

The general form of a linear differential equation with constant coefficients is

$$\sum_{i=0}^{n} a_i \frac{d^i x}{dt_i} = \sum_{j=0}^{m} b_j \frac{d^j v}{dt^j} \qquad (2\text{-}1)$$

where the a's and b's are the constant coefficients, x(t) is the response function, and y(t) is the input function. The equation is linear because the response to a sum of component input functions equals the sum of the responses to each of the component input functions. The highest-order derivative of the response, x(t), that is present in the equation is called the order of the equation. Thus, Eq. 2-1 is an equation of the nth order. The information necessary for a complete solution of the equation is a statement of the initial value of the response and the initial values of its first n - 1 derivatives, as well as specification of the input, y(t). By changing the initial conditions, one obtains a different *solution*. In the classical method of solution, the response can be separated into two parts: (1) a general or homogeneous solution, and (2) a particular solution. The complete *solution* of the differential equation is the sum of the general solution and the particular solution. The general solution always has the form of a sum of exponentials with real and complex arguments; the particular solution has the same form as the input or a sum of the input and its derivatives. The general solution is often called the force-free or transient solution; the particular solution is called the forced or steady-state solution. Each term in the transient solution is called a normal response mode or characteristic of the equation.

The complete solution of a linear differential equation can be represented in general terms by the relationship

$$x(t) = x_p(t) + \sum_{k=1}^{n} A_k e^{p_k t} \qquad (2\text{-}2)$$

where $x_p(t)$ is the particular solution, the p_k's are the roots of the characteristic equation, and the A_k's are polynomial functions of t. If there are no multiple roots, the A_k's are constant-amplitude coefficients. The A_k's and p_k's are, in general, complex numbers that must occur in conjugate pairs if the coefficients a_i (Eq. 2-1) are real.

The term "root" is applied to each of the p_k's because these numbers can be found from the differential equation by treating the differentiating operator d/dt as a real variable, replacing it by the symbol p for convenience, and setting y(t) equal to zero. The algebraic equation that results from making such substitutions in Eq. 2-1 is

$$\sum_{i=0}^{n} a_i p^i = 0 \qquad (2\text{-}3)$$

This equation is known as the characteristic equation. The roots of Eq. 2-3, when determined, give the p_k's of the normal response modes of Eq. 2-2.

The classical procedure for solving constant-coefficient linear differential equations is covered in many textbooks, for example, see Refs. 1, 2, 3, and 4. The use of more powerful tools for treating differential equations, such as Laplace and Fourier transforms, are discussed in par. 2-2.3 through 2-2.3.3. For situations where the input is sinusoidal or is stochastic, additional special techniques are used. These techniques are discussed respectively in par. 2-2.4 and par. 2-2.5. The use of block diagrams and signal-flow graphs is described in pars. 2-2.5.1 through 2-2.5.2.

2-2.3 FREQUENCY-DOMAIN ANALYSIS

2-2.3.1 Laplace and Fourier Transforms

Laplace and Fourier transforms[5] are typical aids for solving linear differential equations that come under the general classi-

fication of frequency-domain analysis. They introduce properties of system performance that enhance the designer's understanding and simplify his task.

The bilateral Laplace transform of a function is defined as follows:

$$\mathcal{L}\left[f(t)\right] \triangleq F(s) \triangleq \int_{-\infty}^{\infty} e^{-st} f(t)\, dt$$

[Direct] (2-4)

where s is the complex frequency variable, $\sigma + j\omega$, and the symbol $\triangleq$ means "equal by definition". The inverse bilateral Laplace transform has the form

$$\mathcal{L}^{-1}\left[F(s)\right] \triangleq f(t) \triangleq \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} e^{st} F(s)\, ds$$

[Inverse] (2-5)

where c is a constant that defines the path of integration.

The single-sided Laplace transform is a useful special case, applicable to time functions that exist only for $t \geq 0$. The transform and its inverse are defined as follows:

$$\mathcal{L}_{+}\left[f(t)\right] \triangleq F(s) \triangleq \int_{0}^{\infty} e^{-st} f(t)\, dt$$

[Direct] (2-6a)

$$\mathcal{L}_{+}^{-1}\left[F(s)\right] \triangleq f(t) \triangleq \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} e^{st} F(s)\, ds$$

[Inverse] (2-6b)

where the subscript + sign indicates that these two transforms apply for positive time only.

The Laplace transform exists for a large class of functions. For existence, it is necessary only that the function f(t) be piecewise differentiable (i.e., finite jumps of the function f(t) are permissible) and be of exponential order (i.e., the integral

$$\int_{-\infty}^{\infty} \left| f(t) \right| e^{-ct}\, dt$$

is finite for any finite value of C)[42].

As already noted, the frequency variable s in the bilateral Laplace transform is a complex variable. When attention is restricted to the imaginary component $j\omega$, the bilateral Laplace transform becomes identical in form with the Fourier transform. Thus, the Fourier transform can be considered to be a special case of the Laplace transform.* The Fourier transform and its inverse are defined by the relationships

$$\mathcal{F}\left[f(t)\right] \triangleq F(j\omega) \triangleq \int_{-\infty}^{\infty} e^{-j\omega t} f(t)\, dt$$

[Direct] (2-7a)

$$\mathcal{F}^{-1}\left[F(j\omega)\right] \triangleq f(t) \triangleq \frac{1}{2\pi j} \int_{-\infty}^{\infty} e^{j\omega t} F(j\omega)\, d(j\omega)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} F(j\omega)\, d\omega$$

[Inverse] (2-7b)

The Fourier transform exists for a more restricted class of functions than the Laplace transform. The requirement for the existence of the Fourier transform is that f(t) be piecewise differentiable and that the integral

$$\int_{-\infty}^{\infty} \left| f(t) \right| dt$$

exist.

2-2.3.2 Useful Theorems

The following theorems are useful for applying the Laplace and Fourier transforms to the solution of differential equations:

Linearity Theorems

(a) $$\mathcal{L}\left[a\, f(t)\right] = a\, F(s) \qquad (2\text{-}8)$$

(b) $$\mathcal{L}\left[\alpha f_1(t) + \beta f_2(t)\right] = \alpha \mathcal{L}\left[f_1(t)\right] + \beta \mathcal{L}\left[f_2(t)\right] \qquad (2\text{-}9)$$

* For reasons of historical development and relative complexity, the Laplace transform is sometimes introduced as a special case of the Fourier transform.

Real Differentiation Theorem

$$(c)\ \mathcal{L}\left[\frac{d^n f(t)}{dt^n}\right] = s^n F(s) - s^{n-1} f - s^{n-2} f^1(0+) - \ldots - s f^{(n-2)}(0+) - f^{(n-1)}(0+) \qquad (2\text{-}10)$$

in which $f(0+) \underset{=}{\mathbf{A}} \lim_{t \to 0} f(t)$, where the limit is approached from positive values of t and

$$f^{(k)}(t) \underset{=}{\mathbf{A}} \frac{d^k f(t)}{dt^k}.$$

Real Integration Theorem

(d)

$$\mathcal{L}\left[\underbrace{\int_{-\infty}^{t} \cdots \int_{-\infty}^{t}}_{\text{(n times)}} f(t)\,(dt)^n\right] = \frac{F(s)}{s^n} + \frac{\int_{-\infty}^{0+} f(t)\,dt}{s^n} + \frac{\int_{-\infty}^{0+}\left[\int_{-\infty}^{t} f(t)\,dt\right]dt}{s^{n-1}} + \ldots + \frac{\int_{-\infty}^{0+}\left[\underbrace{\int_{-\infty}^{t} \cdots \int_{-\infty}^{t}}_{\text{(n-1 times)}} f(t)\,(dt)^{n-1}\right]dt}{s} \qquad (2\text{-}11)$$

Normalization Theorem

$$(e)\ \mathcal{L}\left[f\left(\frac{t}{a}\right)\right] \qquad (as) \qquad (2\text{-}12)$$

This relationship is useful when it is desired to change the time scale of a problem.

Real Convolution Theorem

$$(f)\ \mathcal{L}\left[\int_0^t f_1(t-\tau)\, f_2(\tau)\, d\tau\right] = F_1(s)\, F_2(s) \qquad (2\text{-}13)$$

where τ is a new time variable.

$$(g)\ \mathcal{L}[f_1(t)\, f_2(t)] = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F_1(s \cdot \qquad (2\text{-}14)$$

where the notation $*$ means that $F_1(s)$ is convolved with $F_2(s)$.

$$(h)\ \mathcal{L}[f_1(t)\, f_2(t)] \neq F_1(s)\, F_2(s) \qquad (2\text{-}15)$$

if neither $f_1(t)$ nor $f_2(t)$ is equal to zero.

Real Translation Theorem

$$(i)\ \mathcal{L}[f(t-a)] = e^{-as} F(s) \qquad (2\text{-}16)$$

if $f(t-a) = 0$ for $0 < t < a$

$$(j)\ \mathcal{L}[f(t+a)] = e^{as} F(s) \qquad (2\text{-}17)$$

if $f(t+a) = 0$ for $-a < t < 0$

Final-Value Theorem

$$(k)\ \lim_{s \to 0} s F(s) = \lim_{t \to \infty} f(t) \qquad (2\text{-}18)$$

Initial-Value Theorem

$$(l)\ \lim_{s \to \infty} s F(s) = \lim_{t \to 0} f(t)$$

Theorems (a), (b), (e), (f), (g), (h), and (k) also apply to the Fourier transform.

*Eq. 2-15 merely brings attention to a common error; Eq. 2-14 is the correct form of $\mathcal{L}[f_1(t)\, f_2(t)]$.

2-2.3.3 Solution Procedure

The solution of ordinary linear differential equations is accomplished by means of theorems (a), (b), (c), and (d) of par. 2-2.3.2. Application of these theorems to Eq. 2-1 shows that

$$\left[\sum_{i=0}^{n} a_i s^i\right] X(s) - A(s) = \left[\sum_{j=0}^{m} b_j s^j\right] Y(s) + B(s) \tag{2-20}$$

where A(s) is a polynomial in s depending upon the a's and the initial values of x and its first (n-1) derivatives, and B(s) is a polynomial in s depending upon the b's and the initial values of y and its first (m-1) derivatives. The response transform can be obtained by solving Eq. 2-20 for X(s), i.e.,

$$X(s) = \left[\frac{\sum_{j=0}^{m} b_j s^j}{\sum_{i=0}^{n} a_i s^i}\right] Y(s) + \left[\frac{B(s) + A(s)}{\sum_{i=0}^{n} a_i s^i}\right] \tag{2-21}$$

In words, this equation can be written

$$\begin{pmatrix}\text{response}\\ \text{transform}\end{pmatrix} = \begin{pmatrix}\text{system}\\ \text{function}\end{pmatrix}\begin{pmatrix}\text{input}\\ \text{transform}\end{pmatrix} + \begin{pmatrix}\text{initial condition}\\ \text{function}\end{pmatrix} \tag{2-22}$$

The ratio of the response transform to the input transform when all initial conditions are zero (i.e., when the initial condition function is zero) is called the system function or the transfer function of the system. This function depends only upon the coefficients of the differential equation and is independent of the input and the initial conditions. As will be shown later, the transform of an impulse function is unity. Therefore, a comparison of Eq. 2-22 (with initial condition function set equal to zero) with Eq. 2-20 shows that the transfer function of a system equals the transform of the impulse response of the system for a unit impulse.

Transforming a differential equation enables the analyst to replace the processes of differentiation and integration by simple algebraic processes. Then, the transform X(s) can be found algebraically. Subsequently, the system response x(t) corresponding to the response transform X(s) can be found by using the inverse Laplace transform (see Eq. 2-7). However, this inverse transform usually involves contour integration in the complex s plane. To avoid this integration, tables of transform pairs have been constructed that give the time function corresponding to a given transform directly. A brief list of commonly used transform pairs is given in Table 2-1. More extensive tables can be found in Refs. 5 and 6.

If tables of transform pairs are unavailable, or if the particular transform whose inverse is sought is not listed in the tables, the method of partial fractions may be used to expand the transform into a sum of terms, each of which is readily recognized as the transform of a simple time function. If the transform whose inverse is sought is a ratio of rational polynomials, the roots of the numerator polynomial are called the zeros of the function and the roots of the denominator polynomial are called the poles* of the function. If the poles of the function are not repeated, they are called simple poles. The order of a pole is the number of times the pole is repeated. For a function containing only simple poles, the partial-fraction expansion of the function is

$$F(s) \triangleq \frac{N(s)}{D(s)} = \sum_{k=1}^{n} \frac{K_k}{s - s_k} \tag{2-23}$$

where

$$K_k \triangleq \left[\frac{(s-s_k)N(s)}{D(s)}\right]_{s=s_k} = \left[\frac{N(s)}{D'(s)}\right]_{s=s_k} \tag{2-24}$$

and s_k is the kth root of the denominator polynomial D(s).

* A function F(s) that can be represented by a ratio of polynomials is said to have a pole at $s = s_k$ of order n if $\lim_{s \to s_k} F(s) = \infty$ and if $\left[(s-s_k)^n F(s)\right]_{s=s_k}$ is finite and not zero. The function F(s) is said to have a zero at $s = s_k$ if $\lim_{s \to s_k} F(s) = 0$.

TABLE 2-1. COMMONLY USED LAPLACE TRANSFORM PAIRS.

No.	$F(s)$	$f(t), t \geqq 0$
1	1	$\delta_0(t)$, unit impulse
2	$\frac{1}{s}$	$6\text{-}_1(t)$, unit step
3	$\frac{1}{s^2}$	$\delta_{-2}(t)$, unit ramp
4	$\frac{1}{Ts+1}$	$\frac{1}{T} e^{-t/T}$
5	$\frac{\omega}{s^2+\omega^2}$	sin tot
6	$\frac{s}{s^2+\omega^2}$	cos of
7	$\frac{1}{s^2+2\zeta\omega_n s+\omega_n^2}$	(1) $\zeta<1$: $\frac{1}{\omega_n\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2}t$ (2) $\zeta=1$: $te^{-\omega_n t}$ (3) $\zeta>1$: $\frac{1}{\omega_n\sqrt{\zeta^2-1}} e^{-\zeta\omega_n t} \sin t\,\omega_n \sqrt{\zeta^2-1}t$
8	$\frac{1}{(s+\alpha)^2+\beta^2}$	$\frac{1}{\beta} e^{-\alpha t} \sin \beta t$
9	$\frac{s+\alpha}{(s+a)'+\beta^2}$	$e^{-\alpha t} \cos \beta t$
10	$\frac{1}{s^n}$	$\frac{1}{(n-1)!} t^{n-1}$
11	$\frac{1}{(Ts+1)^n}$	$\frac{1}{(n-1)!} \frac{t^{n-1}}{T^n} e^{-t/T}$

If the transform contains multiple-order poles, the partial-fraction expansion of the function is

$$F(s) \triangleq \frac{N(s)}{D(s)} = \sum_{k=1}^{n} \sum_{j=1}^{m_k} \frac{K_{kj}}{(s-s_k)^{m_k-j+1}} \qquad (2\text{-}25)$$

where

$$K_{kj} \triangleq \frac{1}{(j-1)!} \left\{ \frac{d^{j-1}}{ds^{j-1}} \left[\frac{(s-s_k)^{m_k} N(s)}{D(s)} \right] \right\}_{s=s_k} \qquad (2\text{-}26)$$

and m_k is the order of the pole of $F(s)$ at $s=s_k$.

From Eqs. 2-23 and 2-25, it is obvious that the expansion of a rational function that is inverse transformed produces a sum of exponential terms for the corresponding time function. Terms containing simple poles, as in Eq. 2-23, may be inverse-transformed by the use of transform 4 of Table 2-1. For multiple-order poles with real roots, transform 11 is employed. More commonly, the multiple-order poles appear in complex conjugate pairs; in this case, transforms 8 and 9 are employed, and the time functions are combined to form product terms (exponentials multiplied by a sine or cosine function) representing damped sinusoids.

An alternative to the partial-fraction expansion method is the method of residues. If $F(s)$ has a simple pole at $s=s_k$, then the residue $\phi(s_k)$ is given by the relationship

$$\phi(s_k) = \frac{N(s)}{(s-s_k)\,D(s)} \qquad (2\text{-}26a)$$

and the term of f(t) corresponding to that pole is $\phi(s_k)e^{s_k t}$. The complete time function is the sum of the residues of F(s) multiplied by e^{st}, for all the poles. For multiple-order poles, the residue formula is[43]

$$\phi(s_k) = \frac{1}{(n-1)!}\left[\frac{\partial^{n-1}}{\partial s^{n-1}}(s-s_k)^n \frac{N(s)}{D(s)} e^{st}\right]_{s=s_k} \qquad (2\text{-}26b)$$

where n is the order of the pole. Eq. 2-26b reduces to Eq. 2-26a for n = 1.

Example. The system defined by the equation

$$\frac{d^4x}{dt^4} + 10.65\frac{d^3x}{dt^3} + 89.0\frac{d^2x}{dt^2} + 15.50\frac{dx}{dt} + 27.0x = 27.0y \qquad (2\text{-}27)$$

is initially at rest. At t = 0, a unit ramp input is applied. Find the difference between the input y and the output x as a function of time.

Solution. Since the system is initially at rest, all initial conditions are zero. Transforming Eq. 2-27 results in

$$X(s) = \frac{27.0}{s^4 + 10.65s^3 + 89.0s^2 + 15.50s + 27.0} Y(s) \qquad (2\text{-}28)$$

Let

$$e(t) = y(t) - x(t) \qquad (2\text{-}29)$$

Then, transforming Eq. 2-29 and substituting for X(s) from Eq. 2-28 gives

$$E(s) = \frac{s\left[s^3 + 10.65s^2 + 89.0s + 15.501\right.}{s^4 + 10.65s^3 + 89.0s^2 + 15.50s + 27.0} Y(s) \qquad (2\text{-}30)$$

Determination of the solution of Eq. 2-30 requires that the denominator of the equation be put in factored form. Unfortunately, determination of the roots of equations of order higher than the third is difficult unless the roots happen to be real. One of the methods best suited to paper-and-pencil computations is Lin's method[7]. This is a division technique in which a trial divisor is assumed and refined by repeated trials until a factor is found to the accuracy desired.

Consider an equation of the form

$$s^n + B_{n-1}s^{n-1} + B_{n-2}s^{n-2} + \ldots + B_2s^2 + B_1s + B_0 = 0 \qquad (2\text{-}31)$$

The first step when n is even is to select a trial divisor formed from the last three terms. This divisor takes the form

$$s^2 + \frac{B_1}{B_2}s + \frac{B_0}{B_2} \qquad (2\text{-}32)$$

This is divided into the original equation as follows:

$$s^2 + \frac{B_1}{B_2}s + \frac{B_0}{B_2}\;\Big)\;\overline{s^{n-2} + \ldots}$$

$$\overline{s^n + B_{n-1}s^{n-1} + B_{n-2}s^{n-2} + \ldots + B_2s^2 + B_1s + B_0}$$

$$C_2s^2 + C_1s + C_0$$

$$D_2s^2 + D_1s + D_0$$

Remainder

If the remainder is negligible, then the divisor selected is a quadratic factor of the original equation. If the remainder is not negligible, then a second trial divisor is formed as follows:

$$s^2 + \frac{C_1}{C_2}s + \frac{C_0}{C_2} \qquad (2\text{-}33)$$

where the C's are determined from the preceding division. The second trial divisor is divided into the original equation as was the first. If the remainder is negligible, the second trial divisor is a quadratic factor of the original equation. If not, the process is again repeated. After one factor is found, the method is applied in the same way to the resulting polynomial, which is now of order n-2.

When the highest power of the original equation is odd, a linear factor of the form

$$s + \frac{B_0}{B_1} \qquad (2\text{-}34)$$

is taken as the trial divisor.

This method maybe applied to find the roots of the denominator of Eq. 2-30 as follows:

The first trial divisor is

$$s^2 + \frac{15.50}{89.0}s + \frac{27.0}{89.0} = s^2 + 0.174s + 0.303$$

This is then divided into the original equation to give

$$\begin{array}{r|l} & s^2 + 10.48s + 86.9 \\ \hline s^2 + 0.174s + 0.303 & s^4 + 10.65s^3 + 89.0s^2 + 15.50s + 27.0 \\ & s^4 + 0.174s^3 + 0.303s^2 \\ \hline & 10.48s^3 + 88.7s^2 + 15.50s \\ & 10.48s^3 + 1.8s^2 + 3.18s \\ \hline & 86.9s^2 + 12.32s + 27.0 \\ & 86.9s^2 + 15.12s + 26.3 \\ \hline & 2.80s + 0.7 \end{array}$$

The second trial divisor becomes

$$s^2 + \frac{12.32}{86.9}s + \frac{27.0}{86.9} = s^2 + 0.142s + 0.311$$

Division then yields

$$\begin{array}{r|l} & s^2 + 10.51s + 87.2 \\ \hline s^2 + 0.142s + 0.31 & s^4 + 10.65s^3 + 89.0s^2 + 15.50s + 27.0 \\ & s^4 + 0.14s^3 + 0.3s^2 \\ \hline & 10.51s^3 + 88.7s^2 + 15.50s \\ & 10.51s^3 + 1.5s^2 + 3.27s \\ \hline & 87.2s^2 + 12.23s + 27.0 \\ & 87.2s^2 + 12.38s + 27.12 \\ \hline & -0.15s - 0.12 \end{array}$$

The remainder is such that the greatest error in any term is 1%. This is sufficiently small for this example; so now the denominator may be written in factored form as

$$(s^2 + 0.142s + 0.311)(s^2 + 10.51s + 87.2)$$

The roots of each of these quadratics may now be found by application of the quadratic formula.

At this stage, it is possible to write Eq. 2-30 in the factored form

$$E(s) = \frac{s\left[s^3 + 10.65s^2 + 89.0s + 15.501\right]}{(s^2 + 0.142s + 0.311)(s^2 + 10.51s + 87.2)}\,Y(s) \qquad (2\text{-}35)$$

Since it is desired to evaluate E(s) when y(t) is a unit ramp applied at t = 0, the transform of the unit ramp is found from Table 2-1 and substituted in Eq. 2-35. Since the transform of a unit ramp is $1/s^2$, the result is

$$E(s) = \frac{s^3 + 10.65s^2 + 89.0s + 15.50}{s(s^2 + 0.142s + 0.311)(s^2 + 10.51s + 87.2)} \qquad (2\text{-}36)$$

The inverse transform of E(s) is found by reducing the expression for E(s) into the sum of a number of terms for each of which the transform is known or can be obtained from a table. This means that a partial-fraction expansion of Eq. 2-36 must be made. This expansion * takes the form:

$$E(s) = \frac{k_1}{s} + \frac{K_1}{s-s_1} + \frac{K_1^*}{s-s_1^*} + \frac{K_2}{s-s_2} + \frac{K_2^*}{s-s_2^*} \qquad (2\text{-}37)$$

[K_1 and K_1^* are complex conjugates; K_2 and K_2^* are complex conjugates]

since the roots of each of the quadratic terms are complex conjugates, i.e.,

$$\begin{array}{ll} s_1 = -\alpha_1 + j\omega_1 & s_2 = -\alpha_2 + j\omega_2 \\ s_1^* = -\alpha_1 - j\omega_1 & s_2^* = -\alpha_2 + j\omega_2 \end{array} \qquad (2\text{-}38)$$

The undriven or transient response of any system whose characteristic equation is a linear constant-coefficient differential equation with real coefficients takes the form:

$$c(t) = k_1 e^{-\sigma_1 t} + k_2 e^{-\sigma_2 t} + \ldots + K_1 e^{(-\alpha_1 + j\omega_1)t} + K_1^* e^{(-\alpha_1 - j\omega_1)t} + \ldots + K_N e^{(-\alpha_N + j\omega_N)t} + K_N^* e^{(-\alpha_N - j\omega_N)t} \qquad (2\text{-}39)$$

* It is important to note that a polynomial equation with real coefficients has pairs of conjugate zeros, but this is not the case for polynomial equations in general. Example: $z^2 + (j-2)z - 2j = 0$. This polynomial equation has 2 and $-j$ as the only possible zeros.

where K_i and K_i^* are complex conjugates; i = 1, 2, ..., N.

The constants K_i and K_i^* in Eq. 2-37 (i = 1, 2) are therefore complex conjugates and may be written as

$$K_i = a_i + jb_i \text{ and } K_i^* = a_i - jb_i \qquad (2\text{-}40)$$

Insertion of the expression of Eq. 2-38 and Eq. 2-40 into Eq. 2-37 yields

$$E(s) = \frac{k_1}{s} + \frac{a_1 + jb_1}{(s + a_1) - j\omega_1} + \frac{a_1 - jb_1}{(s + a_1) + j\omega_1} + \frac{a_2 + jb_2}{(s + a_2) - j\omega_2} + \frac{a_2 - jb_2}{(s + a_2) + j\omega_2} \qquad (2\text{-}41)$$

The terms with complex conjugate roots can be combined to yield

$$E(s) = \frac{k_1}{s} + \frac{2a_1(s + a_1) - 2b_1\omega_1}{(s + a_1)^2 + \omega_1^2} + \frac{2a_2(s + a_2) - 2b_2\omega_2}{(s + a_2)^2 + \omega_2^2} \qquad (2\text{-}42)$$

The values of a_1, b_1, a_2, and b_2 are found in the usual manner following Eq. 2-24. The results, in general, are complex numbers and the real part is associated with the a_i terms and the imaginary part with the b_i terms in accordance with Eq. 2-40.

In the example at hand, the quadratic terms in the denominator may be factored using the quadratic formula. The results are

$$a_1 = +0.071 \qquad a_2 = +5.26$$
$$\omega_1 = 0.553 \qquad \omega_2 = 7.72 \qquad (2\text{-}43)$$

K_1 may then be found as

$$K_1 = a_1 + jb_1 = \left[(s - s_1)\frac{N(s)}{D(s)}\right]_{s = -0.071 + j0.553} = \frac{(s^3 + 10.65s^2 + 89.0s + 15.50)}{s(s + 0.071 + j0.553)(s^2 + 10.51s + 87.2)}\Bigg|_{s = -0.071 + j0.553} \qquad (2\text{-}44)$$

Straightforward substitution of $s = s_1$ entails considerable manipulation. This may be simplified by reducing the expression for K_1 to its completely factored form and then employing an evaluation scheme based upon a graphical approach. In factored form, after substitution of $s = s_1$,

$$K_1 = \frac{(s_1 + 0.178)(s_1 + 5.24 - j7.73)(s_1 + 5.24 - j7.73)}{s_1(s_1 + 0.071 + j0.553)(s_1 + 5.26 - j7.72)(s_1 + 5.26 + j7.72)} \qquad (2\text{-}45)$$

The roots of Eq. 2-45 appear in the s-plane as shown in Fig. 2-1.

It is now possible to evaluate K_1 in terms of the length and angle of the phasors* drawn to the root s_1 from the other poles and zeros of the function; i.e.,

$$K_1 = \frac{(0.56\ \angle 79.1°)(9.44\ \angle 61.3°)(8.49\ \angle 302.3°)}{(0.557\ \angle 97.3°)(1.11\ \angle 90°)(8.85\ \angle 305.9°)(9.76\ \angle 57.9°)} = 0.840\ \angle -108.4° = -0.265 - j0.797 = a_1 + jb_1 \qquad (2\text{-}46)$$

Then

$$K_1^* = 0.840\ \angle +108.4° = -0.265 + j0.797 = a_1 - jb_1 \qquad (2\text{-}47)$$

Similarly, K_2 and K_2^* can be found to be

$$K_2 = 2.29 \times 10^{-4}\ \angle 111° = (-0.82 + j2.14) \times 10^{-4} = a_2 + jb_2 \qquad (2\text{-}48)$$

$$K_2^* = 2.29 \times 10^{-4}\ \angle -111° = (-0.82 - j2.14) \times 10^{-4} = a_2 - jb_2 \qquad (2\text{-}49)$$

Substitution of these values of a_1, b_1, a_2, and b_2 and the value of k, into Eq. 2-42 yields

* A phasor is a directed line segment in the complex plane. With the segment's point of origin given, the phasor is defined either by a magnitude and an angle (the symbol ∠ denotes angle) or by the real and imaginary components of its terminal point.

$$E(s) \quad \frac{0.572}{s} + \frac{2(-0.265)(s + 0.071) - 2(-0.797)(0.553)}{(s + 0.071)^2 + (0.553)^2}$$

$$+ \frac{2(-0.82 \times 10^{-4})(s + 5.26) - 2(-2.14 \times 10^{-4})(7.72)}{(s + 5.26)^2 + (7.72)^2}$$

$$\frac{0.572}{s} \quad \frac{-0.530(s + 0.071)}{(s + 0.071)^2 + (0.553)^2} + \frac{0.881}{(s + 0.071)^2 + (0.553)^2}$$

$$\frac{1.64 \times 10^{-4}(s + 5.26)}{(s + 5.26)^2 + (7.72)^2} + \frac{33.04 \times 10^{-4}}{(s + 5.26)^2 + (7.72)^2} \qquad (2\text{-}50)$$

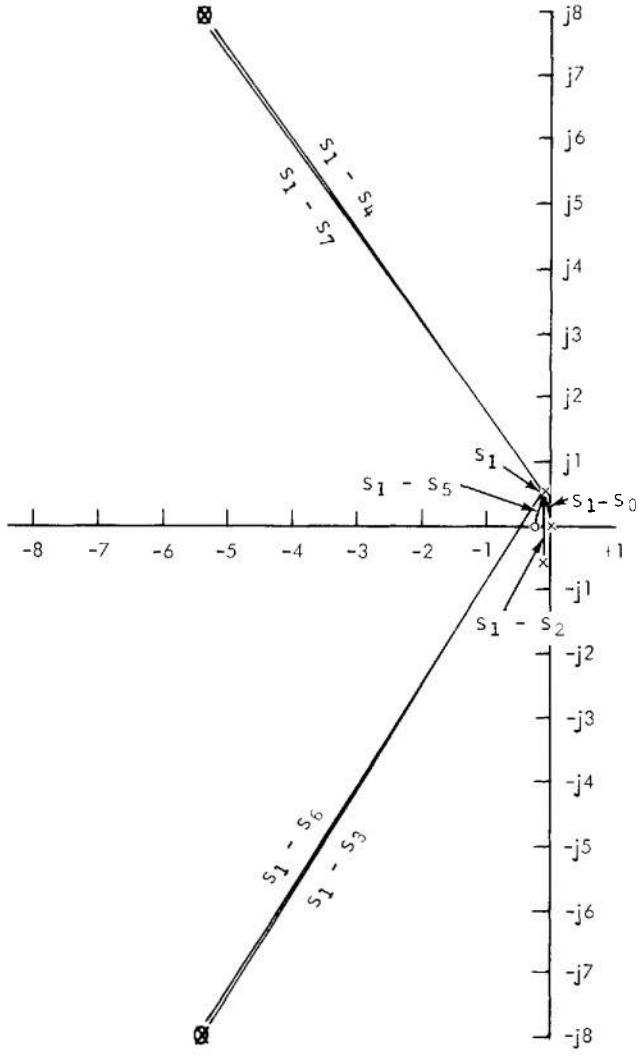


Figure 2-1. Locations of the roots of Eq. 2-45 in the s-plane.

Each of these terms is now in a form that appears directly in the table of transforms. It is, therefore, now possible to write directly

$$e(t) = 0.572 - 0.530 e^{-0.071t} \cos 0.553t + \frac{0.881}{0.553} e^{-0.071t} \sin 0.553t - 1.64 \times 10^{-4} e^{-5.26t} \cos 7.72t \frac{33.04 \times 10^{-4}}{7.72} e^{-5.26t} \sin 7.72t$$

$$= 0.572 + e^{-0.071t} \left[-0.530 \cos 0.553t + 1.593 \sin 0.553t \right] + e^{-5.26t} \left[-1.64 \sin 7.72t - 4.28 \cos 7.72t \right] \times 10^{-4}$$

$$= 0.572 + 1.679 e^{-0.071t} \cos (0.553t - 108.4°) + 4.584 e^{-5.26t} \cos (7.72t - 249.0°) \times 10^{-4} \qquad (2\text{-}51)$$

2-2.4 FREQUENCY-RESPONSE TECHNIQUES

It is often important to find the output response x of a system to a sinusoidal input y. For a sinusoidal input, $A_y \sin(\omega t + \phi_y)$, the output of the system will also be sinusoidal, after the transients have died out, i.e., $A_x \sin(\omega t + \phi_x)$. The amplitude and phase angle of the output relative to the input are then dependent only upon W(s), the transfer function of the system, and can be determined by letting $s = j\omega$ in the transfer function, where ω is the frequency (in rad/sec) of the input sinusoid. The ratio of output amplitude to input is then given by

$$\frac{A_x}{A_y} = \left| W(j\omega) \right| \qquad (2\text{-}52)$$

where A_x is the output amplitude, A_y is the input amplitude, and $W(j\omega)$ is the transfer

function of the system evaluated for real frequencies. The phase angle of the output ϕ_x relative to the phase angle of the input ϕ_y is given by

$$\phi_x - \phi_y = \angle W(j\omega) \qquad (2\text{-}53)$$

where $\angle W(j\omega)$ is the argument (phase angle) of the transfer function.

When the transfer function of a system is evaluated as a function of frequency for a sinusoidal input, the complex function that results is called the frequency response of the system.

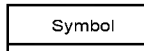
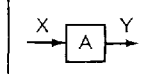
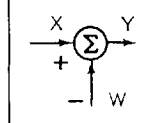
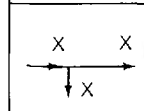
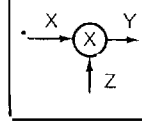
2-2.5 BLOCK DIAGRAMS AND SIGNAL-FLOW GRAPHS

2-2.5.1 Block Diagrams

Eqs. 2-20 and 2-22 demonstrate that, with zero initial conditions, the transform of the output of a system can be expressed in terms of the input transform and the system function. The system function can be thought of as an operator, i.e., the system function operates on the input transform to produce the output transform. In a similar manner, the system operates on the input to produce the output in the time domain, the operation being defined by the convolution integral and depending only upon the impulse response of the system. The concept of an operator is presented pictorially by the technique shown as operational block diagram algebra. The block diagram of a system is the pictorial representation of the mathematical operations involved in the differential equations that describe the system.

Table 2-2 presents a list of symbols used in the block-diagram representation of a system and Fig. 2-2 summarizes some of the reductions that enable one to simplify or reduce the block diagrams of a system. Since the block diagram contains no more information than the differential equations, the manipulation of a block diagram is merely a pictorial process of manipulating the differential equations. The advantage of a block-diagram representation is that the operational relations in a system are emphasized rather than the hardware. By becoming familiar with common block arrangements, the designer can interpret the function of various elements in a system much more rapidly than would be possible from an inspection of the differential equations.

TABLE 2-2. BLOCK-DIAGRAM SYMBOLS.

Symbol	Description	Operation
	variable	———
	operator	Y = AX
	summing point	Y = X − W
	splitting point	X = X
	multiplier	Y = XZ

Example. A servomotor drives an inertial load coupled to the motor through a flexible shaft as shown schematically in Fig. 2-3. The transformed equations of this system are

$$T_m = (J_m s^2 + f_m s)\theta_m + K(\theta_m - \theta_L) \qquad (2\text{-}54)$$

and

$$K(\theta_m - \theta_L) = J_L s^2 \theta_L + T_L \qquad (2\text{-}55)$$

where

T_m = motor-generated torque
J_m = motor moment of inertia
f_m = motor damping
θ_m = angular displacement of the motor end of the shaft
K = shaft stiffness (spring constant)
θ_L = angular displacement of the load end of the shaft
J_L = load moment of inertia
T_L = externally applied load torque

and s is the complex frequency variable. The damping of the flexible shaft is assumed to be negligible. Draw the block diagram of the system and reduce the diagram, keeping the

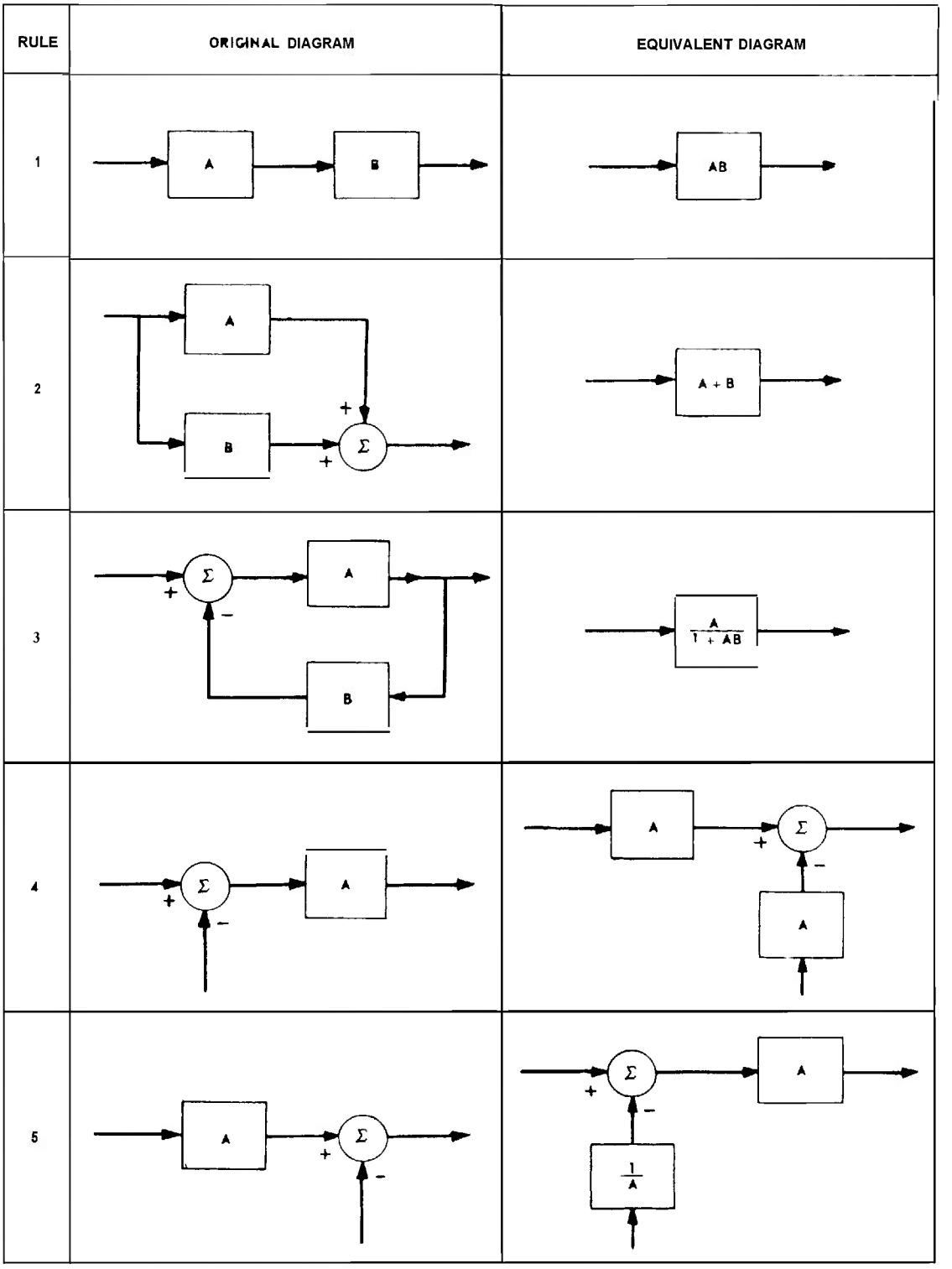


Figure 2-2. Block-diagram manipulation and reduction "rules". (Sheet 1 of 3)

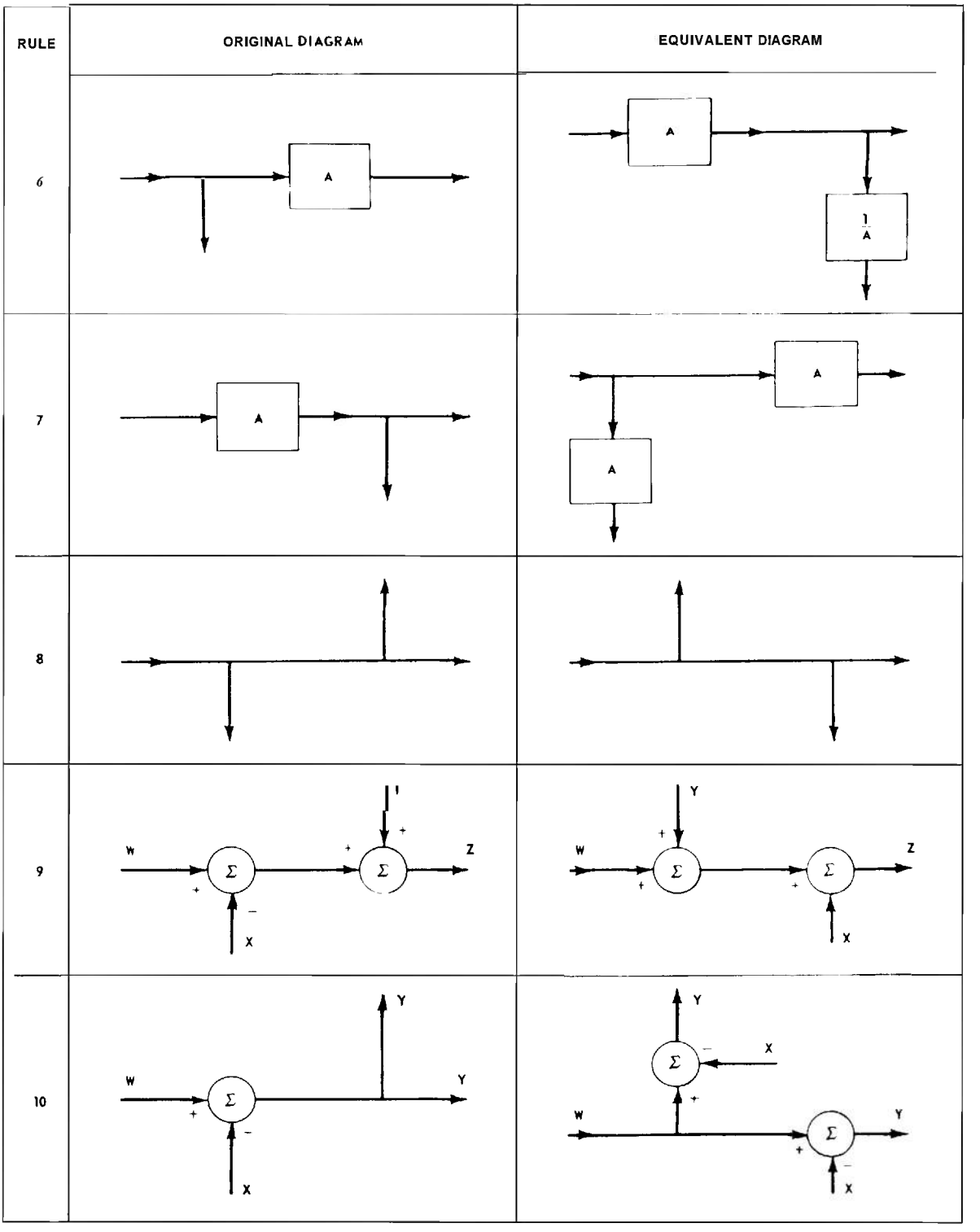


Figure 2-2. Block-diagram manipulation and reduction "rules". (Sheet 2 of 3)

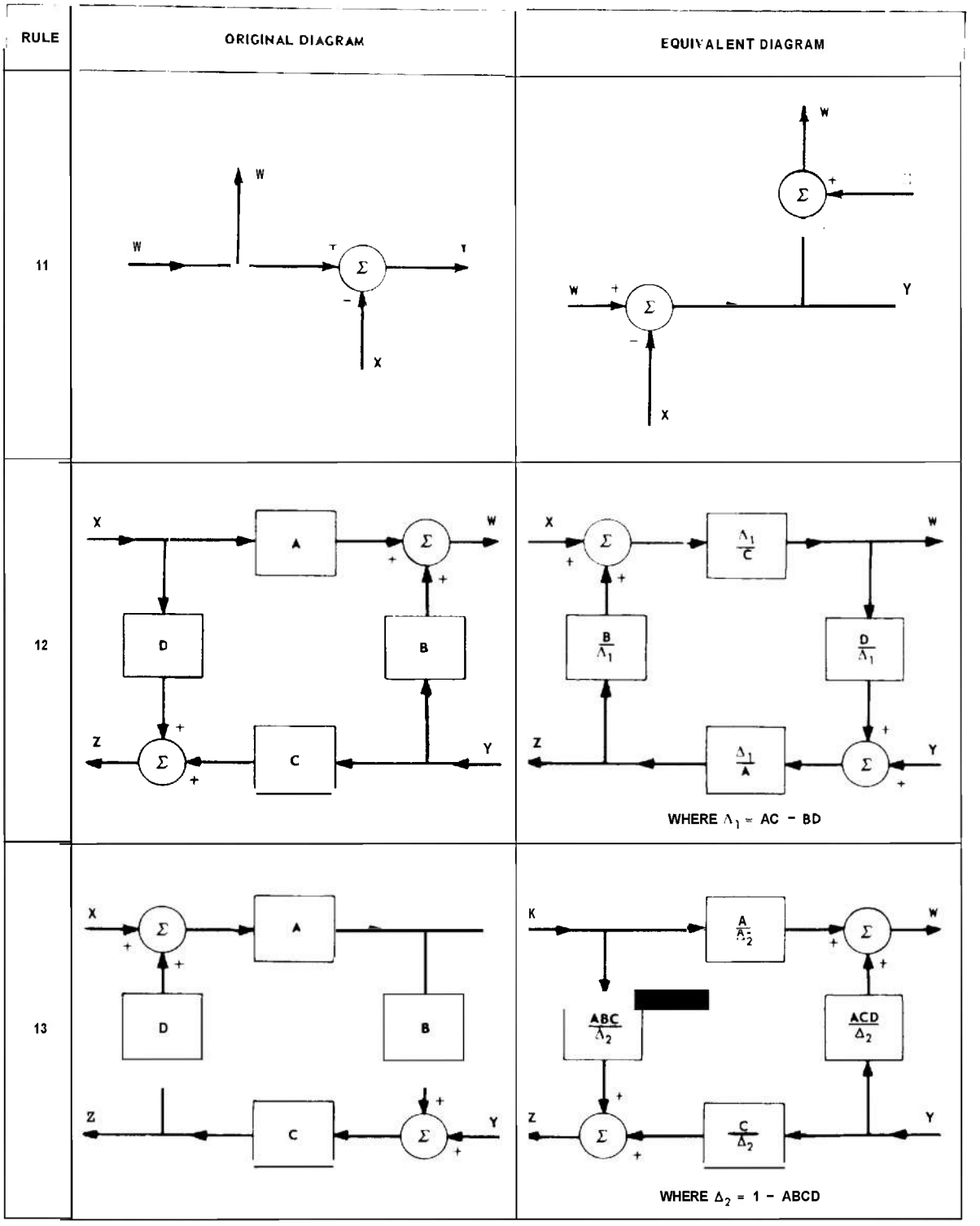


Figure 2-2. Block-diagram manipulation and reduction "rules". (Sheet 3 of 3)

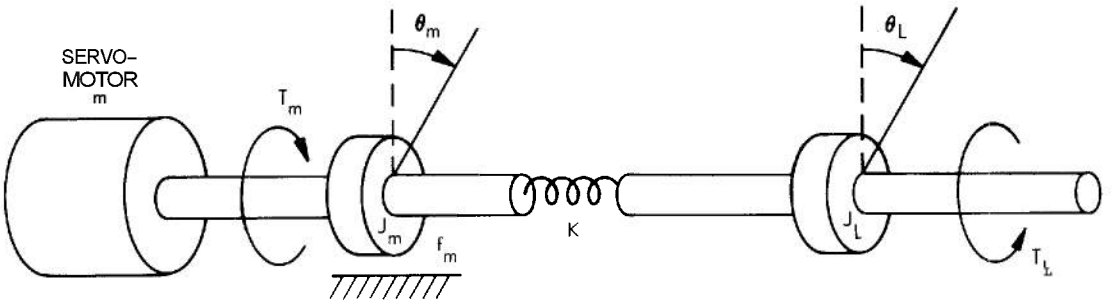


Figure 2-3. Mechanical schematic diagram of a servomotor coupled to an inertial load by means of a flexible shaft.

motor angle θ_m and the load angle θ_L in evidence.

Solution. The block diagram of the system is drawn in its "primitive" form in Fig. 2-4(A). The successive steps necessary to reduce the "primitive" diagram to the desired form are shown in Figs. 2-4(B) to 2-4(I), with the rules used for each step indicated below each step.

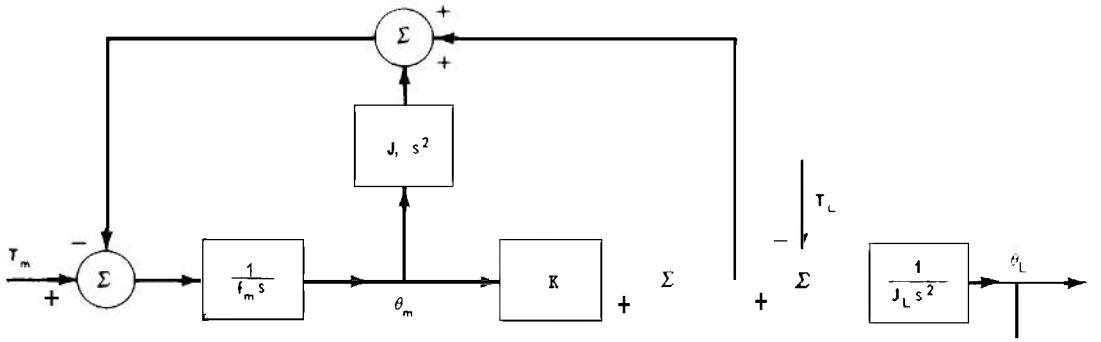
2-2.5.2 Signal-Flow Graphs[8, 9, 44]

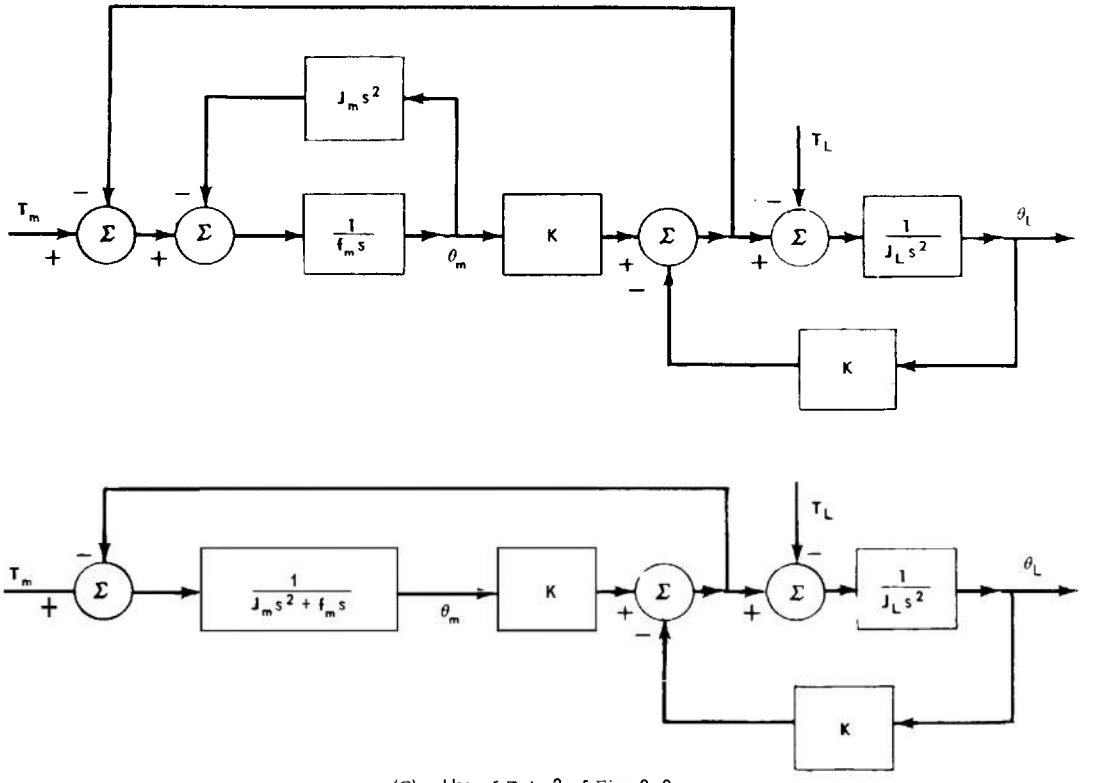
An alternative procedure for representing the differential equations of a system pictorially is Mason's signal-flow graph method. In a signal-flow graph, variables are represented by points called nodes and transfer functions are represented by directed lines or branches called transmittances. The distinction between the summing points and the splitting points of block-diagram algebra is eliminated in the signal-flow graph, The rules for drawing a signal-flow graph are as follows:

(a) Signals travel along branches only in the direction of the arrows.

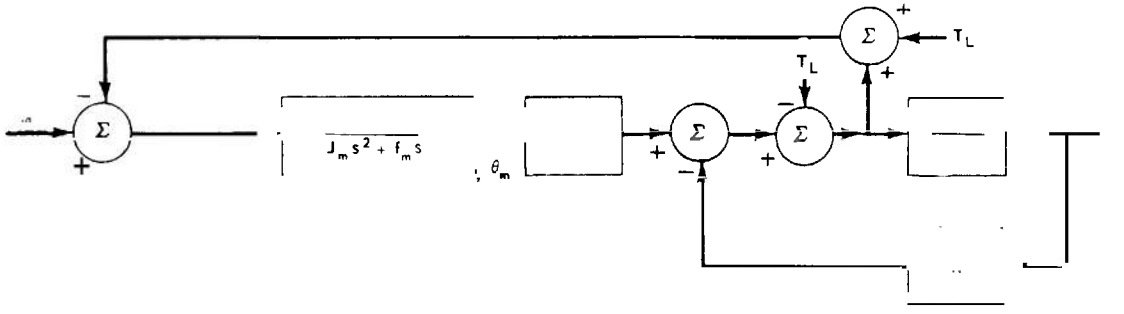
(b) A signal traveling along any branch is multiplied by the transmittance of that branch.

(c) The value of the variable represented by any node is the sum of all signals entering the node.





(C) Use of Rule 3 of Fig. 2-2



(D) Use of Rule 11 of Fig. 2-2

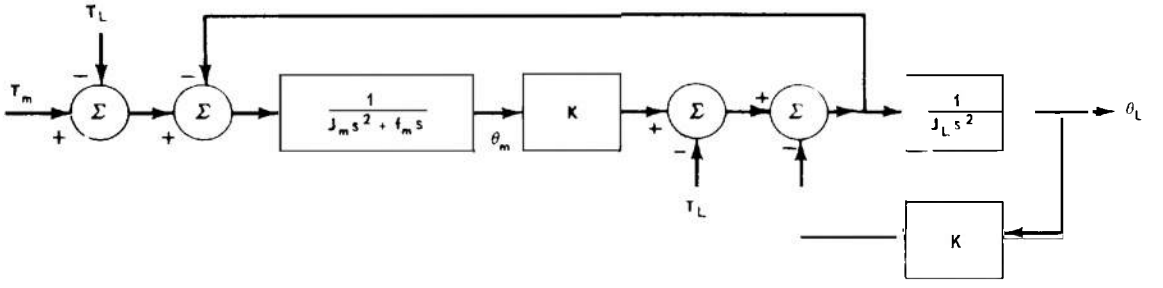
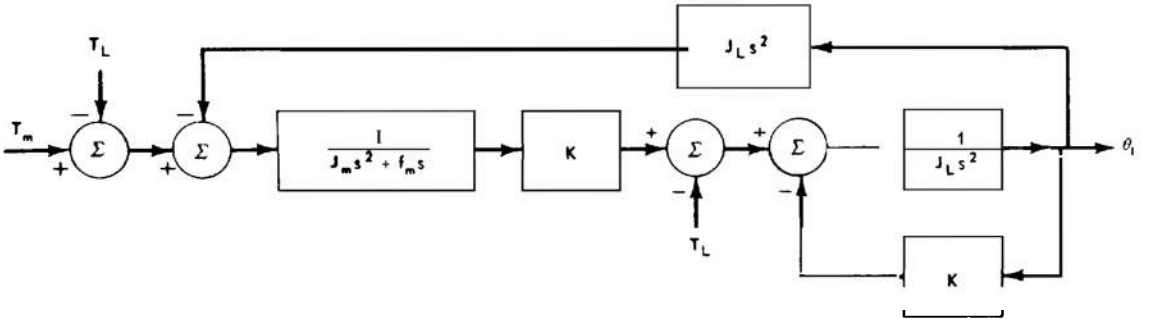
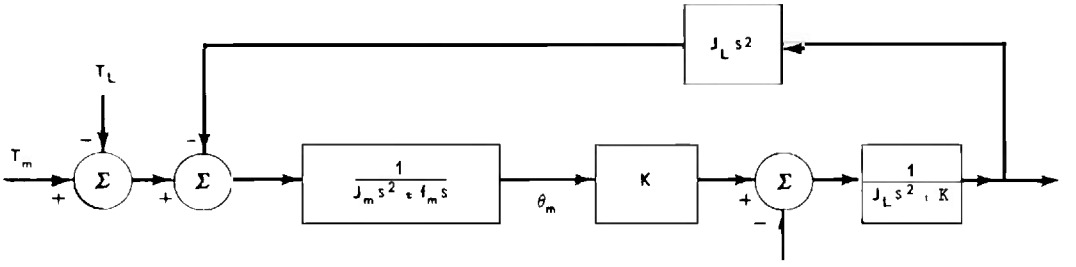
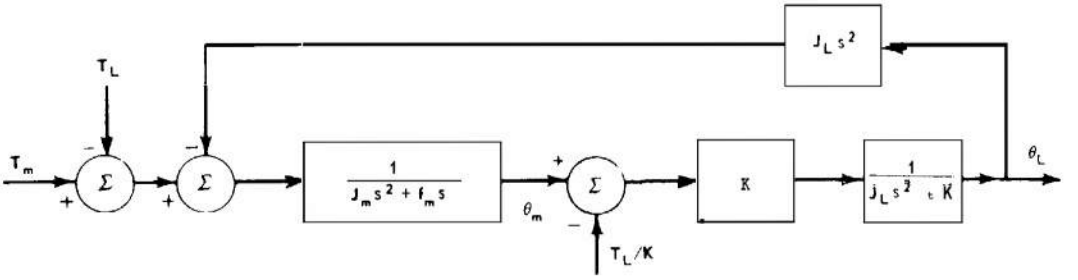


Figure 2-4. Block-diagram examples. (Sheet 2 of 3)

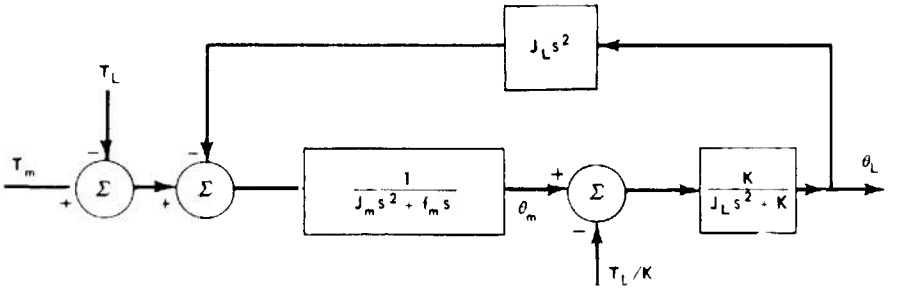


(F) Use of Rules 6 and 8 of Fig. 2-2





(H) Use of Rule 1 of Fig. 2-2



(I) Use of Rule 1 of Fig. 2-2

Figure 2-4. Block-diagram examples. (Sheet *3* of *3)*

(d) The value of the variable represented by any node is transmitted on all branches leaving that node.

Example. As an example of this procedure, the two equations

$$x_1 = t_{01}x_0 + t_{11}x_1 + t_{21}x_2 \qquad (2\text{-}56)$$

$$x_2 = t_{02}x_0 + t_{12}x_1 + t_{22}x_2 \qquad (2\text{-}57)$$

are represented by a signal-flow graph in Fig. 2-5.

For convenience, the signal-flow graph is usually drawn such that no branch enters an input node or leaves an output node. This is accomplished by introducing an additional node connected by a unity-transmittance branch to each input and output node as shown in Fig. 2-5, where the input node is assumed to be x_0 and the output node is assumed to be x_1.

The order of a signal-flow graph is a measure of the number of independent feedback loops and thus indicates the complexity of the system. The order of the signal-flow graph is the minimum number of essential nodes-- those nodes that must be removedto eliminate all feedback paths. A node is removed either by setting the variable associated with the node equal to zero or by deleting all branches leaving the node. Signal-flow graphs of orders one and two are shown in Figs. 2-6 and 2-7, respectively. The signal-flow graph of Fig. 2-5 is of order two, the essential nodes being x_1 and x_2.

The reduction of signal-flow graphs is accomplished by application of the following rules:

(a) Two parallel paths may be replaced by a single path with a transmittance equal to the sum of the two original transmittances (see Fig. 2-8).

(b) Two cascaded paths are equivalent to a single path with a transmittance equal to the product of the two original transmittances (see Fig. 2-9).

(c) The termination of a branch with transmittance t can be shifted one node forward by th? following steps (see Fig. 2-10):

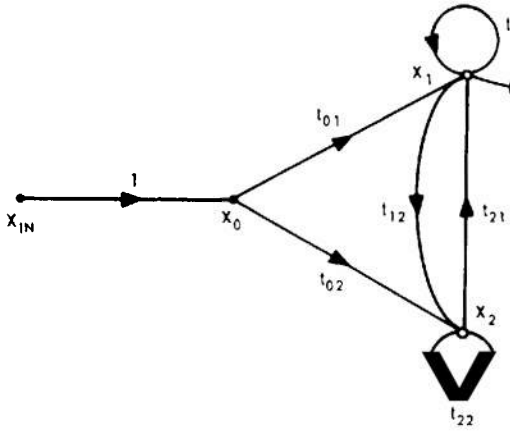


Figure 2-5. Signal-flow graph in three variables.

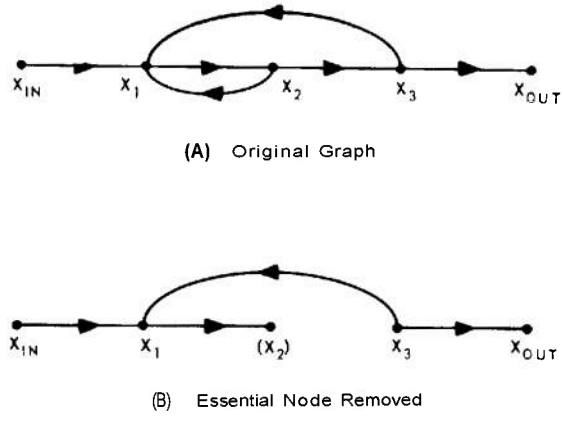


Figure 2-6. Signal-flow graph of order one.

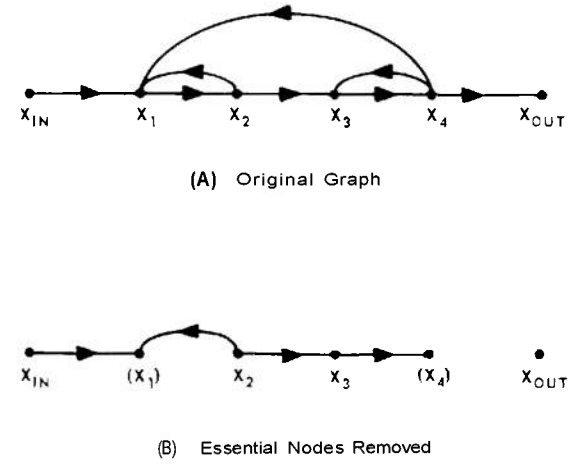
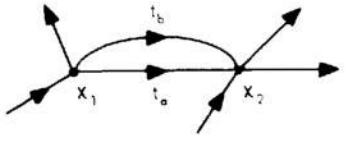
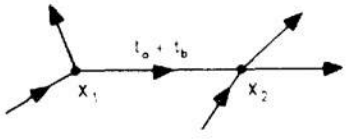


Figure 2-7. Signal-flow graph of order two.

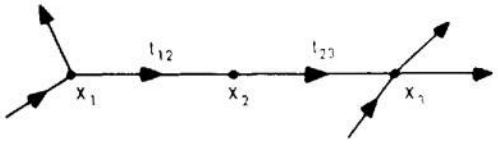


(A) Original Graph

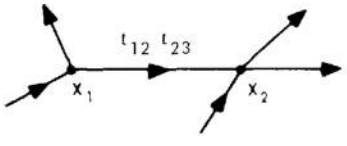


(B) Equivalent Graph

Figure 2-8. Signal-flow graph showing addition of parallel branches.



(A) Original Graph



(B) Equivalent Graph

Figure 2-9. Signal-flow graph showing multiplication of cascaded branches.

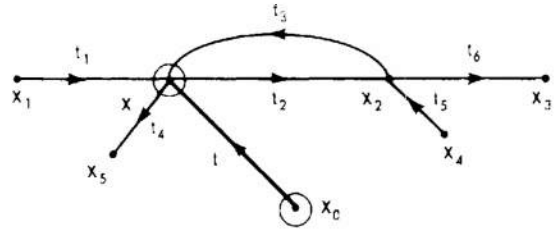
(1) Determine all the branches leaving the original terminating node x of branch t.

(2) Draw new branches from the starting node x_0 of branch t to the terminating nodes of all the branches leaving the terminating node x.

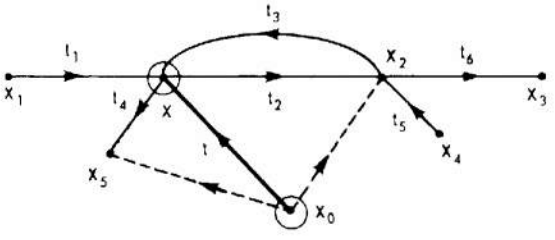
(3) To each of the new branches thus drawn assign a transmittance equal to the product of t times the transmittance from node x to the node on which the new branch terminates.

(4) Eliminate the original branch t.

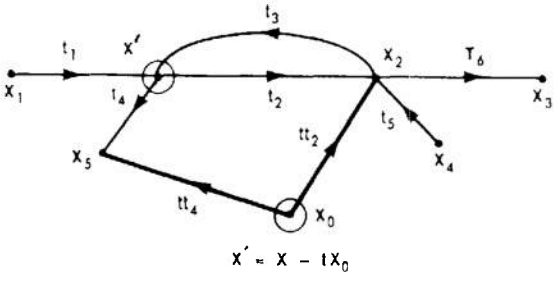
(5) Change the variable of the original node x to $x' = x - tx_0$.



(A) Original Graph - t to be Moved From x to x_2



(B) Steps (1) and (2) - Introduction of New Branches



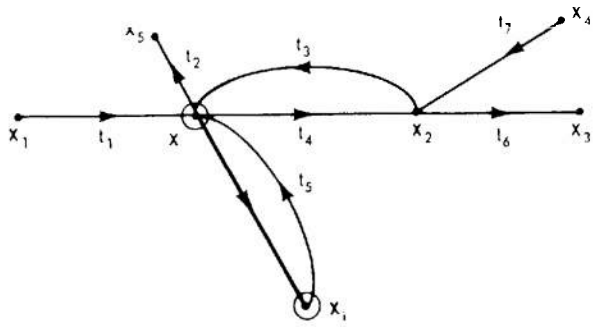
(C) Steps (3), (4), and (5) - Elimination of Old Branch; Labelling of New Branches, Change of Variable at Terminating Node of Old Branch

Figure 2-10. Signal-flow graph showing termination shifted one node forward.

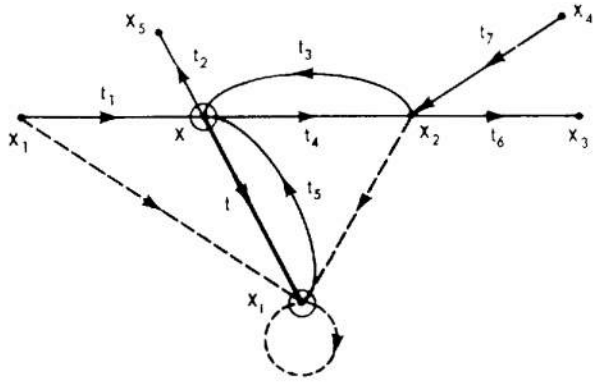
(d) The starting point or origin of a branch with transmittance t can be shifted one node backward by the following steps (see Fig. 2-11):

(1) Determine all the branches entering the original starting node x of branch t.

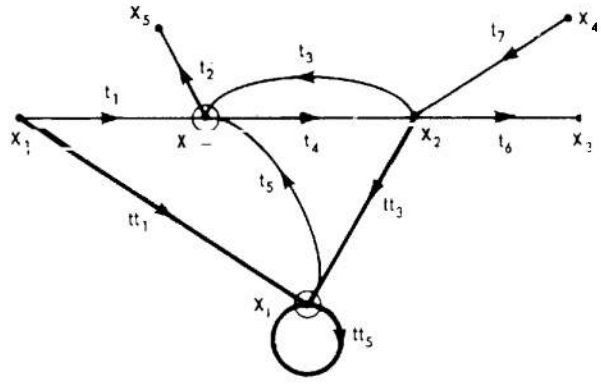
(2) Draw new branches from the starting nodes of all the branches entering starting node x to the terininating node x_i of branch t.



(A) Original Graph - t to be Moved From x to x_1



(B) Steps (1) and (2) - Introduction of New Branches



(C) Steps (3) and (4) - Elimination of Old Branch and Labelling of New Branches

Figure 2-11. Signal-flow graph showing origin shifted one node backward.

(3) To each of the new branches thus drawn assign a transmittance equal to the product of t times the transmittance from the node at which the new branch starts to node x.

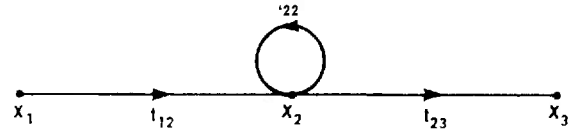
(4) Eliminate the original branch t.

(c) A self-loop with transmittance t of a node x can be removed by dividing the transmittances of all branches entering node x by (1 - t) and eliminating the loop (see Fig. 2-12; in this figure, $t \equiv t_{...}$, where the first subscript denotes the node on which the branch originates and the second subscript denotes the node on which the branch terminates). Note, in rule (c), that a self-loop is created at node x_0 for a branch starting from the terminating node x of branch t and ending on the starting node x_0 of branch t (Fig. 2-10 does not happen to have such a branch). In rule (d), a self-loop is created at node x_i for a branch starting from the terminating node x_i of branch t and ending on the starting node x of branch t.

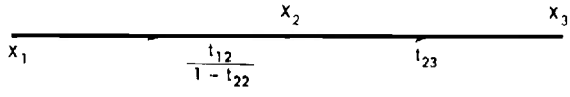
As an example of the reduction of signal-flow graphs, the various steps involved in reducing the second-order signal-flow graph of Fig. 2-5 are shown in Fig. 2-13.

2-2.6 STATISTICAL THEORY [10, 11]

The response r(t) of a linear system to a stochastic input cannot be expressed as a specific function of time. The only way to describe system behavior in the presence of stochastic inputs is in terms of the statistics of the input and the response. Theoretically, an infinite number of statistics is required to describe a stochastic process completely. Practically,' however, only a few statistics are used.

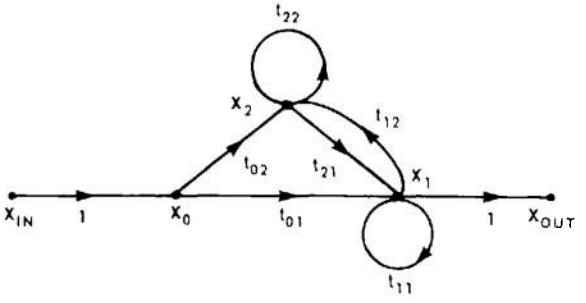


(A) Original Graph

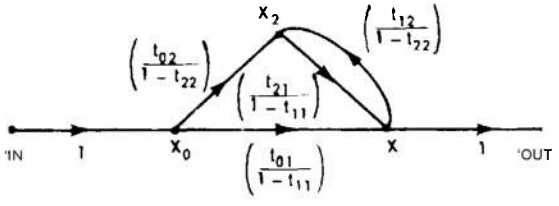


(B) Equivalent Graph

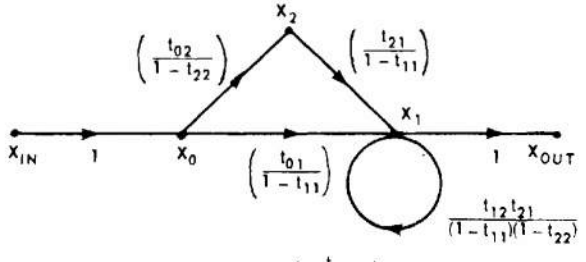
Figure 2-12. Signal-flow graph showing elimination of a self-loop.



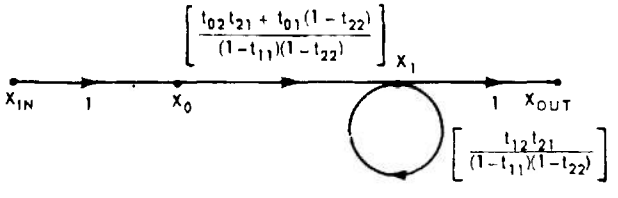
(A) Original Signal-Flow Graph (Second Order)



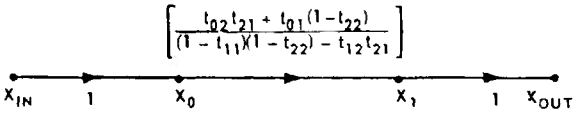
(B) .Reduction to First-Order Graph by Eliminating Self-Loops



(C) Movement of Branch $\left(\frac{t_{12}}{1-t_{22}}\right)$ Termination From Node x_2 to Node x_1



(D) Cascade and Parallel Branches Combined



(E) Reduction to Zero-Order Graph by Elimination of Self-Loop

Figure 2-13. Signal-flow graph showing reduction of second-order graph.

As discussed in Chapter 4 of Ref. 49, probability density functions are direct measures of the chance of occurrence of certain events in a stochastic process. The first probability density function of the stochastic variable r(t)* is denoted and defined as follows:

$p_1(r_1, t_1) \triangleq$ probability density function expressing the probability that the variable has a value r_1 at time t_1

Similarly, the second probability density function is denoted and defined as follows:

$p_2(r_1, t_1; r_2, t_2) \triangleq$ probability density function expressing the probability that the variable has a value r_1 at time t_1 and also a value r_2 at time t_2

In practice, only these first two probability density functions are used. For a stationary stochastic process, the first probability density function is independent of the time t_1, the second probability density function is a function only of the time difference $(t_2 - t_1)$.

Two commonly used probability density functions are the normal distribution and the Poisson distribution. The normal distribution is given by

$$p(r)\,dr = \frac{1}{\sigma\sqrt{2\pi}}\,e^{-\frac{1}{2}\left(\frac{r-\bar{r}}{\sigma}\right)^2} \qquad (2\text{-}58)$$

where $p(r)dr$ is the probability of finding r between r and $r + dr$, $\bar{r}$ is the mean value of r (to be defined below), and σ is the standard deviation of r (to be defined below). The Poisson distribution is given by

$$P(N, \Delta t) = \frac{(\nu\Delta t)^N e^{-\nu\Delta t}}{N!} \qquad (2\text{-}59)$$

* The stochastic response variable r(t) should not be confused with the radial quantity r in the polar coordinate system (r, θ, ϕ) employed in Chapter 4 of Ref. 49.

where $p(N, \Delta t)$ is the probability of finding N events in a time interval Δt, and ν is the average frequency of occurrence of the events.

In general, the average or mean value of a stochastic variable r is given by

$$\bar{r} \triangleq \int_{-\infty}^{+\infty} r\, p(r,t)\, dr \qquad (2\text{-}60)$$

For a stationary stochastic process, the mean value is independent of time and can also be found from

$$\bar{r} \triangleq \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{+T} r(t)\, dt \qquad (2\text{-}61)$$

The mean-square value of a stochastic variable or process is given by

$$\overline{r^2} \triangleq \int_{-\infty}^{+\infty} r^2\, p(r,t)\, dr \qquad (2\text{-}62)$$

For a stationary stochastic process, the mean-square value is also given by

$$\overline{r^2} \triangleq \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{+T} r^2(t)\, dt \qquad (2\text{-}63)$$

The root-mean-square (rms) value is the square root of the mean-squart value.

The variance v of a stochastic process is given by

$$v \triangleq \overline{|r - \bar{r}|^2} \qquad (2\text{-}64)$$

The standard deviation σ is the square root of the variance. It can be expressed in terms of the mean value and the mean-square value as follows:

$$\sigma^2 = \overline{r^2} - [\bar{r}]^2 \qquad (2\text{-}65)$$

In most applications, rms values and mean values are the most common statistics used. To aid in the determination of these quantities, statistics called correlation functions are used. The autocorrelation function $\phi_{rr}(\tau)$ of a stationary stochastic process r(t) is defined as the mean value of the product of the function r at time t by the function r at time $t + \tau$, i.e.,

$$\phi_{rr}(\tau) \triangleq \overline{r(t)\, r(t+\tau)} \qquad (2\text{-}66)$$

$$\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{+T} r(t)\, r(t+\tau)\, dt \qquad (2\text{-}67)$$

The crosscorrelation function $\phi_{ru}(\tau)$ between two stationary stochastic processes r(t) and u(t) is defined as the mean value of the product of the function r at time t by the function u at time $t + \tau$ i.e.,

$$\phi_{ru}(\tau) \triangleq \overline{r(t)\, u(t+\tau)} \qquad (2\text{-}68)$$

$$\lim \int r(t)\, u(t+\tau)\, dt \qquad (2\text{-}69)$$

From the definition of the autocorrelation function (Eq. 2-66), it is evident that the mean-square value of a stochastic process equals the value of the corresponding autocorrelation function with zero argument, i.e.,

$$\overline{r^2} = \phi_{rr}(0) \qquad (2\text{-}70)$$

Useful properties of the correlation functions are as follows:

(a) $\phi_{rr}(\tau) = \phi_{rr}(-\tau)$ [even function] (2-71)

(b) $$|\phi_{rr}(\tau)| \leq \phi_{rr}(0) \qquad (2\text{-}72)$$

(c) $$\lim_{\tau\to\infty} \phi_{rr}(\tau) = 0 \qquad (2\text{-}73a)$$

$$\lim_{\tau\to 0} \phi_{rr}(\tau) = \overline{r^2(t)} \qquad (2\text{-}73b)$$

$$\lim_{\tau\to 0} \phi_{rr}(\tau) > |\phi_{rr}(\tau)| \quad \text{for } \tau \neq 0 \qquad (2\text{-}73c)$$

[i.e., the maximum always occurs at $\tau = 0$]

(d) $$\phi_{ru}(\tau) = \phi_{ur}(-\tau) \qquad (2\text{-}74)$$

(e) $$|\phi_{ru}(\tau)| \leq \sqrt{\phi_{rr}(0)\, \phi_{uu}(0)} \qquad (2\text{-}75)$$

(f) $$\lim_{\tau \to \infty} \phi_{ru}(\tau) = 0 \qquad (2\text{-}76a)$$

$$\lim_{\tau \to 0} \phi_{ru}(\tau) = \overline{r(t)\, u(t)} \qquad (2\text{-}76b)$$

A few examples illustrating the use of autocorrelation functions follow. If r(t) is a rectangular wave with values $+\beta$ or $-\beta$ and with zero crossings located at even points that are Poisson-distributed in time with an average frequency of ν, the autocorrelation function of the process is given by*

$$\phi_{rr}(\tau) = \beta^2 e^{-2\nu|\tau|} \qquad (2\text{-}77)$$

If r(t) is a rectangular wave with amplitude values distributed in any fashion and with zero crossings located at event points Poisson-distributed in time with an average frequency ν, the autocorrelation function of the process is given by

$$\phi_{rr}(\tau) = \sigma^2 e^{-\nu|\tau|} + \bar{r}^2 \qquad (2\text{-}78)$$

where σ is the standard deviation of the amplitude distribution, and r is the meanvalue of the amplitude distribution.

If r(t) is a train of identical finitepulses whose starting points are Poisson-distributed in time with average frequency ν, the autocorrelation function of the process (known as "shot noise") is given by

$$\phi_{rr}(\tau) = \nu \int_{-\infty}^{+\infty} f(t)\, f(t+\tau)\, dt + \bar{r}^2 \qquad (2\text{-}79)$$

where f(t) is the time variation or waveform of a single pulse and r is given by

$$\bar{r} = \nu \int_{-\infty}^{+\infty} f(t)\, dt \qquad (2\text{-}80)$$

If r(t) is pure or "white" noise, the autocorrelation function is given by

$$\phi_{rr}(\tau) = \gamma\, \delta_0(\tau) \qquad (2\text{-}81)$$

where y is a constant that depends on how the process is generated and $\delta_o(\tau)$ is a delta function whose value is unity at $\tau = 0$ and is zero for all other values of τ. Thus, if "white" noise is considered as a limiting case of shot noise generated by exponential pulses of amplitude A and time constant T (where the amplitude approaches infinity and the time constant approaches zero with the area S under the pulse held constant), then the constant y is given by

$$\gamma = \frac{\nu S}{2} \qquad (2\text{-}82)$$

where ν is the average frequency of occurrence of the pulses.

Because the correlation functions are completely defined as functions of a time variable τ, they are Fourier transformable. By convention, $1/2\pi$ times the Fourier transform of a correlation function is called a power spectrum or a power density spectrum. Thus, the power-density spectrum $\Phi_{rr}(s)$ of a stochastic process is defined as

$$\Phi_{rr}(s) \triangleq \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-s\tau} \phi_{rr}(\tau)\, d\tau \qquad (2\text{-}83)$$

The cross-power density spectrum between two stochastic processes r(t) and u(t) is defined as

$$\Phi_{ru}(s) \triangleq \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-s\tau} \phi_{ru}(\tau)\, d\tau \qquad (2\text{-}84)$$

Given the power spectra, the corresponding correlation functions can be found by inverse transformation, i.e.,

$$\phi_{rr}(\tau) = \frac{1}{j} \int_{c-j\infty}^{c+j\infty} \Phi_{rr}(s)\, e^{s\tau}\, ds \qquad (2\text{-}85)$$

$$\phi_{ru}(\tau) = \frac{1}{j} \int_{c-j\infty}^{c+j\infty} \Phi_{ru}(s)\, e^{s\tau}\, ds \qquad (2\text{-}86)$$

* The derivation of Eq. 2-77 is too lengthy to repeat here. See page 221 of Ref. 45 for a complete derivation.

In terms of the power-density spectrum, the mean-square value of a stochastic process can be found by evaluating the following integral:

$$\int_{-\infty}^{+\infty} \Phi_{rr}(s)\, ds \qquad (2\text{-}87)$$

Useful properties of the power spectra are

$$\Phi_{rr}(s) = \Phi_{rr}(-s) \text{ (even function)} \qquad (2\text{-}88)$$

$$\Phi_{ru}(s) = \Phi_{ur}(-s) \qquad (2\text{-}89)$$

With some of the statistics of stationary stochastic processes having been established, the response of a linear system to a stochastic input can now be described. If $\phi_{rr}(\tau)$ is the autocorrelation function of the input r(t) of a linear system whose impulse response is w(t), the autocorrelation function of the output c(t) is given by*

$$\phi_{cc}(\tau) = \int_{-} dt_1\, w(t_1) \int_{-} dt_2\, w(t_2)\, \phi_{rr}(\tau + t_1 - t_2) \qquad (2\text{-}90)$$

The crosscorrelation function between the input and the output is given by

$$\phi_{rc}(\tau) = \int_{-\infty}^{+\infty} dt\, w(t)\, \phi_{rr}(\tau - t) \qquad (2\text{-}91)$$

which can be recognized as a convolution integral.

Extending the description of the stochastic response of a linear system to the frequency domain, if W(s) is the transfer function of the system and $\Phi_{rr}(s)$ is the input power-density spectrum, the output power-density spectrum is given by

$$\Phi_{cc}(s) = W(s)\, W(-s)\, \Phi_{rr}(s) \qquad (2\text{-}92)$$

The cross-power-density spectrum between input r(t) and output c(t) is given by

$$\Phi_{rc}(s) = W(s)\, \Phi_{rr}(s) \qquad (2\text{-}93)$$

or

$$\Phi_{cr}(s) = W(-s)\, \Phi_{rr}(s) \qquad (2\text{-}94)$$

If $\mu(t)$ is another signal and $_{\mu r}(s)$ is the cross-power-density spectrum between $\mu(t)$ and the input r(t), the cross-power-density spectrum between $\mu(t)$ and the output c(t) is given by

$$\Phi_{\mu c}(s) = W(s)\, \Phi_{\mu r}(s) \qquad (2\text{-}95)$$

or

$$\Phi_{c\mu}(s) = W(-s)\, \Phi_{r\mu}(s) \qquad (2\text{-}96)$$

In summary, once the properties of a stochastic process are expressed in terms of correlation functions, the analysis of system behavior is a straightforward problem that can be treated through the use of the definitions and properties of the correlation functions and their transforms, the power spectra. In particular, where rms values are of interest, Eqs. 2-70 and 2-87 are of great use.

2-2.7 NONLINEAR ANALYSIS [12–40]

2-2.7.1 General

All of the techniques of system analysis discussed in previous paragraphs of this chapter are restricted in their application to linear, time-invariant systems. This linearity restriction imposes two limitations on design. First, components must be of high quality if they are to operate in a linear manner when amplitudes and frequencies of signals vary widely. Second, the linearity restriction limits the realizable system characteristics, the types of systems, and the tasks that can be acomplished.

Whereas techniques for the analysis and synthesis of linear time-invariant systems

* See pages 331 and 332 of Ref. 45 for the derivation of this relationship.

are well established and generally adequate to handle most of the problems met in practice, this happy situation does not exist in the case of nonlinear or time-varying systems. A number of techniques are available that give more or less satisfactory results, but no really unified general theory for nonlinear systems exists -- and it is doubtful that it will for many years to come, if ever. Many quite ordinary situations exist for which there are no really satisfactory solution techniques. These factors make the analysis of nonlinear systems very interesting, but sometimes very frustrating.

Before proceeding further, it is in order to define specifically what is meant when a system is termed nonlinear. Unfortunately, this is not easily done. In fact it is necessary to look first at the definition of a linear system and then proceed from there.

The most fundamental characteristic of a linear system is that it obeys the principle of superposition. This principle can be stated in the following terms: The total response of a linear system is the sum of the responses due to all the applied inputs acting individually because each applied input produces a response independent of the response to any other applied input. This same criterion for linearity applies whether or not the system parameters are time varying. Mathematically, a system is linear if the expression relating the input and output variables involves only first powers of the input and output variables and their derivatives.

This principle is usually stated as follows: If an excitation A, produces an effect B, and an excitation A, produces an effect B_2 when each is applied independently, then the system is linear providing that for the simultaneous application of A, and A, in any proportion the effect is made up of B, plus B_2 in the same proportion. Thus,

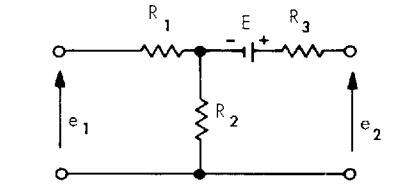
$$\text{if} \quad k_1 A, (t) \rightarrow k, B, (t)$$

$$\text{and} \quad k_2 A_2 (t) = k_2 B_2 (t)$$

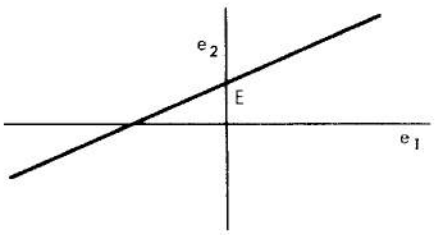
$$\text{then} \quad k_1 A_1 (t) + k_2 A_2 (t) \rightarrow k_1 B_1 (t) + k_2 B_2 (t)$$

At first glance, it might apptar that the output-input relationship for the circuit of Fig. 2-14(A) violates this definition of a linear circuit, whereas one certainly has the firm conviction that such a circuit must be linear since it includes only linear resistors and a battery. Consideration of the output-vs-input curve of Fig. 2-14(B) shows that a simple change in variable would translate the curve to the origin and that in terms of this new variable the definition of superposition as given is indeed valid. It is necessary to recognize this possibility as it is the basis for the study of nonlinear systems by piecewise linear techniques.

Consider for a moment what the consequences are of being fortunate enough to be dealing with a linear system. In addition to the fact that the mathematics associated with linear systems are relatively simple, it should be noted that linear systems allow great freedom for the experimentalist. A truly linear system can be tested with any one of a variety of convenient test signals such as impulses, steps, or sinusoids. Furthermore, the observed system characteristics are independent of the amplitude of the test signal used. Unfortunately, no real physical system is entirely linear and, as a result, attempts to increase the linear range of operation of a



(A) Simple Circuit



(B) e_2 vs e_1 for Simple Circuit

Figure 2-14. A simple circuit and its associated input-output relationship.

system usually lead to a requirement for components with larger power ratings or higher quality. Consequently, in spite of the attractiveness of linear systems from the analysis point of view, the designer is becoming increasingly interested in nonlinear systems -- first, because he is unable to build systems that operate entirely in the linear range, and second, because he can obtain a more satisfactory solution to some problems by the intentional introduction of nonlinear components in a system. A contactor servo might be thought of as a typical example.

Several additional characteristics possessed by a linear constant-coefficient system should be noted. First, the output of a linear constant-coefficient system cannot contain components at frequencies not present in the input. Second, the question of stability is clearly defined and the stability or instability of a system is not dependent on the driving function or any initial conditions. For the general nonlinear systems, however, neither the principle of superposition nor these other characteristics are valid.

Linear systems with time-varying coefficients represent an intermediate case. The principle of superposition can be extended to include this type of system but, on the other hand, it may not be possible to obtain a simple answer to system stability. In fact, the question of stability may have no significance.

The analysis of constant-coefficient linear systems is relatively simple and a variety of techniques has been developed for handling such systems. During the past ten years, transform techniques (see par. 2-2.3.1) have come into wide usage for analyzing constant-coefficient linear systems. In fact, once a correct mathematical representation has been obtained for a constant-coefficient linear system, the use of transform techniques reduces the problem of determining the response of the system to a simple input to a cookbook type of problem.

For time-varying systems, the concepts of operational mathematics still are valid, but the details involved in obtaining answers to specific problems usually become either very involved or impossible to carry out. For nonlinear systems, this whole concept must be discarded because here the principle of superposition no longer applies and application of operational techniques implies validity of the principle of superposition.

One might ask at this point, "Why all the discussion of linear systems when what is really of interest is the definition of a nonlinear system?" The answer is simply that the definition of a nonlinear system is really a negative one. A nonlinear system is simply defined as any system that does not obey the principle of superposition.

As a practical matter, most systems are linear only by assumption, but this assumption leads to a tremendous simplification in the problem of analyzing or synthesizing a system and thus is extremely important. One should not jump to the conclusion, however, that linear systems are good and nonlinear systems are bad. The basic characteristics of many important systems are realized only because some elements in these systems are nonlinear.

In spite of the fact that determination, or even specification, of the performance of nonlinear systems is apt to be rather difficult, control system engineers are becoming more and more interested in this class of systems either because they are confronted with systems that contain nonlinearities they cannot (or cannot afford to) remove, or because they feel that there is a good possibility that they could devise a nonlinear system that would achieve a desired end either more cheaply or more reliably than a linear system.

Basically, the methods that have been developed for analyzing nonlinear systems can be divided into the following three main categories:

1. Methods that can be carried out by an analyst having at his disposal only the ordinary analytic tools
2. Numerical techniques and methods involving the use of modern computers
3. Methods based on extensive experimentation with an actual system

The methods of Category 1 can be further subdivided as follows:

1. Analytic and Quasi-Analytic Techniques

a. Direct solution of nonlinear differential equations
b. Variation-of-parameters technique
c. Piecewise linearization
d. Series solution
e. Perturbation theory
f. Describing-function methods
   (1) Applied to systems with deterministic inputs
   (2) Applied to systems with random inputs

2. Graphical Techniques
   a. Graphical integration
   b. Isocline method
   c. Phase-plane method
   d. Phase-space method

In addition to the foregoing, there are various techniques that have been developed for investigating the stability of nonlinear systems.

Refs. 12 through 36 should be consulted for detailed information concerning these various methods and techniques.

2-2.7.2 Nonlinearities Found in Many Control Systems

The paragraphs which follow describe several types of nonlinearities that are frequently encountered in control-systems work. In addition, some of the system performance characteristics that are uniquely attributable to the presence of a nonlinearity are noted.

2-2.7.2.1 Limiting

The saturation or limiting type of nonlinearity shown in Fig. 2-15 is frequently met in control-systems work. For small signals, the effect or output is proportional to the cause or input, but for signals greater than a critical value, the output ceases to be proportional to the input and finally remains essentially constant no matter how large the input. The solid curve in Fig. 2-15 represents what is sometimes referred to as soft limiting, while the dotted curve represents sharp limiting. In the first case, a smooth transition occurs between the linear and the saturated regions, while in the second this transition occurs abruptly.

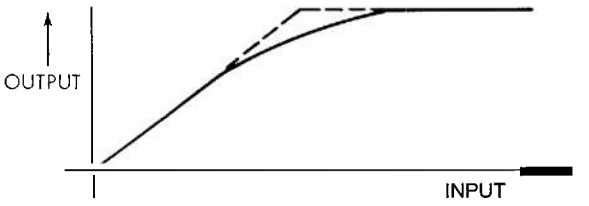


Figure 2-15. Plot depicting the limiting type of nonlinearity.

2-2.7.2.2 Dry Friction

Dry or Coulomb friction is a friction force that is constant in magnitude, regardless of the relative velocity of the moving parts, but reverses sign when the velocity changes sign. This type of friction can be represented as shown in Fig. 2-16. Some Coulomb friction is present in any mechanical system. In those systems that operate with a high nonlinear-friction effect, accurate analysis should include this nonlinear effect. In a well-lubricated system, however, the friction will be approximately proportional to the velocity and thus will not introduce a nonlinearity. This latter type of friction is generally referred to as viscous friction.

2-2.7.2.3 Hysteresis

Hysteresis is a complex type of nonlinearity in which the response of an element is determined by past history as well as by the instantaneous value of the excitation. Fig. 2-17 illustrates this effect, which occurs in electromagnetic circuits and in mechanical devices (such as strain gages and pressure transducers) that utilize materials for which

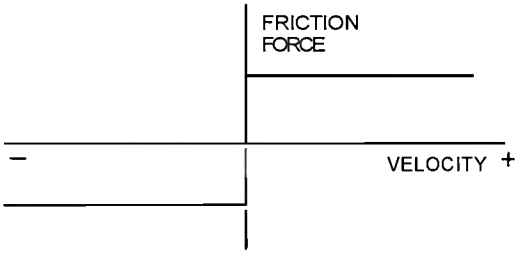


Figure 2-16. Graphical representation of Coulomb friction..

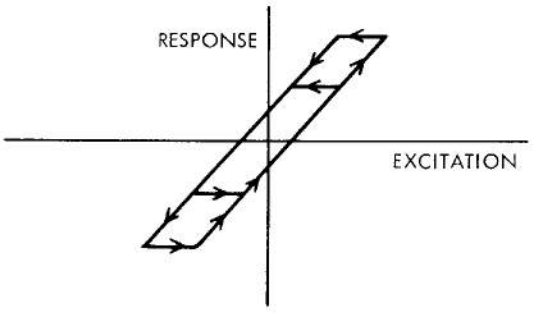


Figure 2-17. Graphical representation of hysteresis.

the stress-strain relationship is determined by the history of strain. Backlash, such as occurs in gearing and mechanical linkages, is somewhat related to hysteresis. Analysis of systems containing backlash is complicated by the fact that changes in the inertia distribution between the driving and driven members lead to significant changes in the influence of the backlash.

2-2.7.2.4 Relays

Relays are used in many control systems because they provide a simple means for realizing a very high amplification. However, the relay is a discontinuous-type amplifier. The simplest representation of such a device is shown in Fig. 2-18. For inputs of magnitude less than A, the output is zero. A positive input greater than A is transformed into a fixed positive output, and a negative input whose magnitude exceeds A is transformed into a fixed negative output. The region from -A to +A is termed "dead-space".

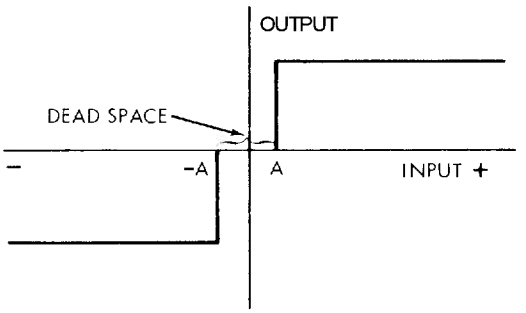


Figure 2-18. Graphical representation of a relay with dead-space but no hysteresis

A more complete representation of a relay would include both dead-space and a hysteresis effect to take into account the fact that the voltage required to switch the relay from the nonenergized position to the energized position is somewhat higher than that at which the relay switches back from the energized to the nonenergized state.

An even more complete model of a relay would include a time delay to account for (1) the fact that the inductance of the relay coil causes the control current to lag behind the applied control voltage, and (2) the time required for the armature to move from one position to the other.

2-2.7.2.5 Diodes

Diodes represent another type of nonlinear device that the control-systems designer may wish to use in order to protect equipment from excessive signals or to achieve special effects. An ideal diode offers zero resistance to the flow of current for one polarity of applied voltage but infinite impedance to the flow of current for the opposite polarity of applied voltage. For many purposes, practical diodes can be treated as though they are ideal.

2-2.7.2.6 Orifices

In one class of hydraulic control systems, the flow of hydraulic fluid in the system is controlled by a valve that consists of several variable orifices. For the case of a sharp-edged orifice, which can usually be assumed in a spool or flapper type of valve, the rate of fluid flow through the valve is proportional to the area of the orifice and to the square root of the pressure drop across it. Because of this basic characteristic, a complete hydraulic valve may insert a significant nonlinearity into a system.

2-2.7.2.7 Products and Transcendental Functions

Control systems are made nonlinear not only by the types of nonlinearities just described but also by the presence of components or of arrangements that introduce

products or powers of the dependent variables or their derivatives. The presence of transcendental functions of the dependent variable also leads to a nonlinear equation because such functions can be expanded as a series of terms of progressively higher powers.

A typical example of a system whose mathematical description involves powers of the dependent variable is that of a mass attached to a nonlinear spring. As a first approximation, this nonlinear spring might be described by the relationship

$$\text{FORCE} = k(1 \pm a^2x^2)\,x \qquad (2\text{-}97)$$

in which k and a are constants describing the spring and x is the deflection. A plus sign would be used in Eq. 2-97 to represent a spring that effectively becomes stiffer as it is deflected while the minus sign would represent a spring that becomes weaker as it is deflected. In this latter case, the mathematical model of Eq. 2-97 applies only for small deflections since for $|x| > \frac{1}{a}$ the force reverses sign.

The differential equation that describes the motion of a constant mass M attached to a spring described by Eq. 2-97 is given by the equation

$$M\frac{d^2x}{dt^2} + k(1 \pm a^2x^2)\,x = 0 \qquad (2\text{-}98)$$

where it is assumed that no friction exists. For nonzero values of a, Eq. 2-98 involves the cube of the dependent variable and is thus a nonlinear differential equation. However, this particular type of differential equation has been studied extensively and its solution can be obtained in the form of elliptic functions.

2-2.7.3 Classification of Nonlinear Systems

The definition of a nonlinear system given in par. 2-2.7.1 was negative in that it did not describe a nonlinear system but, instead, relegated all systems that did not meet the very specific test for linearity to the category of nonlinear systems. This rather unsatisfactory approach is taken because no really good scheme has been devised for classifying nonlinear systems. The present discussion has followed the plan of merely cataloging typical systems without trying to classify them. Examination of the nonlinearities described, however, indicates several schemes of classification that might be employed.

2-2.7.3.1 Continuous and Discontinuous Nonlinearities

From a mathematical point of view, it is sometimes desirable to distinguish between nonlinearities that can be described by continuous curves and those in which the output-vs-input relationship exhibits jumps. This method, then, would distinguish between a limiting type of nonlinearity and a relay.

2-2.7.3.2 Incidental and Essential Nonlinearities

A different scheme of classification might distinguish between (1) those nonlinearities that are introduced because the performance of supposedly linear physical devices deviates from the ideal as a result of mechanical tolerances or the characteristics of materials, and (2) those nonlinearities that the designer deliberately introduces into the system. This scheme, for example, would distinguish between (1) a system that is driven into the saturation region for very large signals but that normally operates in the linear region, and (2) a relay, which does not behave as a linear element for any amplitude of input signal.

2-2.7.3.3 Zero-Memory and Nonzero-Memory Nonlinearities

Another important characteristic of a nonlinearity is whether its instantaneous output is determined uniquely by the instantaneous input, in which case it would be termed a zero-memory or amnesic nonlinearity, or whether its instantaneous output is determined by the history of its inputs, in which case it would be called a nonzero-memory or nonamriesic nonlinearity. A relay with

hysteresis is a typical example of a nonzero-memory nonlinearity since, over a region, the output of the relay depends not only on the instantaneous value of the input but also upon the manner in which the input arrived at its present value.

2-2.7.3.4 Phenomena Peculiar to Nonlinear Systems *37–40*

Nonlinear systems lead to several special problems because they may exhibit phenomena that never occur in a purely linear system. *One* of the most frequently observed phenomena of this type is the limit cycle, an oscillation of fixed amplitude and period but arbitrary wave shape that may be excited under certain conditions. The motion of the escapement in a watch and the voltage in a vacuum-tube oscillator are typical examples of limit cycles. It is basically the nonlinearities in these systems that determine the amplitude of oscillations for, if the systems were actually linear in the ideal sense, the oscillations would grow to unlimited amplitude. Obviously, this would be physically impossible.

Another phenomenon observed in some nonlinear systems is that of self-excitation. This phenomenon can take either of two forms. Systems that break into oscillations when subjected to a very small input signal or disturbance are said to exhibit soft self-excitations. Such systems may become stable when the amplitude of the input signal is increased sufficiently. Hard self-excitation, on the other hand, is exhibited by a system that must be excited with signals of at least some minimum amplitude before it becomes unstable. Systems with quantizers may exhibit either of these types of self-excitation.

Still another peculiarity of nonlinear systems is that the frequencies of the output signal and of intermediate signals in the system are not necessarily the same as the frequency of the input signal. Thus, some nonlinear control systems exhibit subharmonic oscillations with the output oscillating at some odd-order subharmonic of the input frequency.

Another phenomenon that cannot occur in a strictly linear system is the appearance of discontinuous jumps in amplitude as the system excitation is continuously increased in amplitude. When this effect occurs, it is usually accompanied by a hysteresis, with the result that the jump occurs at a different amplitude for increasing signals than it does for decreasing signals.

2-3 SIMULATION TECHNIQUES

2-3.1 GENERAL

Later chapters of this handbook describe both digital and analog computing components, and the combination of such components into digital, analog, or hybrid computers. The paragraphs which follow outline the application of analog and digital simulation techniques for determining the performance characteristics of complex mathematical models.

2-3.2 ANALOG TECHNIQUES

In the process of arriving at a mathematical model for a system, the designer normally utilizes block diagrams as discussed earlier in this chapter (see par. 2-2.5 through par. 2-2.5.2) and again in Chapter 6. Fortunately, the programming of an analog computer follows quite simply as a detailed expansion of the block-diagram representation of a system. To make this expansion, the analyst must represent all operations indicated on the block diagram in terms of those operations that can be performed by the computer, namely: integration, addition, multiplication, and generation of arbitrary functions. Each transfer function in the block diagram must be expanded to show in detail its realization in terms of the basic analog elements. Fortunately, this is a straightforward task and represents no real problem.

After a complete representation has been developed in terms of computing components, appropriate scale factors must be worked out. Scaling involves two distinct problems. The first is concerned with the magnitudes of the variables in the problem and the second with the time the computer takes to obtain a solution. The computer will produce accurate results only if the variables in the computer are

substantially larger than the variations represented by noise in the computing elements. This noise may be broad-band thermal noise generated in resistors, shot noise generated in vacuum tubes, low-frequency noise related to slowly varying offsets in the output of amplifiers, or noise that arises from moving contacts--such as a potentiometer wiper moving over the resistance element of the potentiometer. Other sources of noise are ripple from the power supplies and noise picked up from disturbing sources completely external to the computer. In a well-designed computer, noise from these sources is usually small, with varying amplifier offsets representing tlie major limitation on accuracy.

At tlie other end of the scale, the accuracy of the computation suffers if the magnitude of any computer variable attempts to rise above a maximum set by the design of the element. For example, an amplifier may saturate and thus cease to follow the linear relationship desired between the voltage at its input and that at its output; or the input applied to a function generator may exceed the maximum value for which it was set up, with the result that the desired functional relationship is lost.

The maximum operating voltage used in the majority of the analog computers employing vacuum tube amplifiers is ± 100 volts. In order to achieve the maximum accuracy, the voltages appearing at all points in the computer should be as close to 100 volts as possible without ever exceeding this value. However, since the very nature of solutions usually involves large changes in the variables, some of them will usually approach zero during some parts of a solution. The value of very small variables cannot be determined with high accuracy and, if additional accuracy is required, it may be necessary to rescale the problem and rerun a portion of it.

The question of solution running time must also be considered before the task of programming the computer is completed. Some problems to be studied on the computer may represent physical situations in which the actions of interest take place in microseconds, while in others the time is measured in decades.

Depending on whether the computer is designed for so-called "real-time operation"' or "high-speed repetitive operation", the most satisfactory solution time will be in the range of 10 seconds to one minute for real-time computers or $\frac{1}{10}$ to $\frac{}{100}$ second for high-speed computers. In an analog machine, all elements operate in parallel, so the running time does not increase with the complexity of the problem being studied. The running time depends solely on the gatn of the integrators and may be changed by a factor such as 10 merely by changing the gain of each and every integrator employed by that factor.

Before one can obtain a solution on which to base the selection of scale factors in the computer, he must arrive at some tentative estimates and run a trial based upon these. If any of the signals exceed the maximum allowable or appear to be too small, new scale factors can be chosen and the solution rerun until an acceptable result is achieved.

2-3.3 DIGITAL TECHNIQUES [41]

The effectiveness with which digital computers can be utilized in the study of scientific problems depends as much upon the ease with which the analyst can communicate with the computer as upon the actual characteristics of the computing components of which the computer is made up. These two aspects of a digital computer are generally referred to as its software and its hardware.

In the early stages of digital computer technology, the only programming method available was what has now come to be referred to as machine-language programming. Under this system, the programmer was forced to keep a detailed bookkeeping record of the contents of each memory location and of each transfer of data from a memory location, to the arithmetic unit of the machine, and finally back into another storage location for later use if desired.

As more experience was gained with programming and as appropriate machine hardware changes became possible, symbolic programming techniques were developed. Under these, the programmer was required only to

identify each operation to be performed and each piece of data, but not to make detailed assignments of data to specific storage locations. The first step in obtaining the solution for a problem written in such a language is to have the machine analyze the symbolic program and by means of a compiler program translate the symbolic program into a machine language program.

The development of more and more *sophisticated* programming languages has received a great deal of attention over the past ten years and very powerful languages such as the FORTRAN series are now available. Nevertheless, the conventional approach to the use of the general-purpose computer is still to develop a library of programs, each program solving a specific or standard problem type. Yet, the variety of problem types and engineering situations is so great that the freedom of the engineer is severely restricted by the fixed program library. Ideally, one would like the ease of communication with the computer to be such that the engineer could quickly and economically write a unique program for each engineering situation as it occurs. For this to be feasible, the language for stating the solution must be very efficient, allowing the engineer to describe a solution in the same technical terms he would use in instructing a colleague of his own professional competence.

The development of such problem-oriented languages is now receiving a great deal of attention. One example is COGO (for COordinate Geometry) a system for use in civil engineering problems.

2-4 NUMERICAL TECHNIQUES

2-4.1 GENERAL

Digital computers deal with numbers and are capable of performing simple arithmetic operations at high speed and storing the results. Accordingly, the branch of mathematics known as numerical analysis, which is concerned with the numerical evaluation of mathematical functions and equations, has in recent years seen a great revival of interest and a considerable expansion of techniques.

The methods used for evaluating functions and solving equations in a digital computer may be generally classified as methods of successive approximations, or methods of substitution of an approximate expression for an exact expression. Such approximate expressions may be either power series or sets of tabular differences.

In the methods of successive approximations, or iteration, an approximate solution is substituted in the equation so as to yield a better approximation, and so on. Since the computation involves a closed loop, the possibility of instability exists. Iteration, when stable, is useful in the solution of equations and sets of equations, and in the evaluation of certain functions expressed as equations.

The impetus given to the field of numerical analysis by the computational capacity of the high-speed digital computer has led to the investigation of mathematical fields formerly neglected because of the computational difficulties involved. This, in turn, has led to the application of mathematical tools in new areas of engineering, science, and management. A typical example is the solution of large sets of linear algebraic equations. As is discussed in par. 2-4.6, such sets of equations can frequently be solved by iterative methods. Since such equation sets are usually expressed in the shorthand matrix notation, the method is commonly known as "matrix inversion". The inversion of very large matrices is now practicable with the aid of high-speed digital computers.

Certain logistics problems of the armed services and of large corporations can be expressed mathematically by an operations research technique known as "linear programming". Such factors as the size and location of warehouses, the production capacity of suppliers, and the cost/time characteristics of alternative transportation systems are expressible in terms of sets of linear algebraic equations. These sets of equations can be manipulated by a digital computer so as to achieve an optimum solution in terms including cost or delivery time.

Similar methods applied to the solution of sets of simultaneous linear differential equations have proved equally powerful in the investigation of engineering problems. The problem of the flutter of an aircraft wing is

a typical example. Here, the structural dynamics are expressible by a set of differential equations with many coupling terms and with excitation at numerous points of the set.

The ability of the digital computer to store or compute rapidly the values of a function provides a capability of particular value to the fire control field. Except for trigonometric functions where a geometrical analog is available, generation of functions in an analog computer has been principally accomplished by such inflexible methods as mechanical cams and function potentiometers. Methods to be outlined in par. 2-4.2 offer means of generating analytical or empirical functions, and can readily be extended to functions of two or more variables.

The science of statistics has also been a beneficiary of digital-computer techniques. One of the basic problems of statistics is that of deciding between two (or more) hypotheses on the basis of experimental data (decision theory or tests of significance). Such decisions are based on computations that involve the consecutive multiplication of large numbers of probability distribution functions. The digital computer has so enhanced the facility of performing such computations that they are sometimes carried out "on line"; for example, the production output of a manufacturing plant can be continuously monitored and evaluated statistically to provide decisions to adjust or shut down the production machinery if the deviation of the product from the set standard exceeds certain statistical limits.

The following paragraphs of Chapter 2 discuss the main aspects of numerical techniques in terms of (a) the representation of mathematical functions, (b) numerical differentation, (c) numerical integration, (d) methods for solving differential equations, and (e) methods for solving systems of linear algebraic equations. It should be observed that numerical analysis is partially a science and partially an art. As a result, short of writing a textbook on the subject it would be impossible to indicate the particular circumstances in which even a selected sampling from the vast stock of numerical interpolation, differentation, and integration formulas available would be useful or accurate, or to elucidate the numerical difficulties to which one might be led by uncritical use. Accordingly, the formulas associated with numerical analysis should never be applied blindly.

2-4.2 REPRESENTATION OF MATHEMATICAL FUNCTIONS

One might expect, intuitively, that mathematical functions would be represented in a digital computer by the storage of tabular data, in a manner analogous to the table-look-up procedure employed in hand computations. However, while the storage of functional tables in a digital computer is certainly possible, the high speed of computation and the relatively limited memory capacity that are typical of modern computers make the computation of functions a very attractive procedure. Some functions may be computed from their defining equations (which, in many cases, are differential equations) by iterative techniques. Certain functions, on the other hand, may be readily computed by the use of series approximations.

If a stored table is employed in a digital computer to represent a mathematical function, the storage requirements can be greatly reduced by storing only a few points and using an interpolation formula to approximate the function between these points. Interpolation is also used with input data to reduce the number of points that must be entered. A related process called curve fitting is employed whenever it is known from theoretical considerations that a set of data points should approximate a chosen mathematical function. The best fit between this chosen function and the data can be determined, and the function then used in lieu of the data points.

The paragraphs which follow summarize the pertinent aspects of the aforenoted techniques for representing mathematical functions.

2-4.2.1 Iteration

Iterative or recursive processes are fundamental to numerical methods of analysis. In the application of iteration to the evaluation of a function specified by its defining equation (or equations), one starts with a rough estimate of the value of the function

and then computes successively better approximations. In general, if it is desired to evaluate a function

$$f(x) = 0 \qquad (2\text{-}99)$$

and this equation can be rewritten in the form

$$x = F(x) \qquad (2\text{-}100)$$

the procedure is as follows. Given an estimate $x^{(k)}$ where $x^{(k)}$ represents the kth approximation to the value of the given function $F(x)$, compute $F(x^{(k)})$. Set $F(x^{(k)})$ equal to $x^{(k+1)}$ and repeat the process--computing $F(x^{(k+1)})$, and so on. The computation is terminated when the difference between two successive approximations is equal to or less than the allowable computational error. The evaluation of $\sqrt{N}$ presented in Example 2-1 illustrates the iterative technique. This example was chosen for its simplicity; it should be noted, however, that most defining equations are differential equations.

2-4.2.2 Series Approximation

The representation of functions by series approximations is particularly useful in digital-computer calculations because the function can be generated by a relatively few additions and multiplications. Example 2-2 shows the ease of computing the sine function from a power series.

The Taylor's series expansion is the general expression for a power-series expansion. If a function f(x) is differentiable at a point $x = x_o$, then f(x) can be replaced in the neighborhood of x_o by the power series

$$f(x) = f(x_o) + (x - x_o) f'(x_o) + \frac{(x - x_o)^2}{2!} f''(x_o) + \cdots \qquad (2\text{-}101)$$

or, in compact form,

$$f(x) = \sum_{n=0} \frac{(x - x_o)^n}{n!} f^{(n)}(x_o) \qquad (2\text{-}102)$$

where $f^{(n)}(x_o)$ is the nth derivative of f(x), evaluated at the point $x = x_o$. For computation, the series is truncated after a number of terms, say m terms. The sum of the remaining terms, the remainder, constitutes the error in the approximation. For the special case of a convergent Taylor's series with decreasing terms and alternating signs, the remainder cannot exceed the magnitude of the (m+1)th term, i.e.,

$$|R_m| \leq \left| \frac{(x - x_o)^m}{m!} f^{(m)}(x_o) \right| \qquad (2\text{-}103)$$

where R_m is the truncation error after the mth term. An expression that may be used to determine the truncation error in the general case is

$$R_m = \frac{1}{(m-1)!} \int_o^{x-x} f^{(m)}(x - t)\, t^{m-1}\, dt \qquad (2\text{-}104)$$

By determining the remainder or some bound on the remainder, the maximum error for a given number of terms is known. The computer program may be written to determine this error and to stop adding terms as soon as the error decreases below a desired amount.

2-4.2.3 Interpolation

The preceding paragraph discussed the approximation of functions by means of power series. Another technique, useful when a table of values of a function is available, is interpolation. With this technique, the value of the function at some point intermediate between two known points is approximated by a series of polynominals. In hand computation, only a first-order, or linear, interpolation is normally employed. The greater computational capacity of the digital computer, however, permits the use of higher-order polynomials. For the same accuracy, the higher-order interpolation requires fewer data points in storage.

If the tabular data are given for values of x spaced at equal intervals h, various formulas based on tabular differences can be employed. Newton's formulas are given as

Example 2-1. Iterative procedure for the evaluation of $\sqrt{N}$

An iterative procedure for the evaluation of $\sqrt{N}$ can be obtained if the solution is considered to be the intersection of the curve xy = N and the straight line x = y, as shown by Fig. 1. Start at the point $(x^{(0)}, y^{(0)})$ where $x^{(0)} = N$ and $y^{(0)} = 1$. Successive values of x are taken as the arithmetic mean of the preceding values of x and y, i. e.,

$$x^{(i+1)} = \frac{x^{(i)} + y^{(i)}}{2} \tag{1}$$

The corresponding value of y is

$$y^{(i+1)} = \frac{N}{x^{(i+1)}} \tag{2}$$

It can be readily seen that the solution follows the arrowed path shown in Fig. 1.

A sample calculation for N = 7 is shown in Table 1. For the six places carried, $\sqrt{N}$ = 2.64575. The error is 7×10^{-6}.

Table 1.

Sample Calculation of $\sqrt{N}$ for the Case When N = 7.

i	$x^{(i)}$	$y^{(i)}$
0	7.00000	1.00000
1	4.00000	1.75000
2	2.87500	2.434.78
3	2.65489	2.63664
4	2.64577	2.64573
5	2.64575	2.64575
.	.	
.	.	
.	.	

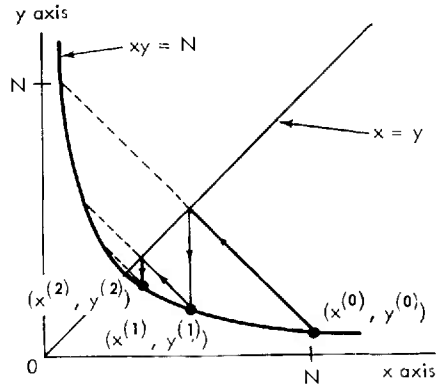
$$x^{(i+1)} = \frac{x^{(i)} + y^{(i)}}{2}$$

$$y^{(i+1)} = \frac{N}{x^{(i+1)}}$$

where

i = 0, 1, 2, 3, ..., i, i+1, ... for the number of computational steps required to achieve the accuracy specified.

Example 2-1 (Continued)



Sample successive approximation

$x^{(2)}, y^{(2)}$ is reached graphically from $(x^{(1)}, y^{(1)})$ by moving initially along a perpendicular to the straight line x = y, and then dropping to the curve xy = N along a line parallel to the y axis. Numerically,

$$x^{(2)} = \frac{x^{(1)} + y^{(1)}}{2}$$

$$y^{(2)} = \frac{N}{x^{(2)}}$$

Figure 1. Graphical representation of the path followed in the computation of $\sqrt{N}$.

Example 2-2. Computationof sin x by means of a power series.

The power series for sin x is

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} \ldots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!} + \ldots \qquad (1)$$

$$n = 1, 2, 3, \ldots$$

If x = 0.5 radian, the approximations for sin x employing one, two, and three terms of the series are, respectively,

$$\left.\begin{aligned} f^{(1)} &= 0.500000 \\ f^{(2)} &= 0.479167 \\ f^{(3)} &= 0.479427 \end{aligned}\right\} \qquad (2)$$

The error is already quite small; inclusion of the fourth term reduces the least significant figure by one unit.

typical. Other formulas of Stirling and Bessel will be found in the literature [46, 47].

If a function f(x) is known at points x_i, evenly spaced by the interval h along the x-axis, then

$$x_i = x_o + ih \quad (i = 0, 1, 2, 3, \ldots). \tag{2-105}$$

The values of the function at x_i are denoted by $f_i = f(x_i)$. The first central difference between f_i and f_o is denoted $\delta f_{1/2}$, and is defined by

$$\delta f_{1/2} = f_1 - f_0 \tag{2-106}$$

Similarly,

$$\delta f_{3/2} = f_2 - f_1 \tag{2-107}$$

and so on. The second, third, etc., central differences are denoted $\delta^2 f$, $\delta^3 f$, etc., respectively. The second differences are the differences between adjacent first differenences, the third differences are the differences between adjacent second differences, and so on. Table 2-3 illustrates the method.

If a new variable m is introduced such that $x = x_o + hm$, Newton's forward-difference formula can be expressed as*

$$f(m) = f_o + m\,\delta f_{1/2} + \frac{m(m-1)}{2!}\,\delta^2 f_1 + \frac{m(m-1)(m-2)}{3!}\,\delta^3 f_{3/2} + \ldots + \frac{m(m-1)\ldots(m-n+1)}{n!}\,\delta^n f_{(1/2)n} \tag{2-108}$$

It is also possible to work backwards from f_o, using Newton's backward-difference formula; this procedure yields

$$f(m) = f_o + m\,\delta f_{-1/2} + \frac{m(m+1)}{2!}\,\delta^2 f_{-1} + \frac{m(m+1)(m+2)}{3!}\,\delta^3 f_{-3/2} + \ldots + \frac{m(m+1)\ldots(m+n-1)}{n!}\,\delta^n f_{-(1/2)n} \tag{2-109}$$

When the tabulated data points x_i are not equally spaced, Lagrangian interpolation by polynomials of any desired degree can be employed. The general form of the Lagrangian interpolation is

$$f(x) = \sum_{j=0}^{n} \frac{(x-x_o)(x-x_1)\ldots(x-x_{j-1})(x-x_{j+1})\ldots(x-x_n)}{(x_j-x_o)(x_j-x_1)\ldots(x_j-x_{j-1})(x_j-x_{j+1})\ldots(x_j-x_n)}\,f_j \tag{2-110}$$

where $f_j = f(x_j)$. See Example 2-3 for an illustrative application of this relationship.

2-4.2.4 Curve Fitting

Where interpolation assumes no knowledge of a functional relationship between data points, curve fitting is the process by which a chosen function is adjusted to best fit a set of data points. The function may be chosen because it appears to fit the data well or, more commonly, because physical reasoning indicates that the data should fit some particular function. While many methods of curve fitting are used--some quite elaborate--only the most commonly used technique, that of the least-squares fit, will be described here.

* It should be noted that, in place of the generalized difference symbol δ used here, some references employ specific difference operators for particular usage, as follows:

$\Delta y(x) = y(x + \Delta x) - y(x)$ = forward-difference operator

$\nabla y(x) = y(x) - y(x - \Delta x)$ = backward-difference operator

$\delta y(x) = y\left(x + \frac{\Delta x}{2}\right) - y\left(x - \frac{\Delta x}{2}\right)$ = central-difference operator

See Section 20.4-2 of Ref. 50 for example.

TABLE 2-3. ARRAY OF TABULAR DIFFERENCES

i	x_i	f_i	δf_i	$\delta^2 f_i$	$\delta^3 f_i$	$\delta^4 f_i$	$\delta^5 f_i$	$\delta^6 f_i$
0	x_0	f_0						
1/2			$f_1 - f_0$					
1	x_1	f_1		$\delta f_{3/2} - \delta f_{1/2}$				
3/2			$f_2 - f_1$		$\delta^2 f_2 - \delta^2 f_1$			
2	x_2	f_2		$\delta f_{5/2} - \delta f_{3/2}$		$\delta^3 f_{5/2} - \delta^3 f_{3/2}$		
5/2			$f_3 - f_2$		$\delta^2 f_3 - \delta^2 f_2$		$\delta^4 f_2 - \delta^4 f_2$	
3	x_3	f_3		$\delta f_{7/2} - \delta f_{5/2}$		$\delta^3 f_{7/2} - \delta^3 f_{5/2}$		$\delta^5 f_{7/2} - \delta^5 f_{5/2}$
7/2			$f_4 - f_3$		$\delta^2 f_4 - \delta^2 f_3$		$\delta^4 f_4 - \delta^4 f_3$	
4	x_4	f_4		$\delta f_{9/2} - \delta f_{7/2}$		$\delta^3 f_{9/2} - \delta^3 f_{7/2}$		
9/2			$f_5 - f_4$		$\delta^2 f_5 - \delta^2 f_4$			
5	x_5	f_5		$\delta f_{11/2} - \delta f_{9/2}$				
11/2			$f_6 - f_5$					
6	x_6	f_6						

Let $y = g(x)$ be a curve fitted by a functional relationship between x and y having the generalized form

$$y = c_1 f_1(x) + c_2 f_2(x) + \ldots + c_n f_n(x) \tag{2-111}$$

where the functions $f_1(x)$, $f_2(x)$, ..., $f_n(x)$ are known. It is desired to satisfy the set of equations

$$\begin{aligned} Y_1 &= c_1 f_1(x_1) + c_2 f_2(x_1) + \ldots + c_n f_n(x_1) \\ Y_2 &= c_1 f_1(x_2) + c_2 f_2(x_2) + \ldots + c_n f_n(x_2) \\ &\vdots \\ y_m &= c_1 f_1(x_m) + c_2 f_2(x_m) + \ldots + c_n f_n(x_m) \end{aligned} \tag{2-112}$$

for the m sets of data points (x_1, y_1) (x_2, y_2), ..., (x_m, y_m). However, in general, each value of y will differ from its functional representation by the "residual" error δ_i where

$$\delta_i = y_i - c_1 f_1(x_i) - c_2 f_2(x_i) - \ldots - c_n f_n(x_i)$$

$$(i = 1, 2, \ldots, m) \tag{2-113}$$

In order to minimize the sum of the squares of the residuals, solutions of the following set of "normal equations" are obtained.

$$\begin{aligned} c_1 \sum f_1^2(x_i) + c_2 \sum f_1(x_i) f_2(x_i) + \ldots + c_n \sum f_1(x_i) f_n(x_i) &= \sum Y_i f_1(x_i) \\ c_1 \sum f_2(x_i) f_1(x_i) + c_2 \sum f_2^2(x_i) + \ldots + c_n \sum f_2(x_i) f_n(x_i) &= \sum Y_i f_2(x_i) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ c_1 \sum f_n(x_i) f_1(x_i) + c_2 \sum f_n(x_i) f_2(x_i) + \ldots + c_n \sum f_n^2(x_i) &= \sum Y_i f_n(x_i) \end{aligned} \tag{2-114}$$

where all summations are from i = 1 to i = m. Methods for the solution of these equations are given in par. 2-4.6. An example showing the application of these equations appears in Example 2-4.

Example 2-3. Sample application of Lagrange's interpolation formula

Given:

x_j	f_j
$x_0 = 13$	$67.8 = f_0$
$x_1 = 16$	$63.2 = f_1$
$x_2 = 32$	$45.4 = f_2$
$x_3 = 36$	$40.3 = f_3$

Find:

f(x) when x = 26 and f(x) when x = 27

Solution:

Use the relationship

$$f(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} f_0 + \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} f_1$$

$$+ \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} f_2 + \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} f_3$$

When x = 26,

$$f(x) = \frac{(10)(-6)(-10)}{(-3)(-19)(-23)}(67.8) + \frac{(13)(-6)(-10)}{(3)(-16)(-20)}(63.2)$$

$$+ \frac{(13)(10)(-10)}{(19)(16)(-4)}(45.4) + \frac{(13)(10)(-6)}{(23)(20)(4)}(40.3)$$

$$= \frac{(600)}{(-1311)}(67.8) + \frac{(780)}{960}(63.2)$$

$$t \frac{(-1300)}{(-1216)}(45.4) + \frac{(-780)}{(1840)}(40.3)$$

$$-31.02975 + 51.35 + 48.53618 - 17.08370$$

$$= 51.77273$$

When x = 27,

$$f(x) = \frac{(11)(-5)(-9)}{(-3)(-19)(-23)}(67.8) + \frac{(14)(-5)(-9)}{(3)(-16)(-20)}(63.2)$$

$$t \frac{(14)(11)(-9)}{(19)(16)(-4)}(45.4) + \frac{(14)(11)(-5)}{(23)(20)(4)}(40.3)$$

$$= \frac{495}{(-1311)}(67.8) + \frac{630}{960}(63.2)$$

$$t \frac{(-1386)}{(-1216)}(45.4) + \frac{(-770)}{1840}(40.3)$$

$$-25.59954 + 41.475 + 48.53618 - 16.86467$$

$$= 47.54697$$

Example 2-4. Application of the least-squares curve-fitting technique to range-vs-time-of-flight data.

Problem:

Fit the following range-vs-time-of-flight source data by a relationship of the form

$$y = c_1 x^2 + c_2 x + c_3 \qquad (1)$$

where

x = target range, in thousands of yards

y = time of flight, in seconds

c_1, c_2, c_3 = constants

Range-vs-Time-of-Flight Source Data

Data-Point Designation i	Target Range x_i (yards)	Time of Flight y_i (seconds)
1	0.6	0.70
2	0.8	0.96
3	1.0	1.24
4	1.2	1.50
5	1.4	1.82
6	1.6	2.12
7	1.8	2.46
8	2.0	2.80
9	2.2	3.16
10	2.4	3.52

Example 2-4. (Continued)

Solution:

Equation 1 can be rewritten in the form of the generalized functional relationship between x and y that is given by Eq. 211; i.e.,

$$y = c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x) \tag{2}$$

where

$$f_1(x) = x^2; \quad f_2(x) = x; \quad f_3(x) = 1 \tag{3}$$

The constants c_1, c_2, and c_3 can be determined by the use of equations that correspond to the generalized relationships expressed by Eq. 2-114. For the problem under consideration, these equations are

$$\left.\begin{aligned}
c_1 \sum_{i=1}^{i=10} f_1^2(x_i) + c_2 \sum_{i=1}^{i=10} f_1(x_i) f_2(x_i) + c_3 \sum_{i=1}^{i=10} f_1(x_i) f_3(x_i) &= \sum_{i=1}^{i=10} y_i f_1(x_i) \\
c_1 \sum_{i=1}^{i=10} f_2(x_i) f_1(x_i) + c_2 \sum_{i=1}^{i=10} f_2^2(x_i) + c_3 \sum_{i=1}^{i=10} f_2(x_i) f_3(x_i) &= \sum_{i=1}^{i=10} y_i f_2(x_i) \\
c_1 \sum_{i=1}^{i=10} f_3(x_i) f_1(x_i) + c_2 \sum_{i=1}^{i=10} f_3(x_i) f_2(x_i) + c_3 \sum_{i=1}^{i=10} f_3^2(x_i) &= \sum_{i=1}^{i=10} y_i f_3(x_i)
\end{aligned}\right\} \tag{4}$$

Application of the relationships given by Eqs. 3 to Eq. 4 yields the following set of equations:

$$\left.\begin{aligned}
c_1 \sum_{i=1}^{i=10} x_i^4 + c_2 \sum_{i=1}^{i=10} (x_i^2)(x_i) + c_3 \sum_{i=1}^{i=10} (x_i^2)(1) &= \sum_{i=1}^{i=10} (y_i)(x_i^2) \\
c_1 \sum_{i=1}^{i=10} (x_i)(x_i^2) + c_2 \sum_{i=1}^{i=10} x_i^2 + c_3 \sum_{i=1}^{i=10} (x_i)(1) &= \sum_{i=1}^{i=10} (y_i)(x_i) \\
c_1 \sum_{i=1}^{i=10} (1)(x_i^2) + c_2 \sum_{i=1}^{i=10} (1)(x_i) + c_3 \sum_{i=1}^{i=10} (1) &= \sum_{i=1}^{i=10} (y_i)(1)
\end{aligned}\right\} \tag{5}$$

The computations on the source data that are required for substitution in Eqs. 5 are summarized in the following tabulation.

Example 2-4. (Continued)

Summary of the Required Computations on the Source Data

i	x_i	Yi	x_i^2	x_i^3	x_i^4	$(y_i)(x_i)$	$(y_i)(x_i^2)$
1	0.6	0.70	0.36	0.216	0.1296	0.420	0.2520
2	0.8	0.96	0.64	0.512	0.4096	0.768	0.6144
3	I.0	1.24	1.00	1.000	1.0000	1.240	1.2400
4	1.2	1.50	1.44	1.728	2.0736	1.800	2.1600
5	1.4	1.82	1.96	2.744	3.8416	2.548	3.5672
6	1.6	2.12	2.56	4.096	6.5536	3.392	5.4272
7	6.8	2.46	3.24	5.832	10.4976	4.428	7.9704
8	2.0	2.80	4.00	8.000	16.0000	5.600	11.2000
9	2.2	3.16	4.84	10.648	23.42 56	6.952	15.2944
10	2.4	3.52	5.76	13.824	33.1776	8.448	20.2752
$\sum_{i=1}^{i=10}$	15.0	20.28	25.8	48.6	97.1088	35.596	68.0008

The substitution of these computations in Eqs. 5 yields the following system of linear equations that can be used to determinc c_1, c_2, and c_3:

$$97.1088\ c_1 + 48.6\ c_2 + 25.8\ c_3 = 68.0008$$

$$48.6\ c_1 + 25.8\ c_2 + 15.0\ c_3 = 35.596$$

$$25.8\ c_1 + 15.0\ c_2 + 10.0\ c_3 = 20.28$$

In matrix form, Eq. 6 becomes

$$\begin{bmatrix} 97.1088 & 48.6 & 25.8 \\ 48.6 & 25.8 & 15.0 \\ 25.8 & 15.0 & 10.0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 68.0008 \\ 35.596 \\ 20.28 \end{bmatrix} \quad (7)$$

Example 2-4. (Continued)

The solution set of this matrix equation is

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0.17803 \\ 1.03439 \\ 0.01709 \end{bmatrix} \tag{8}$$

The application to Eq. 1 of this solution set and the tabulated computations on the source data establishes the following table of computed values for y_i and the resulting residual errors in these computed values. As defined by Eq. 2-113, a negative error means that the computed value of y_i is greater than the actual value of y_i, i. e., the value given in the range-vs-time-of-flight source data.

Summary of Computed Values for y_i and the Resulting Residual Errors δ_i

Data-Point Designation i	Computed Value of Time of Flight $y_{i\ (computed)}$ (seconds)	Error in $y_{i\ (computed)}$ δ_i (seconds)
1	0.702	-0.002
2	0.9 59	0.001
3	1.230	0.010
4	1.515	-0.015
5	1.814	0.006
6	2.128	-0.008
7	2.456	0.004
8	2.798	0.002
9	3.154	0.006
10	3.525	-0.005

2-4.3 NUMERICAL DIFFERENTATION

Numerical differentiation is closely related to the interpolation methods described in par. 2-4.2.3. If a function is represented by interpolating polynomials, the polynomial expression can be differentiated.

Numerical differentiation is very dangerous to use, however, because it is subject to errors that are due to the approximating polynomial of a given function, insufficient data, and many other reasons. As an illustration of this danger, consider the determination of the derivative of a relationship $y = f(x)$ that is given by a table $\{(x_o,y_o), (x_1,y_1), \ldots, (x_n,y_n)\}$ at the point for which $x = \zeta$, where $x_o < \zeta < x_n$. The table is first approximated by a polynomial $P_a(x)$. The derivative $P_a'(x)$ of this approximating polynomial is then evaluated at $x = \zeta$. The resulting number $P_a'(\zeta)$ is used as the derivative of $f(x)$ at $x = \zeta$. Although the approximating polynomial $P_a(\zeta)$ may actually be a very satisfactory fit to $y = f(x)$, the number $P_a'(\zeta)$ may actually be a very poor approximation to $f'(\zeta)$. For example, consider the relationship $y = f(x)$ and its approximating polynomial $P_a(x)$ that is depicted in Fig. 2-19. This figure shows that $P_a'(\zeta)$, the slope of the tangent to $P_a(x)$ at $x = \zeta$, is close to zero but that $f'(\zeta)$ is far from zero. (Observe, however, that although the approximation to $f'(x)$ at $x = \zeta$ is very poor the approximation to $f'(x)$ at $x = \zeta_1$ is very good.)

The various difference formulas (ref. par. 2-4.2.3) can be differentiated to provide suitable numerical differentation formulas. For example, in the case of a given function $y = f(x)$, the differentiation of Newton's (Gregory-Newton) forward-difference formula yields the numerical differentiation formula:::

$$\left.\frac{df}{dx}\right|_{x=} \cong \frac{1}{h}\left(\Delta y_k - \frac{1}{2} + \frac{1}{3}\Delta^3 y_k - \ldots\right) \qquad (2\text{-}115)$$

where

$$x_k = x_o + k\,h \qquad (2\text{-}115a)$$

h = equal intervals at which the tabular values of x are spaced

$$\Delta y_k = y_{k+1} - y_k \qquad (2\text{-}115b)$$

= standard first-order difference

and

$$\Delta^n y_k = \Delta^{n-1} y_{k+1} - \Delta^{n-1} y_k \qquad (2\text{-}115c)$$

= nth-order difference

$n = 2, 3, \ldots$

$k = 0, \pm1, \pm2, \ldots$

As an example of the application of Eq. 2-115, consider the tabular function described by the following set of values for x and y: {(2.0, 0.69315), (2.1, 0.74194), (2.2, 0.78846), (2.3, 0.83291), (2.4, 0.87547)}. Find the derivative at x = 2.1, using Eq. 2-115 and the following forward-difference table.

k	x_k	y_k	Δy_k	$\Delta^2 y_k$	$\Delta^3 y_k$	$\Delta^4 y_k$
0	2.0	0.69315				
			0.04879			
1	2.1	0.74194		-0.00227		
			0.04652		0.00020	
2	2.2	0.78846		-0.00207		
			0.04445		0.00018	
3	2.3	0.83291		-0.00189		
			0.04256			
4	2.4	0.87547				

With h = 0.1, k = 1, $\Delta y_1 = 0.04652$, $\Delta^2 y_1 = -0.00207$, and $\Delta^3 y_1 = 0.00018$, Eq. 2-115 shows that

$$f'(2.1) \cong \frac{1}{0.1}\left(0.04652 - \frac{1}{2}\left\{-0.00207\right\} + \frac{1}{3}(0.00018\}\right)$$

$$\cong \frac{1}{0.1}(0.04652 + 0.00104 + 0.00006)$$

$$\cong \frac{1}{0.1}(0.04762)$$

$$\cong 0.4762$$

* See, for example, Eq. 20. 6-1 in Section 20. 6 of Ref. 50

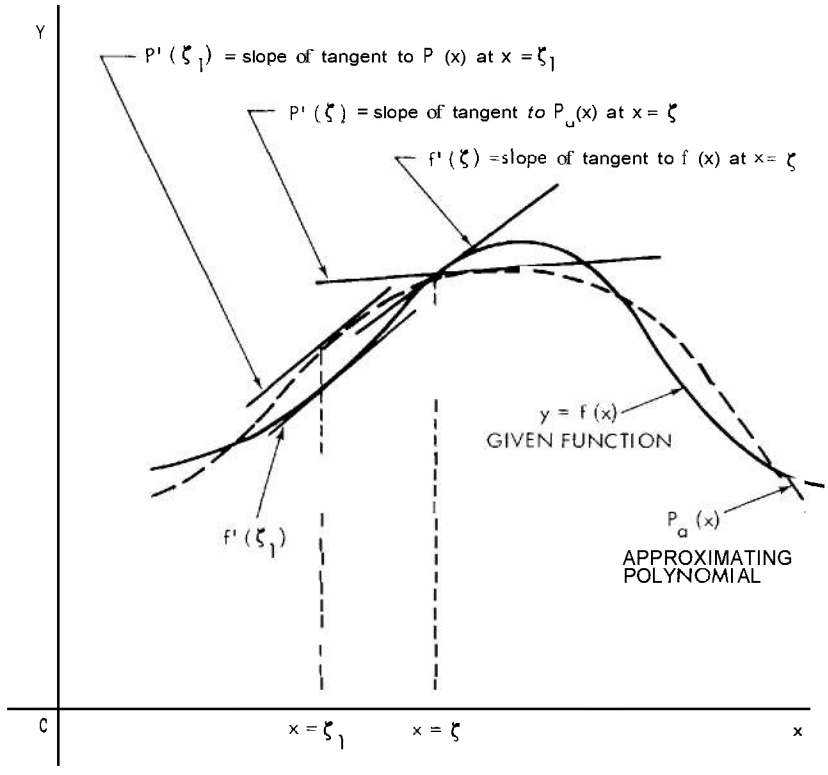


Figure 2-19. The difference between the derivative of a given function and the derivative of its approximating polynomial.

The tabular function used in this example was taken from a natural log table, that is, $f(x) = \ln x$, which yields $f'(x) = 1/x$ for $x > 0$. When the number 2.1 is substituted for x, the result is $f'(2.1) = 1/2.1 = 0.47619$. Thus, the approximation obtained for $f'(2.1)$ is excellent. Such a result cannot always be expected, however, as already observed.

It should be noted that Eq. 2-115 is only one of many possible numerical differentiation formulas.* The particular problem concerned and one's personal experience in using numerical differentiation formulas normally determine which formula is to be used. The choice of an appropriate formula is a subjective process and hence is in the nature of an art rather than a science.

2-4.4 NUMERICAL INTEGRATION

The process of evaluating a definite integral (sometimes known as "quadrature") is a laborious task that has been greatly eased by the availability of digital computers. The basis of numerical integration is inherent in the definition of integration: integration of a function f(x) is accomplished by adding the areas of a series of strips of width Δx and height f(x), as $\Delta x \to 0$. Since it would be necessary to sum a large number of such incremental areas in order to obtain an accurate integration, various formulas have been developed to reduce the number of increments required.

Of the many integration formulas that have been developed,† only one of the best

* For example, Eqs. 20.6-3 and 20.6-4 in Section 20.6 of Ref. 50 give numerical differentiation formulas that result from the differentiation of Stirling's and Bessel's interpolation formulas. See also page 231 of Ref. 10.

† See Chapter IX of Ref. 48.

known, Simpson's rule, will be described. In applying this rule, a parabola is passed through three consecutive equally-spaced points located on the function to be integrated. It can be shown * that the area under the curve is given by

$$A = \frac{h}{3}(y_1 + 4y_2 + y_3) \qquad (2\text{-}116)$$

where the quantities are defined in Fig. 2-20. For an even number n values of x, the area is

$$A \simeq \frac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \ldots + 4f(x_{n-1}) + f(x_n) \right] \qquad (2\text{-}117)$$

where

$$h = \frac{x_n - x_0}{n}$$

Simpson's rule is exact for the integration of polynomials up to the third order. Example 2-5 gives an illustrative application of Simpson's rule.

2-4.5 METHODS FOR SOLVING DIFFERENTIAL EQUATIONS

Since a differential equation describes the behavior of a function by considering infinitesimally small changes, the general method of its solution on a digital computer is intuitively obvious. However, the desire to improve the accuracy of solution and to reduce the amount of storage required has led to the development of rather involved methods of solution. A simple method originated by Euler, two more-complex methods provided by Runge and Kutta, and a predictor-corrector method due to Milne will be described here. Other methods will be found in the literature.

Consider, first, the simple first-order differential equation in the form

$$\frac{dy}{dx} = f(x, y) \qquad (2\text{-}118)$$

both for its own great usefulness and because higher-order equations can be reduced to this form, as will be explained in this paragraph. If the independent variable is divided into increments (not necessarily equal) by the points

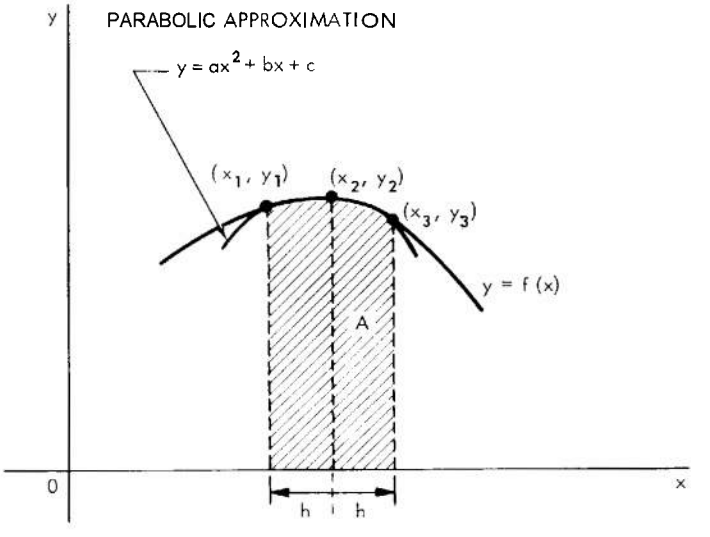


Figure 2-20. Integration by means of Simpson's Rule.

See page 193 of Ref. 48.

Example 2-5. Sample application of Simpson's rule.

Simpson's rule will be used to evaluate $\int_{1.6}^{2.0} f(x)\,dx$ when f(x) is given by the following tabulation for an initial value x_0 and n additional values of x:

n	x_n	$f(x_n)$
0	1.6	12.6894
1	1.7	12.8724
2	1.8	13.0352
3	1.9	13.1943
4	2.0	13.3654

For n = 4, Eq. 2-117 shows that

$$A \approx (h/3)\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + f(x_4)\right]$$

where

$$h = \frac{x_4 - x_0}{n} = \frac{2.0 - 1.6}{4} = \frac{0.4}{4} = 0.1$$

Substitution from the tabulation yields

$$A \approx \frac{0.1}{3}\left[12.6894 + 4(12.8724) + 2(13.0352) + 4(13.1943) + 13.36541\right.$$

$$\approx 0.0333\left[12.6894 + 51.4896 + 26.0704 + 52.7772 + 13.3654\right]$$

$$\approx 0.0333\left[156.39201 = 5.2078\right.$$

$x_0, x_1, \ldots, x_i, x_{i+1}, \ldots$, the value of y at any point, say, $i+1$, may be approximated by extrapolating the value of y at the previous point (i), using the known value of the slope at i; thus,

$$y_{i+1} = \left[\frac{dy}{dx}\right] (x_{i+1} - x_i) + y_i \qquad (2\text{-}119)$$

where y_i is the value of y at $x = x_i$ and y_{i+1} is the value of y at $x = x_{i+1}$. Substitution of Eq. 2-118 in Eq. 2-119 yields

$$y_{i+1} = (x_{i+1} - x_i) f_i + y_i \qquad (2\text{-}120)$$

where f_i is the value of f(x,y) at $x=x_i$, $y=y_i$. Eq. 2-120, known as Euler's formula, has a truncation error with an order of magnitude equivalent to the square of the increment in x.

As an example, the equation

$$\frac{dy}{dx} = y - x \qquad (2\text{-}121)$$

has been solved explicitly in Example 2-6 for values of x between 0 and 0.7, and also by Euler's formula for the same range in x. The evaluation of the Taylor's series for Eq. 2-121 at x = 0 is also shown in Example 2-6. The evaluation of the series expansion is accurate near the point at which the derivatives are evaluated, but requires considerable computational labor.

To apply the Runge-Kutta method, again consider the differential equation of the form

$$\frac{dy}{dx} = f(x, y) \qquad (2\text{-}122)$$

If the solution at some point $x = x_i$ can be determined by the Taylor's series

$$y(x_i + h) = y(x_i) + h f(x_i, y_i) + \frac{h^2}{2} f'(x_i, y_i) + \frac{h^3}{6} f''(x_i, y_i) + \ldots \qquad (2\text{-}123)$$

where $h = x_{i+1} - x_i$, then the Runge-Kutta method determines an expression $y(x_i) + \bar{k}$ that is identical with Eq. 2-123, where

$$\bar{k} = R_1 k_1 + R_2 k_2 + R_3 k_3 + \ldots \qquad (2\text{-}124)$$

and

$$\left.\begin{aligned} k_1 &= h f(x_i, y_i) \\ k_2 &= h f(x_i + \alpha h, y_i + \beta k_1) \\ k_3 &= h f(x_i + \alpha_1 h, y_i + \beta_1 k_1 + \gamma_1 k_2) \\ &\text{etc.} \end{aligned}\right\} \qquad (2\text{-}125)$$

The constants $R_1, R_2, R_3, \ldots, \alpha, \alpha_1, \ldots, \beta, \beta_1, \ldots, \gamma_1, \ldots$ etc., are determined by setting $y(x_i) + \bar{k}$ equal to a specific number of terms of the expansion for $y(x_i + h)$. Except for the second-order expression (which is formed by discarding terms in Eq. 2-123 beyond the h^2 term), the constants are not uniquely determined; moreover the derivations are quite involved. The second- and fourth-order expressions are as follows:

Second-Order

$$\left.\begin{aligned} \bar{k} &= h f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\ k_1 &= h f(x_i, y_i) \end{aligned}\right\} \qquad (2\text{-}126)$$

Fourth-Order

$$\left.\begin{aligned} \bar{k} &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= h f(x_i, y_i) \\ k_2 &= h f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\ k_3 &= h f\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\ k_4 &= h f(x_i + h, y_i + k_3) \end{aligned}\right\} \qquad (2\text{-}127)$$

Example 2-6. The numerical solution of $\frac{dy}{dx} = y - x$ by the exact method and by four approximate methods.

In order to show the application of the methods developed inpar. 2-2.3.2, the exact solution of the differential equation

$$\frac{dy}{dx} = y - x \tag{1}$$

is determined for a starting point of x = 0, y = 4 and an interval in x of 0.1. The same equation under these same conditions is then solved by Taylor's series, by Euler's method, by the Runge-Kutta second-order method, and by the Runge-Kutta fourth-order method.

1. Exact Solution

$$y = ae^x + x + 1 \tag{2}$$

Initial Conditions: x = 0 and y = 4

The substitution into Eq. 2 of these initial conditions shows that a = 3.

For $x_0 = 0$
When $x_0 = 0$, then $e^{x_0} = 1.00000$
Therefore, $y_0 = ae^{x_0} + x_0 + 1$
$= 3(1) + 0 + 1 = 4.00000$

For $x_1 = 0.1$
When $x_1 = 0.1$, then $e^{x_1} = 1.10517$
Therefore, $y_1 = ae^{x_1} + x_1 + 1$
$= 3(1.10517) + 0.1 + 1 = 4.41551$

For $x_2 = 0.2$
When $x_2 = 0.2$, then $e^{x_2} = 1.22140$
Therefore, $y_2 = ae^{x_2} + x_2 + 1$
$= 3(1.22140) + 0.2 + 1 = 4.86420$

For $x_3 = 0.3$
When $x_3 = 0.3$, then $e^{x_3} = 1.34986$
Therefore, $y_3 = ae^{x_3} + x_3 + 1$
$= 3(1.34986) + 0.3 + 1 = 5.34958$

For $x_4 = 0.4$
When $x_4 = 0.4$, then $e^{x_4} = 1.49182$
Therefore, $y_4 = ae^{x_4} + x_4 + 1$
$= 3(1.49182) + 0.4 + 1 = 5.87546$

For $x_5 = 0.5$
When $x_5 = 0.5$, then $e^{x_5} = 1.64872$
Therefore, $y_5 = ae^{x_5} + x_5 + 1$
$= 3(1.64872) + 0.5 + 1 = 6.44616$

Example 2-6. (Continued)

For $x_6 = 0.6$

When $x_6 = 0.6$, then $e^{x_6} = 1.82212$

Therefore, $y_6 = ae^{x_6} + x_6 + 1$

$= 3(1.82212) + 0.6 + 1 = 7.06636$

For $x_7 = 0.7$

When $x_7 = 0.7$, then $e^{x_7} = 2.01375$

Therefore, $y_7 = ae^{x_7} + x_7 + 1$

$= 3(2.01375) + 0.7 + 1 = 7.74125$

The exact solution, to five decimal places, is summarized in Table 1 for values of x from 0 to 0.7 and an interval of 0.1.

Table 1. Exact Solution

x	e^x	y_{exact}
0	1.00000	4.00000
0.1	1.10517	4.41551
0.2	1.22140	4.86420
0.3	1.34986	5.349 58
0.4	1.49182	5.87546
0.5	1.64872	6.44616
0.6	1.82212	7.06636
0.7	2.01375	7.74125

2. Taylor's Series Solution

The Taylor's series through the third-order term is:

$$y(x) \cong y(0) + xy'(0) + \frac{x^2}{2!} y''(0) + \frac{x^3}{3!} y'''(0) + \ldots$$

$$y(0) = 4 \qquad y'' = y' - 1 \qquad y''(0) = 3$$

$$y'(0) = 4 \qquad y''' = y'' \qquad y'''(0) = 3$$

Therefore,

$$y(x) = 4 + 4x + \frac{3}{2} x^2 + \frac{1}{2} x^3 + \ldots$$

For $x_0 = 0$

When $x_0 = 0$,

then $y_0(x) = 4 + 4x_0 + (3/2)x_0^2 + (1/2)x_0^3 + \ldots$

$= 4 + 0 + 0 + 0 + \ldots = 4.00000$

For $x_1 = 0.1$

When $x_1 = 0.1$,

then $y_1(x) = 4 + 4x_1 + (3/2)x_1^2 + (1/2)x_1^3 + \ldots$

$= 4 + 0.4 + 0.015 + 0.0005 + \ldots = 4.41550$

Example 2-6. (Continued)

For $x_2 = 0.2$

When $x_2 = 0.2$,

then $y_2(x) = 4 + 4x_2 + (3/2)x_2^2 + (1/2)x_2^3 + \ldots$

$= 4 + 0.8 + 0.06 + 0.004 + \ldots = 4.86400$

For $x_3 = 0.3$

When $x = 0.3$,

then $y_3(x) = 4 + 4x3 + (3/2)x_3^2 + (1/2)x_3^3 + \ldots$

$= 4 + 1.2 + 0.135 + 0.135 = 5.34850$

For $x_4 = 0.4$

When $x = 0.4$,

then $y_4(x) = 4 + 4x_4 + (3/2)x_4^2 + (1/2)x_4^3$

$= 4 + 1.6 + 0.24 + 0.032 = 5.87200$

For $x_5 = 0.5$

When $x = 0.5$,

then $y_5(x) = 4 + 4x_5 + (3/2)x_5^2 + (1/2)x_5^3$

$= 4 + 2 + 0.375 + 0.0625 = 6.43750$

For $x_6 = 0.6$

When $x = 0.6$,

then $y_6(x) = 4 + 4x_6 + (3/2)x_6^2 + (1/2)x_6^3$

$= 4 + 2.4 + 0.54 + 0.108 = 7.04800$

For $x_7 = 0.7$

When $x = 0.7$,

then $y_7(x) = 4 + 4x_7 + (3/2)x_7^2 + (1/2)x_7^3$

$= 4 + 2.8 + 0.735 + 0.1715 = 7.70650$

The Taylor's Series Solution is summarized in Table 2, together with the error between it and the exact solution.

Table 2. Taylor's Series Solution

x	Y	y-y exact
0	4.00000	0.00000
0.1	4.41550	-0.00001
0.2	4.86400	-0.00020
0.3	5.34850	-0.00108
0.4	5.87200	-0.00346
0.5	6.43750	-0.00866
0.6	7.04800	-0.01836
0.7	7.70650	-0.03475

Note that the error magnitude increases rapidly as the deviation from the point of evaluation of the derivatives (x = 0) increases.

Example 2-6. (Continued)

3. Solution by Euler's Method

Euler's method makes use of the formula

$$y_{i+1} = (x_{i+1} - x_i) f_i + y_i \qquad (4)$$

with $f_i = y_i - x_i$ and initial values $x_0 = 0$ and $y_0 = 4$, and with $x_{i+1} - x_i$ always equal to 0.1.

For $x_0 = 0$

Initial values: $x_0 = 0$ and $y_0 = 4.00000$

For $x_1 = 0.1$

$$\begin{aligned} y_1 &= (x_1 - x_0) f_o = (x_1 - x_0)(y_0 - x_0) + y_0 \\ &= (0.1 - 0.0)\ (4.00000 - 0.0) + 4.00000 \\ &= 4.40000 \end{aligned}$$

For $x = 0.2$

$$\begin{aligned} y_2 &= (x_2 - x_1)\ (y_1 - x_1) + y_1 \\ &= (0.2 - 0.1)\ (4.40000 - 0.1) + 4.40000 \\ &= 4.83000 \end{aligned}$$

For $x_3 = 0.3$

$$\begin{aligned} y_3 &= (0.1)\ (y_2 - x_2) + y_2 \\ &= 0.1\ (4.83000 - 0.2) + 4.83000 \\ &= 5.29300 \end{aligned}$$

For $x_4 = 0.4$

$$\begin{aligned} y4 &= (0.1)\ (y3 - x_3) + y3 \\ &= (0.1)\ (5.29300 - 0.3) + 5.29300 \\ &= 5.79230 \end{aligned}$$

For $x_5 = 0.5$

$$\begin{aligned} y5 &= (0.1)\ (y_4 - x_4) + y4 \\ &= (0.1)\ (5.79230 - 0.4) + 5.79230 \\ &= 6.33153 \end{aligned}$$

For $x_6 = 0.6$

$$\begin{aligned} y_6 &= (0.1)\ (y_5 - x_5) + y_5 \\ &= (0.1)\ (6.33153 - 0.5) + 6.33153 \\ &= 6.91468 \end{aligned}$$

For $x_7 = 0.7$

$$\begin{aligned} y_7 &= (0.1)\ (y_6 - x_6) + y_6 \\ &= (0.1)\ (6.91468 - 0.6) + 6.91468 \\ &= 7.54615 \end{aligned}$$

The Euler's Method Solution is summarized in Table 3, together with the error between it and the exact solution.

Example 2-6 (Continued)

Table 3. Solution by Euler's Method.

x_i	y_i	Error $y_i - Y_{exact}$
0	4.00000	0.00000
0.1	4.40000	-0.01551
0.2	4.83000	-0.03420
0.3	5.29300	-0.05658
0.4	5.79230	-0.08316
0.5	6.33153	-0.11463
0.6	6.91468	-0.15168
0.7	7.54615	-0.19510

4. Solution by the Runge-Kutta Second-Order Method

$$\left.\begin{aligned} y_{i+1} &= y(x_i + h) = y(x_i) + \bar{k} = y_i + \bar{k} \\ \bar{k} &= h f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\ k_1 &= h f(x_i, y_i) \end{aligned}\right\} \quad (5)$$

where h = 0.1.

For $x_0 = 0$

Initial values: $x_0 = 0$ and $y_i = 4.00000$

$$k_1 = h f(x_0, y_0) = h(y_0 - x_0) = 0.1\ (4.00000 - 0) = 0.4$$

$$\begin{aligned} \bar{k} &= h\ f\left(x_0 + \frac{h}{2},\ y_0 + \frac{k_1}{2}\right) \\ &= 0.1\ f\ (0 + 0.05,\ 4 + 0.2) \\ &= 0.1\ (4.20 - 0.05) \\ &= 0.1 \times 4.15 = 0.41500 \end{aligned}$$

For $x_1 = 0.1$

$$\begin{aligned} y_1 &= y_0 + \bar{k} \\ &= 4.00000 + 0.415 \\ &= 4.41500 \end{aligned}$$

$$\begin{aligned} k_1 &= h\ f(x_1, y_1) = h\ (y_1 - x_1) \\ &= 0.1\ (4.41500 - 0.1) = 0.43150 \end{aligned}$$

$$\begin{aligned} \bar{k} &= h\ f\left(x_1 + \frac{h}{2},\ y_1 + \frac{k_1}{2}\right) \\ &= (0.1)\ f(0.1 + 0.05,\ 4.41500 + 0.21575) \\ &= (0.1)\ (4.48075) \\ &= 0.448075 \quad = 0.44808 \end{aligned}$$

Example 2-6. (Continued)

For $x_2 = 0.2$

$$y_2 = y_1 + \bar{k} = 4.415 + 0.44808$$
$$= 4.86308$$

$$k_1 = h\ f(x_2, y_2) = h\ (y_2 - x_2)$$
$$= 0.1\ (4.86308 - 0.2) = 0.1\ (4.66308)$$
$$= 0.466308 \quad = 0.46631$$

$$\bar{k} = h\ f\left(x_2 + \frac{h}{2},\ y_2 + \frac{k_1}{2}\right)$$
$$= (0.1)\ f(0.2 + 0.05,\ 4.86308 + 0.233155)$$
$$= (0.1)\ 4.846235 \quad = 0.4846235$$
$$= 0.48462$$

For $x_3 = 0.3$

$$y_3 = y_2 + \bar{k} = 4.86308 + 0.48462$$
$$= 5.34770$$

$$k_1 = h\ f(x_3, y_3) = h\ (y_3 - x_3)$$
$$= 0.1\ (5.34770 - 0.3) = 0.1\ (5.04770)$$
$$= 0.50477$$

$$\bar{k} = h\ f\left(x_3 + \frac{h}{2},\ y_3 + \frac{k_1}{2}\right)$$
$$= (0.1)\ f(0.3 + 0.05,\ 5.34770 + 0.252385)$$
$$= (0.1)\ (5.250085)$$
$$= 0.52501$$

For $x_4 = 0.4$

$$y_4 = y_3 + \bar{k} = 5.34770 + 0.52501$$
$$= 5.87271$$

$$k_1 = h\ f(x_4, y_4) = h\ (y_4 - x_4)$$
$$= 0.1\ (5.87271 - 0.4)$$
$$= 0.1\ (5.47271)$$
$$= 0.54727$$

$$\bar{k} = h\ f\left(x_4 + \frac{h}{2},\ y_4 + \frac{k_1}{2}\right)$$
$$= 0.1\ (0.4 + 0.05,\ 5.87271 + 0.273635)$$
$$= 0.1\ (5.69635) \quad = 0.56964$$

For $x_5 = 0.5$

$$y_5 = y_4 + \bar{k} = 5.87271 + 0.56964$$
$$= 6.44235$$

Example 2-6. (Continued)

$$k_1 = h\ f(x_5, y_5) = h\ (y_5 - x_5)$$
$$= 0.1\ (6.44235 - 0.5)$$
$$= 0.1\ (5.94235)$$
$$= 0.59424$$

$$\bar{k} = h\ f\left(x_5 + \frac{h}{2},\ y_5 + \frac{k_1}{2}\right)$$
$$= 0.1\ (0.5 + 0.05,\ 6.44235 + 0.29712)$$
$$= 0.1\ (6.18947)$$
$$= 0.61895$$

For $x_6 = 0.6$

$$y_6 = y_5 + \bar{k} = 6.44235 + 0.61895$$
$$= 7.06130$$

$$k_1 = h\ f(x_6, y_6) = h\ (y_6 - x_6)$$
$$= 0.1\ (7.06130 - 0.6)$$
$$= 0.64613$$

$$\bar{k} = h\ f\left(x_6 + \frac{h}{2},\ y_6 + \frac{k_1}{2}\right)$$
$$= (0.1)\ f(0.6 + 0.05,\ 7.06130 + 0.323065)$$
$$= (0.1)\ (6.734365)$$
$$= 0.67344$$

For $x_7 = 0.7$

$$y_7 = y_6 + \bar{k} = 7.06130 + 0.67344$$
$$= 7.73474$$

$$k_1 = h\ f(x_7, y_7) = h\ (y_7 - x_7)$$
$$= 0.1\ (7.73474 - 0.7)$$
$$= 0.1\ (7.03474)$$
$$= 0.70347$$

$$\bar{k} = h\ f\left(x_7 + \frac{h}{2},\ y_7 + \frac{k_1}{2}\right)$$
$$= (0.1)\ f(0.7 + 0.05,\ 7.73474 + 0.351737)$$
$$= (0.1)\ (7.336477)$$
$$= 0.73365$$

For $x_8 = 0.8$

$$y_8 = y_7 + \bar{k} = 7.73474 + 0.73365$$
$$= 8.46839$$

Example 2-6. (Continued)

The solution obtained by the Runge-Kutta second-order method is summarized in Table 4, together with the error between it and the exact solution.

Table 4. Solution by the Runge-Kutta Second-Order Method.

x_i	y_i	k_1	$\bar{k}$	Error $y_i - Y_{exact}$
0	4.00000	0.40000	0.41500	0.00000
0.1	4.41500	0.43150	0.448 08	0.00051
0.2	4.86308	0.46631	0.48462	0.00112
0.3	5.3477 0	0.5047 7	0.52501	0.00188
0.4	5.87271	0.54727	0.56964	0.00275
0.5	6.44235	0.59424	0.61895	0.00381
0.6	7.06130	0.64613	0.67 344	0.00506
0.7	7.73474	0.70347	0.7 3365	0.00651
0.8	8.46839			

5. Solution by the Runge-Kutta Fourth-Order Method

$$
\left.
\begin{aligned}
y_{i+1} &= y(x_i + h) = y(x_i) + \bar{k} = y_i + \bar{k} \\
\bar{k} &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= hf(x_i, y_i) \\
k_2 &= hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\
k_3 &= hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\
k_4 &= hf(x_i + h, y_i + k_3)
\end{aligned}
\right\} \quad (6)
$$

Example 2-6. (Continued)

For $x_0 = 0$

Initial Values: $x_0 = 0$ and $y_0 = 4.00000$

$k_1 = hf(x_0, y_0) = 0.1\ (y_0 - x_0) = 0.1\ (4.0000 - 0)$
$= 0.40000$

$k_2 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}) = 0.1\,f(0 + 0.05, 4.0 + 0.2)$
$= 0.1\ (4.2 - 0.05) = 0.1\ (4.15)$
$= 0.41500$

$k_3 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}) = 0.1f\ (0 + 0.05, 4.0 + 0.2075)$
$= 0.1\ (4.2075 - 0.05) = 0.1(4.1575)$
$= 0.41575$

$k_4 = hf(x_0 + \frac{h}{2}, y_0 + k_3) = 0.1\,f(0 + 0.1, 4.0 + 0.41575)$
$= 0.1\ (4.41575 - 0.1) = 0.1\ (4.31575)$
$= 0.43158$

$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
$= \frac{1}{6}(0.4 + 2(0.415) + 2(0.41575) + 0.43158)$
$= \frac{1}{6}(0.4 + 0.830 + 0.83150 + 0.43158)$
$= 0.41551$

For $x_1 = 0.1$

$y_1 = y_0 + \bar{k} = 4.0 + 0.41551$
$= 4.41551$

$k_1 = hf(x_1, y_1) = 0.1\,f(y_1 - x_1) = 0.1\ (4.41551 - 0.1)$
$= 0.1\ (4.31551)$
$= 0.43155$

$k_2 = hf(x_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}) = 0.1\,f(0.1 + 0.05, 4.41551 + 0.215775)$
$= 0.1\ (4.631285 - 0.15) = 0.1(4.481285)$
$= 0.44813$

$k3 = hf(x_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}) = 0.1\,f(0.1 + 0.05, 4.41551 + 0.224065)$
$= 0.1\ (4.639575 - 0.15) = 0.1\ (4.489575)$
$= 0.44896$

$k_4 = hf(x_1 + h, y_1 + k_3) = (0.1 + 0.1, 4.41551 + 0.44896)$
$= 0.1\ (4.86447 - 0.2) = 0.1\ (4.66447)$
$= 0.46645$

Example 2-6. (Continued)

$$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k3 + k4)$$

$$= \frac{1}{6}(0.43155 + 2(.44813) + 2(.44896) + 0.46645)$$

$$= \frac{1}{6}(0.43155 + 0.89626 + 0.89792 + 0.46645)$$

$$= \frac{1}{6}(2.692218)$$

$$= 0.44870$$

For $x_3 = 0.2$

$$y_2 = y_1 + \bar{k} = 4.41551 + 0.44870$$

$$= 4.86421$$

$$k_1 = hf(x_2, y_2) = 0.1\ (4.86421 - 0.2) = 0.1\ (4.66421)$$

$$= 0.46642$$

$$k_2 = hf\left(x_2 + \frac{h}{2}, y_2 + \frac{k_1}{2}\right) = 0.1f(0.2 + 0.05,\ 4.86421 + 0.23321)$$

$$= 0.1\ (5.09742 - 0.25) = 0.1(4.84742)$$

$$= 0.48474$$

$$k_3 = hf\left(x_2 + \frac{h}{2}, y_2 + \frac{k2}{2}\right) = 0.1f(0.2 + 0.05,\ 4.86421 + 0.24237)$$

$$= 0.1\ (5.10658 - 0.25) = 0.1\ (4.85658)$$

$$= 0.48566$$

$$k_4 = hf(x_2 + h,\ y_2 + k_3) = 0.1f(0.2 + 0.1,\ 4.86421 + 0.48566)$$

$$= 0.1\ (5.34987 - 0.3) = 0.1\ (5.04987)$$

$$= 0.50499$$

$$k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$= \frac{1}{6}(0.46642 + 2\ (0.48474) + 2\ (0.48566) + 0.50499)$$

$$= \frac{1}{6}(0.46642 + 0.96948 + 0.97132 + 0.50499)$$

$$= \frac{1}{6}(2.91221)$$

$$= 0.48537$$

For $x_3 = 0.3$

$$y_3 = y_2 + \bar{k} = 4.86421 + 0.48537$$

$$= 5.34958$$

$$k_1 = hf(x_3, y_3) = 0.1(y_3 - x_3) = 0.1(5.34958 - 0.3)$$

$$= 0.1\ (5.04958)$$

$$= 0.50496$$

Example 2-6. (Continued)

$k_2 = hf\left(x_3 + \frac{h}{2}, y_3 + \frac{k_1}{2}\right) = 0.1f(0.3 + 0.05, 5.34958 + 0.25248)$
$= 0.1(5.60206 - 0.35) = 0.1(5.25206)$
$= 0.52521$

$k_3 = hf\left(x_3 + \frac{h}{2}, y_3 + \frac{k_2}{2}\right) = 0.1f(0.3 + 0.05, 5.34958 + 0.262605)$
$= 0.1(5.612185 - 0.35) = 0.1(5.202185)$
$= 0.52622$

$k_4 = hf(x_3 + h, y_3 + k_3) = 0.1f(0.3 + 0.1, 5.34958 + 0.52622)$
$= 0.1(5.8758 - 0.4) = 0.1(5.4758)$
$= 0.54758$

$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \frac{1}{6}(0.50496 + 2(0.52521) + 2(0.52622) + 0.54758)$
$= \frac{1}{6}(0.50496 + 1.05042 + 1.05244 + 0.54758) = \frac{1}{6}(3.15918)$
$= 0.52590$

For $x_4 = 0.4$

$y_4 = y_3 + \bar{k}$
$= 5.34958 + 0.52590$
$= 5.87548$

$k_1 = hf(x_4, y_4) = 0.1(5.87548 - 0.4)$
$= 0.54755$

$k_2 = hf\left(x_4 + \frac{h}{2}, y_4 + \frac{k_1}{2}\right) = 0.1f(0.4 + 0.05, 5.87548 + 0.273775)$
$= 0.1(6.149255 - 0.45) = 0.1(5.699255)$
$= 0.56993$

$k_3 = hf\left(x_4 + \frac{h}{2}, y_4 + \frac{k_2}{2}\right) = 0.1f(0.4 + 0.05, 5.87548 + 0.284965)$
$= 0.1(6.160445 - 0.45) = 0.1(5.710445)$
$= 0.57104$

$k_4 = hf(x_4 + h, y_4 + k_3) = 0.1f(0.4 + 0.1, 5.87548 + 0.57105)$
$= 0.1(6.44653 - 0.5) = 0.1(5.94653)$
$= 0.59465$

$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \frac{1}{6}(0.54755 + 2(0.56993) + 2(0.57104) + 0.59465)$
$= \frac{1}{6}(0.54755 + 1.13986 + 1.14208 + 0.59465) = \frac{1}{6}(3.42414)$
$= 0.57069$

Example 2-6. (Continued)

For $x_5 = 0.5$

$y_5 = y_4 + \bar{k} = 5.87548 + 0.57069$

$= 6.44617$

$k_1 = hf(x_5, y_5) = 0.1\ (6.44617 - 0.5)$

$= 0.59462$

$k_2 = hf\left(x_5 + \frac{h}{2}, y_5 + \frac{k_1}{2}\right) = 0.1f\ (0.5 + 0.05,\ 6.44617 + 0.29731)$

$= 0.1\ (6.74348 - 0.55) = 0.1\ (6.19348)$

$= 0.61935$

$k_3 = hf\left(x_5 + \frac{h}{2}, y_5 + \frac{k_2}{2}\right) = 0.1f\ (0.5 + 0.05,\ 6.44617 + 0.309675)$

$= 0.1\ (6.755845 - 0.55) = 0.1\ (6.205845)$

$= 0.62058$

$k_4 = hf\ (x_5 + h,\ y_5 + k_3) = 0.1f\ (0.5 + 0.1,\ 6.44617 + 0.62058)$

$= 0.1\ (7.06675 - 0.6) = 0.1\ (6.46675)$

$= 0.64668$

$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \frac{1}{6}(0.59462 + 2(0.61935) + 2(0.62058) + 0.64668)$

$= \frac{1}{6}(0.59462 + 1.23870 + 1.24116 + 0.64668) = \frac{1}{6}(3.72116)$

$= 0.62019$

For $x_6 = 0.6$

$y_6 = y_5 + \bar{k} = 6.44617 + 0.62019$

$= 7.06636$

$k_1 = hf\ (x_6, y_6) = 0.1\ (7.06636 - 0.6)$

$= 0.64664$

$k_2 = hf\left(x_6 + \frac{h}{2}, y_6 + \frac{k_1}{2}\right) = 0.1f\ (0.6 + 0.05,\ 7.06636 + 0.32332)$

$= 0.1\ (7.38968 - 0.65) = 0.1\ (6.73968)$

$= 0.67397$

Example 2-6. (Continued)

$$k_3 = hf\left(x_6 + \frac{h}{2}, y_6 + \frac{k_2}{2}\right) = 0.1f\ (0.6 + 0.05,\ 7.06636 + 0.336985)$$

$$= 0.1\ (7.403345 - 0.65) = 0.1\ (6.753345)$$

$$= 0.67533$$

$$k_4 = hf\ (x_6 + h,\ y_6 + k_3) = 0.1\ (0.6 + 0.1,\ 7.06636 + 0.67533)$$

$$= 0.1\ (7.74169 - 0.7) = 0.1\ (7.04169)$$

$$= 0.70417$$

$$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \frac{1}{6}(0.64664 + 2(0.67397) + 2(0.67533) + 0.70417)$$

$$= \frac{1}{6}(0.64664 + 1.34794 + 1.35066 + 0.70417)$$

$$= \frac{1}{6}(4.04941)$$

$$= 0.67490$$

For $x_7 = 0.7$

$$y_7 = y_6 + \bar{k} = 7.06636 + 0.67490$$

$$= 7.74126$$

$$k_1 = hf\ (x_7, y_7) = 0.1\ (7.74126 - 0.7)$$

$$= 0.70413$$

$$k_2 = hf\left(x_7 + \frac{h}{2}, y_7 + \frac{k_1}{2}\right) = 0.1f\ (0.7 + 0.05,\ 7.74126 + 0.352065)$$

$$= 0.1\ (8.093325 - 0.75) = 0.1\ (7.343325)$$

$$= 0.73433$$

$$k_3 = hf\left(x_7 + \frac{h}{2}, y_7 + \frac{k_2}{2}\right) = 0.1f\ (0.7 + 0.05,\ 7.74126 + 0.367165)$$

$$= 0.1\ (8.108425 - 0.75) = 0.1\ (7.358425)$$

$$= 0.73584$$

$$k_4 = hf\ (x_7 + h,\ y_7 + k_3) = 0.1f\ (0.7 + 0.1,\ 7.74126 + 0.73584)$$

$$= 0.1\ (8.47710 - 0.8) = 0.1\ (7.6771)$$

$$= 0.76771$$

Example 2-6. (Continued)

$$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \frac{1}{6}(0.70413 + 2(0.73433) + 2(0.73584) + 0.76771)$$
$$= \frac{1}{6}(0.70413 + 1.46866 + 1.47168 + 0.76771)$$
$$= \frac{1}{6}(4.41218$$
$$= 0.73536$$

For $x_8 = 0.8$

$$y_8 = y_7 + \bar{k} = 7.74126 + 0.73536$$
$$= 8.47662$$

The results are tabulated in Table 5, together with the error. Note the marked improvement in accuracy over the second-order solution (see Table 4).

Table 5. Solution by the Runge-Kutta Fourth-Order Method.

x_i	y_i	k_1	k_2	k_3	k_4	$\bar{k}$	Error $y_i - y$ exact
0	4.00000	0.40000	0.41500	0.41575	0.43158	0.41551	0.00000
0.1	4.41551	0.43155	0.44813	0.44896	0.46645	0.44870	0.0000
0.2	4.86421	0.46642	0.48474	0.48566	0.50499	0.48537	-0.00001
0.3	5.34958	0.50496	0.52521	0.52622	0.54758	0.52590	0.00000
0.4	5.87548	0.54755	0.56993	0.57104	0.59465	0.57069	-0.00002
0.5	6.44617	0.59462	0.61935	0.62058	0.64668	0.62019	-0.00001
0.6	7.06636	0.64664	0.67397	0.67533	0.70417	0.67490	0.00000
0.7	7.74126	0.70413	0.73433	0.73584	0.76771	0.73536	-0.00001
0.8	8.47662						

In Example 2-6, second-order and fourth-order Runge-Kutta solutions of Eq. 2-121 are included.

A number of "predictor-corrector" methods have been developed. The best known, that of Milne, requires a knowledge of the values of y at four consecutive values of x. These values may be determined by a Runge-Kutta method or other self-starting methods. By Milne's method, a value of y at a new point y_{i+1} is predicted by the formula

$$y_{i+1} = y_{i-3} + \frac{4h}{3}(2y'_{i-2} - y'_{i-1} + 2y'_i) \qquad (2\text{-}128)$$

where y_{i-3}, y_{i-2}, y_{i-1}, y_{i+1} are successive values of y at points on the x axis equally spaced by the interval h, and y'_k denotes the derivative $\frac{dy}{dx}$, evaluated at a point (x_k, y_k). From the predicted y_{i+1} and y'_{i+1}, a corrected value for the new y, denoted $\bar{y}_{i+1}$, is obtained from the formula

$$\bar{y}_{i+1} = y_{i-1} + \frac{h}{3}(y'_{i-1} + 4y'_i + y'_{i+1}) \qquad (2\text{-}129)$$

Once the original four points have been obtained, the computation by Milne's method proceeds more rapidly than does a Runge-Kutta computation of the same step size.

Any of the methods described in this paragraph can be expanded to solve systems of first-order linear equations. A higher-order equation can always be reduced to a system of first-order equations, as follows. Consider the nth-order equation in the general form

$$\frac{d^n y}{dx} = f\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \ldots, \frac{d^{n-1}y}{dx^{n-1}}\right) \qquad (2\text{-}130)$$

Let $z_1 = \frac{dy}{dx}$, $z_2 = \frac{d^2y}{dx^2}$, ..., $z_{n-1} = \frac{d^{n-1}y}{dx^{n-1}}$. Then the following set of first-order equations is equivalent to Eq. 2-130:

$$\left.\begin{aligned} \frac{dz_1}{dx} &= f(x, y, z_1) \\ \frac{dz_2}{dx} &= f(x, y, z_1, z_2) \\ &\cdots\cdots \\ \frac{dz_{n-1}}{dx} &= f(x, y, z_1, z_2, \ldots, z_{n-1}) \end{aligned}\right\} \qquad (2\text{-}131)$$

Thus, the numerical solution of higher-order differential equations is straightforward.

2-4.6 METHODS FOR SOLVING SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS

The standard form of a system of linear algebraic equations, with n equations and variables x_i (i=1,2, ..., n) is

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= c_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= c_2 \\ &\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= c \end{aligned}\right\} \qquad (2\text{-}132)$$

In matrix form, Eqs. 2-132 may be expressed as

$$\begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{vmatrix} = \begin{vmatrix} c_1 \\ c_2 \\ \cdots \\ c_n \end{vmatrix} \qquad (2\text{-}133)$$

The left-hand matrix in Eq. 2-133 is the matrix of coefficients A and is a square matrix with n rows and n columns. The column matrices are represented by X and C, respectively. Eq. 2-133 may then be expressed

$$A X = C \qquad (2\text{-}134)$$

The solutions of Eq. 2-134 are, by matrix algebra,

$$X = A^{-1} C \qquad (2\text{-}135)$$

where A^{-1} is the inverse matrix of A.* The inverse of a square matrix is defined by the the relationship

$$A A^{-1} \triangleq I \qquad (2\text{-}136)$$

where I A $\begin{bmatrix} 1\,0\,0 \ldots 0 \\ 0\,1\,0 \ldots 0 \\ 0\,0\,1 \ldots 0 \\ \cdots \\ 0\,0\,0 \ldots 1 \end{bmatrix}$ is the unit, or identity, matrix.

The solutions to a set of simultaneous linear algebraic equations can thus be obtained by inversion of the matrix of coefficients, followed by multiplication of the inverted matrix by the column matrix of constants. The major operation, that of matrix inversion, can be performed by several methods. The simplest is by the application of Cramer's rule. In matrix form, Cramer's rule states that

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot\cdot \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{|A|_{11}}{|A|} & \frac{|A|_{21}}{|A|} & \cdots & \frac{|A|_{n1}}{|A|} \\ \frac{|A|_{12}}{|A|} & \frac{|A|_{22}}{|A|} & \cdots & \frac{|A|_{n2}}{|A|} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{|A|_{1n}}{|A|} & \frac{|A|_{2n}}{|A|} & \cdots & \frac{|A|_{nn}}{|A|} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \cdot\cdot \\ c_n \end{bmatrix} \qquad (2\text{-}137)$$

where $|A|$ is the determinant of A and $|A|_{ij}$ is the cofactor† of a_{ij} in the determinant $|A|$.

Application of Cramer's rule in digital computation requires a large number of operations. An alternative procedure is the Gauss-Seidel iterative method. The equation set, Eqs. 2-132, may be rewritten in the form

$$\begin{aligned} x_1 &= d_1 - b_{12} x_2 - b_{13} x_3 - \cdots - b_{1n} x_n \\ x_2 &= d_2 - b_{21} x_1 - b_{23} x_3 - \cdots - b_{2n} x_n \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n &= d_n - b_{n1} x_1 - b_{n2} x_2 - \cdots - b_{n,n-1} x_{n-1} \end{aligned} \qquad (2\text{-}138)$$

where $b_{ij} = \frac{a_{ij}}{a_{ii}}$ and $d_i = \frac{c_i}{a_{ii}}$.

By defining the matrices

$$B = \begin{bmatrix} 0 & b_{12} & b_{13} \cdots & b_{1n} \\ b_{21} & 0 & b_{23} \cdots & b_{2n} \\ \cdots\cdots & \cdots\cdots & & \\ b_{n1} & b_{n2} & b_{n3} \cdots & 0 \end{bmatrix} \qquad (2\text{-}139)$$

and

$$D = \begin{bmatrix} d_1 \\ d_2 \\ \cdot\cdot \\ d_n \end{bmatrix} \qquad (2\text{-}140)$$

a simple iterative process may be employed, represented by the matrix iteration equation

* Not every square matrix has an inverse. The value of the matrix A -- considered as a determinant for this operation -- cannot equal zero since, in computing the inverse, division by the determinant is necessary.

† The cofactor is the determinant obtained from $|A|$ by dropping the row and column that contain a_{ij}. The sign of the cofactor is given by $(-1)^{i+j}$.

$$X^{(k+1)} = D + BX^{(k)} \qquad (2\text{-}141)$$

Eq. 2-141 states that an improved matrix $X^{(k+1)}$ can be obtained by multiplying the preceding matrix $X^{(k)}$ by B and adding the result to D. Eq. 2-141 is the original method of Gauss. The improved Gauss-Seidel method divides the matrix B into upper and lower triangular matrices U and L; thus,

$$U = - \begin{vmatrix} 0 & b_{12} & b_{13} & \cdots & b_{1n} \\ 0 & 0 & b_{23} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 \end{vmatrix} \qquad (2\text{-}142)$$

and

$$L = - \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ b_{21} & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & 0 \end{bmatrix} \qquad (2\text{-}143)$$

The matrix iteration equation is

$$X^{(k+1)} = D + UX^{(k)} + LX^{(k+1)} \qquad (2\text{-}144)$$

Eq. 2-144 represents the following process: In the first of Eqs. 2-138, the initial value of all the x's except x_1 is taken as zero. Then $x_1^{(1)} = d_1$. In the second equation, the improved value of x_1 is used, but the remaining x's on the right-hand side are set to zero, so that $x_2^{(1)} = d_2 - b_{21}x_1^{(1)}$, and so on. Both the Gauss and Gauss-Seidel methods converge if the sum of the absolute values of the coefficients b_{ij} is less than or equal to unity in each equation, and is less than unity in at least one equation. This condition can usually be assured by rearranging the equations such that a_{ii} is the largest coefficient.

The Gauss-Seidel method is best suited to automatic computation. The widely-used Crout method is best suited to hand computation.

REFERENCES

1. Wilfred Kaplan, Ordinary Differential Equations, Addison-Wesley, Reading, Mass., 1958.
2. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1944. (Paperback)
3. L. M. Kells, Elementary Differential Equations, 2nd Ed., McGraw-Hill Book Company, Inc., New York, N. Y., 1960.
4. H. B. Phillips, Differential Equations, 3rd Ed., John Wiley & Sons, Inc., New York, N. Y., 1951.
5. M. F. Gardner and J. L. Barnes, Transients in Linear Systems, Vol I, John Wiley & Sons, Inc., New York, N. Y., 1942.
6. V. del Toro and S. R. Parker, Principles of Control System Engineering, McGraw-Hill Book Company, Inc., New York, N. Y., 1960, Chapter 3.
7. Shih-Nge Lin, "Methods of Successive Approximations of Evaluating the Real and Complex Roots of Cubic and Higher Order Equations", J. Math. Phys., Vol. 20, No. 3, Aug. 1941.
8. S. J. Mason, "Feedback Theory – Some Properties of Signal Flow Graphs", Proc. I.R.E., 41, 1144-1156, (1953).
9. S. J. Mason, "Feedback Theory – Further Properties of Signal Flow Graphs", Proc. I.R.E., **44,** 920-926, (1956).
10. W. W. Seifert and C. W. Steeg, Jr., Control Systems Engineering, McGraw-Hill Book Co., Inc., N. Y., N. Y., 1960, Chapters 10-12.
11. W. B. Davenport, Jr., and W. L. Root, An Introduction to the Theory of Random Signals and Noise, McGraw-Hill Book Company, Inc., New York, N. Y., 1958.
12. J. E. Gibson, Nonlinear Automatic Control, McGraw-Hill Book Co., Inc., New York, N. Y., 1963.
13. E. F. Beckenbach, Modern Mathematics for the Engineer, McGraw-Hill Book Company, Inc., 1956, Chapt. 1., "Linear and Nonlinear Oscillations" by S. Lefschetz.
14. R. N. Buland, "Analysis of Nonlinear Servos by Phase-Plane Delta Method", Jour. Franklin Inst. 257, 37-48, (1954).
15. K. Chen, "Quasi-Linearization Techniques for Transient Study of Nonlinear Feedback Control Systems", Trans. AIEE **74,** Part II Applications and Industry, 354-365 (1955).
16. J. E. Gibson, "Nonlinear Systems Design", Part I, Control Engineering, Vol. 5, No. 10 (Oct. 1958) pp. 69-75; Part II, Vol. 6, No. 1 (Jan. 1959), pp. 85-88; Part III, Vol. 6, No. 2 (Feb. 1959) pp. 82-85.
17. P. S. Hsia, "A Graphical Analysis for Nonlinear Systems", Proc. Inst. of Elect. Eng. **99,** Part II, 120-65 (1952).
18. P. Hontoy and P. Janssens, "Topological Analysis Methods for the Solution of Nonlinear Differential Equations. Application to Oscillators Used in Radio Equipment", Rev. H. F. Brussels, 3, 221-244 (1956).
19. A. M. Hopkin, "A Phase-Plane Approach to the Compensation of Saturating Servomechanisms", Trans. AIEE 70, 631-639 (1951).
20. R. E. Kalman, "Phase-Plane Analysis of Automatic Control System with Nonlinear Gain", Trans. AIEE 73, Part II, Applications and Industry, 383-390 (1954).
21. Y.H. Ku, "Nonlinear Analysis of Electro-Mechanical Problems", Journal Franklin Inst., 255, 9-31 (1953).
22. R. E. Kalman, "Analysis and Design Principles of Second and Higher Order Saturating Servomechanisms", Trans. AIEE, **74,** Part II Applications and Industry, 294-310 (1955).
23. Y. H. Ku, "Acceleration Plane Method for Nonlinear Oscillations", Proc. of the Symposium on Nonlinear Circuit Analysis, Polytechnic Inst. of Brooklyn, New York, 129-153, (April 1953).
24. Y. H. Ku, "The Phase-Space Method of Analysis of Nonlinear Control Systems", Trans. ASME, 79, 1897-1903 (1957).

25. Y. H. Ku, "Analysis of Nonlinear Systems with More than One Degree of Freedom by Means of Space Trajectories", Journal Franklin Inst. 259, No. 2, 115-131 (1955).
26. Y. H. Ku, "Analysis of Servomechanisms with Nonlinear Feedback Control", Trans. AIEE 75, Part II Applications and Industry 402-406 (1956).
27. Y. H. Ku, Analysis and Control of Nonlinear Systems, The Ronald Press Co., New York, N.Y., 1958.
28. R. E. Kuba and L. F. Kazda, "A Phase-Space Method for the Synthesis of Nonlinear Servomechanisms", Trans. AIEE 75, Part II Applications and Industry, 282-290 (1956).
29. N. Minorsky, Introduction to Nonlinear Mechanics, J. W. Edwards, Ann Arbor, Mich., 1947.
30. J. G. Truxal, Automatic Feedback Control System Synthesis, McGraw-Hill Book Company, Inc., New York, N. Y., (Chap. 11, Phase Plane Analysis) 1955.
31. J. C. Gille, M. J. Pelegrin, P. Decaulne, Feedback Control Systems, McGraw-Hill Book Co., Inc., New York, N.Y., (Chapters 25-27) 1959.
32. A. A. Andronow and C. E. Chaikin, Theory of Oscillations, translated by S. Lefchetz, Princeton University Press, Princeton, N.J., 1949.
33. J. G. Truxal, Automatic Feedback Control System Synthesis, McGraw-Hill Book Co., Inc., New York, N. Y., (Chapter 10) 1955.
34. R. L. Cosgriff, Nonlinear Control Systems, McGraw-Hill Book Co., Inc., New York, N. Y., 1958.
35. W. W. Seifert and C. W. Steeg, Jr., Control-Systems Engineering, McGraw-Hill Book Co., Inc., New York, N.Y., 1960, Chapters 13 and 14.
36. W. J. Cunningham, Introduction to Nonlinear Analysis, McGraw-Hill Book *Co.*, New York, N. *Y.*, 1958, Chapters 3 and 5.
37. J. G. Truxal, Automatic Feedback Control System Synthesis, McGraw-Hill Book Co., Inc., New York, N.Y., 1955, 559-566.
38. F. H. Clauser, "The Behavior of Nonlinear Systems", Journal. of the Aeronautical Sciences, 23, 411-434 (1956).
39. K. Ogata, "Subharmonic Oscillations of Nonlinear Feedback Control Systems", Trans. ASME, 80, 1802-1808 (1958).
40. Dunstan Graham and Duane McRuer, Analysis of Nonlinear Control Systems, John Wiley and Sons, Inc., New York, 1961, 30-44.
41. A. Ralston and H. S. Wilf, Mathematical Methods for Digital Computers. John Wiley & Sons, Inc., New York, N. Y., 1960.
42. F. B. Hildebrand, Advanced Calculus for Engineers, Prentice - Hall, Inc., New York, N. Y., 1949.
43. R. V. Churchill, Operational Mathematics, McGraw-Hill Book *Co.*, Inc., New York, N. Y., 1958, p. 61.
44. Yutze Chow and Etienne Cassignol, Linear Signal-Flow Graphs and Applications, John Wiley & Sons, Inc., New York, N. Y., 1962.
45. Y. W. Lee, Statistical Theory of Communication, John Wiley & Sons, Inc., New York, N. Y., 1960.
46. Z. Kopal, Numerical Calculus, John Wiley & Sons, Inc., New York, N. Y., 1955.
47. H. T. Davis, Tables of the Higher Mathematical Functions, I, Principia Press, Bloomington, Indiana, 1933.
48. R. H. Pennington, Introductory Computer Methods and Numerical Analysis, MacMillan Company, New York, N. Y., 1965.
49. AMCP 706-327, Engineering Design Handbook, Fire Control Series, Section 1, Fire Control Systems--General,
50. G. A. Korn and T. M. Korn, Mathematical Handbook for Scientists and Engineers, McGraw-Hill Book Company, Inc., New York, N. Y., 1961.

GENERAL BIBLIOGRAPHY OF BOOKS AND PAPERS RELATING TO NONLINEAR SYSTEMS

Describing Functions

1. J. G. Truxal, Automatic Feedback Control System Synthesis, McGraw-Hill Book Co., Inc., New York, N. Y., 1955 (See Chap. 10).
2. R. J. Kochenburger, "A Frequency Response Method for Analyzing and Synthesizing Contactor Servomechanisms", Trans. AIEE 69, Part I., 270-284 (1950).
3. H. D. Greif, "Describing Function Method of Servomechanism Analysis Applied to Most Commonly Encountered Non-linearities", Trans. AIEE 72, Part II, Applications and Industry, 243-248 (1953).
4. E. C. Johnson, "Sinusoidal Analysis of Feedback-Control Systems Containing Nonlinear Elements", Trans. AIEE 71, Part II, Applications and Industry, 169-181 (1952).
5. A. Tustin, "The Effects of Backlash and of Speed-Development Friction on the Stability of Closed-Cycle Control Systems", Journal Inst. Elec. Engrs., (London) Vol. 94, Part IIA, (1947).
6. E. Levinson, "Some Saturation Phenomena in Servomechanisms with Emphasis on the Tachometer Stabilized System", Trans. AIEE 72, Part II, Applications and Industry, 1-9 (1953).
7. R. C. Booton, Jr., "The Analysis of Nonlinear Control Systems with Random Inputs", Proceedings of the Symposium on Nonlinear Circuit Analysis, Polytechnic Inst. of Brooklyn, New York, 369-391, April 1953.
8. R. J. Kochenburger, "Limiting in Feedback Control Systems", Trans. AIEE 72, Part II, Applications and Industry, 180-194 (1953).
9. V. G. Haas, Jr., "Coulomb Friction in Feedback Control Systems", Trans. AIEE 72, Part II, Applications and Industry, 119-126 (1953).
10. M. V. Mathews, "A Method for Evaluating Nonlinear Servomechanisms", Trans. AIEE 74, Part II, Applications and Industry, 114-123 (1955).
11. E. I. Reeves, "Contributions to Hydraulic Control 7 - Analysis of the Effects of Nonlinearity in a Valve-Controlled Hydraulic Drive", Trans. ASME 79, No. 2, 427-432 (1957).
12. S. Demczynski, "Survey of Methods Available for Analysis and Synthesis of Nonlinear Servomechanisms", Electrical Energy I, 279-284 (1957).
13. W. W. Seifert and C. W. Steeg, Jr., Control Systems Engineering, McGraw-Hill Book Co., Inc., New York, N. Y., 1960.
14. J. F. Barrett and J. F. Coales, An Introduction to the Analysis of Nonlinear Control Systems with Random Inputs, Inst. of Elect. Engrs. Monograph No. 154M. London, England, 1955.
15. H. Chestnut, "Approximate Frequency-Response Methods for Representing Saturation and Dead Band", Trans. ASME 76, 1345-1363 (1954).
16. W. W. Seifert, "Experimental Evaluation of Control Systems by Random Signal Measurements", Convention Record of the I.R.E. 1953 National Convention, Part I -- Radar and Telemetry, 94-98, 1953.

Nonlinear Differential Equations

1. N. W. Lachlan, Ordinary Nonlinear Differential Equations in Engineering and Physical Sciences, Oxford University Press, 2nd Edition, 1955.
   [Many applications to physical problems.]
2. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1944. (Paperback)
   [Inexpensive, but very complete catalogue of both linear and nonlinear differential equations.]

3. W. W. Seifert and C. W. Steeg, Jr., Control-Systems Engineering, McGraw-Hill Book *Co.*, Inc., New York, N. Y., 1960, Chapter 3.
[Comprehensive, concise summary.]

4. W. J. Cunningham, Introduction to Nonlinear Analysis, McGraw-Hill Book Co., New York, N. Y., 1958, Chapter 4.
[Treatment from system viewpoint.]

5. Solomon Lefshetz, Differential Equations: Geometric Theory, Interscience Publishers, New York, 1957.
[Definitive work on theory; difficult.]

6. Wilfred Kaplan, Ordinary Differential Equations, Addison-Wesley, Reading, Mass., 1958.
[Simple, readable exposition of fundamental theory; many practice examples.]

PART II
COMPUTING DEVICES USEFUL IN FIRE CONTROL SYSTEMS

CHAPTER 3
THE CLASSIFICATIONS OF COMPUTING DEVICES USED IN FIRE CONTROL SYSTEMS*

3-1 INTRODUCTION

3-1.1 CHARACTERISTICS OF FIRE CONTROL COMPUTERS

The function of a fire control system is, as discussed in Section 1† of the Fire Control Series, to so position a projectile-launching device, or projector, as to cause the projectile to hit the target. This purpose is accomplished by three subsystems: the acquisition and tracking system, the computing system, and the weapon-pointing system. The computing system (generally referred to as simply the "computer" for the sake of convenience) accepts data from the target tracker and from atmospheric and other measurements, computes the required orientation of the projector, and transmits these data to the weapon-pointing system.

The aforenoted functions of a fire-control computer determine its two general basic characteristics:

1. First, a fire-control computer must usually be fast. In many tactical situations, it is important that the time between the detection of a target and the firing of a projectile be minimized. For this reason, it has been found desirable in many fire control systems to incorporate the computer in the tracking loop. In this case, the computation is performed on the same time base as that on which incoming tracking data are received. Such a computer is termed a real-time computer. If, on the other hand, the computer is not incorporated in a data loop, it may operate at speeds either faster or slower than real time.

2. Second, the fire control computer must be extremely accurate. Errors in components tend to accumulate, and usually cannot be reduced by feedback. The only effective overall feedback is obtained from the observation of prior firings. While information obtained in this manner is valuable when the target is fixed or moving at low velocity, this information-transfer process is too slow to be of much help in reducing errors against high-speed targets; in addition, firings necessarily disclose the position of the weapon. By way of contrast, a homing guided missile is continually measuring the error in the missile-target line of sight; thus, computers for homing guided missiles may have accuracy requirements that are much Less stringent than those for fire control systems.

Since the computer must be located in proximity to the rest of the weapon system, it must have qualities of portability, reliability, ease of adjustment, and freedom from disturbances caused by the environment which are commensurate with those of the rest of the system. These qualities are not easy to combine with the requirements for high speed and high accuracy.

3-1.2 CLASSIFICATION SCHEMES

The fire control system designer is faced with the problem of designing a fast, accurate, compact, and rugged computer which will mechanize the mathematical model of the computer portion of the weapon system. To carry out this task, he has the choice of a wide variety of computing devices and systems: some very old, and others just out of

---

* By E. St. George, Jr.

† Fire Control Systems - General (AMCP 706-327).

the laboratory; some complex and some very simple.

For the purpose of discussing the vast field of fire-control computers, it is useful to consider three classification schemes: (1) from the viewpoint of the user, (2) from the viewpoint of the system designer, and (3) from the viewpoint of the component designer. First of all, however, it is desirable to identify the essential features of any computer or computing device.

3-1.3 BASIC COMPUTER CONCEPTS

Excluding direct analogs, in which one physical phenomenon is simulated by another physical phenomenon that has an analogous behavior, all computing processes -- whether they be manual or automatic, digital or analog -- comprise the elements of computation, programming, memory, input, and output. These elements are best illustrated by an analysis of hand computation.

In solving a complex problem by hand computation, the problem must be broken down into simple computations which can be carried out mentally. Unless the problem is quite simple, it is necessary to write down the steps to be followed -- the program. As the computational steps are carried out under the instructions of the program, the results are recorded on paper for use in later stages. This sheet of paper constitutes the memory.

The process of computing may be summarized as (1) transfer of data from the input element to the computation element, (2) performance of a series of computations, with the transfer of intermediate results to and from the memory", and (3) transfer of the final result to the output. The sequence of computations performed and the transfers of data are all under the control of the program, as shown in Fig. 3-1.

For more complex calculations, various aids to computation may be introduced, but the basic concept is not changed. For example, a slide rule, adding machine, or desk calculator may be employed as a computer instead of the human brain. Tables of mathematical functions may augment the paper-and-pencil memory.

When automatic computers are considered, it is found that analog computers perform all parts of a complex calculation simultaneously, so that the memory element disappears completely; also, the programming function is primarily concerned with the interconnections between a large number of computing elements and a large number of inputs and outputs. In a digital computer, on the other hand, the computing element is relatively simple, while the memory may be large and complex, and divided into various categories, dependent primarily upon speed of access. Thus, the basic concept of the computing process applies, with some modification, to all computers from the simplest hand computation to the largest electronic digital computer.

3-1.4 USER CLASSIFICATIONS

From the viewpoint of the user, or operator, it makes little difference whether the computer is digital or analog, electronic or mechanical, as long as it provides the requisite inputs and outputs, and has the required speed and accuracy. The user, therefore, will classify computers primarily by their degree of automaticity. The first classification schemes to be discussed (see pars. 3-2 through 3-4) consider both computing devices that are primarily aids to a chiefly manual computation and computers that are wholly automatic, or almost so. A second classification of importance to the user (see par. 3-7) divides computing devices into special-purpose and general-purpose groups.

3-1.5 DESIGNER CLASSIFICATIONS

From the viewpoint of the system designer, the decision as to the particular type of computer to be employed (i.e., a digital computer, a digital differential analyzer, or an analog computer; see par. 3-5) rests upon a number of interacting factors. Although the designer's own background should, ideally, not influence the decision, it is, practically, often one of the prime factors. However, the decision is influenced, and possibly even forced, by such purely technical considera-

* Additional input data may also be entered at various stages of the computation.

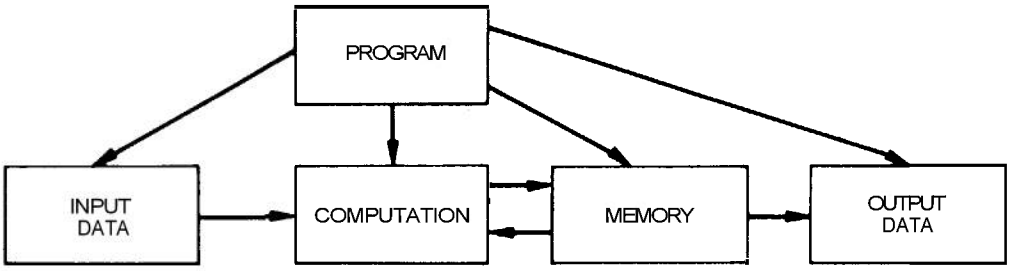


Figure 3-1. The computing process.

tions as the number and form of the inputs to be fed into the computer and of the outputs required, the accuracy required, the specific computations to be performed, the speed at which solutions must be obtained, etc. Also involved, even in the choice of the basic computer type, are such questions as the range of variables to be handled and the related scale-factor questions, the reliability, and the ease with which the computer could be adapted to handle problems involving different operating conditions or even different basic computations from those originally planned.

From the viewpoint of the component designer, computers may best be classified according to the physical means employed to perform the computations (see par. 3-6). While there has been a strong trend toward electronic computers in recent years, electromechanical and fluid-operated computing devices are of great importance, particularly in the specialized fire-control field.

3-2 MANUAL COMPUTING DEVICES

A wide variety of useful aids exists for use in manual computing. Of these, the most useful consists of a pencil and a sheet of paper, which provide the simplest possible auxiliary to the human memory. A natural development from this is the provision of tables of commonly-used functions,

3-2.1 FIRING TABLES

The firing table (see Chapter 3 of Section 1, Fire Control Systems - General) is a basic computing tool in field-artillery fire control. Here, the problem is to orient a gun -- located at a point whose positional coordinates and altitude are known -- *so* as to fire on a target whose positional coordinates arid altitude have been measured by various observational techniques. The use of a firing table is restricted, of course, to circumstances in which sufficient time is available for manual computations. The results of the computations are three pertinent variables:

1. Azimuth of fire
2. Gun elevation angle
3. Time of flight (for fuze settings, and time-on-target applications).

To compute these values, the following data are required:

1. Muzzle velocity
2. Aerodynamic characteristics of the projectile
3. Position of the target with respect to the weapon, specified in terms of:
   a. Range to target
   b. Height of target with respect to weapon
   c. Azimuth to target.
4. Meteorological message, consisting of:
   a. Air pressure and/or air density
   b. Air temperature
   c. Wind velocity
   d. Latitude.

The firing table tabulates weapon elevation angle as a function of range for a given type of weapon and ammunition, under standard atmospheric conditions. To provide corrections for atmospheric variations from standard, unit corrections are listed for each variable. The meteorological message gives the necessary data to acquire the number of units variation from standard, from which the correction is acquired by the product of unit correction and number of units variation. For

the total procedure, the solution is a series of lookups, multiplications and algebraic additions of corrections.

3-2.2 NOMOGRAMS

Other aids to manual computation are based on the nomogram, or alignment chart. The simplest nomogram consists of three parallel scales, A, B, and C in Fig. 3-2, on which are marked any three functions, f(A), f(B), and f(C), of the variables A, B, and C. If A and B are the independent variables, a straight line passing through the selected values of A and B will intersect C at a point determined by

$$f(C) = Rf(B) + (1 - R)f(A) \qquad (3\text{-}1)$$

where R = c/b, as defined in Fig. 3-2.

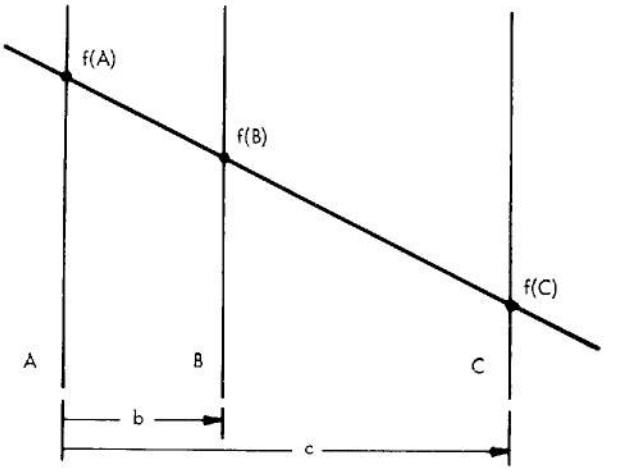


Figure 3-2. Basic nomogram.

If logarithmic functions are chosen, the nomogram may be used to compute products or quotients. This type of nomogram is so useful that the logarithmic scales and the index line are commonly engraved on slides to form the familiar slide rule. A wide variety of computations may be performed by means of special slide rules and a variety of more complex nomograms.

This brief discussion is intended to provide an introduction to topics which are not covered in detail in this handbook. Further information on the involved process of computing firing tables, and details on the construction of various types of nomograms will be found in References 1 through 5.

3-3 MANUALLY OPERATED AUTOMATIC COMPUTERS

The extension of the concept of aiding the manual computation of fire-control data leads naturally to the use of general-purpose digital computers. A simple form of this type of computer is the mechanical desk calculator, which is sometimes employed in fire-control work. This calculator can perform addition, subtraction, multiplication, and division (and in some cases can extract the square root), and can thus supply the arithmetic or computational element in a computing system. The memory and programming are supplied manually.

Any general-purpose digital computer which has sufficient storage capacity for the firing tables can be programmed to solve the field-artillery fire control problem. A portable computer, known as FADAC,* has been developed specifically for the solution of this problem in the field. The FADAC is a typical general-purpose digital computer, and indeed can be readily programmed to perform accounting operations, and other computations. It is distinguished from the fully automatic computers used with fire control systems in two respects:

1. It has manual, rather than automatic, inputs and outputs.
2. It need not operate in real time.

With these differences kept in mind, the descriptions of automatic computers in later chapters may be applied to any of the manually operated automatic computers.

3-4 AUTOMATIC COMPUTING DEVICES

Most of the computers with which one is concerned in fire-control work are automatic

* See Chapter 1 of Section 1 of the Fire Control Series and Chapter 4 of the present section.

Computers that operate with physical rather than mathematical variables as inputs and outputs. Such computers operate in real time and serve as a functional element of a fire control system. Real-time computers are characterized by the same elements (input, program, computation, memory, and output) that are universal to computers, and they may be either digital or analog. Their distinguishing characteristics are an ability to perform computations at the same rate as that at which the input data change, and the provision of equipment to convert the input data into a form acceptable to the computer. Equipment is also provided to convert the computer output into a form suitable for use in positioning the projector.

The requirement for real-time operation makes the fire-control computer a highly specialized design. (General-purpose digital computers generally operate slower than real time; their speed is usually limited by the time of access to the magnetic tape memory most commonly employed for large-volume storage. On the other hand, some electronic analog computers operate considerably faster than real time, making unnecessary the use of drift stabilization in the electronic amplifiers, and making possible the use of cathode-ray-tube output displays.) Real-time analog computers require highly-stable electronic amplifiers and electromechanical elements which have good dynamic response. Real-time digital computers must have high-speed circuitry or redundant elements -- generally both are employed -- and must have rapid-access storage.

Introduction of data to a real-time digital computer is accomplished by means of analog-to-digital converters since the data are generally initially generated in analog form. Conversion of the output data is accomplished by the provision of digital-to-analog converters on the adjustable axes of the projector. The converter outputs are compared with the computer outputs and the differences are employed as the error signals to the projector power servos.

Introduction of data to a real-time analog computer involves only the conversion of the data to a form usable by the computer, generally a voltage or a shaft angle. Control of the projector is usually obtained by means of synchro data transmission.

3-5 DIGITAL, DIGITAL DIFFERENTIAL ANALYZER, AND ANALOG COMPUTING DEVICES

The most basic decision made by the fire-control-system designer in the design of the computing system is the choice of the type of computer to be used -- i.e., whether it will be a digital computer, a digital differential analyzer, or an analog computer. The following paragraphs define, and briefly describe these three classes of computers.

A digital computer is one in which the mathematical variables are represented numerically by discrete physical quantities, and all computations are carried out in numerical form. Typical examples of the discrete quantities employed are the motion of a ratchet actuated by a pawl, the magnetic state (whether magnetized or demagnetized) of a core having a pronounced square hysteresis loop, or the electrical state (left-hand or right-hand transistor conducting) of a transistor bistable circuit commonly referred to as a flip-flop circuit. A digital computer is made up of the elements shown in Fig. 3-1, but the computational element is capable only of addition, subtraction, and detection of the sign of a quantity. All other computations are made up of combinations of these basic operations, with the intermediate results transferred to storage between steps.

A digital differential analyzer (frequently abbreviated DDA) is a special form of digital computer in which the variables are represented by trains of electrical pulses (or other discrete quantities). Each pulse represents an increment of the variable and each has an equal value, whereas in a standard* digital computer only those pulses representing the least significant digit of the number are equivalent to an increment in the variable. The DDA is organized much like an analog computer; i.e., particular elements of the machine are designed to perform a particular mathe-

* In the literature, the two classes of digital computers are sometimes distinguished as DDA and general-purpose (GP) computers, but the latter designation is a misnomer in this situation since the standard computer may be either general-purpose or special-purpose.

matical computation (such as multiplication or integration), and these elements are interconnected to perform the complete computation. Since similar basic components are employed, it is possible to look on the DDA as a standard digital computer with unconventional programming, and depending on redundant computing elements to achieve a high solution speed.

An analog computer is one which employs continuous physical quantities to represent the variables. Analog computers are divided into elements which are made up of electrical and mechanical networks so arranged as to produce particular mathematical functions. A given equation is solved by the interconnection of computing elements in the required pattern. For example, an instrument servo with tachometric feedback can accurately reproduce, as a shaft rotation, the time integral of its input voltage. If a shaped potentiometer is coupled to the output shaft, the sine (or some other function) of the integral can be generated.

A special type of analog computer, usually known as a network analyzer, employs electrical networks whose response is represented by the familiar second-order differential equation

$$e = L\frac{di}{dt} + Ri + \frac{1}{C}\int i\,dt \qquad (3\text{-}2)$$

with many variants, depending on the way in which the elements are combined. Assemblages of such analog networks have proved useful in the analysis of the vibration of complex structures, the transient response of electrical power networks, and the characteristics of many systems with distributed parameters.

In Part II of this section of the handbook, detailed descriptions of computer design principles have been segregated into individual chapters on digital computers, digital-differential-analyzers, and analog computers since these classifications are of most concern to the systems designer.

A chapter has also been devoted to comparisons between these classes of computers. Obviously, a digital computer is more flexible in its application to different problems than an analog computer or DDA since a new program can be entered electrically without the necessity of physically changing electrical or mechanical connections. On the other hand, when the inputs and outputs are in analog form, a digital computer requires additional converter equipment. The reliability of a digital computer is inherently greater than that of an analog device since the digital computer is made up of components with discrete or "yes-no" outputs.

The factors of accuracy, speed, cost, size, weight, and power consumption are interrelated in complex ways for all types of computers. Any particular design is a compromise between these factors, which can often be traded-off against one another. For example, with a given design of digital components, the size of a digital computer is proportional to the product of accuracy and speed.

Further discussion of computer comparisons is reserved for Chapter 8.

3-6 TYPES OF PHYSICAL EQUIPMENT EMPLOYED IN COMPUTERS

A classification of computing devices by the physical means employed to carry out the computation yields the four major classes of electronic, mechanical, electromechanical, and fluid computing devices. Both digital and analog devices are found in all these classes.

Electronic computing devices are defined as those having electrical inputs and outputs, and performing computations by means of electrical networks and electronic amplifiers. Modern high-speed digital computers are almost wholly electronic, with such devices as transistor flip-flops and gates and magnetic-core storage elements predominating. Electronic analog computing devices are employed in computers intended for simulation and related operations, but the limited accuracy in such functions as multiplication has limited the application of purely electronic analog devices in fire control computers.

Mechanical computing devices have mechanical inputs and outputs, and compute by means of mechanical components such as linkages, gearing, springs, and cams. The original digital computers of Pascal and Babbage were all mechanical, and they persist in the common desk calculator. Mechanical analog devices were universally employed in early

fire control computers, but now survive principally as components of electromechanical systems.

Electromechanical computing devices may have either electrical or mechanical inputs and outputs in any combination, but more commonly both inputs and outputs are electrical. In digital computation, electromechanical devices of the punched-card variety are employed mainly in accounting machines, and as input-output devices for general-purpose computers. A variety of other electromechanical devices are employed for digital-computer input-output functions: punched paper tape machines, magnetic tape recorders, electrictypewriters, and plottingboards, for example.

Electromechanical analog computing devices combine the accuracy of mechanical elements with the flexibility of electrical interconnection. A common technique is to convert a signal voltage into a shaft rotation by means of a position servo. Various combinations of linear and aonlinear potentiometers and electromagnetic devices may be coupled to the shaft in orderto multiply orto generate functions. Most analog fire control computers are of this type.

Fluid computing devices (i.e., hydraulic and pneumatic computing devices) are actuated by a fluid-pressure input and produce a fluid-pressure as the output. Fluid devices for use as digital computing components have only recently been developed. They are fast, reliable, and occupy little space, and will become a more important factor in the future. Fluid analog devices have been employed for many years in process-control technology, and more recently in. engine-fuel controls. In general, they have been preferred to electronic systems in those applications where electrical signals would create a fire hazard.

The format of Section 3 is to describe particular physical realizations of computing devices within the chapter on the particular class of computer involved.

3-7 SPECIAL-PUkPOSE AND MULTIPUR-POSE COMPUTING DEVICES

Most of the literature on computers is concerned with the design and operation of general-purpose computers intended for a variety of scientific and business computations. This handbook, on the contrary, is concerned primarily with the design of special-purpose computers.

A special-purpose computer is one designed to solve a fixed set of equations, which are preprogrammed into the machine, to a fixed degree of accuracy. Many special applications also pose requirements as to solution speed, type of input, computer size or weight, and environmental conditions.

As previously stated, the requirements of a special application may lead the designer to prefer one type of computer over another. Once the type of computer has been chosen, the design will usually eliminate the provisions for flexibility in operation that account for much of the cost and complexity of multi-purpose computers. The design of a special-purpose computer places great reliance on the ability of the designer to devise ingenious devices which simplify the equipment, He must also overcome formidable problems associated with limitations on size or difficult environmental conditions. Problems of special-purpose computers are covered in detail in Chapter 11.

REFERENCES

1. D. P. Adams, An Index of Nomograms, published jointly by the Technology Press of the Massachusetts Institute of Technology, Cambridge, Massachusetts and John Wiley & Sons, Inc., New York, N.Y., 1950.
2. G. A. Bliss, Mathematics for Exterior Ballistics, John Wiley & Sons, Inc., New York, N.Y., 1944.
3. J. L. Kelley, E. J. McShane, and F. V. Reno, Exterior Ballistics, The University of Denver Press, Denver, Colorado, 1953.
4. AMCP 706-107, Engineering DesignHandbook, Elements of Armament Engineering, Part Two, Ballistics.
5. AMCP 706-140, Engineering DesignHandbook, Trajectories, Differential Effects, and Data for Proiectiles.

CHAPTER 4

DIGITAL COMPUTERS*

4-1 INTRODUCTION

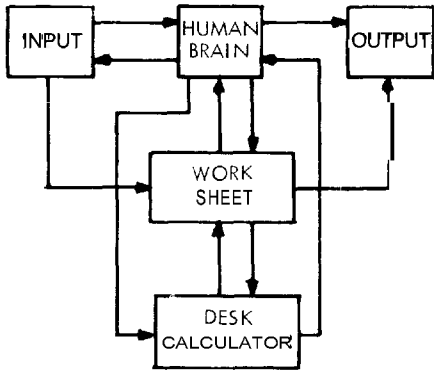
4-1.1 DEFINITION OF A DIGITAL COMPUTER

A digital computer is a calculating machine that, when appropriately programmed, is capable of performing extremely complex numerical mathematics to any accuracy desired. A computer belongs to the "digital" class if it stores and operates upon discrete rather than continuous (analog) quantities. The precision of such a machine is principally determined by the number of digits it is designed to handle,

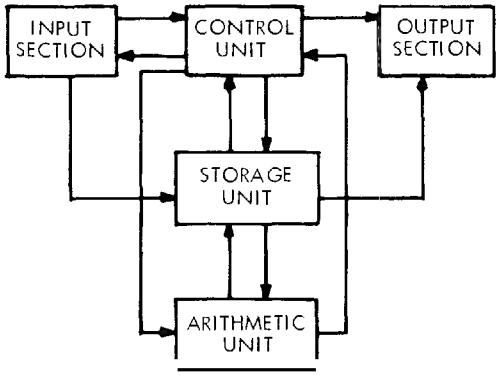
The digital computer employs the basic operations of addition, subtraction, and detection of the algebraic sign of quantity, All Other computations are made up of combinations of these basic operations. Consequently, in order to use the machine for performing more general computations, these must first be reduced to the basicoperations noted and a program or set of instructions must be established to enable the machine to carry out the basic operations in the order required to accomplish the more difficult desired computation.

Some insight into the manner in which a digital computer operates can be gained by comparing the way in which it operates with the way in which a human operator uses a deskcalculator; see Fig. 4-1. In Fig. 4-1(A) the arrows to the human brain represent external input parameters and results from the calculator that must be written down, while the arrows from the brain represent commands and actions. In Fig. 4-1(B) the equivalent flow diagram for a digital computer reflects essentially the same process, with appropriate changes in nomenclature.

A programmed digital computer has certain finite times necessary to perform each of its mathematical operations, and the overall time required for the solution of a complex series is the sum of times used for each part of the series. If the economics of the problem justify added cost and size, very high cyclic rates can be employed and parallel operation (in which all the digits of a number are accepted simultaneously through individual channels rather than serially through a single channel) can be used to reduce the time factor,



(A) Flow Diagram for Simple Computatiori Performed by a Human Operator With the Aid of o Desk Calculator.



(B) Flow Diagram for Digital-Computer Operation.

Figure 4-1. Computation flow diagrams.

* By E. St. George, Jr. and M. M. Miller.

The digital computer is thus not usually a "real time" device; i.e., the solutions are not "in step" with the input parameters. However, modern technology has been developed to the point where digital computers can be used to solve real-time fire control problems,

Incremental digital computers represent something of a cross between digital and analog computers. They operate upon digital values and have a high speed of operation, whichstems from the fact that the computation continuously updates a previously calculated solution, rather than starting each calculation "from scratch". This approach makes use of the fact that calculations upon smoothly varying inputs will have solutions that are smoothly varying. Incremental digital computers include digital differential analyzers (see Chapter 5) and operational digital computers.

4-1.2 NUMBER SYSTEMS

In the design of computing machinery, three number systems are most often encountered. The decimal system, making familiar use of the 10 digits from 0 through 9, is our standard medium for arithmetical calculations and numerical records. The binary system, using the base 2 and employing only combinations of 0's and 1's to express any desired quantity, is most common in digital machines because it offers the most straightforward and economical approach to hardware design. The octal system with the base 8, digits 0 through 7, is closely related to the base-2 system in the hardware necessary to handle it, Furthermore, one using the computer is able to convert numbers from base 2 to base 8 or vice versa merely by inspection. Consequently, the use of base 8 may be encountered, for instance, in test printouts as a means of avoiding the difficulty of handling the large number of 0's and 1's in the pure binary notation.

The sexadecimalnumber systems, using the radix 16, could be used by employing essentially the same designs as binary and octal systems. Apart from this, the radices 3, 10, and 12 are probably the only other ones that have received serious consideration for computing machinery.

Examples of the decimal, binary and octal number systems are included in Information Summaries 4-1, 4-2, and 4-3. Information Summaries 4-4, 4-5, and 4-6 provide conversion rulesanddata that can be used to go from the decimal system to the binary system and vice versa.

4-1.3 FUNCTIONAL PARTS OF A DIGITAL MACHINE [1,3]

As indicated by the digital-computer flow diagram of Fig. 4-1(B), a digital computer must comprise the following five major functional parts:

1. An input section
2. A storage unit
3. An arithmetic unit
4. A control unit
5. An output section

The following paragraphs summarize the essential functions of these five principal computer elements. (More detailed information on these functions is provided in subsequent parts of this chapter,)

Input Section: The input section receives the input data, converts this data into the internal language of the computer, and then transmits the converted data to the appropriate parts of the computer -- primarily to the storage unit. Depending on the characteristics of a particular computer, the input section canreceive the input data in various forms. For example, the data may be in binary-coded decimal form from a typewriter, in analog form, or coded in a special way -- such as on punched cards.

Storage Unit: The storage unit (often called the computer memory) receives data from other elements of the computer, holds it in readiness for subsequent use, and transmits it to appropriate points as directed by the control unit.

Arithmetic Unit: The arithmetic unit must be capable of performing the following specific data-processing tasks:

1. Receive two numbers and be able to distinguish between them.

INFORMATION SUMMARY 4-1. THE DECIMAL NUMBER SYSTEM

Number Rase: 10

Permissible Integers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Integer Multipliers:

$10^{-\infty}$ = zero
.
.

10^{-6} = .000001
10^{-5} = .00001
10^{-4} = .0001
10^{-3} = .001
10^{-2} = .01
10^{-1} = .1
10^{0} = 1

10^{0} = 1
10^{1} = 10
10^{2} = 100
10^{3} = 1000
10^{4} = 10,000
10^{5} = 100,000
10^{6} = 1,000,000
.
.

10^{∞} = Infinity

Example: Decimal Number 732.625

$$
\begin{aligned}
= \quad & 7 \times 10^{2} && \text{or } 700 \\
& +3 \times 10^{1} && \text{or } 30 \\
& +2 \times 10^{0} && \text{or } 2 \\
& +6 \times 10^{-1} && \text{or } 0.6 \\
& +2 \times 10^{-2} && \text{or } 0.02 \\
& +5 \times 10^{-3} && \text{or } 0.005 \\
= \quad & && 732.625
\end{aligned}
$$

INFORMATION SUMMARY 4-2. THE BINARY NUMBER SYSTEM

Number Base: 2

Permissible Integers: 0, 1

Integer Multipliers:

$2^{-\infty}$ = zero

$2^{-4} = \frac{1}{16}$

$2^{-3} = \frac{1}{8}$

$2^{-2} = \frac{1}{4}$

$2^{-1} = \frac{1}{2}$

$2^{0} = 1$

$2^{0} = 1$
$2^{1} = 2$
$2^{2} = 4$
$2^{3} = 8$
$2^{4} = 16$
$2^{5} = 32$
$2^{6} = 64$
$2^{7} = 128$
$2^{8} = 256$
$2^{9} = 512$
$2^{10} = 1024$
$2^{11} = 2048$
$\vdots \quad \vdots$
2^{∞} = Infinity

Example: Decimal 732.625 in Binary

		Binary	Decimal
=	1×2^{9} or	1000000000	512
	$+1 \times 2^{7}$ or	10000000	128
	$+1 \times 2^{6}$ or	1000000	64
	$+1 \times 2^{4}$ or	10000	16
	$+1 \times 2^{3}$ or	1000	8
	$+1 \times 2^{2}$ or	100	4
	$+1 \times 2^{-1}$ or	0.1	0.5
	$+1 \times 2^{-3}$ or	0.001	0.125
=		1011011100.101	732.625

INFORMATION SUMMARY 4-3. ANY NUMBER SYSTEM

Number Base: N

Permissible Integers: 0, etc., up to but not including N

Integer Multipliers:

$N^{-\infty}$ = zero

$N^{-4} = \frac{1}{N \cdot N \cdot N \cdot N}$

$N^{-3} = \frac{1}{N \cdot N \cdot N}$

$N^{-2} = \frac{1}{N \cdot N}$

$N^{-1} = \frac{1}{N}$

$N^{0} = 1$

$N^{0} = 1$
$N^{1} = N$
$N^{2} = N \cdot N$
$N^{3} = N \cdot N \cdot N$
$N^{4} = N \cdot N \cdot N \cdot N$

N^{∞} = Infinity

Example: Decimal Number 732.625 in Number Base N=8

			Decimal	
=	1×8^3	or 1000		512
	$+3 \times 8^2$	or 300		192
	$+3X\ 8^1$	or 30		24
	$+4 \times 8^0$	or 4		4
	$+5 \times 8^{-1}$	or 0.5		0.625
=		1334.5	=	732.625

In the octal (RaseN = 8) System

In the decimal (Base N = 10) System

INFORMATION SUMMARY 4-4. CONVERSION RULES — DECIMAL TO BINARY

To convert a decimal number to a binary number, successively divide the given decimalnumber by 2 and record the remainders of each division. When zero is reached, the remainders taken in reverse order express the binary equivalent of the decimal number.

For example, this process for the decimal number 327 is as follows:

2)327	
2)163	1
2)81	1
2)40	1
2)20	0
2)10	0
2)5	0
2)2	1
2)1	0
0	1

Therefore, the binary equivalent of 327 is 101000111.

INFORMATION SUMMARY 4-5. CONVERSION RULES — BINARY TO DECIMAL

To convert a binary number to a decimal number, use the formula

$$N = 2^{n-1}d_n + 2^{n-2}d_{n-1} + 2^{n-3}d_{n-2} + \ldots + 2^0 d_1$$

where

N = the decimal equivalent of the given binary number
n = the number of digits in the binary number
d_i = the value (0 or 1)of the ith digit (i = 1, 2, . . ., n)
d_1 = the least significant digit

For example, consider the binary number 11011001, which has eight digits. Then,

$$N = 2^7(1) + 2^6(1) + 2^5(0) + 2^4(1) + 2^3(1) + 2^2(0) + 2^1(0) + 2^0(1)$$

$$= 128 + 64 + 16 + 8 + 1$$

$$= 217$$

See Information Summary 4-6 for values of the powers of **2.**

INFORMATION SUMMARY 4-6. BINARY EQUIVALENT OF DECIMAL NUMBERS

Binary Equivalent	2^n Form	Decimal Number
1	2^0	1
10	2^1	2
11	2^1+2^0	3
100	2^2	4
101	2^2+2^0	5
110	2^2+2^1	6
111	$2^2+2^1+2^0$	7
1000	2^3	8
1001	2^3+2^0	9
1010	2^3+2^1	10
1011	$2^3+2^1+2^0$	11
1100	2^3+2^2	12
1101	$2^3+2^2+2^0$	13
1110	$2^3+2^2+2^1$	14
1111	$2^3+2^2+2^1+2^0$	15
10000	2^4	16
100000	2^5	32
etc., with the same number of zeroes as in the exponent of the 2^n form.	2^6	64
	2^7	128
	2^8	256
	2^9	512
	2^{10}	1,024
	2^{11}	2,048
	2^{12}	4,096
	2^{13}	8,192
	2^{14}	16,384
	2^{15}	32,768
	2^{16}	65,536
	2^{17}	131,072
	2^{18}	262,144
	2^{19}	524,288
	2^{20}	1,048,576

2. Carry out such simple arithmetic operations as addition and subtraction. (In addition, a capability for multiplication and division is usually required, and frequently the capability of performing other operations is provided.)

3. Carry out the logical operation of determining which of two numbers is larger.

4. Transmit the processed result of its operations to an appropriate point -- usually the memory.

Control Unit: The control unit oversees each individual operation in the sequence of computer operations that is required to solve a particular problem. In order to carry out this function, the control unit must possess the following capabilities:

1. Supply master timing signals.

2. Control switching between the various computer elements.

3. Initiate each computer operation and sense its completion.

4. Transmit the result of a computer operation to storage, but retain a knowledge of how to find it again.

5. Decide upon the next operation to be performed, based on the results of the preceding operation and any instructions that

have been placed in the storage unit.

6. Receive and interpret stored instructions so as to be able to appropriately apply the foregoing capabilities,

Output Section: The output section receives computed data in the internal language of the computer and then converts these data to a useful output form.

It should be noted that it is in the nature of the control circuitry that general-purpose and special-purpose digital computers differ from one another, A general-purpose computer stores the sequence of required operations -- together with the data that are to be operated upon -- in its own storage unit, and can perform many different operations. A special-purpose computer, on the other hand, is designed for a far more limited capability. For a specified precision, it can usually carry out its specific functions much faster, and with much less hardware, than a general-purpose computer.

4-2 SYSTEM DESIGN

4-2.1 EQUATIONS TO 13E SOLVED

A digital computer can be used to find solutions for linear equations, linear differential equations, matrices, partial differential equations, and the roots of polynomials. It can also solve many other types of equations. In these applications, however, the computer is only able to perform directly the processes of arithmetic. Therefore, to be acceptable to a digital computer, an equation or function must be converted into a numerical approximation. In the case of a trigonometric function, an arithmetic method must be used to obtain an approximation if, for example, the sine of an angle is required. A computer with very large storage could remember a full set of trigonometric tables, but to eliminate the need for storage, it is possible to use the technique of expanding sin' x into a rapidly converging series and substitute the values of 'x into this expression. It is possible to solve differential and integral equations by numerical approximation as well.

As a first step toward designing a digital computer to carry out some particular set of computations, the original equations are broken down into various subroutines, such as finding the square root and taking the sine of a number. An experienced computer programmer working with the designer will be able to specify the way in which the various steps must be interrelated. Eventually, the number of words of input and output data can be determined and a size of memory can be established that will be adequate to contain the problem data, the program, intermediate results, and constants.

The basic layout of the program-computer combination is a blend of system analysis, circuit design, and logical design. The system analysis creates a mathematical model and a set of requirements for its solution. The circuit design creates combinations of reliable components to store information and to operate on information according io fixed rules. Logical design produces a set of wiring diagrams that connect the components into a complete machine. Actually, the three functions overlap considerably; in particular, the logic design goes hand in hand with system and circuit efforts.

4-2.2 USE OF NUMERICA L ANALYSIS AND OTHER MATHEMATICAL TECHNIQUES

In considering how a digital computer solves an equation, or a set of equations, it is instructive to look first at the so-called "brute force" technique. This approach consists of simply trying successively all possible values of the independent variable and thereby determining whether or not there is a solution.

As an example of the "brute force" technique, consider the following case::: -- in which the technique might actually prove to be a practical method of solution. Included in this example are some of the programming tricks by means of which the computation time and the storage requirements of the computer can be reduced.

* Adapted from Chapter 16 of Ref. 1.

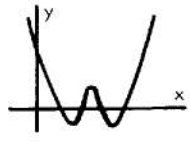
Assume that it is desired to find the roots of the polynominal

$$y = Ax^4 + Bx^3 + Cx^2 + Dx + E \qquad (4-1)$$

by means of a digital computer. (This relationship for a given set of real coefficients would have four real roots; see sketch in Fig. 4-2.) Assume that at the start of the problem only the following knowledge exists:

1. All four roots of the polynomial are real and lie in the region $0 < x < 10$.

2. No two roots differ by as little as 0.001.



$y = Ax^4 + Bx^3 + Cx^2 + Dx + E$

Figure 4-2. Fourth-degree polynomial, all roots real and positive.

Application of the "brute force" technique in this case means evaluating the polynomial on the right-hand side of Eq. 4-1 for successive values of x spaced at intervals of 0.001, starting with zero and continuing until all four roots have been determined. Since a change in the sign of y evidences the presence of a root, the computing' procedure to be used at each of the successive values of x will be as follows:

1. The computer evaluates y and examines its sign to see whether it has changed from the preceding evaluation.

2. If the sign has not changed, the computer evaluates y for the next incremental value of x.

3. If the sign has changed, this means that a root has been found -- to an accuracy determined by the choice' of interval that was made. Therefore, the computer prints out the result.

4. Then, in order to determine whether the computation process should stop, the computer asks itself whether it has yet located all four roots. If the answer is affirmative, computation ceases; if negative, the computer proceeds to evaluate y at the next incremental value of x.

By means of the "brute force" procedure outlined, a typical digital computer could evaluate the specified polynomial- for the requisite values of x (10,000 values maximum) in just a few seconds. Unless this evaluation has to be repeated many more times than this, one might well be willing to sacrifice the few seconds of computing time required, in order to avoid the labor involved in coding a more complex method of solution. Therefore, while the brute-force technique would not generally be used in practice, it is not always an unrealistic method.

It should be noted that a very simple rearrangement of the polynomial would considerably simplify the calculations required of the computer, even if it continued to use the basic brute-force technique that has been outlined. Each evaluation of the polynomial of Eq. 4-1 in its present form requires a minimum of seven multiplications and four additions. (While it could be evaluated by determining x^4, multiplying that result by A, and then starting all over again by determining x^3, and so forth, this would be a wasteful procedure involving a total of ten multiplications and four additions. It would be more economical of effort to start by evaluating x^2 first, followed by x^3 and x^4.) Inasmuch as a multiplication takes much more computer time than an addition, it is worthwhile to attempt to reduce the number of multiplications required in order to evaluate the polynomial. One means of achieving this objective is to divide both sides of Eq. 4-1 by A, thereby yielding

$$z = \frac{y}{A} = x^4 + \frac{B}{A}x^3 + \frac{C}{A}x^2 + \frac{D}{A}x + \frac{E}{A}$$

$$x^4 + Px^3 + Qx^2 + Rx + S \qquad (4-2)$$

Thus, in exchange for adding four divisions that are each performed only once, it has been possible to obviate the necessity for performing one multiplication ($A \cdot x^4$) thousands of times. Eq. 4-2 requires six multi-

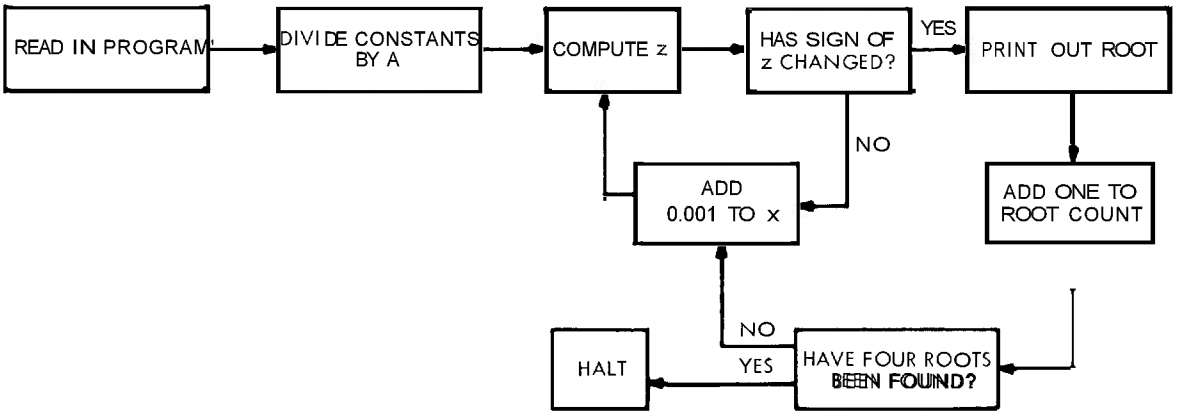
plications and four additions. The number of operations can be still further reduced by rearranging Eq. 4-2 into the form

$$z = S + x\left\{R + x\left[Q + x(P + x)\right]\right\} \qquad (4\text{-}3)$$

This equation requires only three multiplications and four additions. The corresponding computer flow diagram is depicted in Fig. 4-3.

The brute-force technique would not normally be employed by a sophisticated programmer. Instead, some of the techniques known as numerical analysis would be employed. While these techniques are extremely powerful, they have been developed in a pragmatic, rather than a theoretical, context. Numerical analysis thus comprises a body of individual approximation techniques, each having application to a particular class of equations. The choice of method is determined both by the form of the equations to be solved and the capabilities of the computer; this choice thus depends heavily on the experience and ingenuity of the programmer.

The application of numerical analysis to the important problems of the representation of functions, the fitting of empirical data, the solution of linear simultaneous equations, the solution of nonlinear equations, and the evaluation of integrals and differential equations are briefly examined in the remainder of par. 4-2.2. For details of the methods noted, the reader is referred to the bibliography at the end of this chapter.



ORIGINAL FORM OF POLYNOMIAL

$$y = Ax^4 + Bx^3 + Cx^2 + Dx + E$$

MODIFIED FORM EMPLOYED FOR EASE OF COMPUTATION

$$z = \frac{y}{A} = S + x\left\{R + x\left[Q + x(P + x)\right]\right\}$$

Figure 4-3. Flow diagram depicting the steps involved in computing the roots of a polynomial.

Functions are usually computed if at all possible. A typical example is the power series €or sin x:

$$\sin x \quad x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

This series would be expanded to tlie extent necessary to obtain the accuracy desired. When the function is not readily computable, it rnust be stored as a table in the computer memory. To minimize storage space, the table contains a minimum number of values, and intervening values are obtained by interpolation. A number of interpolation formulas are available, usually based on the use of a power polynomial. One of the most useful is the Langrangian interpolation formula:

$$\sum_{j=0}^{n} \frac{(x-a_0)(x-a_1)\cdots(x-a_{j-1})(x-a_{j+1})\cdots(x-a_n)}{(a_j-a_0)(a_j-a_1)\cdots(a_j-a_{j-1})(a_j-a_{j+1})\cdots(a_j-a_n)} f(a_j)$$

$$= \sum_{j=0}^{n} \ell_j(x)\, f(a_j)$$

where the a_i symbols represent the tabulated values of x_i. If equal intervals are employed, the Langrangian coefficient $\ell_i(x)$ may be normalized to permit tlie storage of tables of standard values.

For frequently employed functions, it maybe advantageous to minimize storage by deriving a best-fit polynomial, i.e., one which -- for a limited number of terms -- gives the least error between the approximation and the actual function [35].

Wlien empirical points are given, a curve may be fitted by the method of least squares.

While a number of methods for tlie solution of linear simultaneous equations exist, that due to Crout [36] is the most applicable to computer mechanization.

The solution of nonlinear equations is accomplished by first finding trial solutions that lie on either side of the desired root and then approximating the function in the interval between these solutions by some simple formula, such as a straight line, tu find the first approximation to the root. The process is then repeated (iterated) to achieve any desired degree of accuracy. Methods are available for increasing the rapidity of convergence of interative processes.

Numerical integration has a noteworthy simplicity: by dividing the area under the curve to be integrated into rectangles, the definition of integration can be employed to write

$$\int_{x_0}^{x_n} f(x)\,dx = \lim_{\substack{n\to\infty \\ \Delta x_i \to 0}} \sum_{i=1}^{n} f(\xi_i)\,\Delta x_i, \qquad x_{i-1} \le \xi_i \le x_i$$

The accuracy can be increased by use of a trapezoidal or parabolic approximation, such as Simpson's rule.

Differential equations car be solved by difference techniques, in which one extrapolates by a linear (or more complex) approximation from one point to another along the curve. However, iterative methods of successive substitutions, such as the Runge-Kutta method [37-39] are better suited io computer mechanization.

Since numerical analysis is both an art and a science, the designer of fire control systems will not often be called upon to practice it in person. A study of the references noted, to the extent necessary to intelligently supervise the work of tlie professional programmer, is probably all that will ever be required of him.

4.2.3 ACCURACY AND RESPONSE TIME

The accuracy of a fire control computing system is greatly influenced by the inevitable error at the input. (This is for the case of a dynamic installation where certain analog values are digitized for acceptance by the computer.) The degree of this error is usually known to the designer but is not under his control. This error is the first element in the chain of errors that occur throughout the computing system. If it is assumed that the function approximation is a best choice, the principal determinant of the accuracy of such a computing system is the round-off error: the most basic choice open to the designer in setting this accuracy is then the word size specified for the computer. For example, the effect of word size on accuracy can be illustrated by the following tabulation: it shows that the greater the word length, the greater the accuracy.

Word Length	
In Decimal System	In Binary System
256	100000000
512	1000000000
1024	10000000000
Corresponding Approximate Inherent Error due to Round-off	
0.4%	
0.270	
0.1%	

It is evident that an accuracy of 1 part in 256 (i.e., an accuracy of approximately 0.4%) can be achieved with a 9-bit word, whereas an accuracy of 1 part in 512 (approximately 0.2%) can be achieved with a 10-bit word. Furthermore, to prescribe 0.1% accuracy only in storing or reading out the coded value of some quantity, at least 11 binary bits are needed in the computer word. The addition of a sign bit and the frequently used parity or error-checking bit then establishes a minimum 13-bit word length.

However, if the demands of internal arithmetic will require using numbers larger than 1000, each factor of 2 increasing the size of the number will add another bit. Fortunately, manipulative devices such as the introduction of scale factors or the use of floating-point arithmetic will avoid the condition of overflow. In scale factoring, the operands are multiplied by appropriate scale factors at each juncture and the program keeps track of these factors.

In floating-point arithmetic, the scheme for the decimal system is to express all quantities as numbers between 0.1 and 1 multiplied by some integral power of 10. The equivalent in floating-point binary is to express each number as being between 1/2 and 1 (0.1 and 1.0 binary) multiplied by the appropriate integral power of 2.

Even though the word length may be sufficient to express all input quantities to the desired accuracy, the fact that a finite number of digits is used leads to round-off errors that may become significant if a large number of operations must be performed. For example, the product of two 11-bit numbers is a 22-bit number, but the least-significant 11 bits must be dropped for further computation. The resulting error is called round-off.

Once overflow has been avoided and assurance has been gained that the round-off will not be serious, consideration must be given to errors introduced in truncation. Truncation errors result from the fact that digital computations are carried out in a step-by-step manner with the result that a continuous function is defined only at a succession of discrete points. An increase in the sampling rate of a continuous input function permits a closer approximation of the true function and thus a reduction in this source of truncation error. Likewise, any reduction in the interval at which a variable is defined within the computation reduces the truncation error. However, reduction in the interval requires a larger number of steps to carry out the computation for a specified range of the independent variable and therefore increases the time required to carry out the computation.

The response time of the computer in a fire control system must generally be so rapid that the computer appears to be operating in real time. This may require that the most artful selection of routines be assigned to the program, and may also require that some compromise with accuracy requirements be made.

4-2.4 USE OF SAMPLED-DATA THEORY

If continuous functions of time are to be operated upon mathematically or logically by a digital-computer program and be transmitted from the source to a remote location with minimum interference, or be recorded in digital form, the original analog function must be described in terms of discrete samples.

Slated broadly, the sampled-data theorem[1] says that if the amplitude of a contin-

uous function of time is periodically sampled at a uniform rate that is at least twice the highest frequency of interest in the continuous function, then the sample series will contain essentially all of the information that was in the original analog function. This statement is made, however, on the assumption that the sampling time is infinitely small, and that the frequency spectrum of the analog signal has a finite limit. (If high frequencies are present in a function for which only the low frequencies are of interest, then the high frequencies must first be filtered out before sampling.) The theory says further that the analog function may be recovered by passing the sample series through a suitable filter.

There are certain practical difficulties that complicate the design of workable equipment that takes advantage of the sampled-data theorem. However, a good approximation to theoretical system performance can be obtained through the application of proper design considerations.

From a practical engineering standpoint, the following modifications to the theoretical data-sampling technique are required in the design of workable equipment:

1. The sampling rate must be at least four times the highest frequency of interest contained in the analog function to be sampled.

2. The analog input to the sampling circuit must be attenuated at frequencies above the highest frequency of interest but, since perfect filters do not exist, the attenuation is determined by practical accuracy-tolerance requirements of the system.

3. Since sampling cannot he performed instantaneously, a requirement arises for some "aperture correction" techniques in systems where the ratio of shortest signal period to tlie sampling aperture time is not high enough to make the sampling-time error negligible.

4. Signal-conditioning equipment is often required between the signal source and tlie sampling circuits. In addition to the filtering usually required, there is frequently a need for amplification of signals obtained from transducers and other signal sources to increase the signal to a level suitable for sampling. The low-level end of such preamplifiers usually requires special design to eliminate the effects of stray noise pickup and induced common-mode voltage disturbances. At the output, consideration must be given to the d-c level of the output composite signal, as well as the amplitude of the signal itself, in order to make the analog output of the preamplifier compatible with the sampling circuitry.

A discussion of the effects of the sampling process on the design of the system can be found in Chapter 11.

4-3 THE GENERAL CONFIGURATION OF A FIRE CONTROL DIGITAL COMPUTER

As a good start toward determining the general configuration required for a fire control digital computer, the computing-system analyst should ask himself the following series of questions:*

1. What is the source of the input data that is to be processed by the fire control computer?

2. What kind of input data will be presented to the computer? (Numerical? alphabetical? other?)

3. How can this input data best be translated into the internal language of the computer?

4. What is the rate of input-data flow to the computer from the source?

5. What must be done in the way of processing the input data? (Must it be altered? Must it be sorted or combined in some way with other data? If so, how?)

6. How much time is available for the computer to process the input data?

7. What accuracy is required in the input data and in the processing? Does this accuracy differ markedly in different parts of the computation?

8. What must be the output rate of the processed data?

9. What is the purpose of the output data and in what manner is it to be employed?

10. Into what form should the output data be translated in order to accomplish its purpose most effectively?

* Adapted in part from Ref. 2, which discusses the questions concerning operations to be mechanized in terms of information flow that any system analyst must inevitably ask himself as he approaches an electronic data-processing problem.

11. What effect would a computer error have on the flow of data, and how would it affect the particular operation being performed.

12. Can the computer operation be interrupted for emergencies or for regular periods of preventive maintenance?

13. How can manually entered data best be entered from a human-engineering standpoint? How can the output data be presented so as to be readily readable and understandable?

14. What provisions for internally stored programs should be made to facilitate future programming?

In order to provide an appropriate frame of reference in considering the aforenoted questions, the functional diagram of a hypothetical fire control system given in Fig. 4-4 is re-introduced from Chapter 3 of Ref. 104.

This diagram shows three classes of input to the computer:

1. Command decisions
2. Target data
3. Variations from initial conditions and spotting corrections

Stored in the computer are standard trajectory data. Generated within the computer, prior to final correction, are firing data. The two basic ultimate outputs of the computer are time-of-flight information and corrected firing data.

Residing somewhat innocently at the lower center of Fig. 4-4 is the group of data transmitting elements "introduced between functional elements as required". Such elements at the input and output of the computer must be scheduled and controlled for the effective flow of information into and out of the computing system,

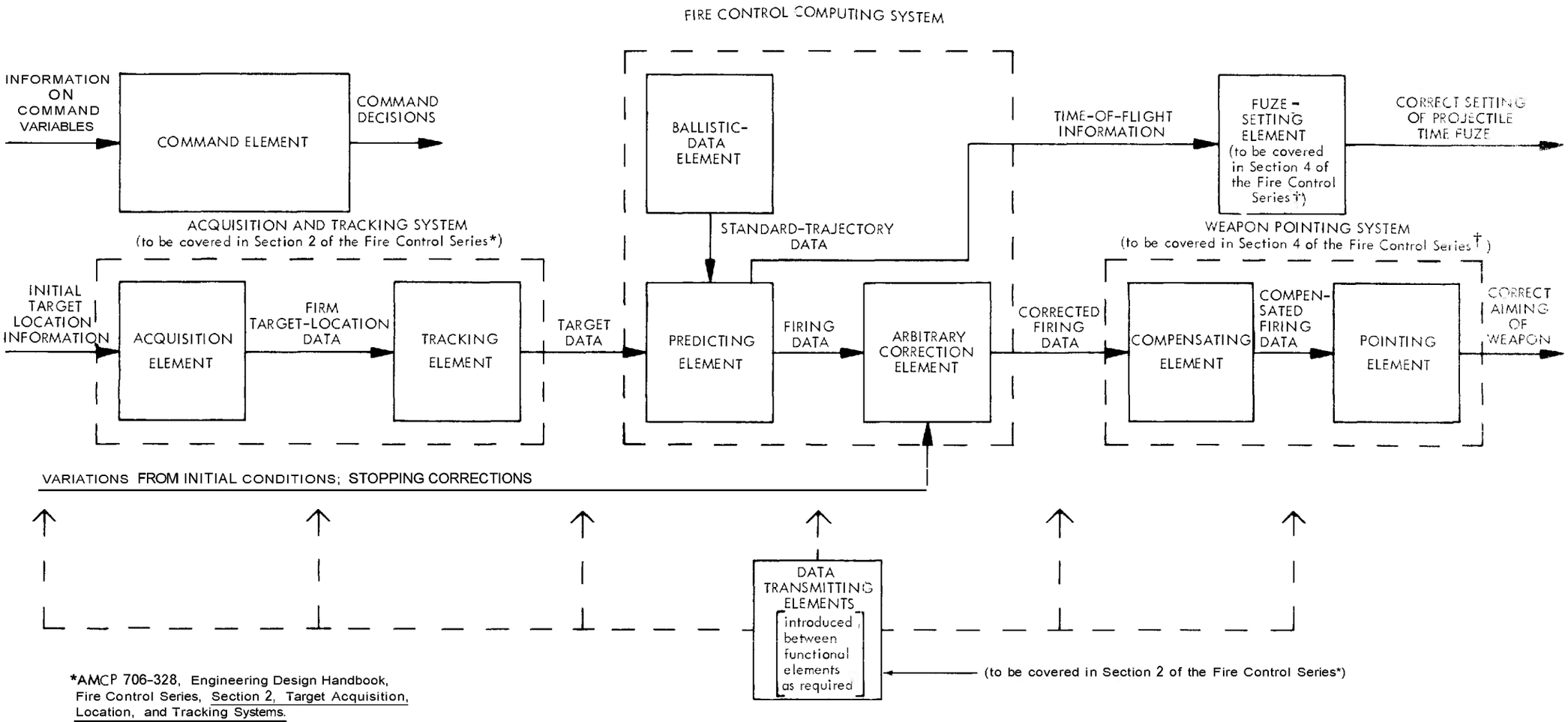
4-3.1 INPUT AND OUTPUT CONSIDERATIONS

By considering for a moment only the overall organization of a digital computer, as depicted in pictorial form by Fig. 4-5, one can approach the general computer problem of accepting real-time data at fixed sampling intervals--remembering that a digital machine is not inherently a real-time device and that the microscopic programmed tasks involved are slaved to a clock. As indicated in Fig. 4-5 (which corresponds to the conventional functional diagram of Fig. 4-1 (B), the computer input data (1) are fed to the input section (2) where they are converted into the internal language of the computer. The translated input data are stored in the input buffer unit (3) until called for, at which time they are transferred to the storage unit (4). Here they are available for processing at the request of the arithmetic unit (5). Processed data go to the storage unit, from which they are transferred through the output buffer (6) into the output section of the computer. The output section then translates these data (reads out the data) into a suitable form (8) for subsequent use. The intermediate buffer (9) between the storage unit (whose memory function is represented by a human brain) and the arithmetic unit (whose data-processing function is represented by an abacus) serves to present information for processing, and to retrieve processed information, at the various rates imposed by the arithmetic processes. The control unit (10) coordinates the activity of the computer in three ways:

1. With regard to the computer's internal operation.
2. With regard to the reception of input data.
3. With regard to the readout of output data.

In carrying out these coordination functions, the control unit schedules operations (as indicated by the clock in Fig. 4-5) and communicates with the other units (as indicated by the speaker horn). As the basis for scheduling, it utilizes a computer program that is either placed in the storage unit for internally programmed computers or is available externally, as indicated by the clipboard (11).

A common problem exists for the mechanization of the input and output sections of the computer. This problem -- referred to as the input/output problem -- stems from the fact that whatever means are employed for passing information into, and out of, the internal portions of the computer require the control and synchronization of these operations with the internal-computer retrieval and transfer operations. The nature of the input/output problem is essentially the same for both the input and output portions of the computer. The complexity that this problem



*AMCP 706-328, Engineering Design Handbook, Fire Control Series, Section 2, Target Acquisition, Location, and Tracking Systems.

†AMCP 706-330, Engineering Design Handbook, Fire Control Series, Section 4, Weapon Pointing Systems.

Figure 4-4. Functional diagram of a hypothetical fire control system that contains all of the functional elenients associated with fire control equipment.

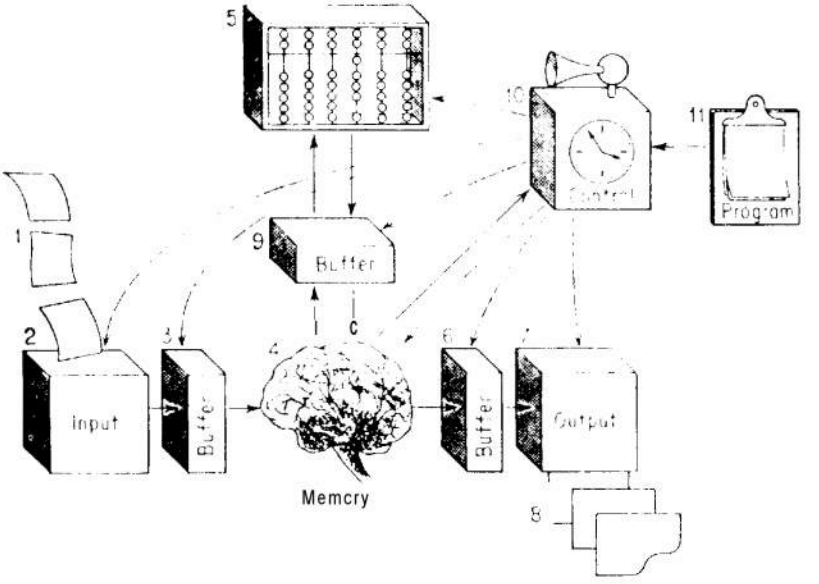


Figure 4-5. Organization of the computer in pictorial form.

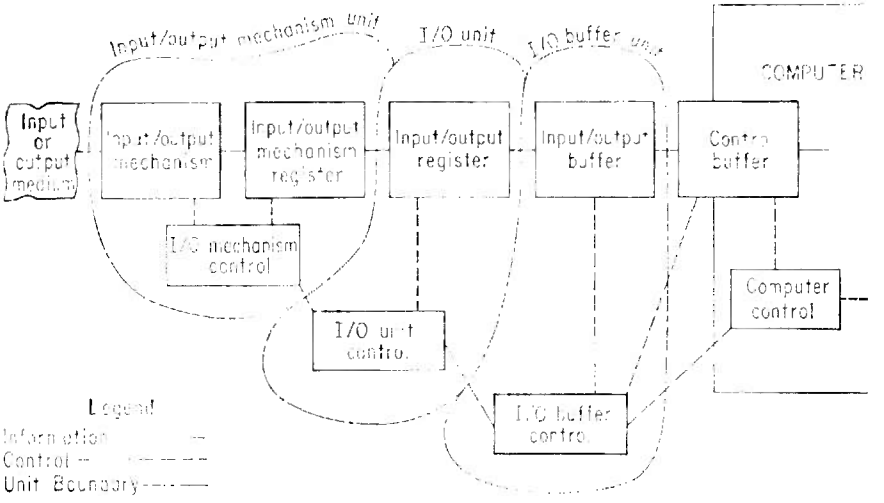


Figure 4-6. Relation of the input/output to the computer.

can attain under extreme conditions is indicated by Fig. 4-6 which shows the functional elements that might be involved for the input and output sections of the computer,

In the case of fire control computation that can be performed in advance of the actual firing of the weapon and then terminated, the input/output considerations are essentially as given in the preceding paragraphs. This applies, for example, to FADAC (Field Artillery Digital Automatic Computer), which is discussed in Part III. Because of the required portability of this equipment and the standardized nature of the computations, the input/output equipment tends to be relatively simple.

Some fire control digital computers, on the other hand, must continuously compute new weapon-positioning information during the course of an engagement. Such computers are commonly called real-time computers and are used, for example, in connection with a moving target. For such computers, the input/output mechanism represented functionally in Fig. 4-6 would typically be a shaft encoder, The cumbersome registering and buffering activity that is depicted dramatizes the unfortunate situation that arises as the difference in operating speeds between the input/output equipment and the internal portion of the computer -- the computer proper -- becomes more disparate. (See Section 15.3 of Kef. *3* for an excellent discussion of the considerations involved when the greatest disparity possible exists and the maximum buffering is required.)

A further complication in a real-time fire control computer arises from the necessity of reading-in data from multiple sources (e.g., elevation, azimuth and range data from a radar tracker). As indicated in Fig. 4-7, multiple inputs are usually fed to independent input/output registers for each source of data. The computer then interrogates each of these registers in turn, *so* that a single

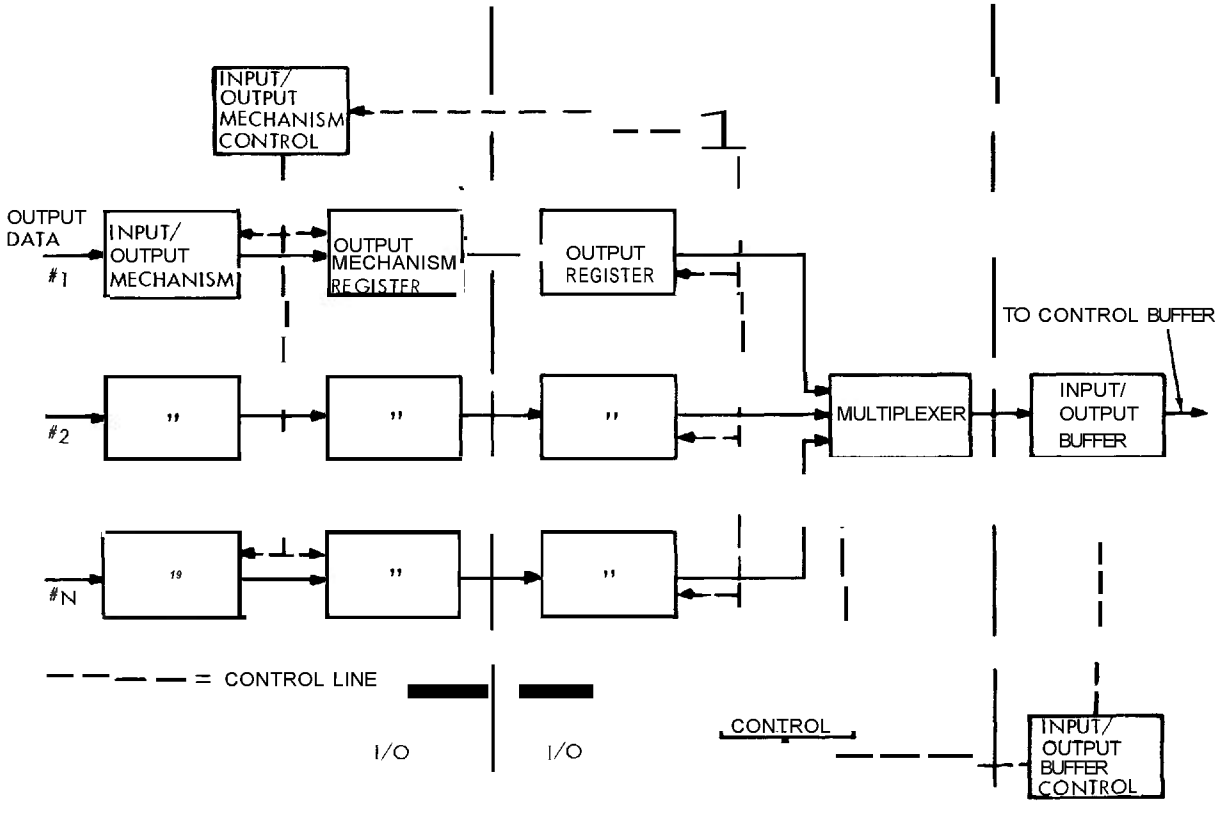


Figure 4-7. Computer input/output configuration for multiple inputs.

data word is entered into the input/output buffer at any one instant, This process is performed by the multiplexer under the control of the input/output control. If one or more data inputs changes more rapidly than the others, provision can be made to interrogate it (them) more frequently with the multiplexer,

It is now evident that the characteristics of the problem and the input/output equipment determine the speed and accuracy requirements for the computer. The range of frequencies encountered in the problem, as noted in par.4-2.4, determines the minimum sampling rate, This rate in turn defines the specifications of the input/output equipment and also the maximum allowable solution time of the computer. At the same time, the accuracy required in the system determines the word length and in some cases may put a constraint on the sampling rate as well,

Having specified sampling rate and word length, the computer designer must adjust a number of parameters in order to achieve his goal. The most important of these are the clock rate, the capacity and access time of the various storage elements available to him, the choice of serial or parallel logic, and the choice of programming schemes. With this variety of choice, there is no uniquely best design; rather, there is a wide opportunity to exercise his judgment and ingenuity to achieve a good design.

The moment the logic designer starts to work with relays, switches, push buttons, and similar devices in order to communicate with a digital computer, he has left the neatly defined area of decision and memory elements whose outputs are defined at every clock pulse and synchronize perfectly with computer functioning. The essentially slow, mechanical pieces of equipment have output signals that may change at any time with respect to the principal computer timing signals, and they tend to "bounce" and provide a more or less random series of "zeros" and "ones" before stabilizing. Fig. 4-8 illustrates what may happen with relay "bounce", which introduces a period of uncertainty that the designer must eliminate from the logic by introducing a delay d_m to prevent the "noisy" contact from affecting the desired signal Q.

4-3.2 COMPUTER SPEEDS

Because any complex calculation requires a very large number of transfers of information into and out of the rnemory units, the access time of storage in the computer is the largest determinant of the speed of processing and of performing arithmetic operations. In turn, the time to perform arithmetic is influenced most by the time required to do addition. Addition time depends on the system of coding used and on the logic used for addition. The choice of components will, of course, influence the logic for addition.

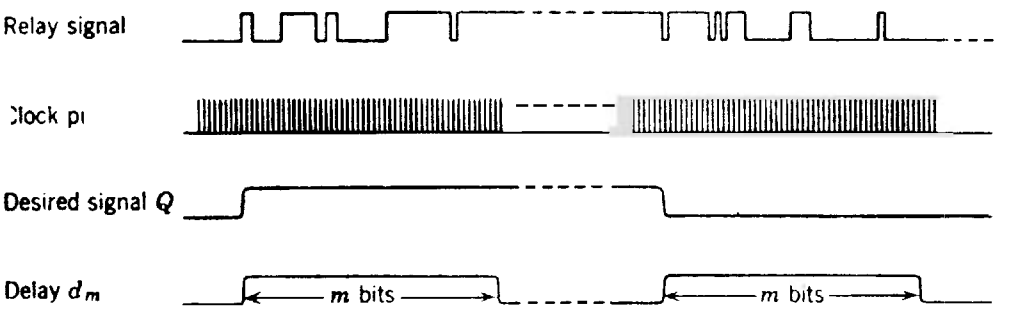


Figure 4-8. Derivation of a smoothed signal from an asychronous signal device,

It is alsonecessary to take into account the time required to enter computer words into, and to withdraw them from, the registers active during addition. In general, addition time is independent of the numbers being added. Time estimates for computations involving multiplication and division can be approached by formulas relating all the factors mentioned.

The rate at which individual bits will be handled by the machine is the pulse repetition rate, or clock rate. The characters to be handled by the computer, coded in electronic form, will be processed at the clock rate set by the designer. The upper limit of the clock rate is determined by the component circuitry used in the computer, Generallyspeaking, the cost of the basic circuitry increaseswith an increase in the clock rate. Clock pulses, for example, must be supplied to the circuits that read from and write into memory, so that information stored is operated on in synchronism withother data in the memory and in other parts of the computer,

Substantial increases in speed can be obtained by designing the computer to perform all operations in parallel, For example, if a 40-bit machine with a serial representation operated at a 1-mc rate, it would take a minimum of 40 microseconds to so much as transfer a number fromone place to another within the machine. If all operations were parallel, thenumber could be transferred in 1 microsecond on 40 separate wires. In a serial-parallel machine, the 40-bit number could be divided into four groups of 10 bits eachand inonly 10 microseconds the number could be sent over four parallel wires. IIere, the reduction of 75% in transmission time might represent a good engineering choice in the light of slower limiting times in other elements of the logic. There is no point in having any circuitry in a computer design that far out-races the rest of the system and then is idle most of the time.

Transistorized digital modules that have operating speeds of up to 5, 10, and even 20-mc are currently available from manufacturers; in addition, 50-mc logic has been reported in the laboratory, Operating speeds of 5 mc or less are more common, however, since they fulfill most requirements and are less costly. Integrated-circuit speeds of up to 10 mc are also available for certain types of logic.

4-4 DETERMINATION OF COMPUTER STORAGE CONFIGURATION

4-4.1 SIZE OF COMPUTER PROGRAM

Determination of the effect of program size on the storage configuration can start easily with the creation of a flow diagram. The flow diagram, similar to the block diagram used for preliminary design and understanding of many types of equipment, is a means for visualizing the computer program by breakingit down into functional units that correspond to different sections of the problem. Ultimately the programmer will carry this fractionating process of the program itself down through the routines, the subroutines, the loops, and finally to the commands, the smallest elements of the program. A typical subroutine is the taking of a square root. A loop (also called a cycle or iteration) consists of repetition of a group of instructions in a routine.

By starting with an example of a loop, the general form of a flow chart or diagram can be readily illustrated, Ref. 76, which should be consulted for additional information, describes a realistic example: that of determining the position of a ballistic missile after each 10 seconds of flight along its trajectory, neglecting the effects of air resistance, etc. (see Chapter 2 of Ref. 104). In this example, at time t, the x and y components of position will be

$$x_i = V_{0x} t_i$$

$$y_i = V_{0y} t_i - (1/2) g t_i^2$$

$$i = 1, 2, 3,$$

where V_{0x} is the initial x component of the velocity and V_{0y} is the initial y component. To be concrete, suppose that $V_{0x} = 2{,}000$ fps, $V_{0y} = 1{,}000$ fps, and $g = 32$ ft/sec^2. Then at timet, (= 10 sec), x ,= 20,000 ft and

$$y, = 10{,}000 - 1{,}600 = 8{,}400 \text{ ft}$$

At time t_2 (= 20 sec), $x_2 = 2{,}000 \times 20 = 40{,}000$ ft and

$$y_2 = 1{,}000 \times 20 - 16 \times 20^2 = 13{,}600 \text{ ft}$$

At time t_3 (= 30 sec), $x_3 = 2{,}000 \times 30 = 60{,}000$ ft and

$$y_3 = 1{,}000 \times 30 - 16 \times 30^2 = 15{,}600 \text{ ft}$$

and so forth. During such a computation, it is clear that the same formulas are used over again, each time increasing t_i by 10 sec. However, the computation should stop when the missile hits the ground, i.e., when y_i is zero. In the present case, Table 4-1 shows that this condition exists at a point in the interval between t = 60 second and t = 70 sec. Computation in this loop is therefore stopped when t = 70 sec, as shown by Fig. 4-9 -- the flow chart of the process. This flow chart employs the i notation, where t_i t_1 represents the next time around and i + 1 → i means that for the next iteration the old ith values are replaced with the new (i + 1)th values.

TABLE 4-1. COMPUTATION OF THE TRAJECTORY OF A MISSILE.

i	t_i		
1	10	20,000	8,400
2	20	40,000	13,600
3	30	60,000	15,600
4	40	*80,000*	14,400
5	50	100,000	10,000
6	60	120,000	2,400
7	70	140,000	-8,400

With this illustration it is seen that there are four basic ingredients to a recursion code:

1. A set of instructions, called the iteration instructions, that are to be reused.
2. Another set of instructions that modifies the original set each time around.

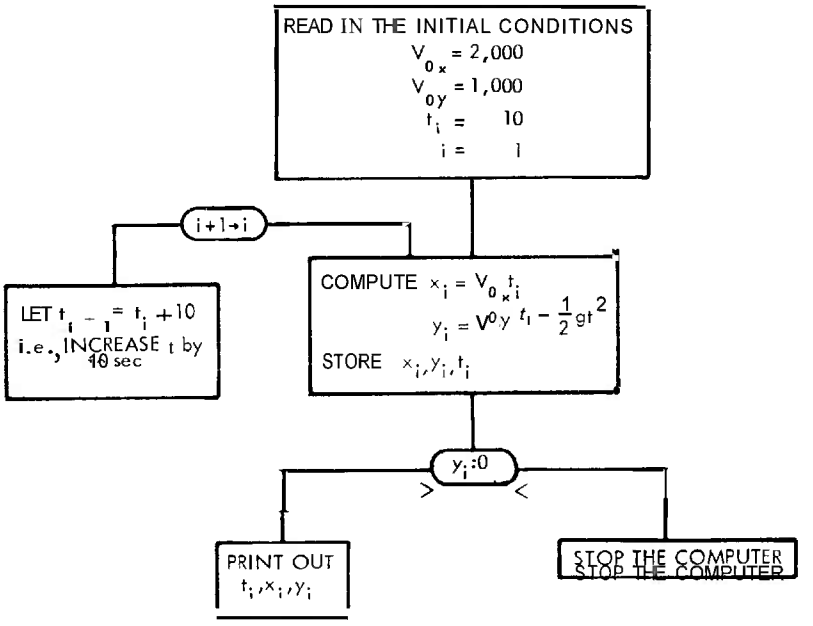


Figure 4-9. Flow chart for computation of missile trajectory.

3. A set of instructions, often called a tally, that determines when to exit, or break out of the loop, and appropriately notifies the computer.

4. A set of instructions that sets up the initial conditions and starts the loop.

In addition, a loop or recursion code often contains a set of instructions that resets the loop so that it may be used again by the computer at some future time, A generalized loop can be indicated by the flow diagram of Fig. 4-10.

Sometimes the tally consists of instructions for determining whether or not the result of each iteration is smaller than some given number, as occurs often in function computations, At other times, the tally may just count the number of iterations until the desired number have been accomplished.

Fig. 4-11 shows the flow diagram for instruction modification in a loop. Here a long column of numbers was previously placed in consecutive memory addresses, the last of which is address 077. The same add instruction is used for successive additions, but it is modified before each addition so as to add the contents of the next successive address to the partial sum each time around.

Fig. 4-12 illustrates the use of loops within loops, as in the computation of sin x by means of the infinite series

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \cdots$$

at intervals of 1/ 0 radian from 0 to $\pi/2$, to eight decimal places. In the figure, loop A forms $x^n/n!$ by multiplying a partial product successively by x/P_i. Loop B adds or subtracts this result to or from the partial sum and increases n by 2 until the partial sum becomes correct to eight significant figures. Loop C increases x by 0.01 and continues to compute the next value of sin x.

A subroutine is a subcode that is written only once but may be used at different times and places during the computation of a program. Fig. 4-13 illustrates the simple case of two points from which the program can jump to the subroutine, through the A connectors, and return through the appropriate exit route via the B connectors.

Construction of the complete flow diagram will identify the number of program steps (including any advisable accuracy checks) and will establish the type of orders required for the computations. Throughout this process of refining the program there may be constant compromise between speed and accuracy, between serial and parallel operation, and between short-term (register) or intermediate (buffer) or long-term (memory) storage requirements, each affecting the ultimate storage configuration,

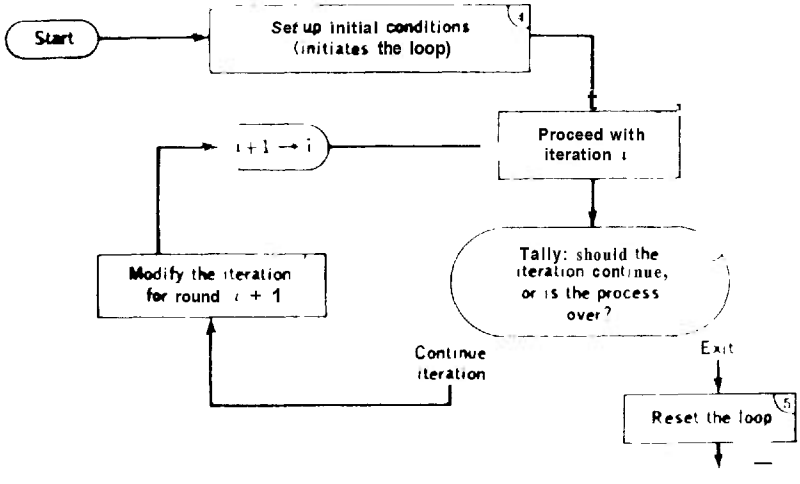


Figure 4-10. Flow chart of generalized loop.

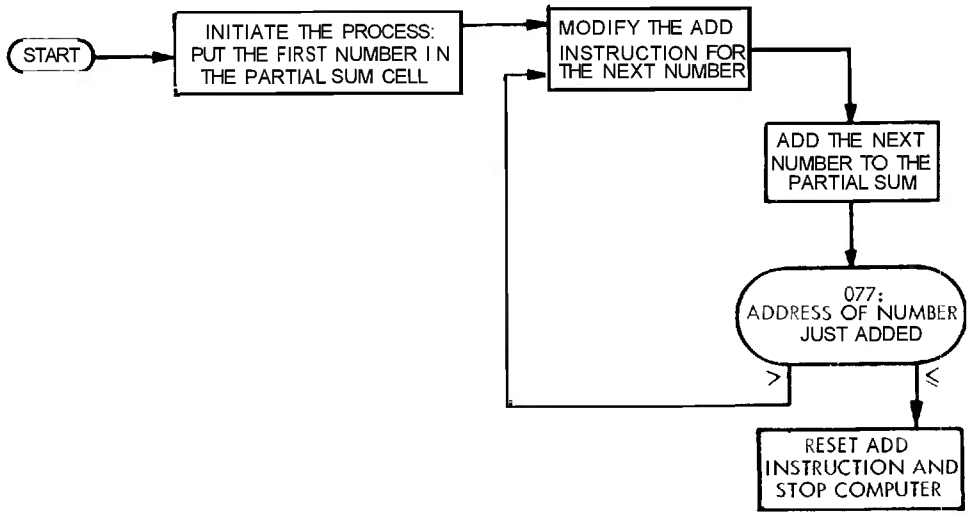


Figure 4-11. Flow chart for instruction modification.

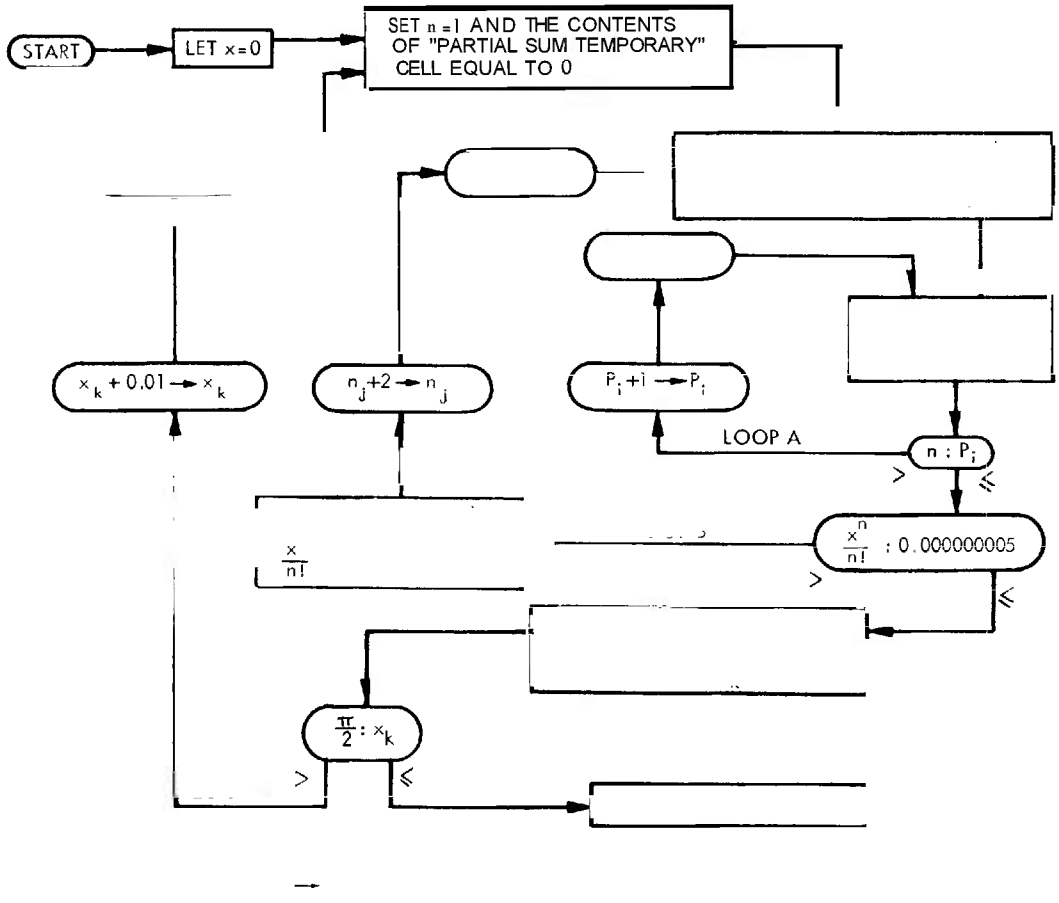


Figure 4-12. Flow chart of loops within loops.

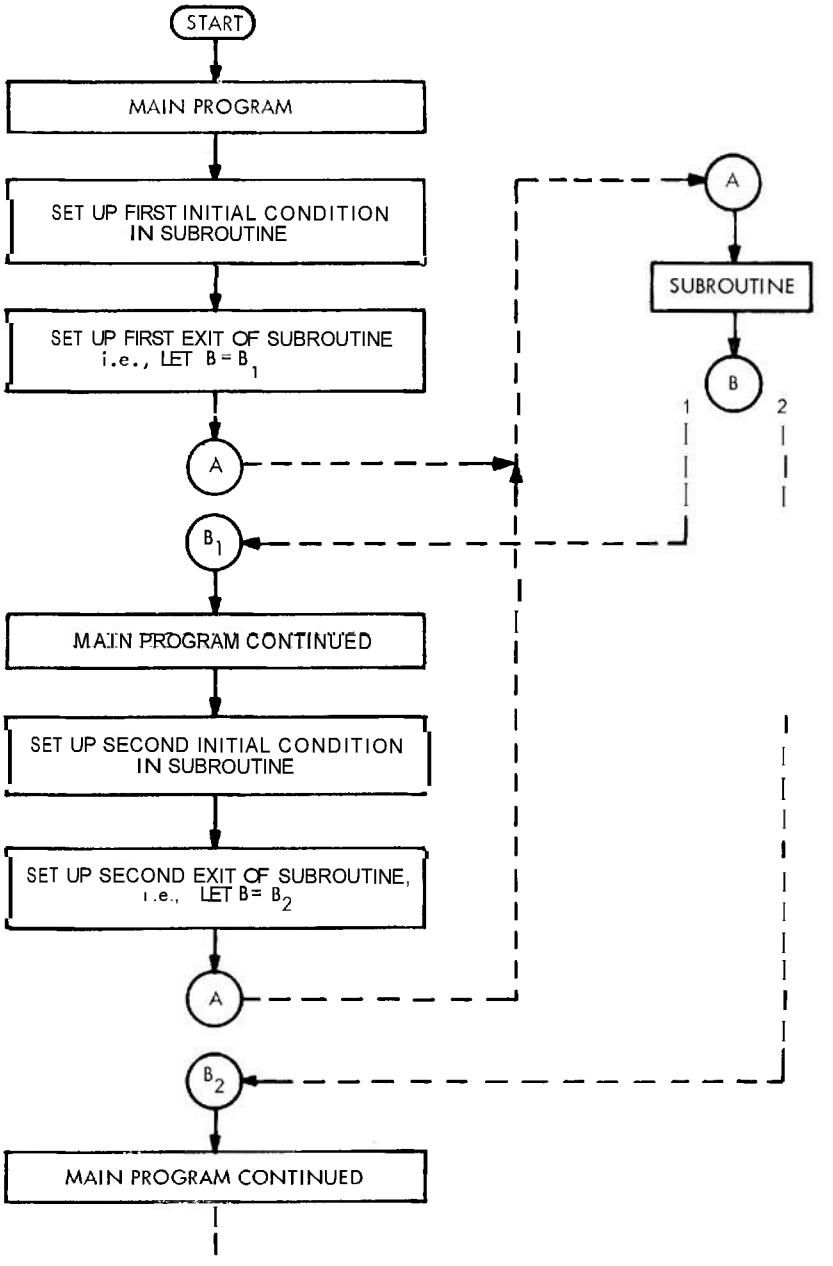


Figure 4-13. Flow chart for setting up initial conditions and different exits of a subroutine.

4-4.2 CODING SYSTEM AND WORD LENGTH

The actual assignment of an order to the computer for each step in the program is called coding. The finished code is a coniplete list of instructions, or orders, and their equivalent numbers, since ultimately the computer mechanism deals only with numbers, Instructions and actual data quantities are indistinguishable from each other except by interpretation, The computer memorizes instructions and quantities as the contents of addresses in its memory. Instructions explicitly involve only addresses and tell the computer what to do with the contents of these addresses. The structure of the program set up for the computer, the choice of a coding system to comniunicate with the computer, and the size of the word -- the string of binary digits -- that represent the storage capacity of each memory address are extensively interrelated. Some of the basic factors are briefly reviewed for their effect on computer storage configuration.

In thedesign of a computer, the number of bits reserved for an address places an upper limit on the number of words in the addressable memory of the computer, If an address is denoted by n bits, no more than 2^n words can be contained in the addressable memory, In the choice of a coding system, a typical format for four types is as follows:

43 bit word: 4 addresses of 9 bits, 1 instruction of 6 bits, and 1 sign bit, Memory 5I2 words, max.

43 bit word: 3 addresses of 12 bits, 1 instruction of 6 bits, and 1 sign bit. Memory 4096 words, max.

43 bit word: 2 addresses of 18 bits, 1 instructionof 6 bits, and 1 sign bit. Memory 262,144 words, max.

43 bit word: 1 address of 36 bits, 1 instruction of 6 bits, and 1 sign bit. Memory 68,7I9, 476,736 words, max.

Naturally, with two-address or one-address systems, shorter word lengths are common, with the lower limit largely determined by the numerical accuracy required in problem solution.

In order to describe the additional memory elements required beyond those used in the computer memory for central storage (see Fig. 4-14), the functions of the computing unit will be reviewed. This unit has two functions (see Fig. 3-15):

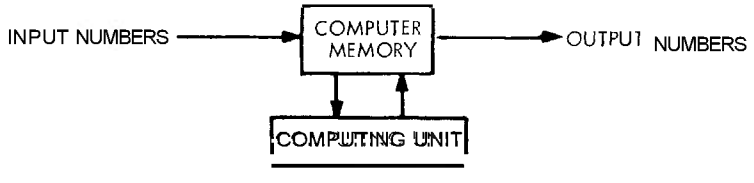


Figure 4-14. Memory and computing unit.

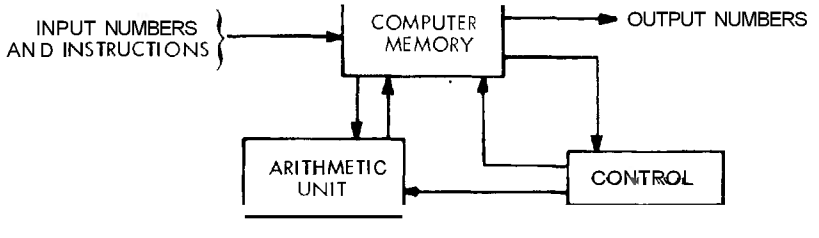


Figure 4-15. Arithmetic unit and control.

1. To obtain instructions from the memory and interpret them (done by the control unit).

2. To perform the actual operations (done by the arithmetic unit).

The control unit must perform two functions (ref. Fig. 4-16):

1. Interpret the instructions (done by the instruction decoder).

2. Tell the arithmetic unit what to do (done by the control generator).

After an instruction has been executed, the control generator produces signals that enable the next instruction to go from the computer memory to the instruction decoder,

As shown in Fig. 4-17, the control generator also commands the input-output selector, Through appropriate buffering memory, this unit feeds input and output information to and from the main memory.

Fig. 4-18 points out several other basic memory elements. When an arithmetic operation is performed, the result is formed in the accumulator (high-speed register memory) of the arithmetic unit. In order that the instruction decoder may be able to refer to the current instruction during the time that control signals are being set up, the instruction (word) being executed is stored in a special memory cell, the instruction register.

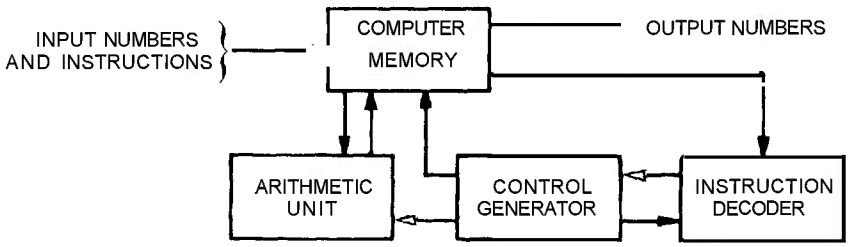


Figure 4-16. Instruction decoder and control generator. (Solid-headed arrows indicate information; hollow-headed arrows indicate control signals.)

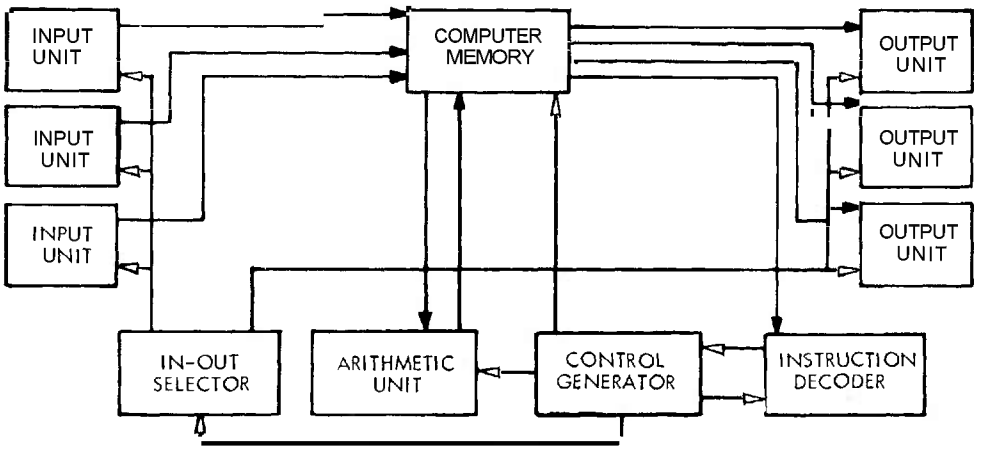


Figure 4-17. Input and output functional units.

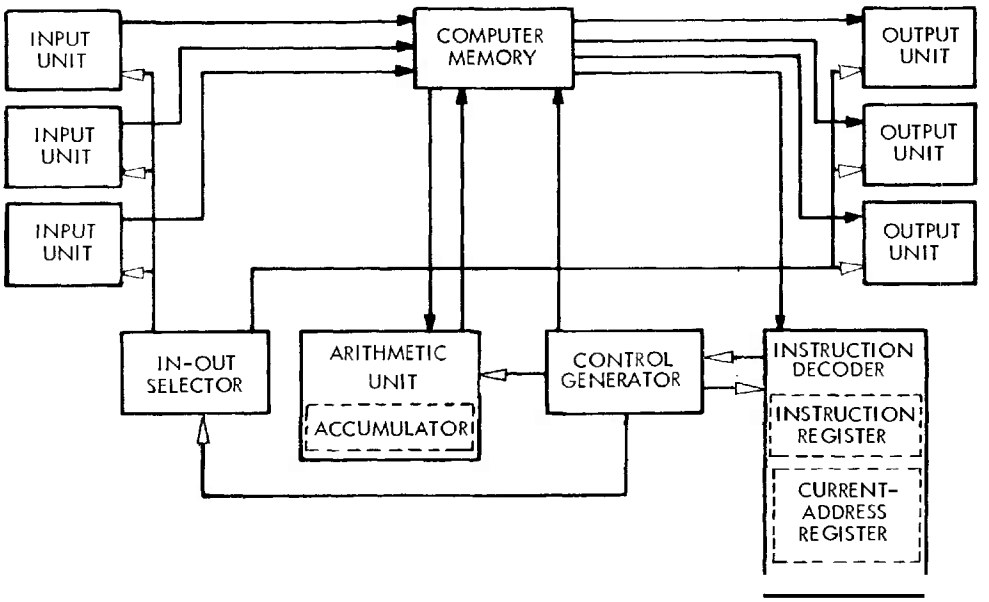


Figure 4-18. Accumulator, instruction register, and current-address register.

The current-address register usually contains the memory address from which the instructions being executed came. This covers the situation where the address of the present instruction was given as part of the previous instruction, and the situation wherein the next instruction is the next higher (or otherwise related) address.

In designing a digital computer, the engineer's first task can be considered to be that of choosing the proper coding and programming system for the purpose at hand. The logical design of the computer circuitry follows. The electronic design, which would include the design of the memory configuration, constitutes the third step. Various parts of a large program can be stored in a relatively slow memory system -- such as a magnetic tape or drum -- and then transmitted to a high-speed (and high-cost) memory when the actual computations in this part of the program are to occur. Another factor tending to hold down the size of the main memory system evolves from the skill of the programmer; for example, his ability to use temporary storage for intermediate results that, once computed, are not used again in the problem -- thus requiring the main memory to store only permanent or constant numbers.

The relative speeds of common types of digital memories are listed in Table 4-2.

Unfortunately, throughout the digital computer literature, the reader will be called upon to distinguish between instruction code and machine language code. In the digital machine language codes, the yes-no bits can be associated in many ways to represent characters in machine language. Consider, for example, the following five machine language codes:

TABLE 4-2. ORDER OF MAGNITUDE OF MEMORY ACCESS TIME.[102]

Memory System	Access Time
Magnetic Tape	5 msec, plus time to position tape
Magnetic Drum	10 msec to 1 sec
Acoustic Delay Line	50 psec to 50 msec
Magnetic Core	500ns* to 50 μsec
Diode Capacitor	1 μsec
Flip-flop Register	10ns* to 10 μsec

* ns = nanosecond = 10^{-9} sec

Machine Language Code	Binary Notation	Decimal Number
Natural Binary Code	101011011	(347)
Natural Binary Decimal Code	0011 0100 0111	(347)
Excess Three Binary Decimal Code (X53)	0110 0111 1010	(347)
Odd Parity NBDC	10011 00100 00111	(347)
Odd Parity X53	10110 00111 11010	(347)

The naturalbinary decimalcode is easier to translate from human language, but sacrifices the efficiency of natural binary arithmetic. The X53 code has the advantages of simple translation plus easier arithmetic, the fact that a digit and its 9's complement are complementary, and the fact that no decimal digit including zero is coded as 0000. The two parity checking codes provide a means ofprotecting againstthe loss of pickup of a singlebit, The extraparity bit is used to adjust the totalnumber of 1's in each binarydecimal bit to be - in these cases - odd. (Even, however, can be used.)

It should be noted that there are billions of possible character codes. These'are representative samples only,

The choice of how many characters will make up a machine word, or whether the computer will handle only fixed word lengths or variable wordlengths will be another multiplying factor in determiningthe overall storage configuration.

4-4.3 SUBROUTINES, REQUIREMENTS FOR TEMPORARY STORAGE

Once the number of words of input and output data have been determined as required by the program, and a necessary coding system for the machine language has been set, an examination of the subroutines will do two things:

1. Reduce the absolute size required of storage capacity, by eliminating the memorizing of tables whenever a numerical approximation is more economical.
2. Achieve reduced but sufficient memory space to provide storage locations for intermediate results and constants used in calculations.

4-4.4 DATA STORAGE REQUIREMENTS

The necessity to store standard trajectory data within the computer for reference as required can be considered to be a nominal requirement for the solution of a fire control problem. An examination of the magnitude of this type of requirement will enable the total computer storage configuration to be determined.

4-4.5 EXAMPLE OF FADAC MEMORY

The FADAC* general-purpose transistorized digital computer operates serial by bit, parallel by function, and allows 12,800 one-word execute (add, subtract, etc.) operations per second, The word length is 33 binary digits, including parity bit, sign bit, and 31 binary digits for absolute numerical value.

The memory is a rotating magnetic disc, 6000 rpm nominal, Storage totals 4096 words, 32 channels of 128 words each. 28 channels are permanent storage (read only) and 4 channels are for working storage, There are two 16-word high-speed loops for rapid access, five 1-word registers for arithmetic operations and control, and one 2-word register for output display-information storage, The functional diagram of the FADAC System appears in Fig. 4-19, in which the memory elements are identified.

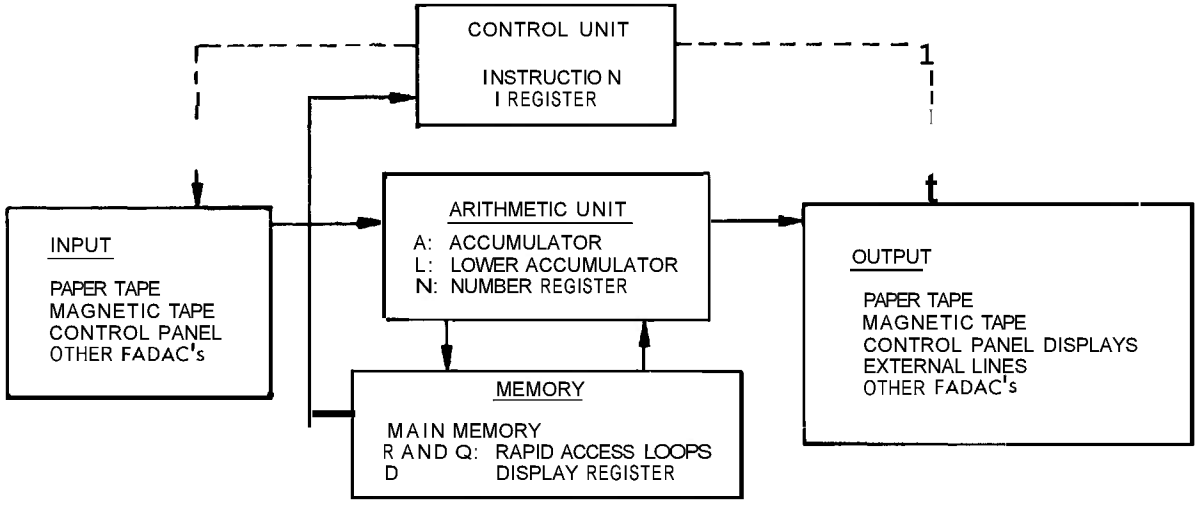


Figure 4-19. Functional diagram of FADAC system.

* Abbreviation for Field Artillery Digital Automatic Computer; see par. 1-3.4 of Ref. 104 for background information relating to FADAC.

4-5 FLEXIBILITY REQUIREMENTS

4-5.1 SPECIAL-PURPOSE VERSUS GENERAL-PURPOSE COMPUTERS

The general-purpose computer usually has a large number of input and output channels, and these channels may each consist of a number of bits in parallel rather than a single bit. Such a computer generally has a storage capacity of many thousands of computer words, and the various input, output, and memory devices are all accessible to the programmer through instructions that enable him to select whatever device he needs next. In addition to the basic instructions – add, subtract, multiply and divide – many others will be available, such as, for example:

Jump - Specifies the location of the next instruction and directs the computer thereto.

Shift right (or left) - In the case of binary numbers, this is a shift to the right (or left) of the number of bit positions specified in the instructions and effectively multiplies the number by 2^{-n}, (or 2^{n}), where n is the number of bit positions shifted, Those bits that run off the right-hand end of the word are discarded; those added to the left are zeros.

Transfer - To transmit, transport, exchange, read, record, store or write data – whatever the computer operation that is next required.

These instructions form the computer "repertoire" and provide the programming flexibility required for different types of problems. As a rule, the larger the repertoire, the easier the programming task. For example, if a computing machine has a "multiply" instruction, it can be stated directly. Otherwise, successive additions must be programmed – generally by iteration,

Further, general-purpose machines may have instructions with two, three, or four addresses, in turn requiring logic for as many as four accesses to memory for a single order, A typical order for a four-address machine might be "divide the operand from address A by the operand from address B, store the quotient in address C, leave the remainder in the accumulator, and obey next the instruction in storage location E",

Special-purpose computers may have the larger part of their programs built-in, literally soldered in place, but the sacrifice in flexibility will almost always yield a substantial increase in speed., Designed to solve only one problem, or to perform relatively few types of calculations, the special-purpose computer needs very little capability for "talking" with the operator, At the same time, it may require laborious reconstruction when its mission changes. The general-purpose machine, faced with a shift in duty, would require only modification of the information in its memory, This would be achieved by preparing a new set of instructions and reading these into the memory,

4-5.2 CHOICE OF BUILT-IN COMPUTER OPERATIONS

As mentioned previously, the basic operations are only addition, subtraction, and detection of the sign of a quantity, together with operations of transfer to and from storage, input, and output. While it is conceivable that problems could be programmed using only these basic operations, such a computer would be most inflexible and difficult to program. At the very least, such operations as multiplication, division, and decision as to which of two quantities is the larger would be programmed as built-in operations. Further elaboration is not generally wired in place, but programmed as a permanent subroutine, Operations of this type commonly include integration, interpolation, and function-generating equations.

A special-purpose computer, by definition, has all operations built in. Provision for minor adaptations, however, is usually necessary, and there is a sequence of major and subroutines as in the general-purpose computer.

In an attempt to ease the burden of programming, general-purpose computers have been provided with elaborate programming schemes that permit English-language instructions and in some cases, the entering of equations in almost the original mathematical form.

4-5.3 CHOICE OF PROGRAMMING SYSTEMS

In the case of external programming, each operation required of the computer is under the control of some external device. In the simple case of a punched card program, the computer "senses" a new card for each operation, Changing the program requires only a new arrangement of punched cards. However, the degree of sophistication available with an external program alone is rather narrow. For example, take the unattractive case of punchingenough cards to allow an iterative process whose duration cannot be predicted in advance. Secondly, consider the dilemma that exists when the program requires one branching operation if a number happens to be positive and another branching operation if a number happens to be negative.

A so-calledplugged-program can be instituted through the use of a plugboard control panel. With a plugboard, the actual physical wiring of the computer is changed for each new set of computations. To minimize error and to speed the changeover, whole plugboards can be interchanged. In general, practical physical limitations place a maximum of about 100 steps that can be contained before the mass of wires becomes unwieldy, although there is no theoretical limit as to how far the designer might go.

A fully stored-program computer stores instructions and data interchangeably. There is a list of instructions a given computer can perform and these can be combined and sequenced by the programmer for a wide range of operations. The computer may perform arithmetic and other operations on instructions in the program as well as on items of data.

In programming, it is often desirable to be able to prepare a program in a "problem-oriented" or "user's" language instead of directly in machine language. This can be accomplished through the aid of a "compiler" program, or translator program, designed for a particular class of problems. For engineering and scientific work, FORTRAN is one of the most widely used compiler languages. FORTRAN stands for FORmula TRANslation. [97-99]

The manner in which a compiler program can be used to solve trajectory equations is illustrated in Fig. 4-20. In this case, the mathematical problem is set up in a special form and then fed to an off-line support computer that solves the problem and ultimately produces either a magnetic tape or a perforated Mylar tape that contains the required coordinate data in fire control computer language, This output tape is loaded into the memory of the fire control cornputer for subsequent processing,

The basic steps in the overall programmingprocess *(see* Fig. 4-20) can be described briefly as follows:

1. The programmer prepares a flow chart that outlines the steps to be executed in the solution of the given problem. As a rule, the more complex the problem, the more detailed the flow chart. Careful analysis of the flow chart after it has been prepared by the programmer can be undertaken by the fire control computer designer to check *for* conformance to system requirements.

2. The program is then written in compiler language on a standard coding sheet. Each line on the coding sheet represents a single compiler statement. The format on the coding sheet varies with the type of compiler.

3. A punchedcard is prepared for each compiler statement. When all the cards are punched, they are put together in sequence—thereby comprising the basic "source" program. (This assembly of cards is called the "source deck".)

4. Control cards, peculiar to the support computer, are added to the source deck. These cards tell the support computer what operations to perform, where to store certain information, what type of output to generate, and other control information. Data cards (e.g., constants) may be added to the source deck, or may actually be entered as part of the source deck.

5. The punched-card program is read into the support-computer memory by means of a high-speed card reader. The compiler program and other computer routines are usually stored on magnetic library tapes that are mounted at different tape stations in the computer room. When the support computer is operated, the source program is translated and assembled into machine language to ob-

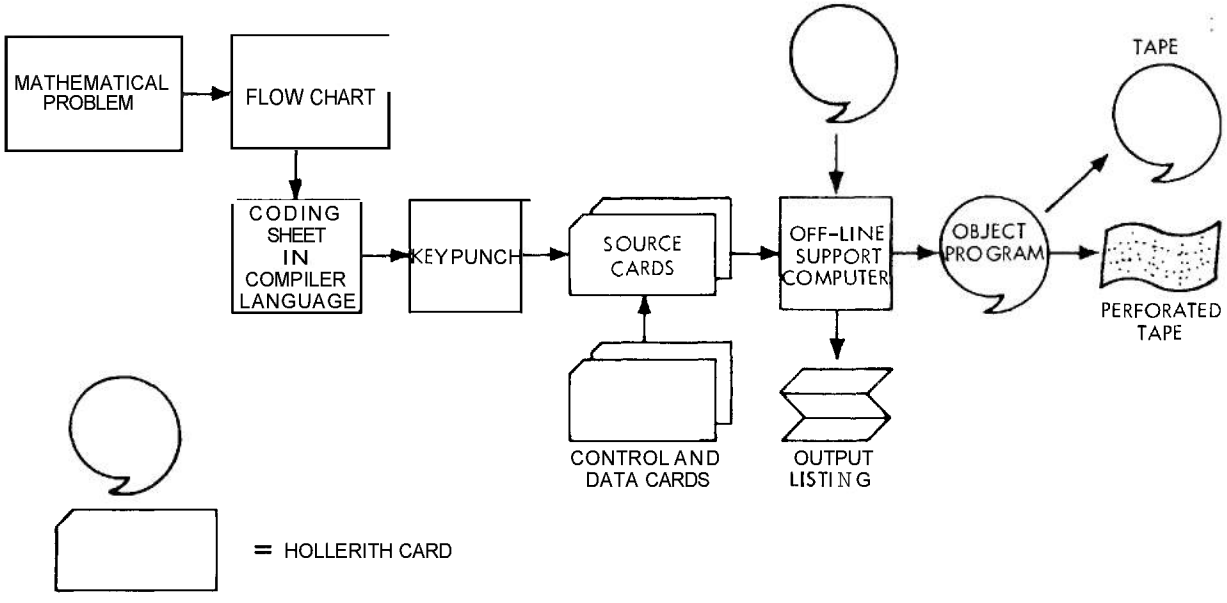


Figure 4-20. The basic programming process.

tain the "object" program, which is then processed to produce the desired output — generally in the form of either a magnetic tape or a perforated Mylar tape. An output listing is provided on a high-speed printer so that the designer can evaluate the program and any compiler-generated error messages.

Fig. 4-21 illustrates a simple program that was written in FORTRAN compiler language and processed on an RCA 301 support computer. The prime objective is to solve the equation

$$y = x^2 + 0.0008356$$

for different values of x under the following conditions:

a. The values of x are pre-punched in Columns 1 to 10 on 80-column cards and are floating-point numbers.

b. The number of data cards is unknown, but it is known that x will never be greater than 9999.0. (A special card is provided that has a value of x>9999.0 to indicate the last card in the series.)

A secondary objective is to point out the values of x and y and the card count on the support-computer output listing. (See Table 4-3 for an explanation of the various FORTRAN compiler statements that appear in Fig. 4-21.)

4-6 COMPUTER TYPES

4-6.1 SYNCHRONOUS AND ASYNCHRONOUS

Most digital computers are synchronized by means of clock pulses as a basic means of timing all the activity in the system. Clock-pulse signals delivered to every flip-flop identify bit times and are used to keep information being written into and read out of memory in synchronism with other data in the memory and in other parts of the computer. In addition, appropriate pulses control all the transfer and exchange of information throughout the machine. A computer slaved to a clock is known as a synchronous machine.

Clock frequencies may be generated by a crystal oscillator, with subfrequencies scaled down through frequency dividers, or the basic pulses may be generated by a multivibrator. Other repetitive sources such as

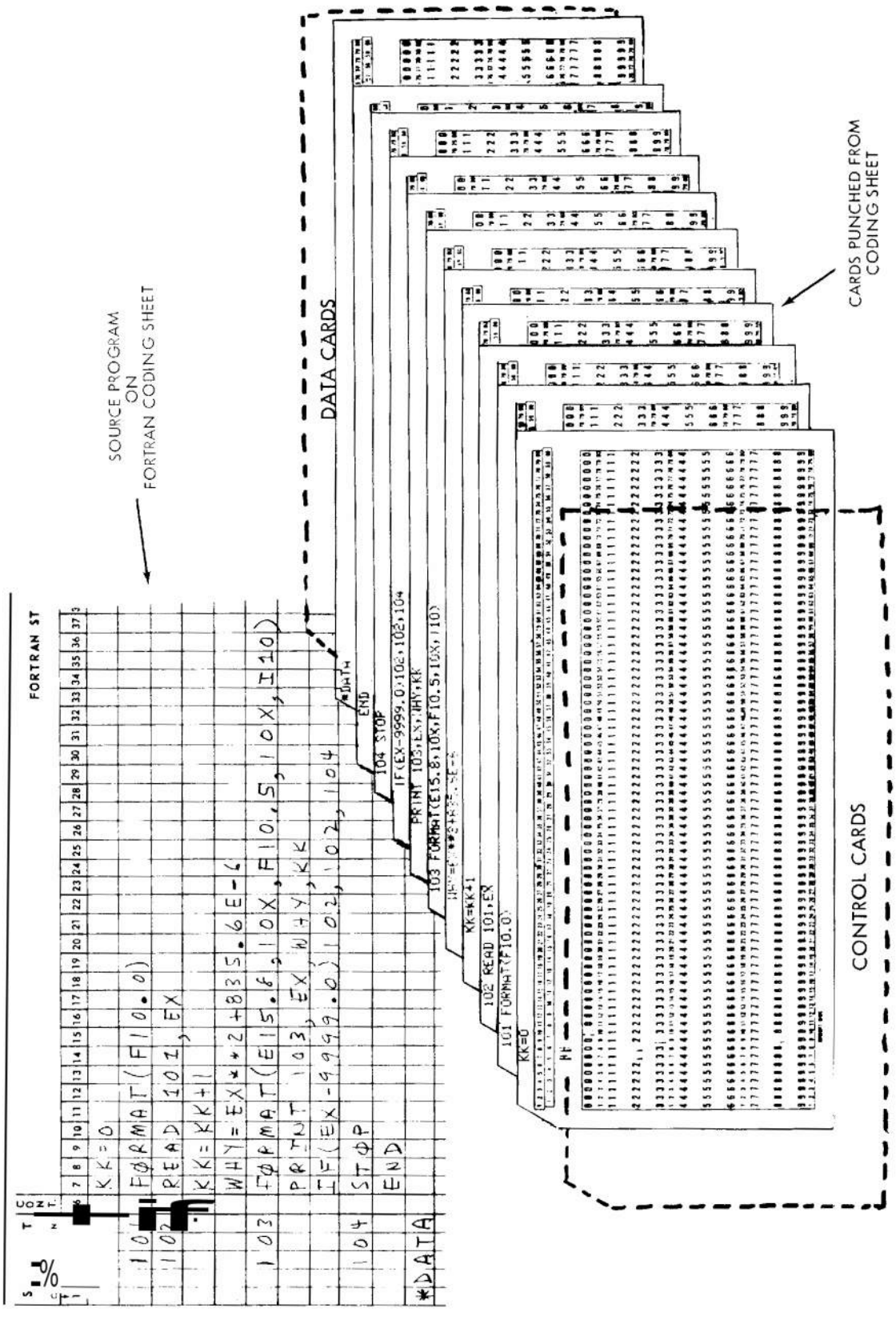


Figure 4-21. An illustrative FORTRAN prog arm. (sheet 1 of 2)

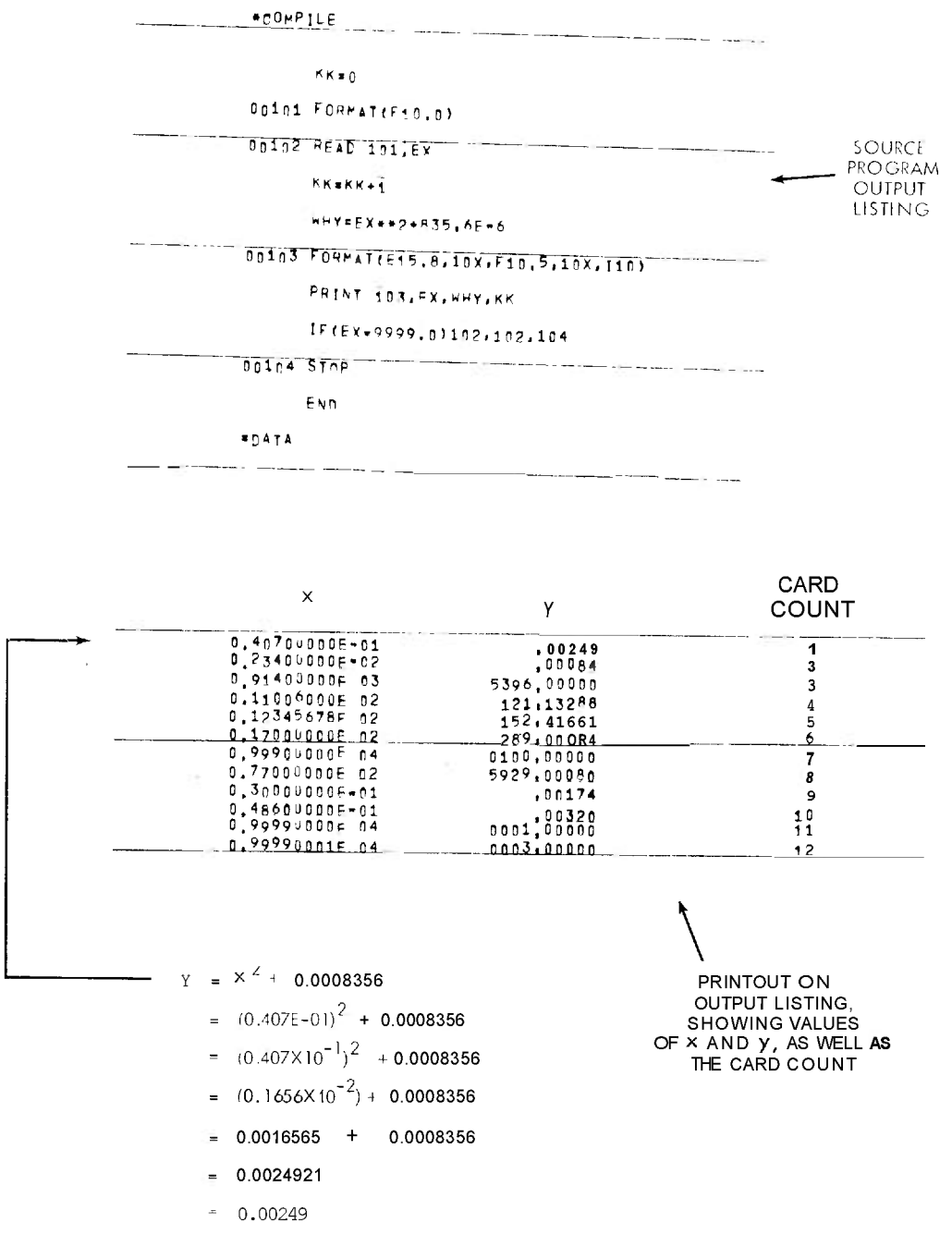


Figure 4-21. An illustrative FORTRAN program. (sheet 2 of 2)

TABLE 4-3. INTERPRETATION OF THE FORTRAN COMPILER STATEMENTS IN FIG. 4-21.

KK=0

This equation sets the card count equal to zero. Since the card count is a positive integer (whole number), an integer variable was selected. In FORTRAN, a variable may be written with up to six alphanumeric characters, the first character being a letter. Integer variable names must*: begin with an I,J,K,L,M, or N. Ked-variable names can begin with any other letter of the alphabet. There can be only one variable on the left-hand side of the equation.

101 FØRMAT (F10.0)

A FØRMAT statement is used in conjunction with a card READ statement to specify the type and arrangement of the data field to be entered as input. In this case, 101 is the statement address or number, and F10.0 states that the input is a floating-point variable of 10 digits, with no digits to the right of the decimal point. (F specifies floating point, 10 specifies a 10-digit width, and .0 specifies zero digits to the right of the decimal point.) If an input card has a decimal point, it takes precedence over the FØRMAT statement. Capital "oh's" are slashed to distinguish them from zeros. Also, parentheses are required around the variable identifier F10.0.

102 READ 101,EX

This statement is used to read in a data card that contains a real variable labeled EX. The 102 is the address associated with the READ Statement, 101 is the address of the associated FØRMAT statement, and EX is the variable name arbitrarily assigned to x.

KK=KK+1

This statement updates the card count by a unit of one.

WHY=EX**2+835.6E-6

This statement sets up the equation

$$y = x^2 + 0.0008356$$

The variable y is represented by the symbolic WHY, x is represented by EX, a double asterisk denotes an exponential, and E-6 is the same as 10^{-6} so that 0.0008356 can be represented as $835.6 \times 10^{-6} = 835.6E-6$.

*The word "must" as used here means that the statement is a rule of FORTRAN.

TABLE 4-3. INTERPRETATION OF THE FORTRAN COMPILER STATEMENTS IN FIG. 4-21. (cont.)

103 FØRMAT (E15.8, 10X, F10.5, 10X, I10)

This FØRMAT statement is used in conjunction with a PRINT statement to specify the type and arrangement of the data field to be outputted. The 103 is the address associated with the FØRMAT statement; E15.8 specifies that the first output will be an exponential of up to 15 characters, with eight digits to the right of the decimal point; 10X means "to skip 10 spaces" on the same line; F10.5 means that the second output will be a floating-point number, with five digits to the right of the decimal point; 10X means "to skip 10 spaces" on the same line; and, finally, I10 specifies that the third output will contain an integer with up to 10 digits.

PRINT 103, EX, WHY, KK

This statement results in the printing of the variables identified in the preceding FØRMAT statement (address 103). The first output to be printed is x; the second output on the same line is y; and the third output on the same line is the card count.

1F (EX-9999.0) 102, 102, 104

This is an arithmetic 1F statement that states that "if x minus 9999.0 results in a minus value, branch to address 102; if x minus 9999.0 results in a zero value, branch to address 102; and, if x minus 9999.0 results in a positive value, branch to address 104. In essence, this means that "if $x \leq 9999.0$, read in another data card, if $x > 9999.0$, stop the program."

104 STOP

This statement is used to stop the machine for operator action.

END

The last statement in a FORTRAN source program must be an END statement. It signals the compiler that the work of translation is completed. (Remember, the entire source program is read into the support-computer memory, compiled, and then executed.)

*DATA

This statement identifies the following cards as data cards.

a timingtrack on amagnetic memory device can be used. When a magnetic drum is employed as a storage element, the timing track employed as the lock generator provides an ideal means of synchronizing the mechanical and electronic elements to each other, Care must be exercised that at the time of start-up the basic frequencies throughout the machine are in phase or in synchronism.

A synchronous machine has the advantage that registers may be cleared on a repetitive cycle, each operation takes a known length of time, and random noise pulses are discriminated against, As a consequence, most computers are basically synchronous, For an example of asynchronous operation, consider such a simple device as a counter fed by a shaft-driven pulse generator. Such a device is asynchronous since pulses are generated at random intervals, depending on the rotation of the shaft.

There is a larger cycle in a synchronous computer that encompasses the time required to complete an addition and transfer the result to storage, This add time is commonly 2 to 20 clock pulses, depending on the logical design,

Often, synchronous computers are required to operate with asynchronous data inputs. To accomplish this, the input data are held in a register until such time as the computer cycle can accept them.

4-6.2 WHOLE-TKANSFER AND INCREMENTAL COMPUTERS

While most digital computers operate on whole numbers, extremely useful special-purpose computers may employ the technique of counting increments to obtain a rapid representation of an answer, Incremental computers perform computations by a series of updating operations, This approach is particularly useful for applications where a continuous output solution is required for continuously varying input parameters, and where conventional electronic or electromechanical means do not provide the requisite zeroing precision. Incremental computers have often been described as hybrid since they operate like analog computing systems with digital-computer accuracy. A substantial saving in computer hardware can be achieved by computing at speeds appropriate to the dynamics of each stage of computation.

The logic at the inputs is made capable of following rapidly varying phenomena. The logic involved in the majority of mixing and output computations takes advantage of the smoothing inherent in integrations at the input.

4-6.3 OPERATIONAL COMPUTERS

Operational computers employ a separate computing element for each mathematical operation in the problem. They are exceedingly fast since equipment need not be time-shared between operations, By the same token, operational computers are wasteful of equipment, and are thus used only for relatively simple problems – for example, simple counters or integrators. While whole-transfer logic may be employed, incremental operation is more often encountered,

An operational computer employing incremental logic, and capable of being programmed for a variety of mathematical problems, is known as a digital differential analyzer. This class of computer is sufficiently important to the fire control field to form the subject of a separate chapter (see Chapter 5).

4-6.4 COMPUTERS AS SERVO ELEMENTS

Special-purpose computer logic can be made to operate as the correction-computing element of a servo if the input to the servo or the feedback element (or both) is digital, In a manner comparable to conventional analog servos, the computer subtracts the output from the input to obtain the error, and may also be programmed with the expressions for the filter networks required to stabilize the servo. At some point in the circuit, digital-to-analog conversion of the error signal is provided. The converter may be quite rudimentary – as, for example, a three-position relay – but in order to reduce the tendency to oscillate, several steps are usually provided *so* as to form a quasi-proportional system (see Fig. 4-22).

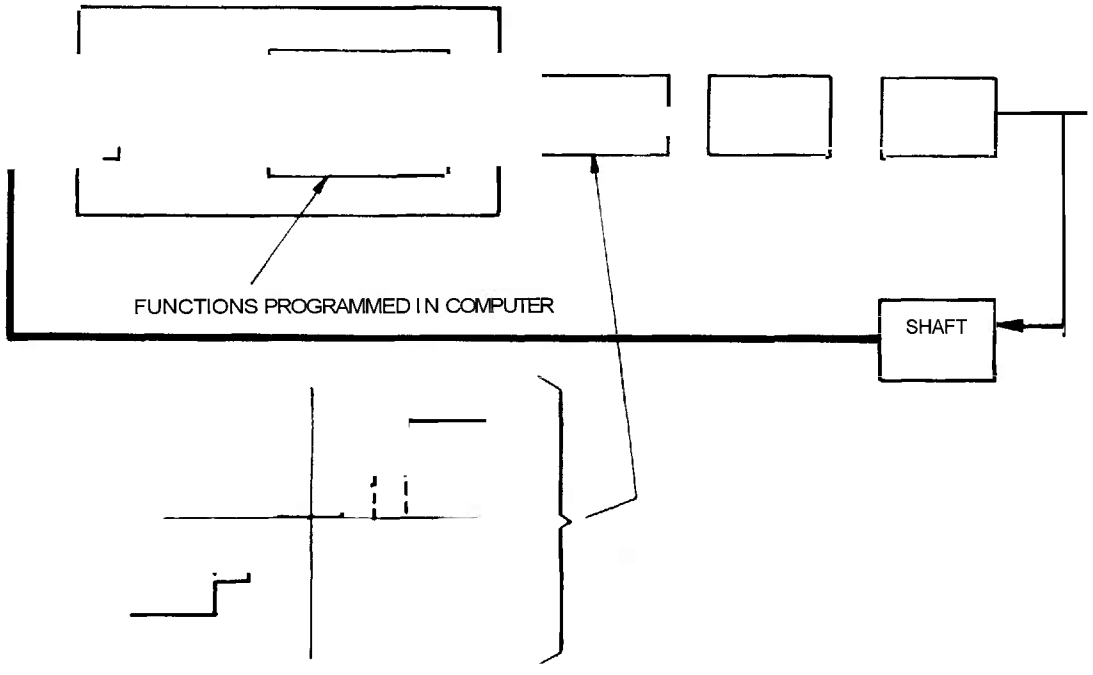


Figure 4-22. Typical digital servo.

4-7 TYPICAL DIGITAL COMPUTER

The description below of a typical computer, which starts with the main storage (memory unit) and works through to the control unit - based on the functional block diagram of Fig. 4-23, is a modified condensation of pertinent portions of Chapter 11 of Ref. 4. For a more complete discussion, the reader should consult this referenced source. It should be noted that the terminology used is merely that employed by the source and is not universally accepted in describing all computers. Any specific machine may have more or fewer registers than discussed here, but the basic concepts discussed apply to all types.

As a rule, main storage devices are not satisfactory for the temporary storage of numbers undergoing arithmetic operations or controlling the sequence of operations. Three storage registers, each adapted for speed and capacity, are used for storing numbers that enter into the computations.

The first storage register, the S-register, serves the primary function of storing the multiplicand (or divisor) so that it is not necessary to refer repeatedly to main storage during multiplication or division. Numbers are transferred from the main storage to the S-register over a set of parallel wires, one for each binary digit.

The second register, the accumulator, is used for binary addition or subtraction. The third register, called the multiplier-quotient or M-Q register, is used for storing the multiplier during the multiplication or the quotient during division.

For addition and subtraction, numbers are taken from appropriate locations in the main storage and sent to the accumulator. They are sent through the S-register because this path is needed anyway for other purposes. For multiplication, it is necessary as a first step to cause the multiplier to be transmitted from the main storage to the M-Q register. The accumulator and M-Q register are both capable of shifting the numbers in them to right or left. Then, at the start of the actual multiplication process, the multiplicand is obtained from the main storage and placed in the S-register to be added in over-and-over

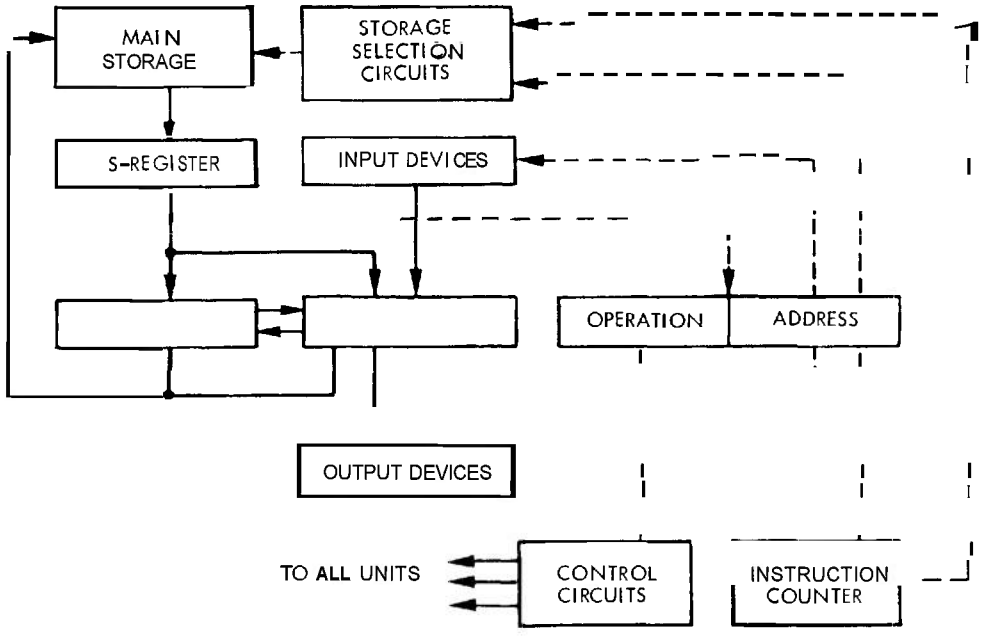


Figure 4-23. Typical arrangement for a stored-program computer.

fashion in the accumulator, which shifts to the right one step after each addition. As the product is built up, it is shifted into M-Q register, At the conclusion of the multiplication process, the double-length product is stored with its high-order digits in the accumulator and its low-order digits in the M-Q register. If it is desired to retain the entire product, two storage locations in the main storage are required and two program steps are used to transfer it from the two registers. Division is substantially the reverse of multiplication. The dividend is placed in the accumulator (or the accumulator and the M-Q register if a double-length dividend is required) and the divisor is stored in the S-register. As the division process proceeds, the digits in the accumulator are shifted to the left, with the result that the final remainder appears in the accumulator and the quotient in the M-Q register. In Fig. 4-23, all paths used for the data and results are indicated by solid lines,

The problem is now to control the transmission of data between the main storage and the three registers in the arithmetic portion of the machine. This function is to be accomplished through the use of numbers representing program steps, and these numbers are to be stored in the main storage along with the numbers representing data. The instruction counter, the operation-address register, and the control circuits are the major units that are employed for accomplishing this purpose.

The instruction counter has two functions, First, it keeps track of the program step that the computer is executing at any given time, Normally, a pulse is sent to the instruction counter at the conclusion of each arithmetic operation to step it up by one count; for altering or repeating a program, however, the contents of the address part of the operation-address register may be transferred to the instruction counter to replace the number there, The second purpose of the instruction counter is to control the storage-selection circuits when a number representing a program step is being sensed in the main storage. The paths to and from the instruction counter are indicated in Fig. 4-23 by dotted lines, as are all of the paths that transmit information pertaining to the control or programming of the computer,

Before describing the function of the operation-address register, the meaning of the term "address" must be explained, In its narrowest sense, an "address" is a number

that represents a storage location in the main storage, Usually, each location is assigned one of a series of consecutive numbers from zero to the storage capacity of machine. Then, when an address is sent to the storage-selection circuits, access is gained to the storage location represented by that address. By this definition, the number in the instruction counter is more than an abstract number used for counting program steps; rather, it is an address also, because it prescribes the storage location from which a number representing the program step is to be taken. An address can be used to designate other things. For example, it is used to specify the desired input or output mechanism when sending information to or from the computer, Also, the address specifies the number of shifting steps that are to take place in a shift operation.

The operation-address register is used for storing the "instruction", which has been previously referred to as the number that represents the program step. An instruction consists of two parts, known as the operation part and the address part. The operation part specifies the operation to be performed, which may be an arithmetic operation such as add or multiply, or which may be any one of a long list of other operations such as the transfer of a number from one place to another or the causing of a magnetic tape unit to rewind, This part of the instruction causes the computer to perform the indicated operation by means of control circuits. As the name implies, the address part of the instruction specifies the addresses of the operands when the main storage is involved or the input-output device, the number of shifts, and *so* on, as the case may be in other types of operations. Incidentally, the address part of the register is a counter as well as a register and is used for keeping track of the shifts in a shift instruction, or during a multiplication or a division,

During operation, the computer alternately comes under the control of the instruction counter and the operation-address register, To visualize the sequencing of the computer functions, assume that the program is initially stored in the main storage with at least the first few instructions in the lowest numbered addresses, The various items of data may be at any desired addresses. If the instruction counter is initially at zero, the control circuits first cause the instruction at address zero to be taken from the main storage and sent to the operation-address register. (The fact that the path is through the S-register is incidental.) Normally, the instruction is rewritten at address zero so that it may be used again. The computer then performs the operation indicated by the digits in the operation part of the address register, Upon completion of the first operation, control is returned to the instruction counter, which has in the meantime been stepped from zero to one, The instruction at address one is now caused to be sent from the main storage to the operation-address register, after which the second instruction is executed under control of this register, and *so* on. In other words, each program step consists of two parts: (1) the securing of the instruction and (2) the execution of the instruction. Reference to the main storage may be made during each part; in (1) the storage-selection circuits are under the control of the instruction counter, while in (2) they are under the control of the address part of the operation-address register,

The basic problem of altering a program or repeating portions of it is solved in the stored-program computer by a "jump" (sometimes given other terms such as "branch" or "transfer") instruction. The jump instruction causes the address part of the instruction, which is in the operation-address register, to be sent to the program counter to replace the number there. The result is that the uniform sequence of addresses from which instructions are obtained is terminated, and a jump is made to some other address, Then, because the program counter received one pulse to be counted for each program step, the selection of instructions from sequentially numbered addresses is resumed at the new address and is continued until another jump instruction is encountered.

A second important feature of the instructions in a stored-program computer is that they are indistinguishable from the data, The programmer must keep track of which is which. Occasionally a certain amount of confusion results, but it is useful to be able to perform arithmetic operations on instructions. The addition or subtraction of a constant from the address part of an instruction

is an operation that is performed frequently when using subprograms. Another example of the usefulness of the feature is in the storage of tables when the arguments form a uniform sequence. To find the address of the value corresponding to any given argument, it is sufficient to perform a simple computation on the argument and then use the result as the address part of an appropriate instruction. A time-consuming searching process is thereby avoided,

The input and output devices are shown connected through the M-Q register in Fig. 4-23. That the M-Q register is used in this way is incidental; it just happens to be convenient, However, some temporary storage of some sort is usually needed between the input and output devices and the main storage because the various units are not synchronized with one another, When an instruction calls for a number to be sent from main storage to an output device, for example, the output device may not at that particular instant be prepared to accept it, Then when the output device is ready to accept the number, the timing in the arithmetic part of the computer may not be at the right point for transmission. Another factor, which is probably even more compelling is the fact that the form of the number may be different in the two places, Both the timing and the change-of-form problems can be solved through the use of "buffer" storage, as it is sometimes called,

The objective to be accomplished by the control circuits in a stored-program computer is the causing of all the individual units of the computer to perform in such a manner that the instructions in the main storage are sensed in the proper sequence and executed. In general, the units are controlled by sending pulses to them over a set of wires that may be called "command lines", Each command line is for a specific purpose, such as transferring a number from one register to another, shifting the number in a register, resetting a flip-flop, or any one of a multitude of other functions, Usually, it is necessary to send pulses, appropriately sequenced in time, over several different command lines to execute any one instruction, The circuit arrangement to be used in any given case for distributing the control pulses on the command lines depends in large measure on the organization of the computer as a whole, and in existing machines great variations will be found when comparing one computer with another.

For information on actual computer system design using transfer equations, the reader is referred to Ref. 100. For information on the detailed design of control circuits, counters, and other types of computer networks, the reader is referred to Ref. 101.

4-8 LOGICAL DESIGN

The logical design of a digital computer refers to the design of switching networks that can perform the mathematical operations desired. These operations, reduced to their essentials, are made up of a few basic logical propositions, For this reason, the algebra of symbolic logic – termed Boolean algebra – is the basic tool of the designer of logical systems. The specific application of Boolean algebra to switching networks is known as switching algebra. For more complete discussions of Boolean algebra and its applications, see Refs. 3, 8, 9, 10, 11, 79, and 87.

One point should be clearly understood with regard to the use of Boolean algebra; namely, it leads to a minimal number of logic elements, but not necessarily to the "best" circuit design in terms of operating performance. What the algebra does provide is a convenient means of representing a switching circuit without drawing the circuit. Also, and probably more important, is the fact that it provides a means for quickly finding a multitude of different circuits that will perform any desired switching function, With a little practice, the circuit designer can thereby possess a powerful tool to aid him in finding a "good" circuit, even though it may not be the "best" one.

The subject of logical design of digital computers is much too complex a subject to cover with any degree of thoroughness here, For such coverage, the reader should consult Refs, 4 and 77 through 86. Certain of these references merit particular note, as follows.

Ref. 79 is concerned primarily with relay switching but gives a good treatment of switching algebra and covers some electronic switching applications. Ref. 80 is a more recent book that gives a thorough coverage of up-to-date switching theory applications. This covers switching algebra, switching

components, the various minimization methods, and many of the design criteria concerning both contact networks and electronic switching circuits. It also has a very good treatment of the synthesis of sequential circuits, including pulsed sequential circuits. Ref. 81 covers switching algebra and the simplification of logical functions, but goes much further into the actual logical design of digital computers. It covers the derivation and manipulation of the logical equations for memory elements, input-output equipment, arithmetic units, and control units. Ref. 4 is concerned more with actual circuitry than with the derivation and manipulation of the logical algebraic equations, Ref. 86 goes into some details of actual digital computer circuitry.

The paragraphs which follow cover very briefly some of the concepts of switching algebra and their application to logical design. Although the switching-theory applications are illustrated in terms of relay-contact networks, it should be noted that switching algebra can be applied to electronic switching networks as well,

In a digital computer, an element of a complete circuit of elements may have either one of two values, For example, the output of an electronic circuit may be a voltage that is either high or low, Also, the output of a relay contact network may be the presence or absence of a connection to ground. Thus, a switching variable can represent either the variation of a particular element of a switching system, or it can represent the resultant variation produced by a group of elements, It is convenient to assign values to the switching variable, represented by the digits 0 and 1. The digit 0 could represent either a closed circuit or an open circuit, a high voltage or a low voltage; and vice versa for the digit 1. (It does not really matter which assignment is made, as long as one is consistent,) A prime indicates the inverse.

Table 4-4 is a list of postulates and theorems of switching algebra. Most of the postulates and theorems have dual forms that are stated together. The principle of duality is that any expression in switching logic can be converted to its dual by interchanging both the digits 0 and 1 and the operations add and multiply.

Note that postulate 2' does not correspond to the rules of ordinary algebra. One will notice that $(x + x)$ is equal to x and that $(x)(x)$ is equal to x, both of which are unexpected results, It is obvious then that the addition and multiplication symbols used in switching algebra are not quite those that are used in ordinary algebra.

Consider, for a moment, simple contact networks such as those obtained when using relay switching networks and let the digit 0 be the value of an open circuit and the digit 1 be that of a closed circuit. Furthermore, assign the contacts of each relay involved a capital letter – denoting, for example, the contacts of one relay by the capital letter A for all contacts on that relay that are normally open and A' for all contacts on that relay that are normally closed. Then an expression can be written for the transmission of a contact network in terms of the letters representing the contacts on the various relays involved in the network. This is illustrated in Fig. 4-24 which shows the two types of symbols that are often used to represent the contacts of a relay. Fig. 4-24(A) shows the normally open contacts, i.e., those contacts that leave an open circuit until the relay coil is energized and are then closed, Fig. 4-24 (B) shows the symbols used for normally closed contacts on a relay, i.e., those contacts that are closed until the relay coil is energized and are then opened. The lower symbols shown for both cases in Fig. 4-24 are

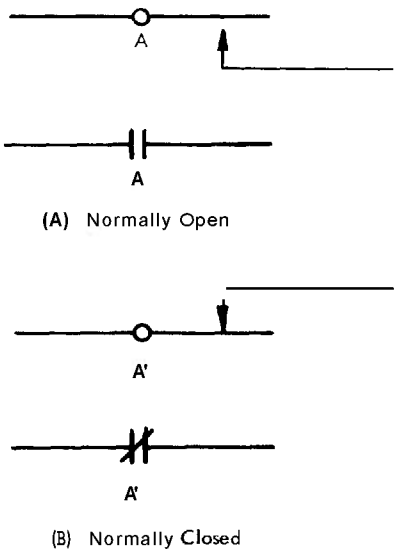


Figure 4-24. Representation of relay contacts.

TABLE 4-4. POSTULATES AND THEOREMS OF SWITCHING ALGEBRA.[11]

POSTULATES

(1) $X = 0$ if $X \neq 1$
(1') $X = 1$ if $X \neq 0$

(2) $0 \cdot 0 = 0$
(2') $1 + 1 = 1$

(3) $1 \cdot 1 = 1$
(3') $0 + 0 = 0$

(4) $1 \cdot 0 = 0 \cdot 1 = 0$
(4') $0 + 1 = 1 + 0 = 1$

(5) $0' = 1$
(5') $1' = 0$

THEOREMS

(6) $X + 0 = X$
(6') $X \cdot 1 = X$

(7) $1 + X = 1$
(7') $0 \cdot X = 0$

(8) $X + X = X$
(8') $X \cdot X = X$

(9) $(X)' = X'$
(9') $(X')' = X$

(10) $X + X' = 1$
(10') $X \cdot X' = 0$

(11) $X + Y = Y + X$
(11') $X \cdot Y = Y \cdot X$

(12) $X + XY = X$
(12') $X(X + Y) = X$

(13) $(X + Y')Y = XY$
(13') $XY' + Y = X + Y$

(14) $X + Y + Z = (X + Y) + Z = X + (Y + Z)$
(14') $XYZ = (XY)Z = X(YZ)$

(15) $XY + XZ = X(Y + Z)$
(15') $(X + Y)(X + Z) = X + YZ$

(16) $(X + Y)(Y + Z)(Z + X') = (X + Y)(Z + X')$
(16') $XY + YZ + ZX' = XY + ZX'$

(17) $(X + Y)(X' + Z) = XZ + X'Y$

(18) $(X + Y + Z + \ldots)' = X'Y'Z' \ldots$
(18') $(XYZ \ldots)' = X' + Y' + Z' + \ldots$

(19) $f(X_1, X_2, \ldots, X_n, +, \cdot)' = f(X_1', X_2', \ldots, X_n', \ldots +)$

(20) $f(X_1, X_2, \ldots, X_n) = X_1 f(1, X_2, \ldots, X_n) + X_1' f(0, X_2, \ldots, X_n)$
(20') $f(X_1, X_2, \ldots, X_n) = [X_1 + f(0, X_2, \ldots, X_n)][X_1' + f(1, X_2, \ldots, X_n)]$

(21) $X_1 \cdot f(X_1, X_2, \ldots, X_n) = X_1 \cdot f(1, X_2, \ldots, X_n)$
(21') $X_1 + f(X_1, X_2, \ldots, X_n) = X_1 + f(0, X_2, \ldots, X_n)$

(22) $X_1' \cdot f(X_1, X_2, \ldots, X_n) = X_1' \cdot f(0, X_2, \ldots, X_n)$
(22') $X_1' + f(X_1, X_2, \ldots, X_n) = X_1' + f(1, X_2, \ldots, X_n)$

the ones most commonly used by logical designers, Fig. 4-25 shows three typical simple networks. Fig. 4-25(A) shows a series connection of normally open contacts of relays denoted A and B. The transmission function is

$$T = A \cdot B = AB$$

Fig. 4-25(B) shows the parallel connection in which the transmission function becomes

$$T = A + B$$

These two expressions may be checked by reference to the list of postulates and theorems of switching algebra given in Table 4-4 and allowing the variable A and B to take on all possible values. If both variables A and B are zero, which would mean that the contacts were open by our previous definition, then the transmission of the network is certainly open or 0. To check, one may use postulate 2. If A = 0 and B = 1, the transmission is still 0 by inspection, as would also be obtained from postulate 4. The same result occurs if A = 1 and B = 0. If both contacts are closed, i.e., A = B = 1, then the transmission is certainly 1, which is given by postulate 3.

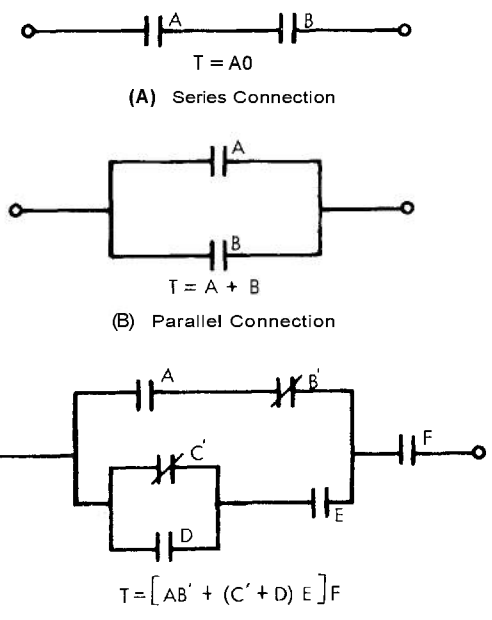


Figure 4-25. Three simple contact networks.

Within this definition, it is seen that a series connection of elements, or groups of elements (a single letter such as A could stand for a network of elements as well as a single contact), results in the multiplication sign. A parallel connection results in the addition sign. Fig. 4-25(C) shows the result of a series-parallel connection, The correctness of the transmission function can be checked from Table 4-4.

The usefulness of switching algebra to the simplification of switching networks can be shown by a simple example, Consider the network of Fig. 4-26(A). It is not entirely obvious that this network can be simplified further. However, by direct application of theorem 16' the transmission expression immediately becomes that of Fig. 4-26(B) and results in the network shown, yielding a reduction in number of contacts by one-third. The reader can verify the correctness of the second network by constructing a truth table for both networks. A truth table is a table in which there is a column for each of the variables involved and the rows of the table constitute all possible combinations of the values of 0 and 1 that the network of variables can have. Finally, a column is tabulated that gives the value, 0 or 1, of the transmission function for each row. Table 4-5 is a truth table for the networks shown in Fig. 4-26. The variable A' could have been included in the table also, but the information would have been redundant since when A = 0, A' = 1 and vice versa.

The construction of a truth table from the requirements for a switching network stated in words is usually the first step in the synthesis of the network. For example, suppose that the requirements for a particular network were that the transmission is to be 1 (i.e., the circuit is to be closed) for the following four conditions:

1. When C is energized but A and B are not energized,
2. When B and C are energized but A is not energized,
3. When A and B are energized but C is not energized.
4. When A, B, and C are all energized.

The requirements for the network as stated would result in the truth table shown in Table 4-5. From the truth table, the transmission function could be written directly by using the four rows in which the value of T appears as 1, that is,

$$T = A' B' C + A' B C + A B C' + A B C$$

The next step inlogical design would be to minimize the transmission function as much as possible. In a simple case like this, the postulates and theorems presented in Table 4-4 could be used to minimize the numbers of terms in the transmission function directly, resulting in the expression

$$T = A B + A' C$$

The resulting network, of course, is that of Fig. 4-26(B).

The preceding discussion pertains to combinational circuits – i.e., circuits whose outputs are determined at any time by the particular combination of inputs at that time. Switching circuits can also be designed so that the outputs at any time are determined by the past history of the inputs; such a circuit is termed sequential and has many uses in digital computer design. In a combinational circuit, each input combination determines a unique output condition. In a sequential switching circuit, however, the output conditions are determined jointly by the sequence in which input signals occur as well as by their combination. It is apparent, then, that one of the characteristics of the sequential switching circuit is the presence of memory elements.

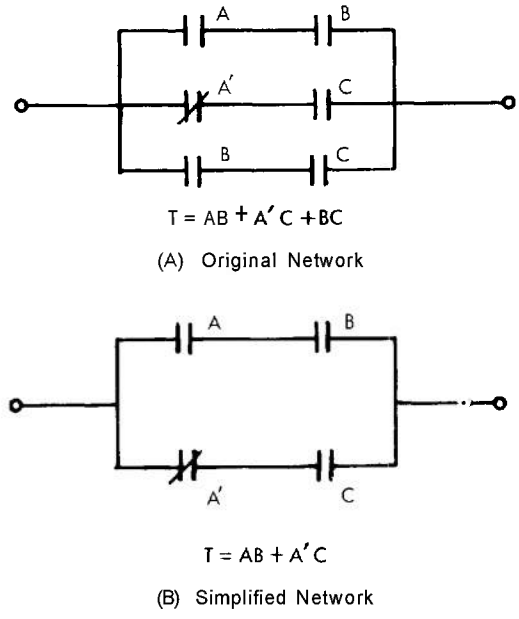


Figure 4-26. Simplification resulting from application of theorems.

TABLE 4-5. TRUTH TABLE FOR FIGURE 4-26.

A	B	C	T
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Two-terminal sequential switching circuits may be attacked by the use of time charts and sequence diagrams as outlined in Ref. 79. These time charts and sequence diagrams are a means for indicating graphically the action taking place in a circuit and how these actions are related to each other in time. By the use of such techniques, the designer of relay circuits can synthesize a circuit by determining the operate and release times of both primary and secondary relays from inspection of the time charts or sequence diagrams. The conventional technique of combinational relay circuitry can then be used to develop the particular control circuit for the secondary relays and eventually the output of the circuit. Ref. 80 presents a thorough treatment of sequential-circuit analysis and synthesis as applied both to relay contact networks and electronic switching systems.

Switching algebra can be used not only for analysis of networks but also for synthesis. The synthesis procedure is the more difficult and more important part of logical design. There is not space available here to go into the various available minimization methods with enough detail to be of any benefit to the reader. Therefore, the reader is referred to Refs. 79 through 90.

4-9 COMPUTER NUMBER SYSTEMS

4-9.1 BINARY SYSTEM

The binary system provides one reliable method of representing numbers with electronic circuits that recognize only two voltage levels. There are many electronic devices with two stable states that may represent 1 and 0. The performance of arithmetic in the binary system is simple, and this system requires less equipment than does the decimal,

Choosing a representation for a binary digit in a computer involves relatively straightforward choices, such as a signal or no signal, a signal on one of two different lines, or a positive or a negative signal to represent a 0 or a 1.

(A brief discussion of number systems of various radices appears in par. 4-1.2, Number Systems.)

4-9.2 BINARY CODES

Many computers have been built to utilize a number system that constitutes a compromise between the binary and decimal classifications, Such a system falls in the class called binary-coded-decimal. It is basically decimal, but each decimal digit is represented by, or encoded with, several binary digits. There are many kinds of binary-coded-decimal systems possible for representing a decimal digit with a minimum of four binary bits by using only 10 and ignoring the remaining 6 of the possible 16 different combinations. It should also be noted that, very often, different codes are used in the transmission of data than are used in the computer itself, and often there are different codes used in the memory of the computer than are used in the arithmetic unit.

4-9.2.1 Reflected Binary (Gray) Code

A Gray code is frequently used in analog-to-digital converters to minimize readout ambiguity since only one digit changes at any one time as the count progresses from zero to full scale (see Chapter 7, Analog-digital Conversion Techniques). In Gray codes, the maximum ambiguity is plus or minus one least significant bit, whereas in natural binary code it is possible for errors of many digits to occur as the encoder hovers at the boundary between two successive natural binary numbers. Table 4-6 illustrates four examples of binary Gray code systems.

4-9.2.2 Decimal Codes

Gray codes share the problem of difficult reading that are inherent in natural binary code, and hence some one of the more than 29 billion possible decimal codes is often used in the input-output devices peripheral to the computer itself. Almost all data pertaining to problems the computer must solve are best checked in decimal notation at the time of design, at the time of data and instruction entry, at the time of checking intermediate results, and at the time of recording final solutions. Therefore, in those cases where a keyboard machine is used as an input-output device, where a quick-look digital display is

TABLE 4-6. GRAY CODES.

	Code 1	Code 2	Code 3	Code 4
0	0 0	0 0 0	0 0 0 0	0 0 0 0
1	0 1	0 0 1	0 0 0 1	0 0 1 0
2	1 1	0 1 1	0 0 1 1	0 1 1 0
3	1 0	0 1 0	0 0 1 0	0 1 1 1
4		1 1 0	0 1 1 0	0 0 1 1
5		1 1 1	0 1 1 1	0 0 0 1
6		1 0 1	0 1 0 1	0 1 0 1
7		1 0 0	0 1 0 0	0 1 0 0
8			1 1 0 0	1 1 0 0
9			1 1 0 1	1 1 1 0
10			1 1 1 1	1 1 1 1
11			1 1 1 0	1 1 0 1
12			1 0 1 0	1 0 0 1
13			1 0 1 1	1 0 1 1
14			1 0 0 1	1 0 1 0
15			1 0 0 0	1 0 0 0

desirable, or even where cards or tape are used to feed stored data into the computer memory, a form of decimal code may be brought into play as a compromise between machine-readable and human-readable languages.

Of the many possible 4-bit codes, relatively few have the property that values or weights can be assigned to the 4 bits with the decimal digit being represented equal to the sum of the weights; three of the more useful 4-bit weighted codes are shown in Table 4-7.

Of some 71 known weighted 4-bit codes, 18 are self-complementing -- such as the 2421 code in Table 4-7. The 8421 is one of the most straightforward 4-bit codes because each decimal digit is represented in a conventional binary system. A disadvantage of the code is that it is not self-complementing. A self-complementing decimal code is one in which the 9's complement of each decimal digit may be obtained by changing the 1's to 0's and the 0's to 1's in the coded representation of the digit.

One nonweighted code that is often used is the excess - 3 code – so-called because it may be generated by adding a binary 3 to each digit representation in the conventional 8, 4, 2, 1 code. This code is shown in Table 4-8. It is a self-complementing code and has the

TABLE 4-7. LISTING OF THREE OF THE 4-BIT WEIGHTED BINARY-CODED-DECIMAL SYSTEMS.

Decimal	8421	2421 *	5421
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

* A self-complementing code

TABLE 4-8. EXCESS - 3 CODE

0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

further advantage that all decimal digits have at least one "1" in the representation *so* that zero and the condition of no digit at all may be distinguished. In many computers, a redundancy bit is used for checking purposes and when this is done the advantage of the excess - 3 code with regard to the representation for zero is largely nullified.

4-9.2.3 Error-detecting and Correcting Codes

The brute-force method of assuring greater computer accuracy is through duplication of calculations and of transfers throughout the machine. Rather than duplicating transfer operations and equipment, it is possible to attach extra bits to each block of data being transferred in such a way that these bits make it possible to detect and correct many errors. A common method is the addition of a "parity" bit, whose value is made 1 or 0 as required to make the bit total in the character always odd or even, Errors can be immediately detected by examining for parity as often as necessary. The choice of odd or even for parity will depend on the particular effect the most probable kind of error will have in a given machine.

The designer must then decide what action to take when an error is detected. Special codes make it possible for the computer to detect and correct certain errors automatically.

In any code composed of binary bits, if a single error in a bit combination can produce another bit combination that is also in the code scheme then the error cannot, in general, be detected. In order to detect the presence of a single error in the bits of a code, it is necessary that the code be such that at least two changes must be made in the bits of the code when changing from the representation of one digit to the representation of any other digit. By adding even more binary bits to the binary representation of a decimal digit, a code can be made error-correcting as well as error-detecting. Such a code requires that at least three changes in

the bit combination be made when changing from therepresentation of one to the representation of any other digit, In this case, a single error will produce a bit combination that can be recognized to contain an error. Furthermore, the individual bit in error can be determined. When two errors occur simultaneously, the resulting bit combination will be recognized as not corresponding to any digit, but the changing of one bit may produce a bit combination that corresponds to one of the digits that is not the desired'digit. Such a code would detect two errors but correct only one. Double-error correcting, triple-error correcting and more powerful schemes maybe devised through the use of codes requiringstill more changes in going from the representation of one digit to another. For the error-detecting, error-correcting, and double-error-detecting codes, a minimum of 5, 7 and 8 bits, respectively, is necessary, It is obvious that more equipment is necessary to implement error-detecting or corresponding codes in a computer. The fact that more equipment means a higher probability of failure requires that a careful study be made before deciding what particular code should be used.

Detailed discussions of error-correcting codes may be found in Ref. 79.

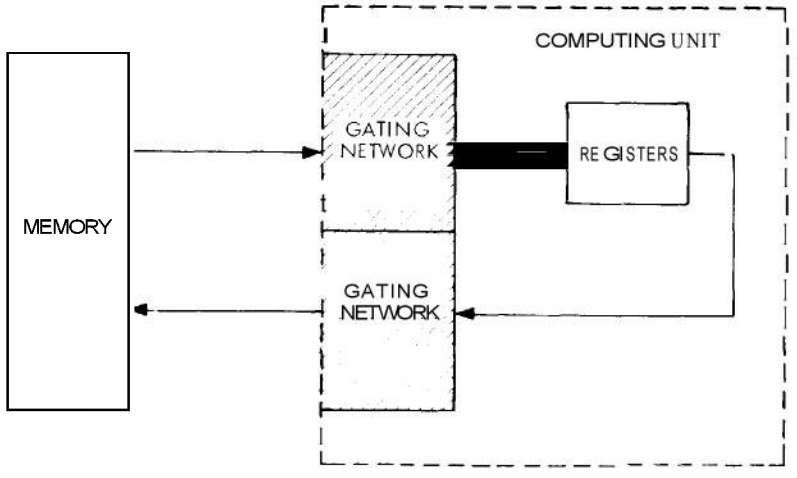
4-10 CLASSES OF COMPUTER LOGIC

The discussion of serial and parallel logic that follows is based on information given in Chapters 7 and 9 of Ref. 2 and Chapter 15 of Kef. 76. For further information on this subject, the reader should consult these excellent sources.

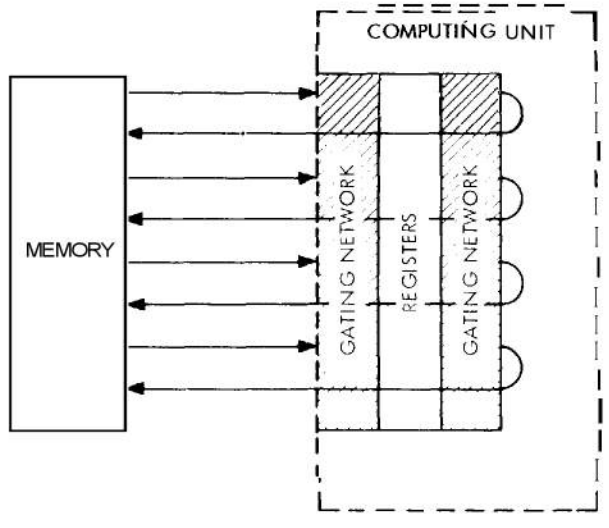
The computer designer has a choice not only of the code used to represent numbers in his computer, but also of whether the coded numbers are tobe operated on in serial or parallel form (see Fig. 4-27), or in some combination thereof. If an arithnietic operation is performed serially, the necessary equipment may be relatively simple, for the logical equations are dependent on only a few bits of each word at one time. The unit of computation time is then one word-time, or the the time it takes for a word to shift serially through the arithmetic unit. If a computation is performed in parallel, the equipment necessary is more complicated, for the most significant digits of a number may be a function of all of the less significant ones. As a result, some memory-element input equations will be very complicated and may be functions of a great many variables. In addition, there must be an equation for each digit in the answer, whereas in a serial unit one equation determines all the various digits of the answer, one by one, over the word-time, The parallel arithmetic unit, though it is more complicated and therefore more expensive than its serial counterpart, is also faster. A complicatedoperation maybe carried out in one bit-time or a few bit-times in *a* parallel machine, as compared with a word-time for a serial unit. If a word contains forty bits, this may mean an increase in speed by a factor of ten to forty.

In a parallelcomputer, all the bits of a word are operated upon and must be available simultaneously; in a serial computer, they are operated on sequentially, one at a time. A magnetic drum, on which bits and words are scanned in sequence by a read-write head, is inherently a good serialmemory device. However, it may also be used as a parallelmemorybyrecording all n bits of a word in n separate channels on the drum and by reading them simultaneously. Note that it may then be necessary to make every other bit in each channel a space bit because of the overlap that occurs in writing on the drum. Because of this, and because the speed inherent in parallel arithmetic operations may be lost as a result of the drum access time, drum memories are usually employed to store serial information.

Because a core memory is inherently able to make any bit available in a few clock-pulse times, at most, it is usually used in a parallel computer where quick access to information is important. It is, however, also possible to employ a magnetic-core memory to read out and write in words one bit at a time so that they may be handled serially by the computer. In such a memory, it is very desirable that the computer clock-pulse interval and the interval between bits in a word read from or written into the memory be the same. If they are not the same, for example, if a bit comes from the memory



(A) Serial Computer



(B) Parallel Computer

Figure 4-27. Serial vs parallel computer.

every two clock-pulse times, the arithmetic logic must be arranged accordingly. Even if they are the same, there may be a delay of several bit-times between the time an address is presented tothe memory and the time the least significant bit of the selected word is available. Thelogical designer must allow for this delay.

A magnetic tape generally contains several channels, recorded side by side, each with its own read-write head. One or two of these channels are reserved for clock pulses and word markers. The remaining channels are used for data. If there is but one more, data can only be recorded serially. If there are twenty or more, data may be recorded in parallel. If there are fewer than twenty, some series-parallel arrangement of data is indicated. For example, with ten data channels it will take two tape clock-pulse times to read a complete twenty-bit binary word; with four data channels in a decimal digit, a word ten digits long would require ten tape clock-pulse times.

The discussion of static and dynamic logic that appears in the six paragraphs

which follow is extracted from Chapter 13 of Ref. 3. For further information on this subject matter, the reader should consult this excellent source.

Information can be stored in two ways:

1. Using dynamic storage, an electrical waveform, bearing information by virtue of its shape, may be preserved in toto by entering it into a delay of some sort. This delay emits the original waveform some time later without any significant change other than attenuation and tolerable distortion. Delay-line serial memories are one type of dynamic storage.

2. Using static storage, digital information in the form of one of a multiplicity of choices of states may be stored in a multistable device by setting such a device to one of its alternate states. Thus, a four-position switch may store one-out-of-four or quaternary information by the way in which it is set.

It should be noted that the intent of dynamic storage is to maintain the information in its original form. The information-bearing wave phenomenon is made to persist by interposing a transmission path that hinders its transit. It is the nature of such a device to cause degradation of the wave form so tliat it must be repeatedly amplified and reshaped to resemble its original form.

Static storage is a mapping of the information into a number of devices tliat have as many possible states as there are possibilities for each "piece" of information. Hence, for binary information, bistable devices are appropriate.

Some elements have a tendency over a period of time to lose the information stored in them. This property is called volatility. The Williams tube, an electrostatic storage device, leaks the charge indicating a 1 from one spot (storage element) to another in a matter of fractions of a second. Frequent regeneration cycles are required to maintain the information without loss. Historically, this was the first high-speed storage device to find use in automatic computers. Because of its volatility, however, it is no longer popular as a memory device since nonvolatile devices are now available.

Devices whose elements are not subject to deterioration in the discrimination between two states over long periods of time – days, months, or years – are called nonvolatile storage elements.

If scanning the elements to retrieve the information causes the inforrnation to be removed from the elements, they are said to have destructive read-out. Core memories, for instance, requirethat each core be set to 0 to be read out. Destructive read-out elements can be used to construct a nondestructive-remembering memory; in that case, the remember cycle includes a read phase and rewrite phase.

4-11 PREDOMINANT LOGICAL COMBINATIONS

4-11.1 GATES

The logical block symbols for the basic AND-OR gates used in computer logic are shown in Fig. 4-28. An AND gate provides a 1-state or HIGH output only when all inputs are HIGH; an OR gate provides a HIGH output when one or more of the inputs are HIGH. With AND-OR gates, voltage levels for HIGHs (1 state) and LOWs (0 state) will vary with the type of circuitry used, typically from +3 to +12 vdc for HIGHs and ground to -6 vdc for LOWs. Assemblies of these primary boxes or blocks can be made so as to manipulate voltages to perform arithmetic or the editing functions of the computer program.

The recommended IRE symbols for AND-OR logic gates are shown on line (a) of Fig. 4-28; the various alternative symbols shown in lines (b) through (e) are still common in some systems. The logical truth table for the AND-OR logic is given in Table 4-9.

By inserting an inverter inside a logic element such as an AND gate or an OR gate, is is possible to obtain NOT AND and NOT OR functions, which are referred to respectively as NAND gates and NOR gates. The logical block symbols for these basic elements are shown in Fig. 4-29. The corresponding logical truth table is given in Table 4-10. For the NAND gate, the output F is LOW only when both inputs A and B are HIGH. For the NOR gate, the output F is HIGH when A or B or both are LOW. A "bubble" on an input or an output line indicates a LOW condition; the absence of a

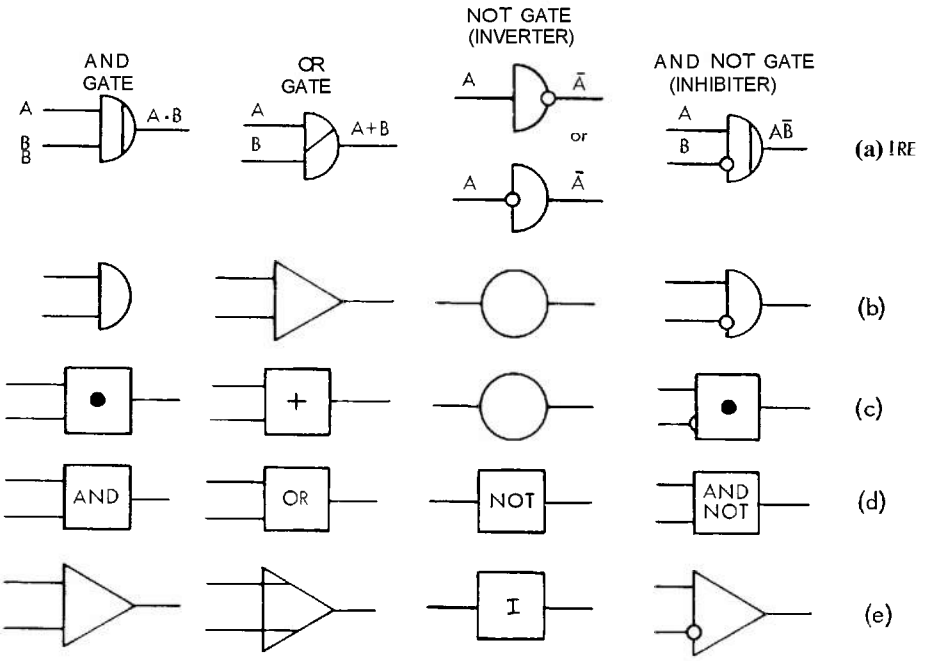
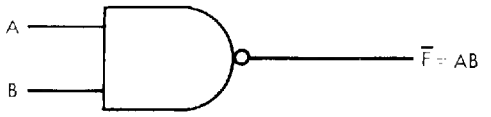
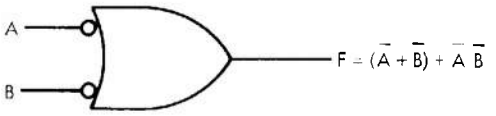


Figure 4-28. Logical symbols for inverters, inhibiters, and two-input AND-OR gates.



F =OUTPUT OF THE NAND GATE AND IS LOW (INDICATED BY THE BAR) FOR THE NAND CONDITION

(A) NAND Gate



F = OUTPUT OF THE NOR GATE AND IS HIGH (INDICATED BY THE ABSENCE OF A BAR) FOR THE NOR CONDITION

(E) NOR Gate

Figure 4-29. Logical symbols for two-input NAND-NOR gates.

TABLE 4-9. LOGICAL TRUTH TABLES.

Input A	Input B	Output for		
		And	Or	And Not
0	0	0	0	0
0	1	0	1	0
1	0	0	1	1
1	1	1	1	0

TABLE 4-10. TRUTH TABLE FOR NAND-NOR LOGIC.

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

"bubble" indicates a HIGH condition. Normally, for diode-transistor micrologic, a HIGH has a voltage level of +5 vdc and a LOW is at ground level.

A complete description of Boolean algebra and logic symbols, together with their use and application, is given in references 2 and 79.

4-11.2 FLIP-FLOPS

The flip-flop, or bistable multivibrator, is a basic single-bit storage element, and is characterized by two stable states, one of which can represent a 1 and the other a 0. Flip-flops are normally provided with two inputs as shown in Fig. 4-30. An impressed pulse or level on one of these inputs will produce the 1 state, while a similar pulse or level impressed on the others will produce the 0 state. These inputs are often designated "Set" and "Reset". A third "Complement" input may be provided. A pulse impressed on the complement input transfers the flip-flop to the state opposite to its original state. Two outputs are provided from the flip-flop, denoted "1" and "0". These outputs can usually control a number of other circuits. (The number of circuits controlled is known as the "fan-out" ratio.) A level change appearing at the "1" output denotes that the flip-flop is set, while a level change appearing at the zero output indicates that it has been reset.

Complementing flip-flops may be interconnected with AND gates to form a binary counter. A typical example is shown in Fig. 4-31. With flip-flops, gates and delay lines, a number of shift registers and other types of digital circuits can be constructed.

4-11.3 ADDERS AND SUBTRACTORS

The basic principles of binary addition are illustrated in Information Summary 4-7. Implementation can be accomplished in a number of ways, depending on the type and amount of logic used. Typical schemes are described in the paragraphs which follow.

4-11.3.1 Half-adder

The half-adder adds two binary digits and produces a sum and a carry output. The name "half-adder" derives from the fact that it does not provide for a carry from the previous set of digits added, i.e., it does only half the job needed for binary addition. Three forms of half-adder logic are shown in Fig. 4-32 and the form of binary addition is given in Table 4-11.

4-11.3.2 Full-adder

A full-adder receives as its input the augend bit, the addend bit, and the carry bit produced by the addition of the preceding bits. A combination of two half-adders and a mixer (or) as shown in Fig. 4-33 produces "full addition".

4-11.3.3 Accumulator

An accumulator is a device for adding multiple-digit numbers. *(See* Information Summary 4-7 for the governing rules and an

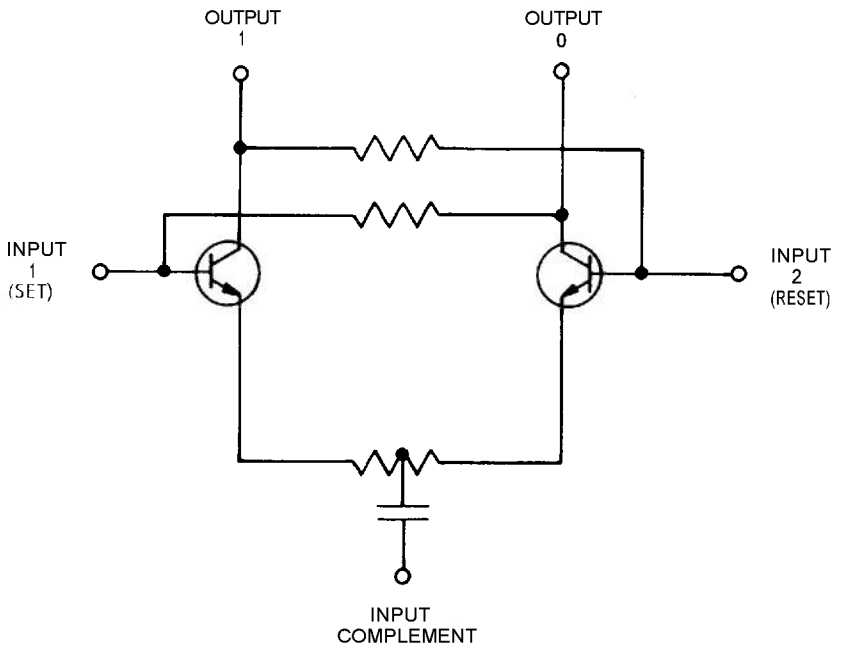


Figure 4-30. The basic flip-flop.

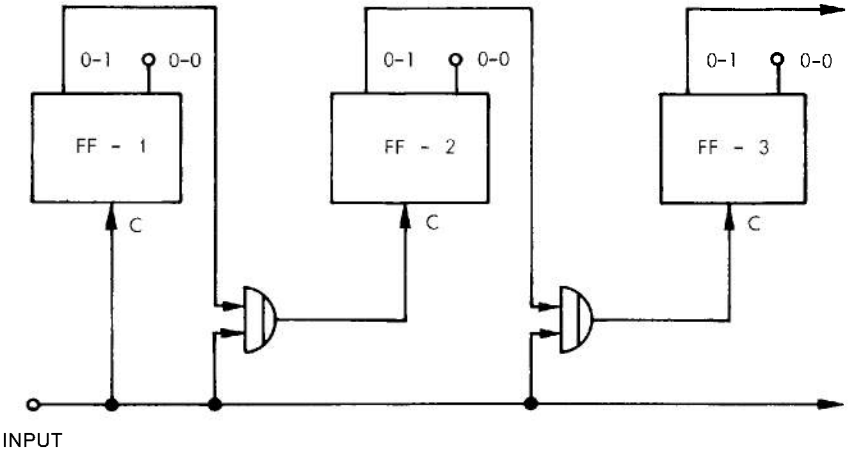


Figure 4-31. A typical binary counter.

example of their application.) The mechanics of the accumulator depend on the coding used in the computer and whether serial (digit-at-a-time) or parallel (all-at-once) operation is used. A complete accumulator consists of a register for the augend, a register for the addend, an adder to produce the sum, a register to hold the sum (frequently the initial register used for the augend), and control logic to guide the operation. A serial-character natural binary accumulator is shown in Figure 4-34.

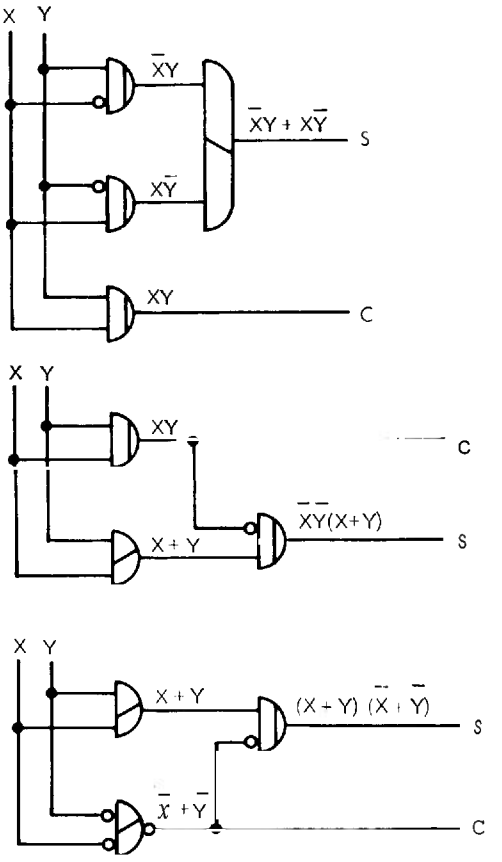


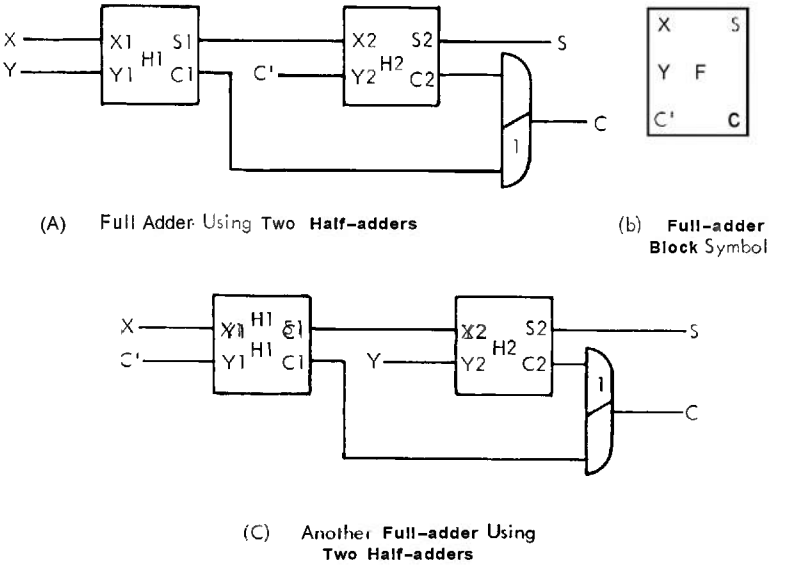
Figure 4-32. Forms of half-adder logic.

TABLE 4-11. BINARY ADDITION.

Augend (X)	0	0	1	1	
Addend (Y)	0	1	0	1	
Sum (S)	0	1	1	1	0

X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Note that the sum is 1 when either X or Y is 1, but not both. There is a carry C only when both X and Y are 1.



(A) Full Adder Using Two Half-adders

(b) Full-adder Block Symbol

(C) Another Full-adder Using Two Half-adders

Figure 4-33. Full-adder using two half-adders.

INFORMATION SUMMARY 4-7. THE RULES OF BINARY-ARITHMETIC ADDITION AND AN EXAMPLE OF THEIR APPLICATION

The rules:

```
    0 + 0 = 0
    1 + 0 = 1
    0 + 1 = 1
    1 + 1 = Two
          = 0 + carry
          = 10
1 + 1 + 1 = Three
          = 1 + carry
          = 11
```

Example:

The addition of decimal numbers 491 and 118	=	carry	1
			491
			118
			609

The binary equivalents of the decimal numbers involved are as follows:

```
491 = 111101011
118 =   1110110
```

Therefore,

The corresponding binary addition	=	carries	1111111
			111101011
			1110110
			1001100001 answer

which equals:

2^9	or	512
$+2^6$	or	64
$+2^5$	or	32
$+2^0$	or	1
		609 as before.

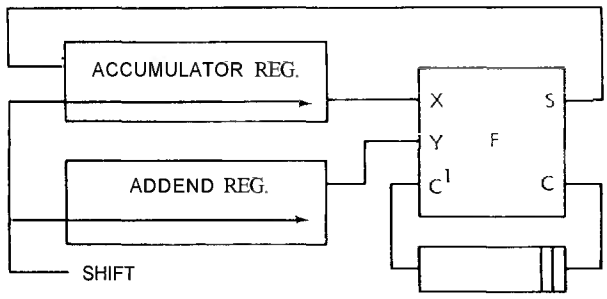


Figure 4-34. Serial binary accumulator.

4-11.3.4 Serial and Parallel Adders

In serial operation, the binary digits of the two numbers to be added together are applied serially in time to the two input lines of the adder, and it is usually necessary that the two input numbers be applied "in phase", i.e., with corresponding digits of the two numbers appearing on the two input lines simultaneously. Serial operation is almost always conducted with the digits appearing in ascending order of significance to maintain the most simple and straightforward mechanism. The speed of addition in a serial computer is usually set by factors that have little relationship to the adder; rather, it is the clock rate in turn a function of the type of storage—that establishes the speed of serial addition.

In parallel operation, the "in phase" requirement of presenting the digits to be added is met almost automatically. The increase in speed through parallel operation is not necessarily n times serial operation, if n orders are involved, nor will a practical parallel machine require n times the equipment. Parallel operation does not necessarily imply that addition of all orders of two numbers is accomplished simultaneously. Memories for both systems would be closely comparable in size, and the total increase in speed by parallel operation would be limited by the dead time required to send numbers to and from memory and by other housekeeping functions.

4-11.3.5 Simultaneous Carry Techniques

In the methods that have been mentioned for handling the carries, either with adders or with accumulators, the carry was "propagated" from one order to the next. It is worthwhile in some applications to employ simultaneous carry with the orders in groups of three or four to increase carry propagation speed. The amount of equipment required for simultaneous carry in all orders might be impractical. Standard discussions of carry techniques appear in Ref. 4.

4-11.3.6 Subtractors

A subtractor may be designed in a manner quite similar to that used for an adder, except that the concept of "borrow" replaces the "carry" concept. Logically, however, the occurrence of distressing cases where the subtrahend is larger than the minuend -- together with the fact that the rules for binary subtraction are significantly more complicated than those for addition -- leads to having the computer perform subtraction by addition, either directly or by using complements. An adder-subtractor performs the double function of creating a sum or difference and, in a separate channel, develops the carry or borrow. Subtraction accumulators count in reverse of addition accumulators.

In some, perhaps most, computers it has been found more convenient to perform subtraction through addition of the complement representation of numbers instead of through the use of a subtractor.

4-11.4 MULTIPLIERS AND DIVIDERS [2,3,4]

Multiplication is in some ways the most important operation to be mechanized, because of its complexity and because it must often be carried out with such speed that it greatly influences the design of the entire arithmetic unit. The multiplying operation (see Information Summary 4-8) usually involves facilities for the simultaneous storage, addition, and shifting of several numbers, and these facilities are also employed

INFORMATION SUMMARY 4-8. THE RULES OF BINARY-ARITHMETIC MULTIPLICATION AND AN EXAMPLE OF THEIR APPLICATION

The rules:

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Example:

$$\left.\begin{array}{l}\text{The multiplication}\\ \text{of decimalnumbers}\\ \text{24 and 3}\end{array}\right\} = \begin{array}{r} 24 \\ \times\ 3 \end{array} = \begin{array}{r} 24 \\ +24 \\ +24 \end{array} = 72$$

The binary equivalents of the decimal numbers involved are as follows:

$$24 = 11000$$
$$3 = 11$$

Therefore,

$$\left.\begin{array}{l}\text{the corresponding}\\ \text{binary.}\\ \text{multiplication}\end{array}\right\} = \begin{array}{r} 11000 \\ \times\ 11 \\ \hline 11000 \\ +11000 \\ \hline \end{array}$$

$$= 1001000 \text{ answer}$$

which equals:

$$\begin{array}{ll} 2^6 & \text{or } 64 \\ +2^3 & \text{or } 8 \\ \hline & 72 \text{ as before.} \end{array}$$

in the mechanization of the other arithmetic operations. The simplest serial, binary multiplier might consist of three storage registers: two of normal length for the multiplier and the multiplicand, and the third of double length to store the finalproduct. One of the principal problems in designing a multiplier is that of controlling and sequencing the various additions, multiplications, and shifts necessary to obtain the product.

The desired speed for multiplication should be based on the speed of addition or subtraction and upon the expected frequency with which multiplication is to be encountered. As pointed out by Phister[2], if it is expected in the average problem that there will be ten additions and subtractions for every multiplication, a reduction in multiplication time from ten add times to one add time may increase the complexity of

the multiplication logic by a factor of ten, but cannot even reduce total computing time by a factor of two.

Division may be mechanized in at least three distinct ways: by the common and familiar trial-and error method, by using a nonrestoring algorithm, *or* by making use of an iterative procedure.

Although multiplication and division are somewhat complicated to carry out even with pencil and paper, multiplication and division by ten in the decimal number system are very easy; it is simply a matter of moving the decimal point. Similarly, in the binary number system, a movement of the binary point to the right or to the left corresponds *to* a multiplication or division by some power of two. In a parallel computer the mechanism for this is, of course, very simple. In a serial computer, the shifting of a number in a register is somewhat more complicated. Shift registers are discussed in considerable detail by Phister[2], Flores[3] and Richards[4].

4-11.5 MATRIX MEMORIES

A major problem in the design of memory devices is the means of selection or access to a particular storage element in order to "read" the state of the element or to "write" into the element. This selection canbe achieved entirely by means of matrix switches using logical elements such as diodes, or part of the selection can be built into the memorv. As the capacity of the memory increases, the complexity of the selection equipment increases, a primary cause of the development of a variety of memory systems. Bistable magnetic cores are capable of performing most of the logical operations of digital computers (aswell as most of the storage). Core circuits have the advantages of reliability, long life, compactness, and light weight; some of the newer cores are inexpensive and require relatively low power consumption.

4-11.6 COUNTERS

Counters are most often used in computers as indices or timers. In addition to the basic clock pulses that establish synchronism between parts of the computer, it is necessary to have other timing signals for use in organizing, sequencing, or identifying data or operations. These timing signals may be obtained as output:; of a counter that changes state every clock pulse or every time some event takes place that is to be identified by the timing circuits.

Modern high-speed computers mainly use binary counters, consisting of a set of bistable storage elements each of which transfers back and forth between its two stable states upon the reception of pulses. With the decimal number system, elements having ten stable states are used; and each time a given element changes froni the state representing 9 to the state representing 0, a pulse is sent to the next element.

Counters in both binary and decimal systems have been adapted to counting only forward, only backward, either forward or backward, or to count up or down from a preset value.

4-11.7 ARITEIMETIC LJNITS

For combinations of logical elements into complete arithmetic units, the reader is directed to Refs. 2, 3, 4, 13, and 34, plus pertinent literature more recently listed in professional society and trade bibliographies.

4-12 CIRCUIT COMPONENTS

4-12.1 VACUUM TUBES

In the period from 1919, when Eccles and Jordan invented the basic circuit used as a flip-flop or trigger or multivibrator, up through about 1945, when Eckert and Mauchley built ENIAC (the first electronic digital computer; ref. the Introduction to this handbook) under the sponsorship of the Ballistic Kesearch Laboratories of the Ordnance Corps, U.S. Army, vacuum-tube digital circuits were increasingly being perfected. ENIAC used 19,000 tubes and consumed nearly 200 kilowatts of power. The solid-state circuits now almost universally used are closely analogous to the tube circuits of older computers. The generally greater reliability, longer life, smaller size, and lower power requirements of solid-state circuitry have virtually eliminated vacuum-tube use in computers.

4-12.2 SEMICONDUCTORS

The field of transistor and diode logic-circuit design evolved with marked technological and economic changes. Refs. 13 and 14 constitute a good foundation in the field, and Refs. 46 through 59 deal with transistor computer circuitry. The availability, performance, and cost of actual semiconductor devices the designer might consider for the solution of a particular problem are factors that can require very close liaison between engineering and procurement groups.

Ref. 60 describes a basic circuit using one transistor, one capacitor, and three resistors from which a complete digital computing system may be economically constructed.

More recently, integrated circuits have been used in computer design and provide advantages not achievable with conventional transistors and diodes, and relatively bulky passive elements such as resistors and capacitors. Extremely low power consumption, low supply voltage, reduced size, and reduced cost make integrated circuits extremely feasible for certain applications (see Refs. 91 through 96).

The manner in which basic logical elements are combined to form circuits, such as identity comparators, is described in Information Summary 4-9. Logical design using basic AND-OR circuits is given first, followed by logical design utilizing NAND-NOR integrated circuits. Truth tables and Karnaugh maps are provided to illustrate the method of optimizing circuit design. For convenience, a Fairchild 930diode-transistor micrologic (DTμL) gate and inverter was selected as the basic logic element for the comparator design. Other types of micrologic, however, could have been used as well.

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS

Identity comparators are used to compare the contents of two or more stages or registers and to provide an output only when all of the corresponding inputs are equal. For example, in the case of the two-stage comparator shown in Fig. IS 4-9.1 an output F will be obtained only when the contents of the A Register are equal to the contents of the B Register. No output F will be obtained if any one of the stages A_i is not equal to its corresponding stage B_i. These conditions are illustrated by the truth table and Karnaugh map in Fig. IS 4-9.1 for each stage, where a 1 is a logical HIGH and a 0 is a logical LOW. For an n-stage identity comparator, the Boolean equation can thus be written as

$$F = (A_1B_1 + \bar{A}_1\bar{B}_1)\ (A_2B_2 + \bar{A}_2\bar{B}_2).....(A_nB_n + \bar{A}_n\bar{B}_n) \qquad \text{(IS 4-9.1)}$$

which states that the output F will be HIGH only when A_1 and B_1 are both HIGH or both LOW, and concurrently A_2 and B_2 are both HIGH or both LOW, and so forth.

DESIGNING WITH AND-OR LOGIC

With the use of inverters and passive AND and OR gates, Eq. IS 4-9.1 could be implemented as shown in Fig. IS 4-9.2. Symbolically, a dot represents an AND gate, a plus sign represents an OR gate, and an open arrowhead represents an inverter for obtaining the complement of the input variable. Other symbols could be used to represent these gates and inverters, depending on the particular drafting standards employed.

DESIGNING WITH NAND-NOR LOGIC

When designing with diode-transistor micrologic, such as Fairchild flatpack integrated circuits, the inverting action of the circuit itself must be taken into consideration. This, in effect, influences the manner in which the Boolean equation is implemented. To illustrate this point, consider a typical DTpL 930 gate, similar to the one shown in Fig. IS 4-9.3. This gate can be represented schematically as a 4-input diode network and a pair of NPN transistors con-

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS (Cont)

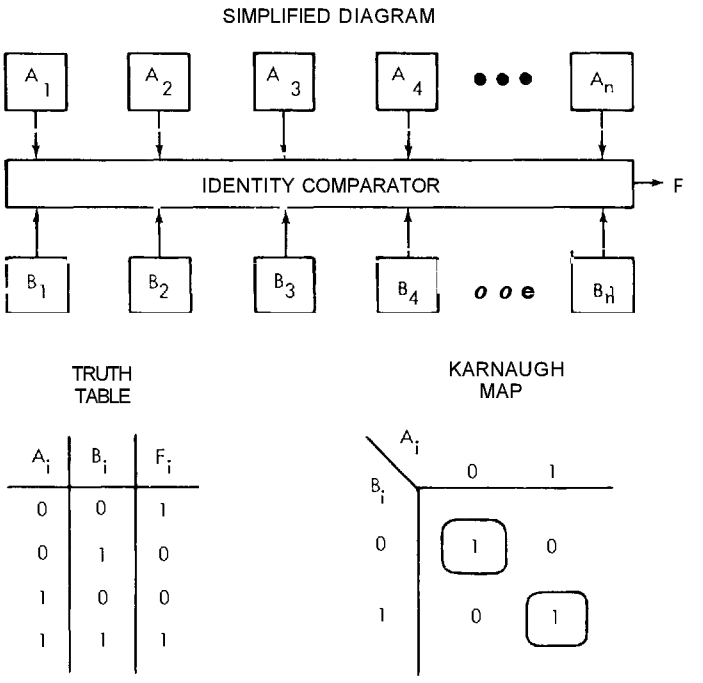


Figure IS 4-9.1. Simplified diagram, truth table, and Karnaugh map for an n-stage identity comparator.

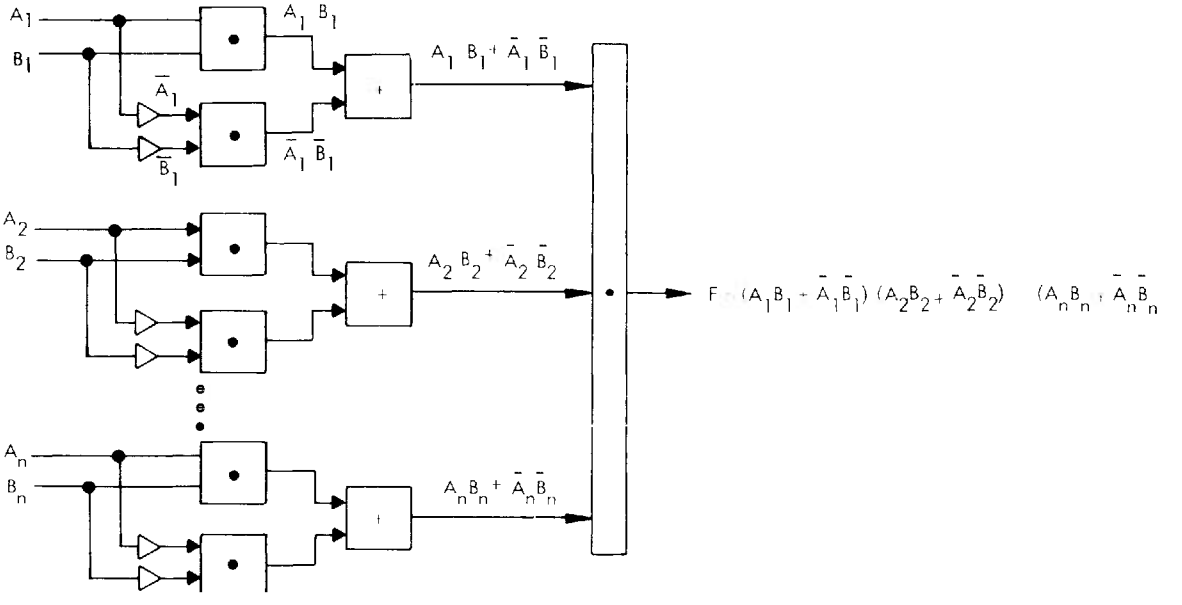


Figure IS 4-9.2. An n-stage identity comparator based on the use of AND-OR logic.

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS (Cont)

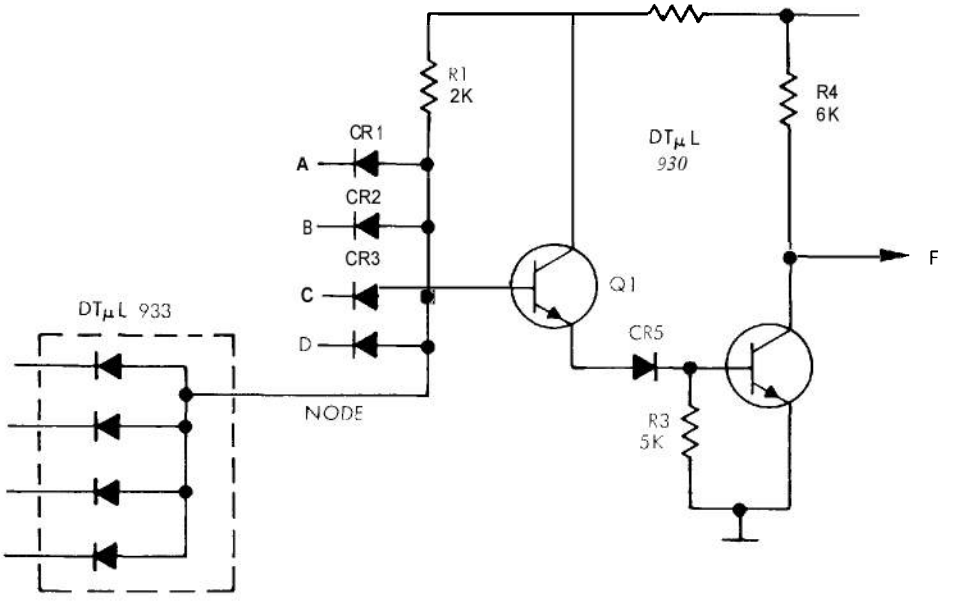


Figure IS 4-9.3. Schematic representation of a single DTpL 930 gate with a DTμL 933 extender.

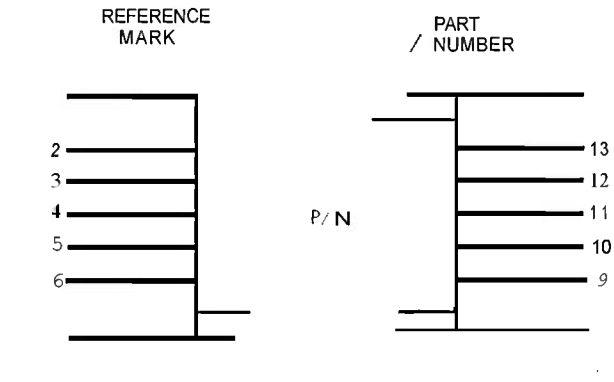
nected in cascade. Two such gates are contained in a single chip. The input network is expandable to 8 diodes by the addition of a diode cluster, such as a DTμL 933 *(see* Fig. IS 4-9.3).

The circuit operation tor the gate itself is as follows. When inputs A, B, C, and D are all IIIGH (+5 vdc), diodes CR1 through CR4 are back-biased so that a IIIGH appears on the base and collector of Q1. Transistor Q1 will then conduct to ground from $+V_{cc}$ through R2, Q1, CR5, and R3. The voltage drop across R3 will cause Q2 to conduct and saturate, so that the output F is LOW (ground potential). If any input A, B, C, or D goes LOW, its associated diode will be forward-biased, thereby placing the base of Q1 at ground. This condition causes Q1 to cut off, thereby cutting oft Q2, so that the output F goes IIIGH ($+V_{CC}$). In effect, the output is LOW when all inputs are HIGH, and the output is IIIGH when one or more inputs are LOW.

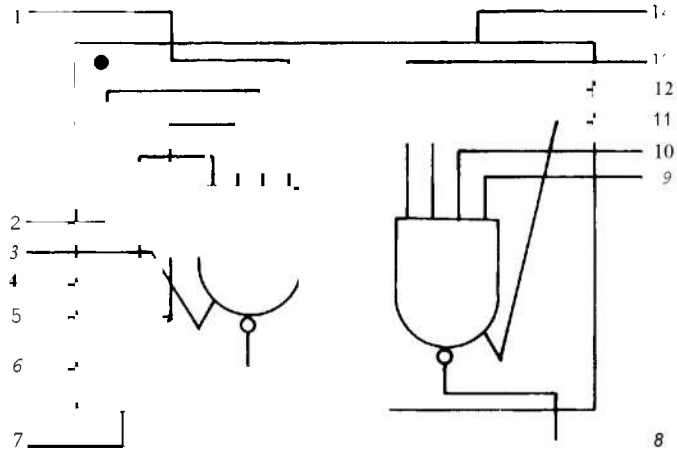
The input loading (fan-in) for this particular gate is one unit load for each input. The output loading (fan-out) is eight unit loads. One unit load is defined as approximately 1.3 milliamperes. For large variations in temperature (-55°C to +125°C), the fan-out should be reduced to approximately six unit loads. At 25°C, the noise immunity of the gate is approximately +1 vdc. As the temperature increases, the noise immunity decreases.

The physical structure of the Fairchild 930 monolithic chip is shown in Fig. IS 4-9.4. Internal diodes, resistors, and transistors are all constructed of semiconductor material, interconnected and tied through "traces" to spring terminals. These terminals are

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS (Cont)



(A) Top View of the Chip



(B) Logic Diagram of the Dual Gate (normally shown as a NAND gate)

Figure IS 4-9.4. Physical structure and actual pin connections of the DTpL 930 monolithic chip.

usually tinned anti then impulse-soldered to printed-circuit boards. Removal of a chip is accomplished with the aid of a razor-type penknife. As a rule, pin 7 is connected to ground and pin 14 is connected to a +5 vdc source (V_{cc}). If an input line is unconnected, it will act the same way as if a +5 vdc level were present (logical HIGH).

In terms of logic, the DTpL 930 can be used either as a positive NAND gate or a negative NOR gate, depending solely on the way in which the input and output levels are interpreted. When the DTpL is used as a NAND gate, the circuit provides a LOW output only when all the inputs are HIGH. Otherwise, the output is IIIGH. These conditions can be

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS (Cont)

expressed by the Boolean equations

$$\overline{F} = ABCD \qquad \text{(IS 4-9.2}$$

and

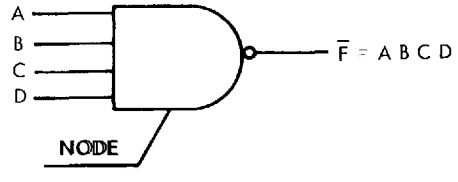
$$F = (\overline{A} + \overline{B} + \overline{C} + \overline{D}) + \overline{A}\overline{B}\overline{C}\overline{D}$$
$$= \overline{A} + \overline{B} + \overline{C} + \overline{D}$$
$$= ABCD \qquad \text{(IS 4-9.3)}$$

The logic symbol for a 4-input NAND gate is in Fig. IS 4-9.5 (A). A "bubble" on the output line indicates a logical LOW; the absence of a "bubble" on an input line indicates a logical HIGH.

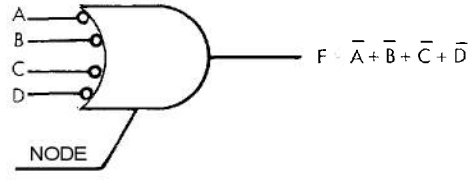
When the DT μL is used as a NOR gate, the circuit provides a HIGH output whenever one or more inputs are LOW. If all inputs are HIGH, then and only then will the output be LOW. These conditions can be represented by the same Boolean Eqs. IS 4-9.3 and IS 4-9.2 above, respectively. The logic symbol for a 4-input NOR gate is shown in Fig. IS 4-9.5(B). In this case, the "bubble" on the input line indicates a logical LOW, and absence of a "bubble" on the output line indicates a logical HIGH.

The use of a DTpL 930 gate as an inverter is illustrated in Fig. IS 4-9.5(C). The output is LOW when the input is HIGH, and vice versa. One input line is normally utilized for this function.

Identity comparators with diode-transistor micrologic can now be designed, as shown in Figs. IS 4-9.6 and IS 4-9.7, by using the basic Boolean Eq. IS 4-9.1. For convenience, only 2-bit and ti-bit identity comparators are illustrated, but the design can be applied to anynumber of bits or stages, provided loading requirements are not exceeded. Implementation that is achieved primarily with a single type of gate, such as the Fairchild 930, simplifies design and maintenance objectives, and usually affords spare input lines. The ultimate selection of the actual chips, however, generally depends on the designer and other factors.



(A) NAND Gate

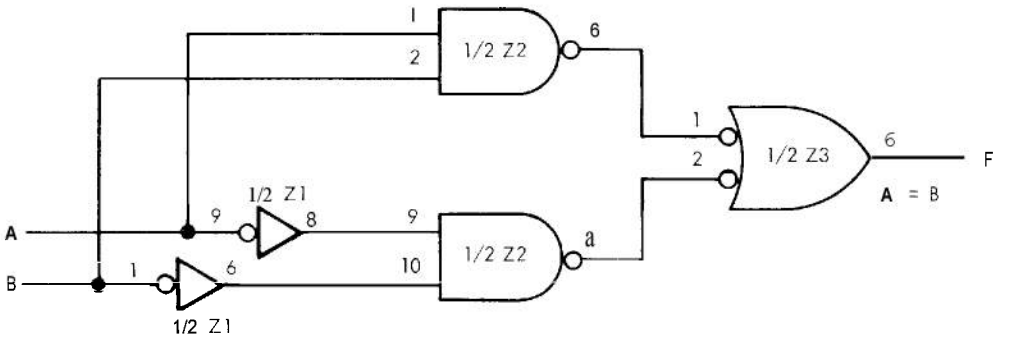


(B) NOR Gate

(C) Inverters

Figure IS 4-9.5. Logic symbols for the DTpL 930 gate.

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS (Cont)



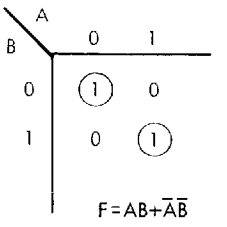
Note:

This identity comparator provides a logical HIGH output when the two input bits are both HIGH or both LOW. For example, when A and B are both HIGH, 22-6 will be LOW and 23-6 will be HIGH. When A and B are both LOW, 22-8 will be LOW and 23-6 will again be HIGH.

(A) Logic Diagram

INPUTS		OUTPUT
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

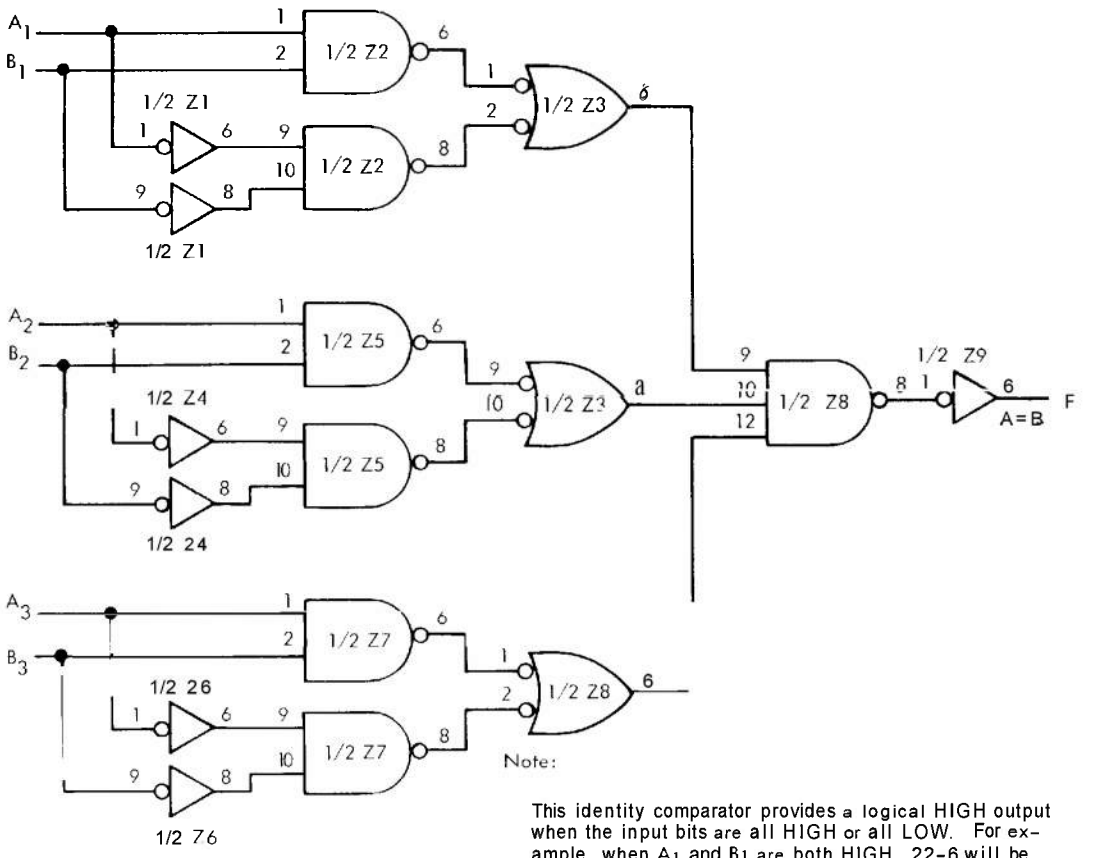
(B) Truth Table



(C) Karnaugh Map

Figure IS 4-9.6. A two-bit identity comparator (A=B).

INFORMATION SUMMARY 4-9. THE LOGICAL DESIGN OF IDENTITY COMPARATORS (Cont)



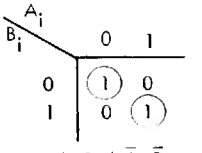
Note:

This identity comparator provides a logical HIGH output when the input bits are all HIGH or all LOW. For example, when A_1 and B_1 are both HIGH, 22-6 will be LOW. With a LOW input on Z3-1, Z3-6 will be HIGH. To obtain a HIGH output at Z9-6, all the inputs to 28-9, -10, and -12 must be HIGH. If any input to 28-9, -10, and -12 is LOW, then Z9-6 will be LOW — stating that an A bit is not equal to a B bit.

(A) Logic Diagram

INPUTS A_i	INPUTS B_i	OUTPUT F_i
0	0	1
0	1	0
1	0	0
1	1	1

(B) Truth Table (for each stage)



$$F_i = A_i B_i + \bar{A}_i \bar{B}_i$$

$$F = (A_1 B_1 + \bar{A}_1 \bar{B}_1)(A_2 B_2 + \bar{A}_2 \bar{B}_2)(A_3 B_3 + \bar{A}_3 \bar{B}_3)$$

(C) Karnaugh Map

Figure IS 4-9.7. A six-bit identity comparator (A=B).

4-12.3 MAGNETIC DEVICES

Refs. 62, 63, and 102 cover some of the first work done in the use of magnetic-core elements for logical switching. Ref.64 is an excellent bibliography of literature pertaining to magnetic circuits and materials, covering magnetic cores and films, ferrites, magnetic metals, multi-aperture magnetic devices, twistors, the ferristor and the parametron. A survey of magnetic devices is given in Ref. 17. In addition to the simple toroidal magnetic core, there are magnetic devices having a more complicated geometry. A two-aperture device, or transfluxor is described in Kef. 70, three-aperture core logic is described in Kef. 72, and an 8-rung "laddie" is described in Ref. 71.

4-12.4 NEW DEVELOPMENTS

An interesting type of large-capacity storage can be constructed using ferroelectric materials such as barium titanate. These materials can be polarized by a sufficiently large potential difference and, since the direction of polarization can be reversed by reversing the direction of the voltage, binary information can be stored. The direction of polarization can be sensed by applying a voltage pulse of specified polarity because a relatively large current is required to change the polarization while little current is required if the polarization is not changed.

The memory cells are constructed by evaporating rectangular electrodes on a plate that is a single crystal of fercoelectric material. A sixteen-cell unit is shown in Fig. 4-35. For writing, a cell is selected by the coincidence of two voltages E, each of which is half of that necessary to change the polarization of the dielectric. When reading, the full switching voltage 2E is applied to one electrode and the otlier is grounded through a load resistor. If the read signal causes a reversal of the polarization, a voltage pulse will appear across the load resistor indicatingthe initial state of the cell. An alternate reading system uses the output wires as the primaries of a transformer as shown in Fig. 4-35(F).

Ferroelectric memories are easy to manufacture since the electrodes are evaporated directly onto the dielectric and are quite small. The electrodes can be 10 mils wide and 10 mils apart. The ferroelectric

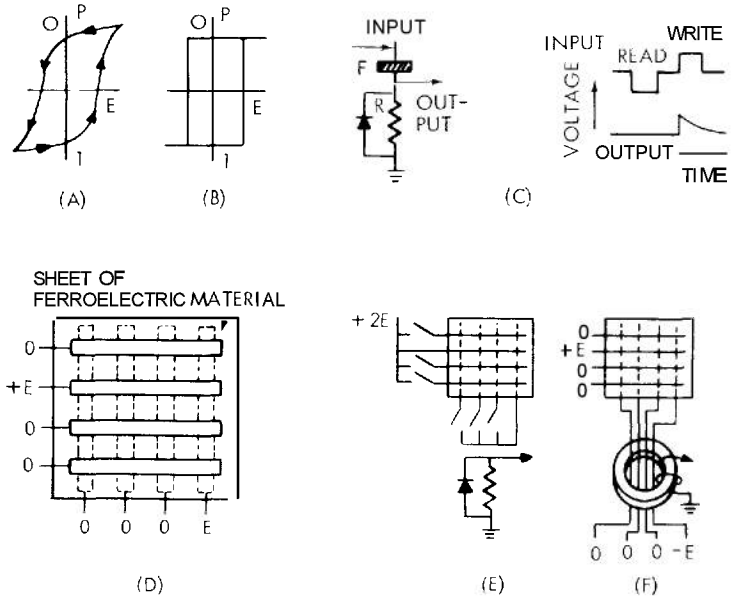


Figure 4-35. Ferroelectric storage.

can be 10 mils or less thick. This corresponds to a bit density of nearly 250,000 bits per cubic inch.

At present, ferroelectrics have several disadvantages. The rectangularity of the hysteresis loop is poor and the size and shape of the loop depends on the frequency of the applied voltage. Also, the ferroelectric properties of the material can be damaged by repeated reversals of the polarization and these properties are sensitive to temperature. Because of these difficulties, ferroelectrics have not been used in any computers; however, they are still the subject of intensive development effort.

The tunnel diode 75. is a solid-state device, with a negative-resistance region in its characteristic curve, that can be used in microwave digital circuits for amplification and gating. By utilizing this device, repetition rates (i.e., 1/unit time interval)of 1,000 to 3,000 megacycles, or more, may be realizable. Thus its potential advantage is very great speed in a comparatively reliable solid-state device. At present, there are disadvantages – associated with problems of reproducibility and uniformity of characteristics as well as with the many circuit-design problems introduced by the high speed – that must be overcome.

The tunnel diode consists of a junction between extremely heavily doped (doped to "degeneracy") n-type and p-type semiconductors. It depends for its operation on the quantum-mechanical phenomenon known as tunneling, from which the diode derives its name. Tunneling is an effect in which an electron can "tunnel" through a potential barrier, even though it does not have sufficient energy to "surmount" the barrier, provided the barrier is "thin" enough.

4-13 STORAGE

4-13.1 SEQUENTIAL-ACCESS STORAGE

The addresses of a sequentialstore are scanned continuously, with a particular address becoming available once each cycle. This type of storage generally provides large amounts of inexpensive memory, but has a relatively long access time.

4-13.1.1 Magnetic Sequential Storage

The most important types of sequential access storage involve magnetic recording on drums, discs, and tape. In order to keep the reading (or writing)in synchronism with other elements of the computer, the drums, discs, or tapes usually have one channel reserved for clocking or timing pulses. In some cases – e.g., where a drum constitutes the main memory of a computer – thetiming pulse channel is the basic clock pulse generator for the computer.

4-13.1.1.1 Magnetic Drums

A magnetic drum is a metal cylinder that revolves about its axis and has information recorded on its surface, as shown in Fig. 4-36. Information is written onto or read off of the drum through heads mounted close to the surface of the drum.

Extremes in drum capacity are a small-size 2 X 2 inch drum storing 20,000 binary digits, contrasted with a 4-foot drum storing 20,000,000 bits. A capacity of 100,000 bits is common, and may be obtainedin a drum 6 X 6 inches, with a storage density of 50 bits per inch around the circumferance and 20 bits per inch along the axis.

4-13.1.1.2 Magnetic Discs

Storage on magnetic discs produces a cross-breed between sequential storage (on each disc)with the random feature of selecting one disc from a continuously revolving stack in a manner of a juke box. Conventional disc storage ranges from4,000 words of 40 bits to units storing severalmegabits.

4-13.1.1.3 Magnetic Tape

A magnetic tape is a flexible plastic or metal strip from 0.001 to 0.010 inch thick, from 0.25 to 4 inches wide, and may be up to 2500 feet in length for a typical installation. Fig. 4-37 illustrates the method of feed, read-write, drive, and take-up for handling magnetic tape.

Tape storage has been used inthe form of endless loops, providingmany of the characteristics of a magnetic drum, as shown in Fig. 4-38.

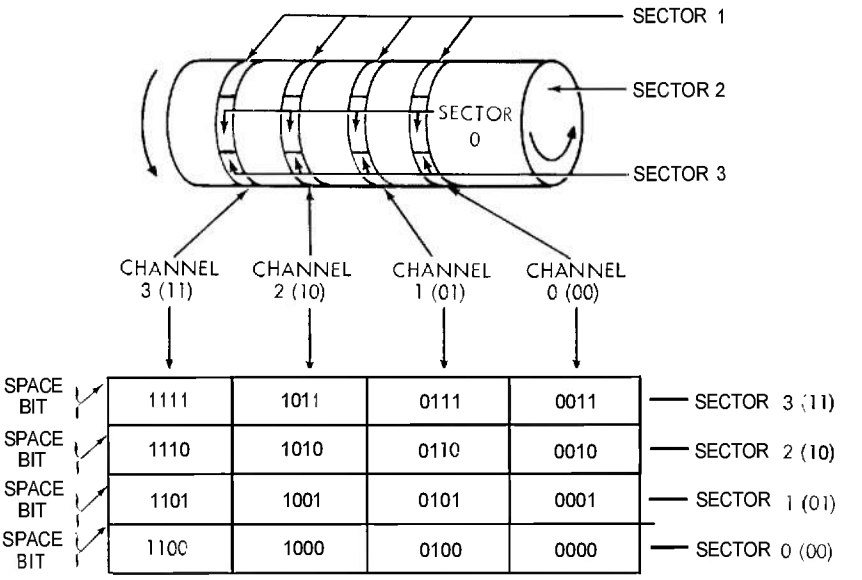


Figure 4-36. Arrangement of a hypothetical, sixteen-word serial memory on the surface of a drum.

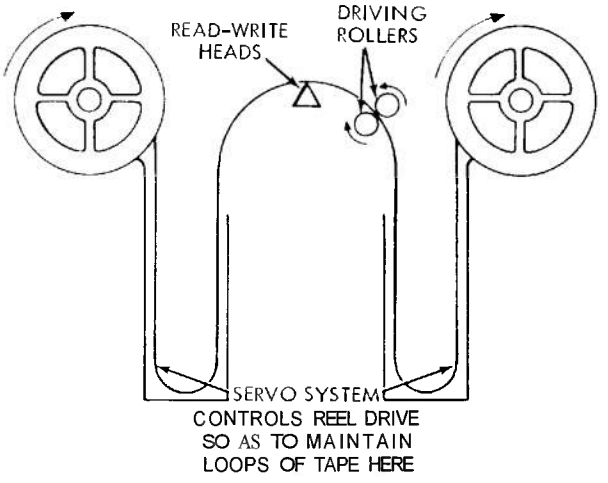


Figure 4-37. Typical reel system for magnetic tape.

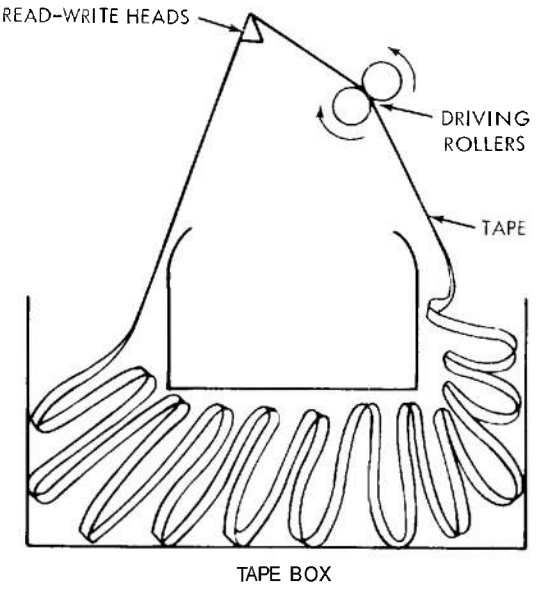


Figure 4-38. Method of providing an endless tape.

4-13.1.2 Delay-line Storage

Considerable use has been made in the past of acoustic (ultrasonic) delay lines to provide reliable storage of information repeated over and over, as shown in Fig. 4-39.

In the acoustic delay line, coded information is introduced serially into a medium, which may be mercury, by loud-speaker action. A second transducer picks up the pulses delayed by the transmission time through the medium. These received pulses can be reformed and sent again through the delay line. In this way, the information store is retained sometimes for days at a time without losing a pulse. A higher-speed variation of the delay line uses a quartz crystal. By having mechanical vibrations transmitted through a crystal polygon rebound from 15 or more sides, a reasonably long delay may be packaged in a small volume.

Lumped-constant delay lines, using a number of inductors and capacitors to create a transmission line with a low propagation velocity, represent a very special approach that involves resolution problems requiring large numbers of different small components. This type has inherent losses and requires additional amplifiers to create long delays.

Magnetostrictive delay lines find wide application as serial memories. Their advantages are light weight and low power consumption. Special packaging is required for extreme environmental conditions, however. These delay lines use the principles whereby some materials deform when a magnetic field is applied, and conversely distort a surrounding magnetic field when the material is strained. In this type of delay line, a magnetostrictive wire is held between two damping elements to prevent reflections and is magnetized by a transmitting coil at one end to deform the wire. The deformation travels down the wire, and the strain is transduced into an electrical pulse at the other end. Pulses are stored in the wire in the form of strain waves.

Fig. 4-40 shows a typical serial memory that is available as a standard off-the-shelf item. A simplified schematic diagram is also shown. Physically, a magnetostrictive delay line is coiled and secured inside a metallic case that is then attached by standoffs to a printed-circuit board containing the logic. The delay line illustrated has an operating frequency range of 1 mc and will store up to 2048 bits at that rate.*: The maximum power dissipation of the whole unit is 3.4 watts.

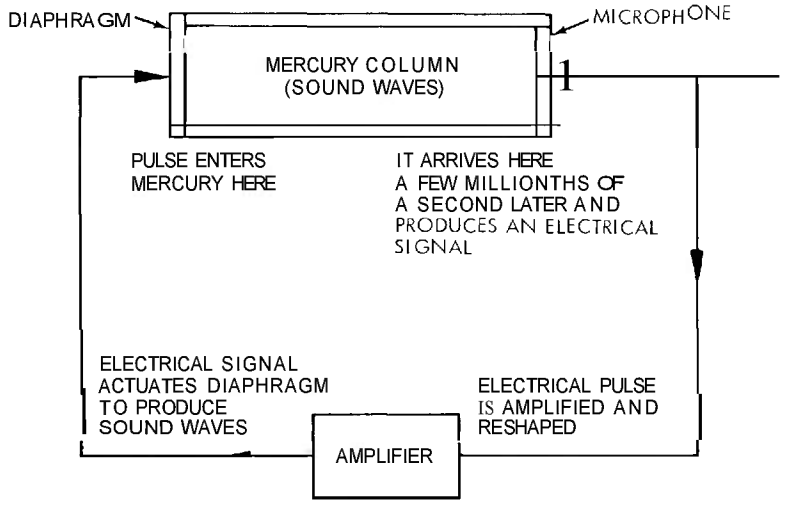
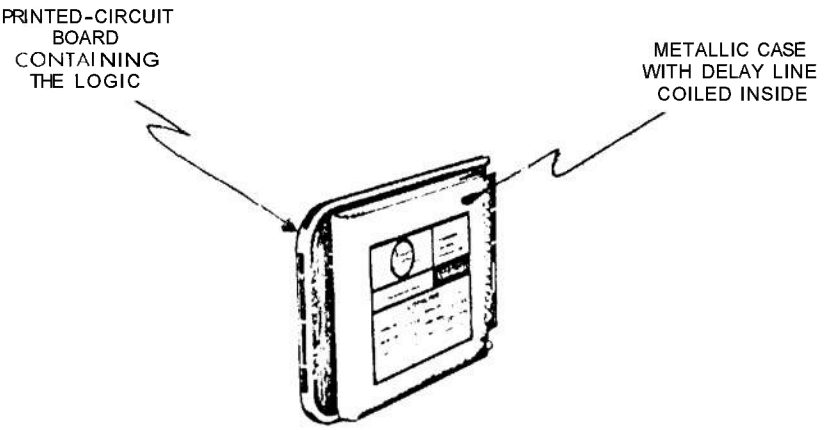


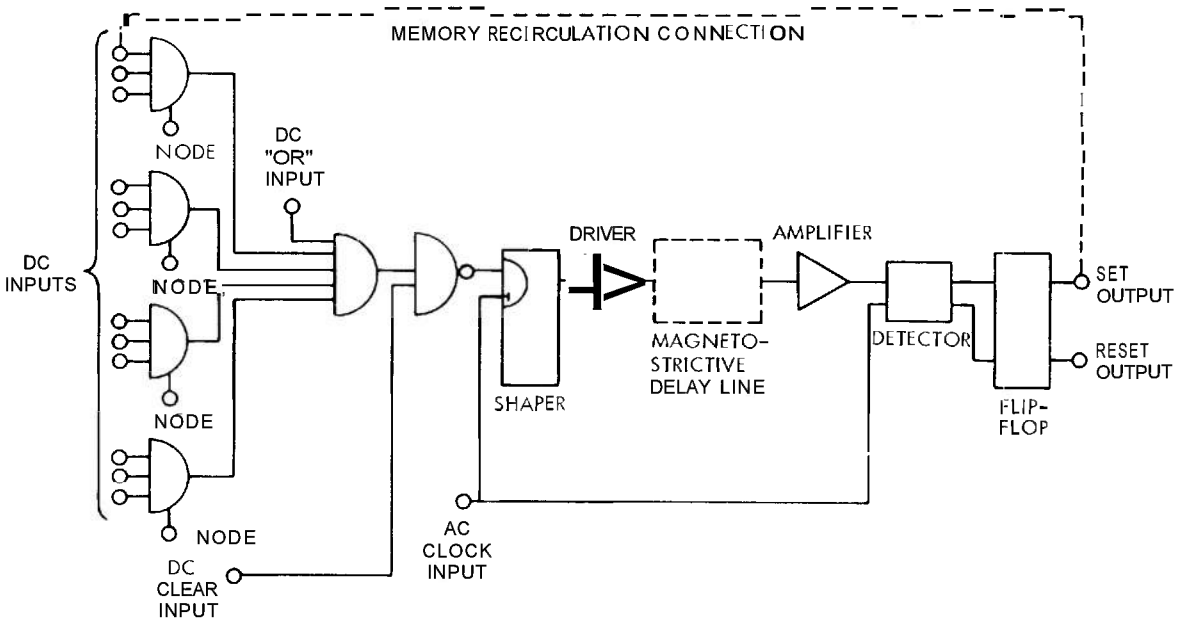
Figure 4-39. Acoustic delay line.

*At 1 mc, a 2048-microsecond delay will store 2048 bits of data, in accordance with the relationship

$$\text{Total pulse delay } (\mu\text{sec}) = \frac{\text{storage capacity (bits)}}{\text{clock frequency (mc)}}$$



(A) External view



(B) Simplified Schematic Diagram

Figure 4-40. A typical serial memory utilizing a magnetostrictive delay line.

Logically, the serial memory may be considered to be a shift register whose length is equal to the specified number of digits of delay and shifts at the clock rate. Continuous storage is achieved by feeding the serial-memory output back to the input and recirculating the stored data. Logical inputs are provided for entering information into the memory, removing information from the memory, modifying memory contents, forming circulating serial adders, etc.

An excellent presentation on magnetostrictive delay lines appears in Ref. 102. The application of these delay lines in airborne serial memories is covered in Ref. 93.

Serial memories with a "zero-temperature-coefficient" glass delay line have been built for operation with high-frequency logic elements. Their use has not been as widespread, however, as magnetostrictive delay lines -- probably because of cost and frequency restrictions, such as short lead length.

4-13.1.3 Punched Paper Tape and Cards

Punched paper tape and cards are useful for very-long-access-time storage, for instruction storage, and as an intermediate input-output medium. The nature of their use is summarized in Figs. 4-41, 4-42, and 4-43.

Both tape and cards provide a method of holding files of information outside the computer. Cards have the advantage of being easily rearranged, added to, or deleted from in order to vary programs and routines or to up-date a data file conveniently. Tape can be read at rates of a few hundred characters (2,800 bits) per second. At the upper limit of card handling – 1000 cards per minute – an equivalent bit rate of 16,000 bits per second is obtainable.

4-13.1.4 Photoelectric Storage

By using photographic storage of opaque and transluscent binary bits arranged not unlike the punched-hole arrays on cards, very high packing densities can be achieved – in the order of 50,000-100,000 bits per square inch. A number of physical configurations 'have been developed including reels of film

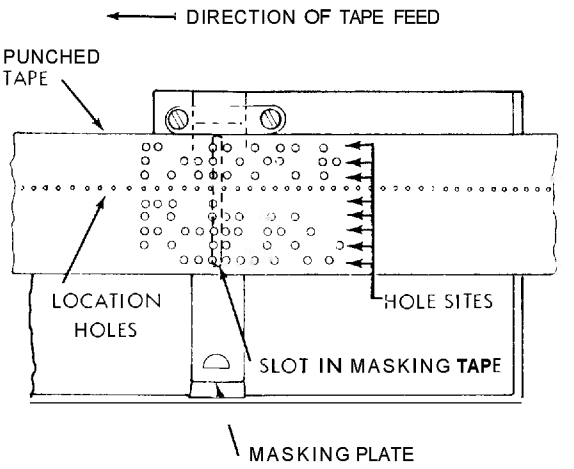


Figure 4-41. Reading of punched paper tape.

(usually 35 mm in width), large disks, plates a few inches square, and film strips or chips. A given photographic element is normally scanned sequentially, but it is conceivable that the photoelectric scan of information would take place at several million bits per second. Except for reels and disks, the photographic elements may be stored in a manner that permits mechanical selection – giving a combined random-sequential access to a very large memory bank.

4-13.2 RANDOM-ACCESS STORAGE

4-13.2.1 Magnetic Core and Other Coincident-current Devices

The very-common, very-high-speed magnetic core memory forms a good example of the class of random access storage devices. Fig. 4-44 shows a "plane" of cores, each typically between 0.1 and 0.4 inch in diameter. Each core is of square-loop magnetic material (as described in par. 4-12.4), and the coils indicated in the figure are commonly of only a single turn so that the complete matrix may be woven from wires threading horizontally and vertically through the cores.

A pair of the X- and Y-coordinates determines a set of cores, and a carefully controlled signal level is used to affect the Oor

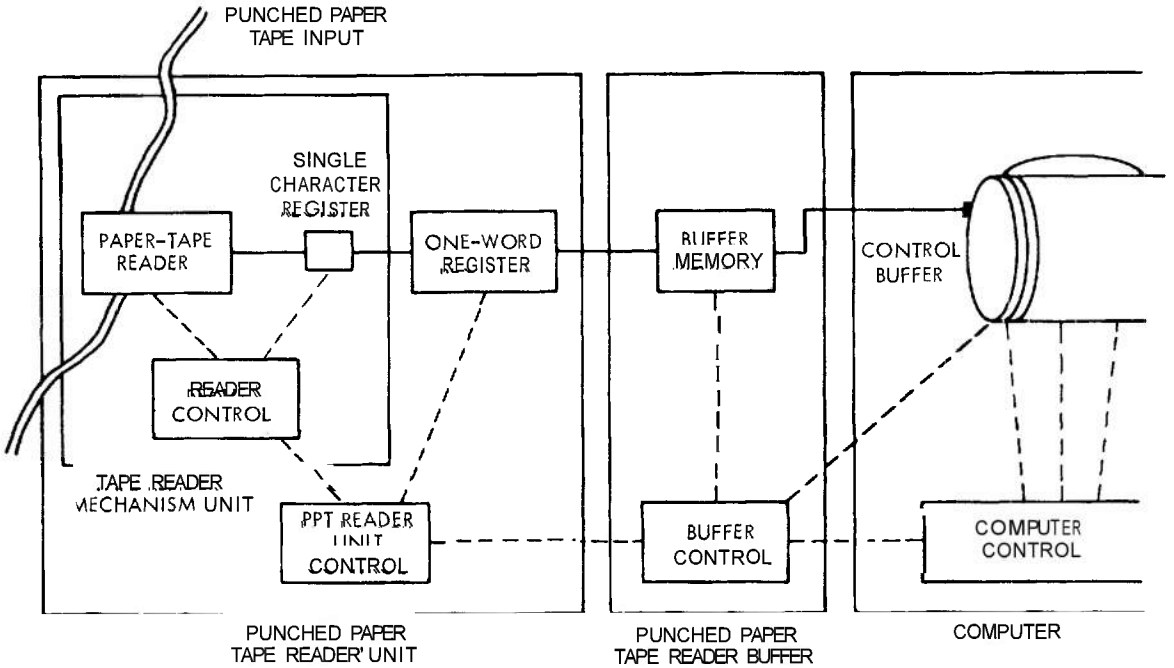
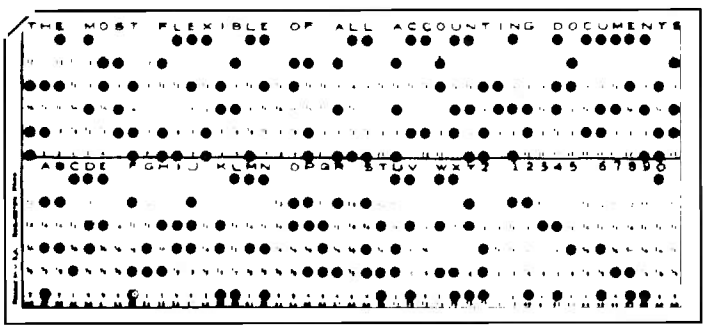
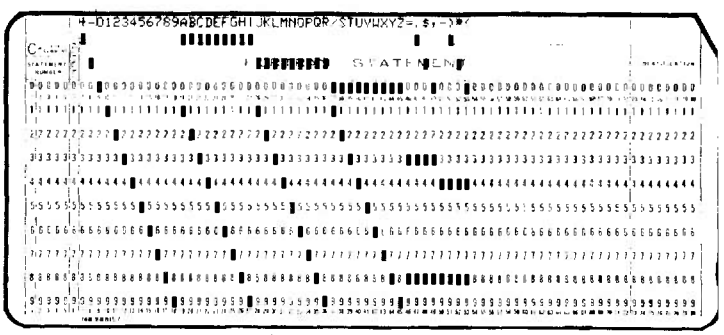


Figure 4-42. Typical arrangement for reading punched paper tape.



(A) 90 Column



(B) 80 Column

Figure 4-43. Punch cards.

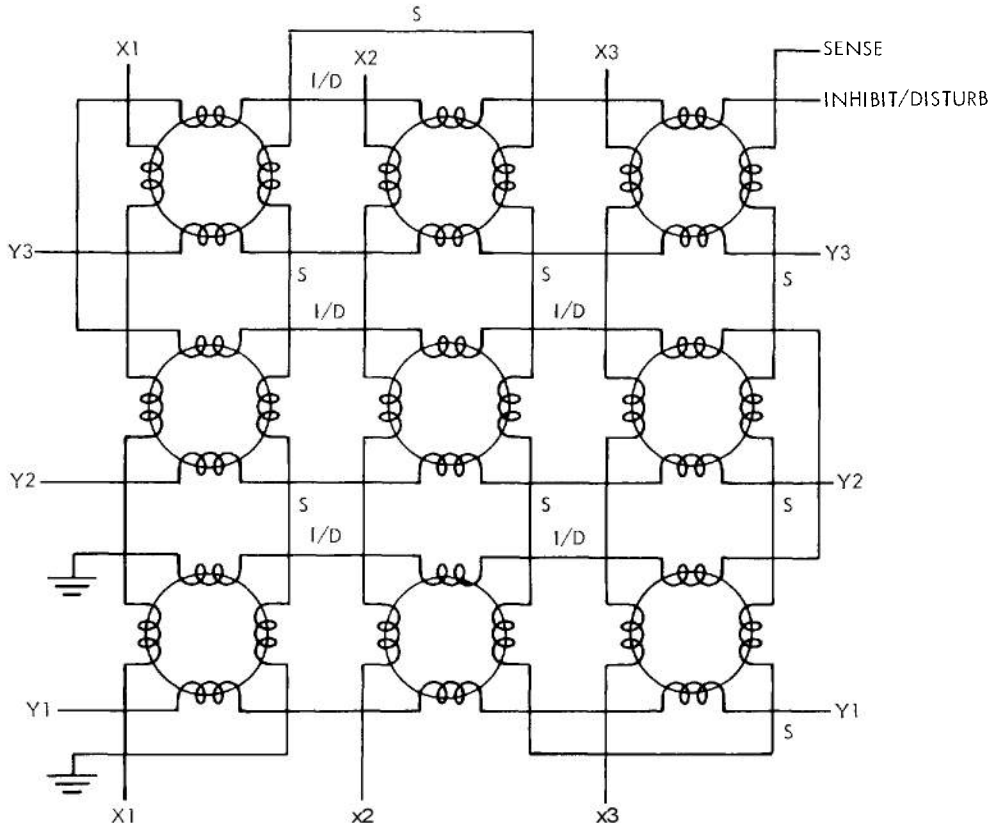


Figure 4-44. Corner of a core matrix.

1 level of one specific core where the coordinates intersect. The addition of many planes and a Z-axis creates a core memory stack, one module of which typically has a word capacity between 1,000 and 4,000. Solid slabs of ferrite material have been designed to act like an array of cores, by creating arrangements of holes and printing conductive windings on the essentially nonconductive ferrite material. A matrix of cores can also be created through techniques of depositing films of magnetic- material, insulation, and copper grids on glass. Such deposited circuits may have extremely high speeds.

A recent development in magnetic memory components is fabricated of pressed ferrite materials and takes the form of small rectangular solids, each nominally 0.085 X 0.050 X 0.050 inch. These blocks have two nonintersecting orthogonal holes: one for storage and one for interrogate. Magnetic domains in the common volume of material between the holes are shared, and a change in flux linkage around one hole will cause a change in the flux linking the other hole. Spacing between the holes is such that interference is reversible, producing a nondestructive memory element. The direction of flux around the storagehole determines if a 1 or a 0 was stored. Abipolar sense voltage induced by the change in stored flux is observed on the sense conductor in the storage hole. Very short access times are reported for these elements.

4-13.2.2 Diode-capacitor Storage

A bank of capacitors with associated diodes can store information of a thousand bits or so for periods of a few seconds. A capacitor is charged to store a 1, and is discharged for a 0.

4-13.2.3 Cathode-ray-tube Electrostatic-mosaic Storage

In earlier digital machines, devices such as the Williams tube, the barrier grid tube, and the selectron were developed for storing charges on plates that were scanned for reading and writing by a cathode-ray beam. These typically would store 1,024 bits with random access time as small as 10 seconds. They required particularly precise and noise-free control circuitry.

4-13.2.4 Photoelectric Storage

Random-access photoelectric storage has been developed in two forms. In one of these, a cathode-ray tube is employed as a light source. Data are recorded on photographic plates, each bit being recorded on a separate plate. To permit parallel readout, the light beam from the tube face is split by an optical system so as to fall on corresponding areas of all plates. The light passing through a plate is then focused in a photo-cell. This memory system has been employed in telephone central offices.

A memory employing a photocell matrix excited by a matrix of electroluminescent cells has been built experimentally. Electrical feedback from each photocell to its corresponding electroluminescent cell is used to hold information in the memory.

4-13.2.5 Ferroelectric Storage

Ferroelectric crystals, such as barium titanate, retain residual electrical polarization on application and release of a voltage. This property is similar to the hysteresis-loop characteristic of magnetic materials used in core memories; therefore, ferroelectric crystals may be adapted as high-speed storage devices. The technique is, however, still in the developmental stage.

4-14 CONSTRUCTION PRACTICES [91–96]

4-14.1 COMPONENT SELECTION

Because of the large number of components used in a typical computer, the highest component reliability is demanded. Present-day computer techniques are primarily concerned with transistors and integrated circuits. In each case, care must be taken in the mechanical design to ensure reliable connections and strain relief. Extreme cleanliness both in the elimination of dust and in removing all traces of chemicals used in processing is a necessity. Fortunately, such reliable components are now available.

In the case of transistors, manufacturing tolerances are so broad that measurement of critical parameters on all units is usually necessary. For some critical applications, selection of units is necessary. Complete inspection of other components, such as resistors and capacitors, is not practiced except in the case of military applications in which the highest reliability is required.

Pulsed operation introduces certain problems that are not encountered in conventional circuits. In vacuum-tube circuits, it was found that a cathode interface was formed under steady pulsed operation, which materially reduced the emission. Special cathodes have been developed for this service. Transistors for pulsed service are usually required to operate in the saturated region. For this type of operation, the saturated collector-to-emitter voltage should be minimized since this voltage times the collector current is the major part of the power dissipated in the transistor. The design of the transistor must also minimize the storage time, i.e., the time required to dissipate the minority carriers collected at the base-collector junction.

Resistors and capacitors are conventional, but types that have minimum inductance are employed. Pulse transformers, when used, employ square-loop cores which have low leakage reactance when saturated.

4-14.2 PACKAGING TECHNIQUES (MINIATURIZATION)

Techniques that employ conventional electronic components, mounted and connected by methods that minimize waste space, are classified as miniaturization techniques. The term also comprises efforts to reduce the volume of the components themselves. Microminiaturization, on the other hand, describes techniques that eliminate the cases and/or leads of the components. Carrying

the concept further, the techniques of molecular electronics eliminate the separate components as mechanical units.

The combination of miniaturization techniques with the modular concept has become standard practice in present-day computers. Printed-circuit cards are used extensively, usually fitted with a connector to facilitate maintenance. A number of ingenious designs have been worked out by the manufacturers to prevent insertion of a card in a wrong socket, to lock the cards in place, and to remove cards easily.

While circuit cards vary widely in size, a standard 0.1-inch grid system has been adopted by the industry to facilitate layout and mechanized production. Most modern components conform to this system. Automatic machinery is available for the bending and insertion of component leads in prepunched holes. The card material is usually a glass-fiber-base epoxy resin material in military equipment that must withstand a humid environment. Occasionally glass or ceramic are used. In commercial equipment, a paper-base phenolic resin is common.

Interconnections are sometimes made by hand-wiring, but more-uniform and reliable results are obtained with photo-etched wiring. Connections may be hand soldered, but with close attention to details and careful inspection good results are obtained with dip soldering.

Connectors for module cards must be rugged and reliable. The use of printed connectors should be restricted to applications where size and weight are paramount considerations since ruggedness and life are somewhat compromised.

The number of components per module is limited by maintainability considerations. About six transistors, with associated circuitry, is a practical limit. Thus, a flip-flop or binary counter element is a viable module, while a complete counter would be difficult to maintain as a module. Conversely, a single diode AND gate would be wasteful as a modular element; the usual practice is to combine several gates on a single card.

Considerable reduction in size has been obtained with welded construction. In this technique, components are interconnected by spot welding of their leads, without support of a board. The completed assembly is then encapsulated, the final module usually having the form of a rectangular block with the interconnecting leads projecting from one face. Such welded assemblies have shown high reliability in severe environments; however, the production costs are much higher than with the conventional technique.

4-14.3 MICROMINIATURIZATION

The Micromodule program, sponsored by the U.S. Army, has been used extensively. Individual ceramic wafers are employed as a substrate for individual components (occasionally more than one component is included on a wafer). Since the size is standardized, stacks of wafers may be readily interconnected to form modules which are then usually encapsulated. Micromodular construction has the benefits of size reduction (about 500,000 components per cubic foot) and reliability improvement, yet is amenable to automatic production.

More recently, there has evolved the technique of molecular electronics, particularly integrated circuits. This technique employs a semiconductor substrate on various portions of which resistors, capacitors, diodes, and transistors may be formed. Resistors, for example, are formed by applying two ohmic contacts to the substrate; diodes and transistors are formed by forming rectifying junctions. Unwanted conducting paths are etched away, and desired paths are vacuum deposited. The result may be either a complete logic element in the space occupied by a single transistor or a single micromodule.

REFERENCES

BASIC REFERENCES

1. Harry H. Goode and Robert E. Machol, System Engineering, New York, McGraw-Hill Rook Company, Inc., 1957.
2. Montgomery Phister, Jr., Logical Design of Digital Computers, New York, John Wiley, 1958.
3. Ivan Flores, Computer Logic, The Functional Design of Digital Computers, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1960.
4. R. K. Kichards, Arithmetic Operations in Digital Computers, Princeton, N. J., D. Van Nostrand, 1955.
5. George R. Stibitz and Jules A. Larrivee, Mathematics and Computers, New York, McGraw-Hill Rook Company, Inc., 1957.
6. Douglas B. Netherwood, "Logical Machine Design: A Selected Bibliography," Parts I and II, IRE Transactions of the Professional Group on Electronic Computers, Vol. EC7, No. 2, June 1958, pp. 155-178; Vol. EC8, No. 3, Sept. 1959, pp. 367-380.
7. John S. Murphy, Basics of Digital Computers, 3 Vols, New York; John F. Rider, 1958.

BOOLEAN LOGIC

8. B. Beizer and S. W. Leibholz, Engineering Applications of Boolean Algebra, New York. Gage Publishing Co., 1958.
9. F. W. Veitch, "A Chart Method for Simplifying Truth Fractions," Proceedings of the Association for Computing Machinery, Vol. 5, Feb. 1952.
10. M. Karnaugh, "Synthesis of Combinational Logic Circuits," Communications and Electronics, No. 9, Nov. 1953, pp. 593-599.
11. Samuel H. Caldwell, Switching Circuits and Logical Design, New York, John Wiley, 1958.

COMPONENTS AND CIRCUITS

12. J. Millman and H. Taub, Pulse and Digital Circuits, New York, McGraw-Hill, 1956.
13. R. K. Richards, Digital Computer Conponerits ant] Circuits, Princeton, N. J., D. Van Nostrand, 1957.
14. A. I. Pressman, Design o' Transistorized Circuits for Digital Computers, hew York; John F. Rider, 1959.
15. G. W. Booth and T. P. Bothwell, "Basic Logic Circuits for Computer Applications," Electronics, Vol. 30, March 1957, pp. 196-200.

MAGNETICS AND MEMORIES

16. S. J. Begun, Magnetic Recording, New York, Rinehart, 1949.
17. J. A. Rajchmain, "Magnetics for Computers: A Survey of the State of the Art," RCA Review, Vol. XX, No. 1, March 1959, pp. 92-135.
18. Robert E. McMahon, "Transistorized Core Memory," IRE Transaction of the Professional Group on Instrumentation, T'ol. 16, No. 2, ,Tune 1957, pp. 153-156.

PROGRAMMING

19. D. D. McCracken, Digital Computer Programming, New York, John Wiley, 1957.
20. Joachim Jeenal, Programming for Digital Computers, New York, McGraw-Hill, 1959.

BLOCK LOGICAL DESIGN

21. J. C. Alrich, "Engineering, Description of the Electro Data Digital Computer," IRE Transactions of the Prof. Group on Electronic Computers, Vol. EC4, March 1955, pp. 1-10.

22. E. S. Hughes, Jr., "The IBM Magnetic Drum Calculator Type 650; Engineering and Design Considerations," Western AIEE-IRE-ACM Computer Conference, Feb. 1954 at Los Angeles, pp. 140-154.
23. H. D. Ross, Jr., "The Arithmetic Element of the IBM 701 Computer," Proceedings IRE, Vol. 41, Dec. 1953, pp. 1287-1294.
24. W. Bucholtz, "System Design of the IHM 701 Computer," Proceedings IRE, Vol. 41. Oct. 1953, pp. 1262-1275.
25. Leiner, Notz, Smith, and Weinberger, "System Design of the SEAC & DYSEAC," Transactions of the Professional Group on Electronic Computers, Vol. EC3, No. 2, ,June 1954, pp. 8-22.
26. A. W. Banks, "The Logical Design of an Idealized General Purpose Computer," Journal of the Franklin Inst., March 1956, pp. 297-314; April 1956, pp. 421-436.
27. Astrahan and Rochester, "The Logical Organization of the New IBM Scientific Calculator," ,Journal of the Association for Computing Machinery, May 1952.

COMPUTER SPECIFICATION

28. Controllership Foundation, Business Electronics Reference Guide, Vol. 4, New York, Controllership Foundation, 1959.

SYMBOLS

29. RCA Service Company, The Language and Symbology of Digital Computer Systems. Camden. N. J., RCA Corporation, 1959.

GENERAL

30. Control Engineering Staff, The Use of Digital Computers in Science in Business and Control, New York, Control Engineering Magazine, 1958.

PERIODICALS

31. Transactions of the Professional Group on Electronic Computers. Quarterly by the Institute of Radio Engineers, New York.
32. Journal of the Association for Computing-Machinery. Quarterly by the ACM, New York.
33. Communications of the Association for Computing Machinery. Monthly by the ACM, New York.

NUMERICAL ANALYSIS

34. W. W. Seifert and C. W. Steeg, Jr., Control Systems Engineering, New York, McGraw-Hill, 1960.
35. Cecil Hastings, Jr., Approximations for Digital Computers, Princeton, N. J., Princeton University Press, 1955.
36. P. D. Crout, "A Short Method for Evaluating Determinants and Solving Systems of Linear Equations with Real or Complex Coefficients," Trans. AIEE, Vol. 60, pp. 1235-1240, 1941.
37. C. Runge, Uber die numerische Auflösung von Differentialgleichungen," Math. Ann., Vol. 46, p. 167, 1895.
38. K. Heun, "Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhangigen Veranderlichen," Zeit. f. Math. u. Phys., Vol. 45, p. 23, 1900.
39. W. Kutta, "Beitrag zur naherungsweisen Integration totaler Differentialgleichungen, Zeit. f. Math. u. Phys., Vol. 46, pp. 435-453, 1901.

GENERAL REFERENCES

40. A. D. Booth, Numerical Methods, London, England, Butterworth's Scientific Publications, 1955.
41. S. Gill, "A Process for the Step-by-step Integration of Differential Equations in an Automatic Digital Computing Machine," Proc. Cambridge. Phil. Soc., Vol. 47, pp. 96-108, 1951.
42. F. B. Hildebrand, Introduction to Numerical Analysis, New York, McGraw-Hill Book Company, Inc., 1956.
43. Z. Kopal, Numerical Calculus, New York, John Wiley & Sons, Inc., 1955.
44. W. E. Milne, Numerical Calculus, Princeton, N. J., Princeton University Press, 1949.

45. W. B. Scarborough, Numerical Mathematical Analysis, 2nd Ed., Baltimore, Md., Johns Hopkins Press, 1950.
46. J. Linvill, "Nonsaturating Pulse Circuits Using Two Junction Transistors", IRE Proc., Vol. 43, July 1955, pp. 826-34.
47. J. Ebers and J. Moll, "Large Signal Behavior Of Junction Transistors". IRE Proc., Vol. 42, December 1954, pp. 1761-72.
48. J. Moll, "Large-Signal Transient Response Of Junction Transistors", IRE Proc., Vol. 42, December 1954, pp. 1773-84.
49. R. Henly and J. Walsh, "The Application Of Transistors To Computers", IRE Proc., Vol. 46, June 1958, pp. 1240-54.
50. L. Hunter, Ed., Handbook Of Semiconductor Electronics, McGraw-Hill Book Co., N. Y., 1956.
51. On The Design Of A Very High-Speed Computer, Rept. No. 80, Univ. of Illinois, Digital Computer Lab., October 1957.
52. G. Prom and ti. Crostiy, "Junction Transistor Switching Circuits For High-Speed Digital Computer Applications", IRE Trans. on Electronic Computers, Vol. EC-5, December 1956, pp. 192-6.
53. R. Befer et al., "Surface Barrier Transistor Switching Circuits", IRE Convention Record, Part 4, 1955, pp. 139-45.
54. J. Harris, "Direct-Coupled Transistor Logic Circuitry", IRE Trans. on Electronic Computers, Vol. EC-7, March 1958, pp. 2-6.
55. H. Yourke, "Millimicrosecond Transistor Current Switching Circuits", IRE Trans. on Circuit Theory, Vol. CT-4, September 1957, pp. 236-40.
56. S. Buckman et al., A Sooke Transistorized Germanium Flip-Flop And Gate For Airborne Digital Equipment, GE Rept. No. R58EML22, Utica, 1958.
57. Engineering Research Associates, Inc., High Speed Computing Devices, McGraw-Hill Book Co., N. Y., 1950.
58. R. Richards, Digital Computer Components, D. Van Nostrand Co., Princeton, 1959.
59. D. Masher, Operation and Specification of Transistors for Direct-coupled Logic Circuits, U. S. Gov't Research Rept., Signal Corps, PB133052, Vol. 30, October, 1958, p. 219.
60. T. Cloot, "A Basic Transistor Circuit For the Construction of Digital Computing Systems," Metropolitan-Vickers Gazette, Vol. 29, November 1958, pp. 298-306.
61. C. Becker et al., "Molecular Storage and Read-out With Microwaves," IRE National Convention Record Vol. 6, Part 4, 1958, pp. 255-62.
62. J. Rajchman and A. Lo, "The Transfluxor -- A Magnetic Gate With Stored Variable Setting," RCA Review, Vol. 16 No. 2, June 1956, pp. 303-11.
63. J. Rajchman and A. Lo, "The Transfluxor," IRE Proc., Vol. 44, March 1956 pp. 321-32.
64. W. Morgan, "Bibliography of Digital Magnetic Circuits and Materials," IRE Trans. on Electronic Computers, Vol. EC-8, No. 2, June 1959, pp. 148-58.
65. W. Carroll and R. Cooper, "Ten Megapulse Transistorized Circuits for Computer Application," Semi-conductor Products, Vol. 1, July-August 1958, pp. 26-30.
66. C. Hoover et al., "Fundamental Concepts in the Design of The Flying Spot Store," Bell System Technical Journal, Vol. 37, September 1956, pp. 1161-94.
67. T. Greenwood and R. Staelher, "A High-speed Barrier Grid Store," Bell System Technical Journal, Vol. 37, September 1958, pp. 1195-223.
68. "Electroluminescence From Barium Titanate," NBS Technical News Bulletin, Lo~42, October 1958, pp. 206-7.
69. E. Grabbe et al., Handbook of Automation and Control, Vol. 2, John Wiley and Sons, N. Y. 1959.
70. H. Abbot and J. Suran, "Multihole Ferrite Core Configyrations and Applications," IRE Proc., Vol. 45, August 1957, pp. 1081-93.
71. U. Gianola and T. Crowley, "The Laddic -- A Magnetic Device for Performing Logic," Bell System Technical Journal, Vol. 47, January 1959, pp. 385-425.
72. H. Crane, "A High Speed Logic System Using Magnetic Elements and Connecting Wire Only," IRE Proc., Vol. 47, January 1959, pp. 63-73.

73. D. Looney, "Recent Advances In Magnetic Devices for Computers," Journal of Applied Physics, Supplement to Vol. 30, No. 4, April 1959, pp. 385-425.
74. J. Raffel "Operating Characteristics of a Thin Film Memory," Journal of Applied Physics, Supplement to Vol. 30, No. 4, April 1959, pp. 605-15.
75. L. Esaki, "New Phenomenon in Narrow Ge p-n Junctions, "Phys. Rev., Vol. 109, January 15, 1958, pp. 603-604.
76. R. S. Ledley, Digital Computer and Control Engineering, New York, McGraw-Hill, 1960.
77. M. Aronson, The Computer Handbook, Instruments Publishing *Co.*, Pittsburgh, 1955.
78. C. Smith, Electronic Digital Computers, McGraw-Hill Book Co., N. Y., 1959.
79. W. Keister et al., The Design of Switching Circuits, D. Van Nostrand Co., Princeton, 1951.
80. S. Caldwell, Switching Circuits and Logical Design, John Wiley and Sons, N.Y., 1958.
81. M. Phister Jr., Logical Design of Digital Computers, John Wiley and Sons, N. Y., 1958.
82. R. Higonnet and R. Grea, Logical Design of Electrical Circuits (translation), McGraw-Hill Book Co., N. Y., 1958.
83. W. Humphrey Jr., Switching Circuits With Computer Applications, McGraw-Hill Book Co., N. Y., 1958.
84. C. Florida, "Digital Computer Adding And Complementing Circuit", Electronic Engineering, Vol. 30, July 1958, pp. 429-35.
85. D. Netherwood, "Logical Machine Design: A Selected Bibliography", IRE Trans. on Electronic Computers, Vol. EC-7, June, 1958, pp. 155-78. September 1958, p. 250.
86. R. Richards, Digital Computer. Components And Circuits, D. Van Nostrand Co., N. Y., 1957.
87. C. Shannon, "A Symbolic Analysis Of Relay And Switching C'ircuits", AIEE Trans., Vol. 57, 1938, pp. 713-23.
88. C. Shannon, "The Synthesis Of Two-Terminal Switching Circuits", Bell System Technical Journal, Vol. 28, 1949, pp. 59-98.
89. D. Huffman, "The Synthesis Of Sequential Switching Circuits", Journal of Franklin Institute, 257, 1954, pp. 161-90, 275-303.
90. A. Rurks and I. Copi, "The Logical Design Of An Idealized General Purpose Computer", Journal of Franklin Institute, March and April 1956.
91. D. K. Lynn, C. S. Meyer, and D. J. Hamilton, Eds., Analysis and Design of Integrated Circuits, McGraw-Hill Hook Company, Inc., New York, N. Y., 1967.
92. G. W. A. Dummer, Ed., Solid Circuits and Microminiaturization, The MacMillan Company, New York, N. Y., 1964.
93. E. Keonjian, Micropower Electronics, The MacMillan Company, New York, N. Y., 1964.
94. S. N. Levine, Principles of Solid-state Microelectronics, Holt, Hinehart, and Winston, Inc., New York, N. Y., 1963.
95. Integrated Electronics Lecture Series, Electronic Components Laboratory, U.S. Army Electronics Command, Fort Monmouth, New Jersey, 1066.
96. AFSC Design Handbook 1-8, Microelectronics, Headquarters, Air Force Systems Command, Andrews Air Force Base, Washington, D. C., 1967.
97. Gordon B. Davis, An Introduction to Electronic Computers, McGraw-Hill Book Company, New York, N. Y. 1965.
98. Decima M. Anderson, Computer Programming, Fortran IV, Appleton-Century-Crofts, Division of Meredith Publishing Co., New York, N. Y., 1966.
99. Philip M. Sherman, Programming and Coding Digital Computers, John Wiley & Sons, Inc., New York, N. Y., 1963.
100. Thomas C. Bartee, Irwin L. Lebow, and Irving S. Reed, Theory and Design of Digital Machines, McGraw-Hill Book Company, New York, N. Y., 1962.
101. John N. Warfield, Principles of Logic Design, Ginn and Company, Boston, Mass., 1963.
102. Computer Industry Annual, 1967-68, *Computer* Files, Inc., Subsidiary of Computer Design Publishing *Corp.*, West Concord, Massachusetts, 1967.

103. T. M. Dundon, How to Specify Magnetostrictive Delay Lines for Digital Applications, Report No. 153, Honeywell Computer Control Division, Framingham, Mass., 1963.

104. AMCP 706-327, Engineering Design Handbook, Fire Control Series, Section 1, Fire Control Systems -- General.

CHAPTER 5
DIGITAL DIFFERENTIAL ANALYZERS

5-1 INTRODUCTION

As a general rule, the solution requirements for fire control computers (particularly where prediction techniques are involved) are very time limited. Accordingly, fire-control solutions must be processed at the highest possible speed consistent with the necessary degree of both accuracy and precision. The solution of a reasonably complex set of differential equations can require an unreasonably large number of iterations and an inordinate amount of computational equipment. Therefore, whenever there is a large number of differential equations to be solved, it becomes desirable to investigate the use of other than a standard digital computer? (see Chapter 4). Requirements for long-term drift stability, accuracy, and a wide range of variables lead quite often to the digital differential analyzer (DDA) as a likely candidate for this portion of the computational workload.

The use of digital differential analyzers --because of their great speed advantage in solving differential equations, coupled with modern mechanization techniques--appears to offer a promising alternative to the standard-digital-computer approach. An attractive application for a high-speed DDA is as part of a hybrid system comprised of a DDA section and a standard-digital-computer section. In this system, the DDA section would process high-speed differential-equation calculations and thereby alleviate the load on the standard section.

Potentially, the DDA can iterate a differential equation faster than a standard digital computer since the latter wastes time doing housekeeping tasks, memory-transfer instructions, indexing, etc. In most DDA designs, this potential has not been completely realized for two reasons:

1. Most DDA's have been serial machines rather than parallel.
2. Most DDA designs use a fixed independent variable increment for solving differential equations rather than a variable increment (which is used by most sophisticated standard digital computers).

The digital differential analyzer is an incremental computer consisting of a collection of digital integrators interconnected in such a way as to solve integro-differential equations. A DDA is permanently programmed--insofar as a problem solution is wired into the configuration of computing units--but the modular basis of the configuration leads to simplicity of design, and ease of maintenance and programming. It should be pointed out that a DDA solves differential equations; it does not derive them. The basic digital-integrator computing units can be used not only as integrators but also as switching devices, limiters, and generators of special functions, and can be programmed for algebraic computations such as multiplication and division. A DDA integrates by means of a digital process involving the overflow of registers; the effect is similar to solving a differential equation stepwise by finite differences.

The detail required in programming DDA's can be greatly simplified over that for other digital machines because the DDA is particularly susceptible to a block-diagram approach in programming. It is interesting that some commercial general-purpose digital computers have been provided with the software to establish operation as a DDA. This possibility is sometimes of interest

* By E. St. George, Jr. and A. Kezer, based on the references given at the end of this chapter

† See par. 3-5 of Chapter 3 for the background of this terminology.

during the design phase of a special-purpose computer.

The immediately following paragraphs on the concept of the digital differential analyzer are based on Ref. 3, which should be consulted for further information. (Also informative in this connection is Ref. 18, which presents a relatively complete discussion of DDA theory, operation, mechanization, and programming.) This remarkably illustrative description shows how a DDA computes successive values of a function by means of successive differential additions.

The problem chosen is to compute a table of values for a function $y = f(x)$. If one starts with a given initial value of y, $y_o = y(x_o)$, then

$$y(x_1) = y(x_o) + \left[y(x_1) - y(x_o)\right] \quad (5\text{-}1)$$

$$= y(x_o) + (\Delta y)_{x_o} \quad (5\text{-}2)$$

Similarly,

$$y(x_2) = y(x_1) + (\Delta y)_{x_1} \quad (5\text{-}3)$$

and so forth. In other words,

$$y(x_{i+1}) = y(x_i) + (\Delta y)_{x_i} \quad (5\text{-}4)$$

Now suppose that

$$y = e^x \quad (5\text{-}5)$$

Then

$$dy = e^x dx = y dx \quad (5\text{-}6)$$

or, approximately,

$$(\Delta y)_{x_i} = y(x_i)\,(x_{i+1} - x_i) = y(x_i)\,\Delta x \quad (5\text{-}7)$$

Thus, one can compute successive approximate values of e^x by means of the relationship

$$y(x_{i+1}) = y(x_i) + y(x_i)\,\Delta x \quad (5\text{-}8)$$

For the above illustration, the smaller one makes Δx, the more accurate the results will be. However, this is not always true; for example, when $y = \epsilon$ or because of the limitations of calculating techniques, etc.

By pausing for a moment before setting up a computing unit to mechanize Eq. 5-8, one can explore -- and then dismiss -- what could be an obstacle in the computations. Specifically, multiplication can produce a double-length result, i. e., an n-digit number multiplied by an m-digit number can result in a number with m + n digits. In a digital computer, the register to record the result of a multiplication is usually of double-word length (or two registers) wherein the first word or register records the most significant bits (major product) and the second word or register records the least significant bits (minor product). In the present problem, where Δx is small, the major part of the multiplication $y(x_i)\Delta x$ will be zero. Thus, errors will result when the minor product, which contains very significant figures in this problem, is dropped. It would therefore be better to work with the double-length extension of $y(x_i)$. However, this is not particularly desirable for what is supposed to be a small special-purpose computer.

By examining the procedure more carefully, it is possible to ascertain how the use of a double-length accumulator can be avoided. As has been noted, each time $y(x_i)\Delta x$ is formed the major product is zero; but certainly the major part of $y(x_i)$ must change eventually. This must occur during the addition $y(x_i) + y(x_i)\Delta x$ and will be the result of a carry from the minor part into the major part. In other words, as one accumulates in the minor part, one eventually propagates a carry, or overflow, from the minor part into the major part of $y(x_i)$. If one is working in binary, the carry can only be a 1 and the major part of $y(x_i)$ can change at most by 1 during any iteration. Hence, one does not need *a* double-length accumulator at all; rather, one needs a single-length accumulator that simply accumulates successive minor parts of $y(x_i)\Delta x_i$ and a counter that holds the major part of $y(x_i)$ and adds 1 to the major part of $y(x_i)$ each time there is an overflow from the minor accumulator. The accumulated minor parts are referred to collectively as the residual. *

True multiplication is not essential in finding $y(x_i)\Delta x$, for one can always take $\Delta x = 1/2^q$. One then need only shift $y(x_i)$ by q posi-

* *See* the subsequent discussion of errors in par. 5-5

tions to the right to form $y(x_i)\Delta x$. But, since there will not be a major part in the accumulator, one can always choose Δx so that the shifting of $y(x_i)$ need not have to occur actually (although of course, it does occur virtually).

To illustrate these points, consider a 4-bit binary word and let $\Delta x = 0.0001$ (see Fig. 5-1). Figure 5-1(A) illustrates the accumulator that is to hold the residual and the counter that is to hold the major part of the function value y. The circle represents the component wherein y is virtually but not actually shifted; this is simply a gate that at the proper time passes y to be accumulated with the previous residual to form the new residual. The dash-dot lines represent the "true" juxtaposition of the double length value of y. For this illustrative problem, start with $x_0 = 0$, and $y_0 = e^0 = 1$ preloaded into the counter. Then $y(x_0) = 01.00$ and $y(x_0)\Delta x = 00.00\ 0100$, where 0100 is the residual. Since

$$y(x_1) = y(x_0) + y(x_0)\,\Delta x \quad 01.00\ 0100$$

the 01.00 remains unchanged in the counter and 0100 is put into the accumulator (see Fig. 5-1(B)). With $y(x_1)\Delta x = 00.00\ 0100$ again, one obtains 0100 + 0100 = 1000 as the residual (by binary arithmetic), and 01.00 remaining still unchanged in the counter as $y(x_2)$. Then, $y(x_3)\Delta x = 00.00\ 0100$, whence 1000 + 0100 = 1100 is the new residual and 01.00 remains unchanged as $y(x_3)$ in the counter. Next, $y(x_4)\Delta x = 00.00\ 0100$, whence 1100 + 0100 = carry 1 + 0000; now the counter is increased by 1, putting 01.01 in the counter as $y(x_5)$ and leaving 0000 as the new residual. One then continues with $y(x_5)\Delta x = 00.00\ 0101$, and so forth. The result of each step and the graph of the function so calculated are shown in Fig. 5-1(B).

The progress of the plotted output in Fig. 5-1 illustrates a problem to be kept in mind, namely, that the slope of the function cannot be steeper than 45 degrees. By suitable scaling (see par. 5-4), the slope of a function can be adjusted so that it does not exceed 45 degrees within the range of computation.

5-2 LOGICAL CIRCUITRY

A generalized form of the basic integrator circuit for DDA's (see Fig. 5-2) may be considered as consisting of a Y register, or counter; an accumulator register R; and a means for adding the contents of Y to R. An addition occurs each time a pulse is applied to the gate at the point marked Δx. The content y of the Y register is altered by the Δy input which may be either a single pulse to add unity to y or a much larger number. As the number in Y is repeatedly added to R, the R register will overflow from time to time. Each time an overflow occurs, a pulse will appear on the Δz output line. The Δz overflow, or carry, from the accumulator is the differential value that is available to be added to some functional value and thus continue the calculation. For example, in the configuration of the example in Fig. 5-1, the carry of $y\,\Delta x$ was fed back to the Y register of the same computing unit to create the next point in the table of $y = e^x$. The relationship between the number of Δz pulses and Δx pulses will be

$$\Delta z = \frac{1}{B^n}\, y\,\Delta x \qquad (5\text{-}9)$$

where

y = the number in Y

B = the radix (base) of the number system in use

n = the number of orders in the registers

The problem of computing sin x and cos x will illustrate how two computing units can be combined. Let $y_1 = \cos x$ and let $y_2 = \sin x$; then the differential equations are:

$$dy_1 = -\sin x\,dx = -y\,dx \qquad (5\text{-}10)$$

and

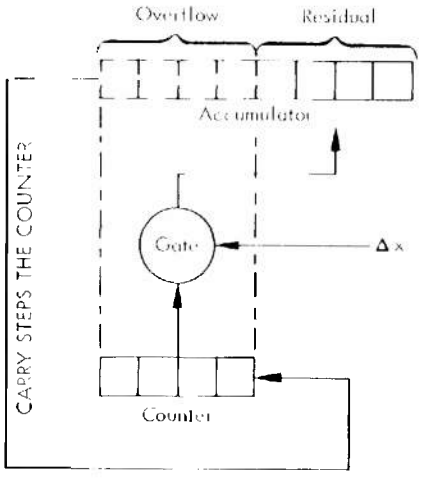
$$dy_2 = \cos x\,dx = y_1 dx \qquad (5\text{-}11)$$

In the same manner developed for the example of par. 5-1, successive approximate values can be computed from the relationships

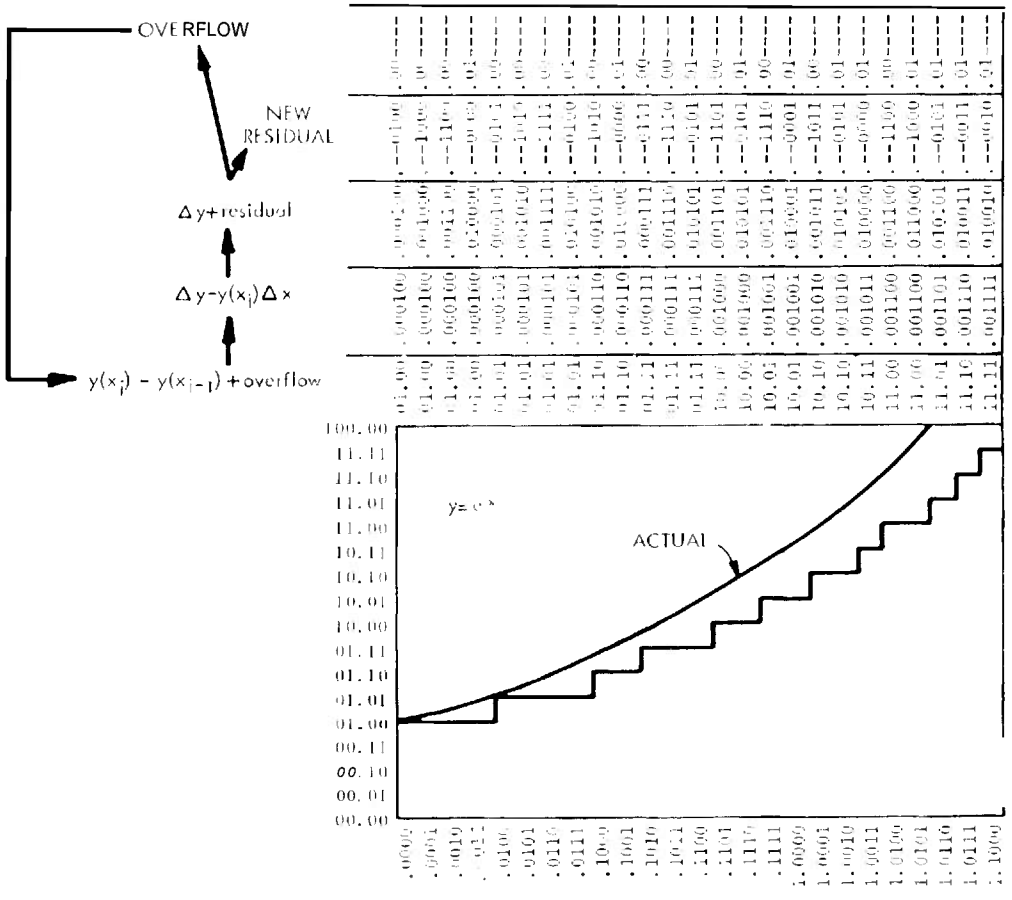
$$y_1(x_{i+1}) = y_1(x_i) + \Delta y_1\,;\ \Delta y_1 = -y_2\,\Delta x \qquad (5\text{-}12)$$

and

$$y_2(x_{i+1}) = y_2(x_i) + \Delta y_2\,;\ \Delta y_2 = y_1\,\Delta x \qquad (5\text{-}13)$$



(A) Functional Diagram of the Basic DDA Computing Element



(B) Example of the Computational Steps for a DDA Solution of y = e^x (Δx = 0.0001)

Figure 5-1. Concept of the digital differential analyzer

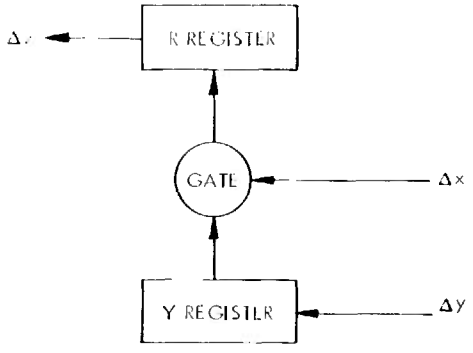


Figure 5-2. The DDA computing unit.

The arrangement of two DDA computing units shown in Fig. 5-3 can then be set up to compute y_1 and y_2.

A variety of mathematical operations can be performed with combinations of integrators. If, for example, it is desired to multiply two numbers, $y_1 y_2$, one can solve for the expression $y_3 = y_1 y_2$. (The y-subscript form has been employed to emphasize the use of the Y register in the basic computing unit.) Since

$$dy_3 = y_1 dy_2 + y_2 dy_1 \qquad (5\text{-}14)$$

then

$$y_3(x_{i+1}) = y_3(x_i) + \Delta y_3 \qquad (5\text{-}15)$$

$$y_2(x_{i+1}) = y_2(x_i) + \Delta y_2 \qquad (5\text{-}16)$$

$$y_1(x_{i+1}) = y_1(x_i) + \Delta y_1 \qquad (5\text{-}17)$$

and

$$\Delta y_3 = y_1 \Delta y_2 + y_2 \Delta y_1 \qquad (5\text{-}18)$$

which can be computed by the configuration of Fig. 5-4, using three computing units. Only the counter, or Y register, of the third unit is used, and the two inputs to it must be electronically arranged so that they do not step this counter at precisely the same time.

Generating a square also uses three units. Again, only the counter in the third unit is used, as shown in Fig. 5-5. From the relationship

$$y = x^2 \qquad (5\text{-}19)$$

it follows that

$$dy = 2x\,dx \qquad (5\text{-}20)$$

$$y(x_{i+1}) = y(x_i) + \Delta y \qquad (5\text{-}21)$$

and

$$\Delta y = 2x_i \Delta x \qquad (5\text{-}22)$$

The first counter is loaded with the constant 2 and never changes.

The amount of hardware required to put together many integrators, or computing units, particularly if many significant figures were incorporated in the computational accuracy, might appear to be very substantial. However, modern LSI (large-scale integrated) circuits have made all parallel machines quite feasible. An alternative classical approach is to use a serial system in which all the arithmetic operations required in the "gating" functions can be performed by a single set of arithmetic units. In a serial

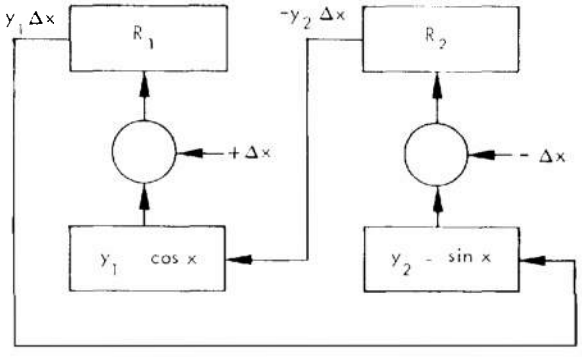


Figure 5-3. An illustrative combination of two DDA computing units.

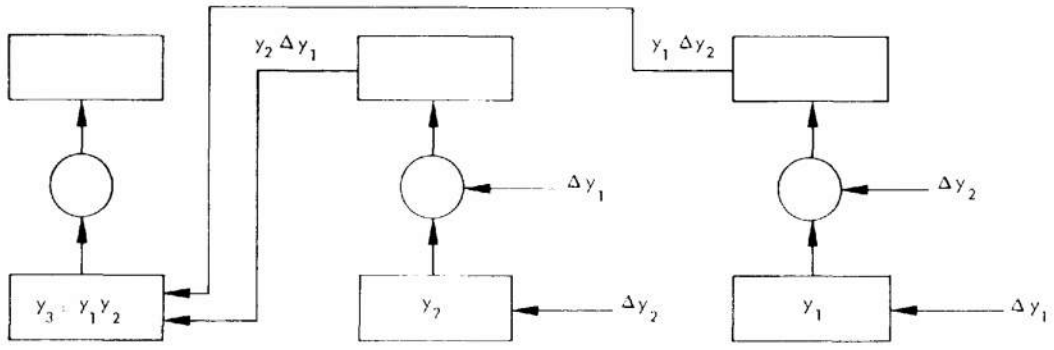


Figure 5-4. The computation of $y_3 = y_1 y_2$.

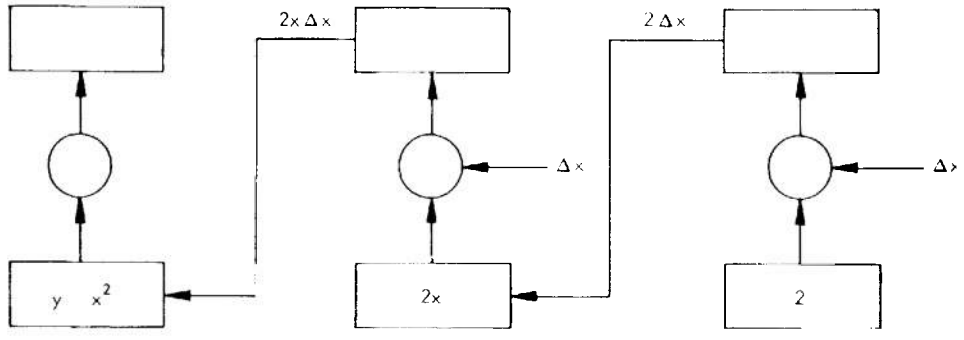


Figure 5-5. The computation of $y = x^2$.

system, the contents of the Y registers and R registers are merely held in storage locations, the contents of which are passed in serial fashion through the control and arithmetic elements of the computer. Magnetic-drum and delay-line types of storage have been used effectively in this application. Also, LSI shift registers may be effectively utilized. Jackson[7] presents a simplified block diagram of a magnetic-drum system to illustrate the principle. As shown in Fig. 5-6, five channels or tracks of information are stored on the drum. The contents of the Y, R, and ΔY lines are processed in a computational unit that is controlled by the address line, L. The operations are all synchronized by a permanently recorded clock line, C. Consider the Y channel: the Y registers of all integrators are recorded in this channel in a serial manner. One drum revolution is called a major cycle, while the iteration or processing of each integrator is termed a minor cycle. If the machine contains M integrators, then M minor cycles constitute a major cycle. Within each minor cycle, the digits contained in the Y register being processed are presented in serial, with the least-significant digit first. Thus, in a machine of M integrators that handles a maximum of N digits, the Y channel consists of MN digits around the circumference of the drum, with the digits of integrator 2 following those of integrator 1, etc. The R registers of the integrators are contained in the R line in a similar manner. The R and Y registers are in parallel on the two lines; i. e., the R register of integrator 23 occupies the same position on the R line as does the Y register of integrator 23 on the Y line.

The computation unit operates on the digits contained in the Y and R registers exactly as the action of the integrator was defined; i. e., it causes y_i to be added to or subtracted from r_i, and it adds the summation of the Δy_i

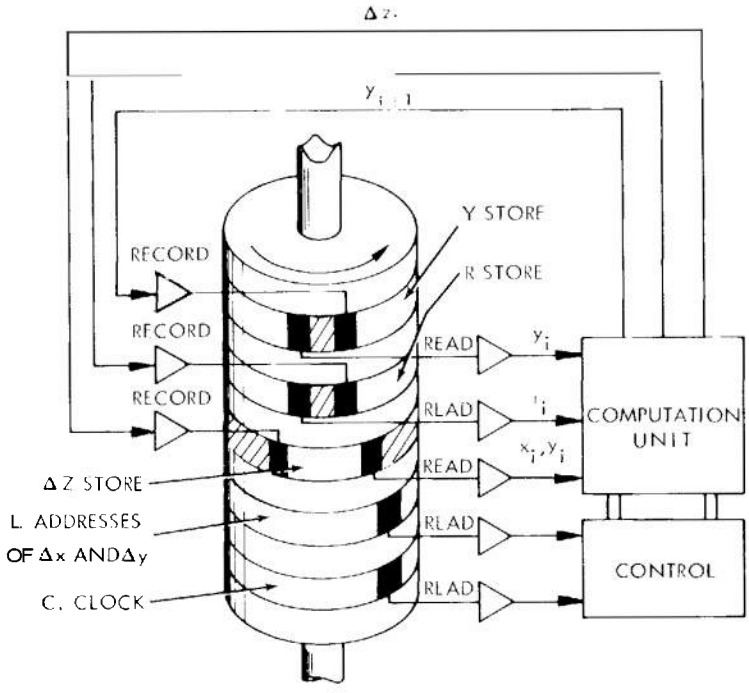


Figure 5-6. Simplified representation of DDA operation.

inputs to y_i to form y_{i+1}. This is done in a serial manner and the digits of r_{i+1} and y_{i+1} are re-recorded on the drum as soon as they are formed. The output of the integrator being processed is recorded on the ΔZ line.

The use of the ΔZ line to make the outputs of all integrators available to any particular integrator now must be considered. Suppose that a machine uses 20 digits for the Y and R registers and contains 20 integrators. In other words, assume that there is a string of integrators each taking 20 pulse-times to pass through the read or record circuit. Also assume that the ΔZ line is very short where information remains on the drum for only 19 pulse-times and that it is then read off and recorded again by the recording head that originally recorded the data. Then, as the first integrator is processed, its output--consisting of either one pulse or no pulse--is transferred to the ΔZ line and is recorded on the drum at that instant. Twenty pulse-times later the second integrator is ready to deposit its output on the ΔZ line. Since the ΔZ line represents a delay of only 19 pulse-times, the first integrator's output will have come off the line, gone back on the line, and moved over into the second position. This causes the second integrator's output to be recorded on the ΔZ line directly behind the first integrator's output. The third integrator coming 20 pulse-times later will, of course, place its output on the ΔZ line after those of integrators 1 and 2, and so on. After 19 integrators have been processed, the ΔZ line will contain the most recent outputs of all of them. Since the ΔZ line completely recycles during the time that any integrator is passing through the computation unit, it makes the outputs of all the integrators available to the integrator that is being processed. This is obviously a simplified example because, in general, DDA's contain many more integrators than they carry digits per integrator. In this case, two or more ΔZ lines or more than one reading head on the ΔZ line are required. However, this example does illustrate the processing scheme common to all systems.

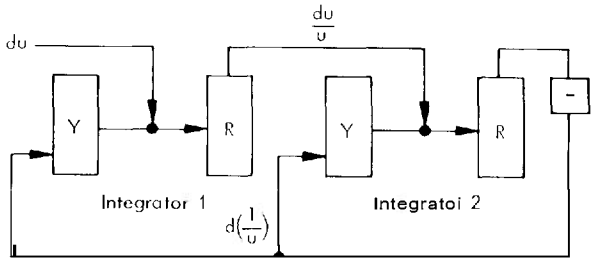
To pick the dy inputs for an integrator, a coincidence circuit observes the ΔZ line and the address channel L. When a pulse appears on the address line, the coincidence

circuit causes the contents of the ΔY line at that instant to be fed to an up-down counter that sums the dy inputs. In DDA's to date, provision has been made for summing from 7 to as many as 15 dy inputs to a single integrator. In most machines, the dy addresses are offset one integrator space so that they are summed during the integrator period prior to the one to which it applies.

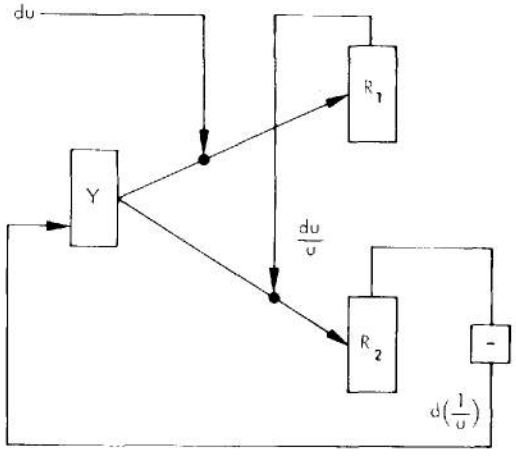
In most machines, the dx input to an integrator is restricted to a single input. If the sum of several variables is desired as the variable of integration, some other means of summing the increments must be used. Usually, however, the dx address consists of a binary number denoting the pulse position in the ΔY line of the desired input. This binary number is used to set a counter two integrator times ahead of the integrator to which it applies. Then, during the next integrator period, the counter counts down and causes a dx register to observe the ΔY line at the proper instant to pick up the desired dx input for the integrator to be processed next.

In addition to improvements in machine organization that can be effected by the use of recent integrated-circuit technology, several additional improvements can be made over classical DDA's. One method [9] is the combination of several integrators that contain the same variable and share the Y register and adder elements with multiple R registers. Another improvement is the use of either quasi-floating-point arithmetic or pure-floating-point arithmetic in which, instead of scaling a fixed-point machine to handle the worst-case range of variables, the DDA is scaled to handle a nominal range. When a variable exceeds this range, critical integrators are automatically rescaled to modify the increment size of the variable. Since the iteration time of a DDA is dependent on the variable size, this technique also allows the parallel DDA to run at the optimum speed at each region of its solution.

As an example of the technique of integrator sharing, consider the case of the solution of 1/u. Fig. 5-7(A) shows the conventional interconnection of two integrators that would be used for this solution. Both integrator 1 and integrator 2 store the variable 1/u in the Y registers. Therefore, the corresponding shared-integrator arrangement of Fig. 5-7(B) has two R registers, where register R_1 is identical with the R register of integrator 1 and register R_2 is identical with the R register of integrator 2. This shared-integrator DDA arrangement requires one adder, one Y register, and two R registers, whereas the conventional DDA arrangement requires two adders, two Y registers, and two R registers.



(A) Conventional Implementation



(B) Shared-integrator Implementation

Figure 5-7. Conventional and shared-integrator DDA implementations of 1/u.

5-3 SOLUTION OF DIFFERENTIAL EQUATIONS

Par. 5-2 has discussed how the output of the R register of the basic integrator circuit (see Fig. 5-2) is considered to be

$$\Delta z \quad y\Delta x \qquad (5\text{-}23)$$

Usually the finite increments Δx, Δy, and Δz are replaced by the differentials dx, dy, antl dz. The characteristic equation of the device then becomes

$$dz = ydx \qquad (5\text{-}24)$$

or

$$z = \int_{x_o}^{x} ydx + z_o \qquad (5\text{-}25)$$

There is yet another way of looking at this problem. If it is assumed that y is an integer arid Δx is plus or minus one unit, then the digits contained in the register R represent the fractional part of

$$\frac{1}{B^n} \sum y\Delta x \qquad (5\text{-}26)$$

where €3 is the radix of the number system used and n equals the number of digits in the registers. Considered in this way, the summation of the Δz outputs is then the integral part of the preceding expression. Thus, this constant of proportionality canbe included in the characteristic equation of the device, so that

$$z' = \frac{1}{B^n} \int_{x_o}^{x} ydx + z_o \qquad (5\text{-}27)$$

where z' is an integer. The fractional part of expression 5-26, which remains in tlic R register, is neglected and represents a round-off error. Note that this error is always less than Δx, which has been assumed to be plus or minus one unit.

The step from finite increments to differentials in a purely incremental device may jar the mathematicians among the readers. Admittedly, it is inaccurate but it seems to be customary in discussing DDA's and *is,* perhaps, a carry-over from the mechanical integrators which were truly continuous devices. Tt is used here because it makes the general explanation simpler and is fairly accurate if second-order effects are neglected. The reader is reminded, however, that although the digital integrator can be considered to be functionally equivalent to the mechanical integrator, its discrete nature introduces errors; these are discussed in par. 5-5.

It has been pointed out how, in DDA operation, the increment mav bc macle a sufficientlv small part of the variable so that the incremental *e*quation closely approximates the integral equation

$$z = \int_{x_o}^{x} ydx \qquad (5\text{-}28)$$

or the differential equation

$$dz = ydx \qquad (5\text{-}29)$$

or the derivative equation

$$\frac{dz}{dx} = y(x) \qquad (5\text{-}30)$$

This establishes the individual integrator as a device that solves a first-order differential equation. The useful fact is that if the Y register is loaded with a second- (order derivative antl the ***Ax*** input is the increment in the independent variable, then the Δz output generates the first-order derivative

$$\left(\frac{d^2y}{dx^2}\right) dx = d\left(\frac{dy}{dx}\right) \qquad (5\text{-}31)$$

It is generally possible to isolate the highest-order derivative in a differential equation. For example, in the case of the equation

$$\frac{d^2w}{dt^2} - w\frac{dw}{dt} - \sin w = 0 \qquad (5\text{-}32)$$

one may solve for the highest-order derivative as

$$\frac{d^2w}{dt^2} = w\frac{dw}{dt} + \sin w \qquad (5\text{-}33)$$

The differential of this quantity is

$$d\left(\frac{d^2w}{dt^2}\right) = d\left(w\frac{dw}{dt}\right) + d(\sin w) \qquad (5\text{-}34)$$

$$\left(\frac{dw}{dt}\right) + \frac{dw}{dt}\, dw + d(\sin w) \qquad (5\text{-}35)$$

From a network of integrators, such as that shown in Fig. 5-8, the first-order derivative and the dependent variable can then be obtained from the second-order derivative--which is itself a function of the first-order derivative, as shown by Eq. 5-33.

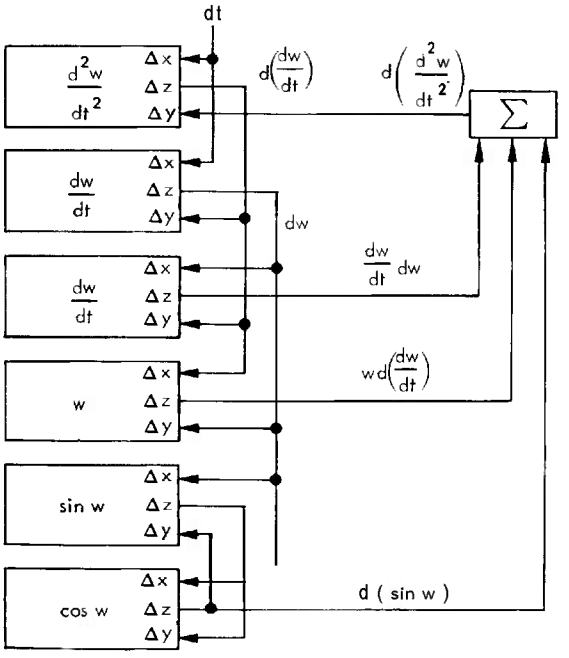


Figure 5-8. Connections employed for the solution of $\frac{d^2w}{dt^2} - w\frac{dw}{dt} - \sin w = 0$.

In this general approach, it is assumed that the second-order derivative is available to load the Y register of the "first" integrator. This is integrated against t to obtain the first-order derivative, which is integrated in turn to obtain w. Note that the first-order derivative is needed twice, and hence a duplicate count in another Y register is generated. Integrated against w, this element produces the term $\frac{dw}{dt}$ dw. By using the d($) output of the "first" integrator as the Δx input to the integrator containing the dependent variable w, the first term on the right-hand side of Eq. 5-35 is obtained. The generation of d(sin w) uses the network already shown in Fig. 5-3. Note that the generation of $\frac{d^2w}{dt^2}$ depends on the knowledge of w, and the generation of w depends on the knowledge of $\frac{d^2w}{dt^2}$. The feedback connection makes this arrangement of integrators a closed system driven by the dt input. It might be said that the feedback mechanizes the equal sign in Eq. 5-35 since it applies the constraint that forces the system to equalize the two sides of the equation. This represents an important feature in the DDA solution of differential equations.

Just as a set of initial conditions is needed to specify the starting point in solving differential equations, the initial conditions or starting point for a DDA must also be specified. Initial values are placed in all the Y registers. The equation is solved once for the machine at a specific point, and then the machine takes over. The running solution produced by the machine is then accurate and up-to-date within the limitations imposed by the discrete nature of the integrators.

5-4 SCALING[2,22]

Scaling is required to fit the variables of a problem to the numerical range of the DDA. Efficient scaling for a problem requires that the maximum absolute value of all variables be known or carefully estimated to prevent the inadvertent overflow of a Y register during the running of a problem. If too-generous an estimate of maximum values is given, the solution requires more time than is necessary. With well-established values, it should be possible to scale the variables so as to make the most efficient use of the DDA's precision and yet keep the computing time to a minimum.

Jackson[2] presents a set of relationships for scaling the DDA as follows. First, let capital letters represent actual or problem values and let lower-case letters represent machine values. Further, assume that for each quantity there is a scale factor S, *so* that B^S represents one unit of the quantity to the machine, where B is the radix of the number system used. In other words, the following relations exist:

$$dx = B^{S_x} d\lambda \qquad (5\text{-}36)$$

$$dy = B^{S_y} dY \qquad (5\text{-}37)$$

$$dz = B^{S_z} dZ \qquad (5\text{-}38)$$

in which x, y, and z represent machine values; X, Y, and Z represent actual values; and B is the radix of the number system used in the DDA. For each integrand Y of the integrator, there is a positive integer m such that B^{m-1} is less than the maximum absolute value of Y and this maximum absolute value is less than or equal to B^m. This relationship can be stated in the form

$$B^{m-1} < |Y|_{max} \leq B^m \qquad (5\text{-}39)$$

Now it is preferable to have the output relationship of an integrator in the form dZ=YdX, in terms of the actual values. However, it was established earlier that the characteristic equation of the integrator is $dz=(1/B^n)ydx$ in terms of the machine values, where all quantities are integers and n represents the number of digits in the Y register. If the values from Eqs. 5-36 through 5-38 are substituted in the characteristic equation, it follows that

$$B^{S_z} dZ = \frac{1}{B^n} B^{S_y} Y \cdot B^{S_x} dX \qquad (5\text{-}40)$$

and, if the relation dZ = YdX is to hold, then

$$B^{S_y + S_x - n - S_z} = 1 \qquad (5\text{-}41)$$

or

$$S_y + S_x = n + S_z \qquad (5\text{-}42)$$

This first scaling equation establishes a relation between the number of digits used in the Y register, the scale factors of the variables of integration, and the scale factor for the output variable--as it should, since it was derived from the characteristic equation.

As stated earlier, the *Y* register must be capable of holding the integrand y of the integrator at all times during the computation. From the definition of scaling factor it is known that for each unit of Y, the Y register will have to hold the number B^{S_y}. Also, it was noted that at some time during the computation the integration would be almost as big as B^m. Therefore, if it is not to overflow, the Y register must be capable of holding a number as large as $B^m \cdot B^{S_y}$ or B^{m+S_y}. In other words,

$$m + S_y \leq n \qquad (5\text{-}43)$$

This is the second scaling relation, and it determines the number of digits required for an integrator in terms of the maximum value of the integrand and the scale of the integrand input. It also is known, for any particular machine design, that there is some maximum number of digits available, say N. Therefore, this last relation can be expanded to

$$m + S \leq n \leq N \qquad (5\text{-}44)$$

Another useful but dependent scaling relation can be obtained from the two preceding relations. From Eq. 5-42, $S_y = n + S_z - S_x$; from Eq. 5-44, $S_y \leq n - m$. Therefore, $n + S_z - S_x \leq n - m$, or $S_z - S_x \leq -m$, or finally

$$S_x - S_z \leq m \qquad (5\text{-}45)$$

In recapitulation, the three scaling relationships are

$$\left.\begin{aligned} S_y + S_x &= n + S_z \\ m + S_y &\leq n \leq N \\ S_x - S_z &\geq m \end{aligned}\right\} \qquad (5\text{-}46)$$

where m is the smallest integer satisfying $B^{m-1} < |Y|_{max} \leq B^m$, n is the number of digits in the Y register, and N is the maximum number of digits available in the Y register. These equations define the scaling relations necessary for scaling any single integrator. The extension to a system of integrators is, on the surface, simple and straightforward. For compatible operation, all the variables contributing to a particular input must be at a common scale. For instance, all dy inputs to a particular integrator must have the same scale. Violation of this rule results in multiplication by powers of B (which can be used to advantage at times). It is also clear that if the dx output of one integrator is used as an input to another integrator, then, in general, the two must be of equal scale.

All this may seem axiomatic, and indeed it is, but it must be stressed because, while the scaling relations are straightforward, scaling is the most difficult and important phase of programming a DDA. Part of the difficulty results from the fact that the scaling relations involve inequalities. In general, a large number of sets of scaling factors exists that will satisfy any particular system of integrators. The direct approach is to scale the problem once and then adjust the scaling factors until an efficient scaling is found. Usually, there are two possible criteria for fixing the scaling. Unfortunately, they are incompatible. One may require a particular variable to a certain precision, thus fixing its scale and establishing all others. On the other hand, one can fix the time of computation -- thereby fixing the scale of the independent variable. This brings out an important feature of the DDA: the ability to trade time for accuracy, or vice versa. The result of scaling a problem is the determination of the register lengths for every integrator in the machine. Then a correctly scaled problem may be stepped up in accuracy or in speed of computation by readjusting all integrator lengths by the same amount.

Once the scaling of the variables has been determined, it is possible to determine the initial conditions, i.e., the initial value of each integrand, and to express these in terms of the machine values from the scaling relations.

It can be seen that a major disadvantage of a conventional DDA organization is the use of fixed-point arithmetic, in which the scaling is based upon the maximum value that each variable can assume. If some of the variables vary over a large range, an extremely small independent increment may be required to maintain the accuracy. As the size of the increment decreases, the number of iterations increases. If a standard digital computer were restricted to fixed-point arithmetic, it would have the same type of scaling problems encountered by the DDA. There are several possible techniques that can be employed to overcome the scaling problems of fixed-point arithmetic. One, of course, is to implement a fully-floating-point machine. Another, more practical, approach is to use a quasi floating point (i.e., multiple scale) that is a compromise between fixed-point and floating-point arithmetic. If the complete range of a variable is divided into several sub-ranges and each sub-range is scaled to fit the full length (number of bits) of the fixed-point word, the DDA may compute with the scale corresponding to the particular sub-range in which the variable happens to lie. When the variable changes to a different sub-range, the DDA must then switch to the corresponding scale. Each scale therefore uses fixed-point arithmetic, but the switching from scale to scale as the variable changes magnitude simulates the effect of a floating point.

5-5 ERRORS IN THE DDA

The operation of the digital integrator has been characterized by the integral equation

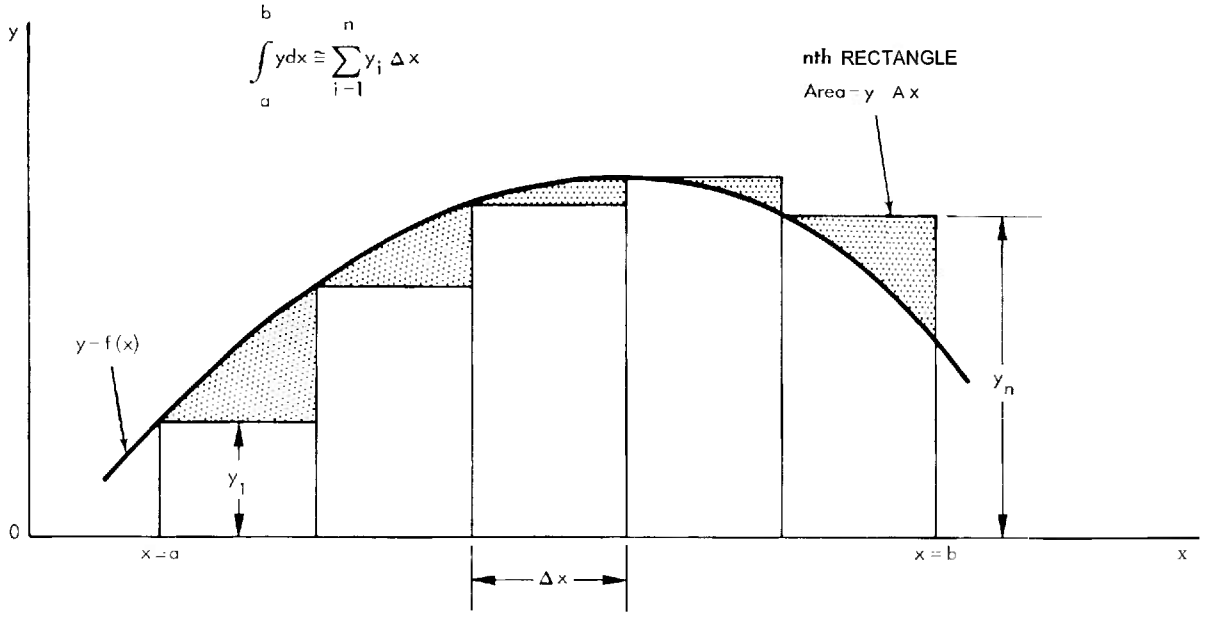
$$z = \int_{x_o}^{x} y\,dx \qquad \text{[Eq. 5-28]}$$

Over a range from x = a to x = b, this integral can be approximated by a finite sum as follows when the range interval has been divided into n equal parts; see Fig. 5-9(A):

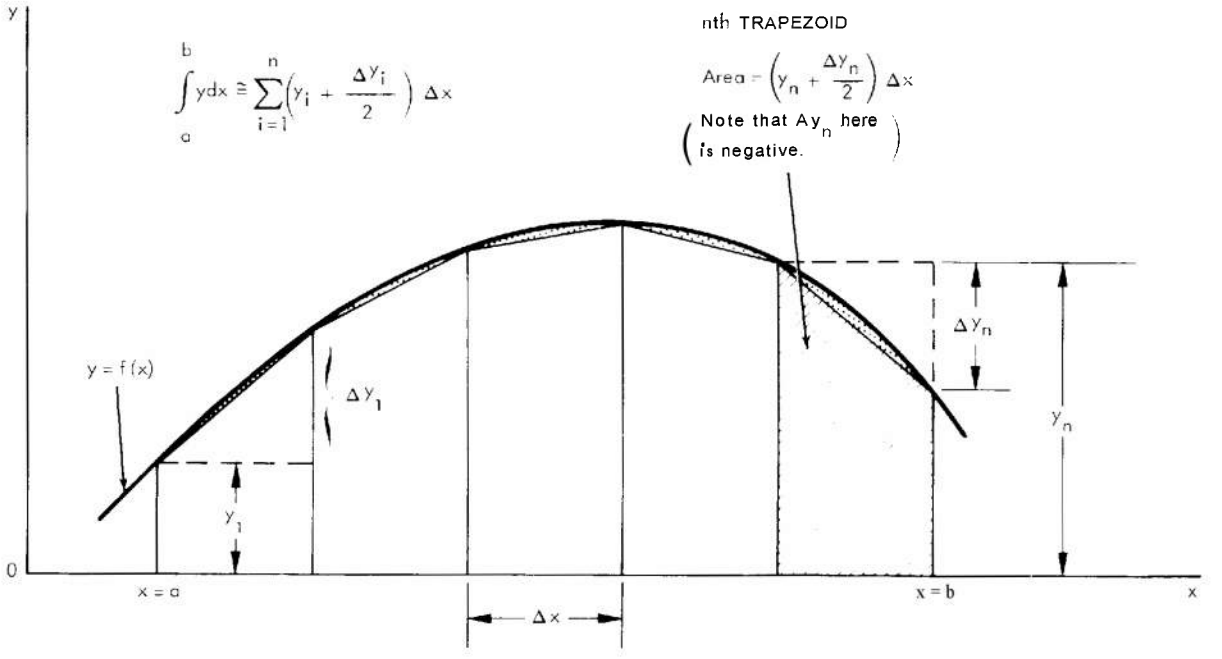
$$\int_a^b y\,dx \approx \sum_{i=1}^{n} y_i \Delta x \qquad (5\text{-}47)$$

The error due to this approximation can be made as small as desired by making Δx small enough or, by what is the same thing, n large enough. The error incurred by using a finite Δx is known as the truncation error. The method of integration as outlined here is known as Euler or rectangular integration because the integral is approximated by summing the rectangular areas $y_i \Delta x$. This is the crudest form of integration and the truncation error can be quite large unless Δx is made very small. However, making Δx smaller means that the machine must run longer to cover a given range. Thus, there is a practical limit to the reduction possible in the size of the increments Δx and it is desirable to be able to reduce the truncation error without having to reduce Δx further. This can be done by refining the method of integration.

A large improvement, without complicating the circuitry, can be made by using what is commonly known as trapezoidal integration; *see* Fig. 5-9(B). In this method, the integral is approximated by summing the



(A) Rectangular Integration



(B) Trapezoidal integration

NOTES:

1. ▨ denotes the truncation error.
2. Δx is made large here for illustrative purposes.

Figure 5-9. Truncation errors associated with rectangular arid trapezoidal integration.

areas of trapezoids $(y_i + \frac{\Delta y_i}{2})\Delta x$. A further reduction of the truncation error may be made by going to forms of parabolic integration that approach Simpson's rule; however, the complexity of circuitry and the number of storage registers required rises rapidly.

An estimate of the errors incurred by these two methods, determined by Courant, is $(1/2)M_1(b - a)\Delta x$ for rectangular integration and $\frac{1}{12}M_2(b-a)(\Delta x)^2$ for trapezoidal integration, where M_1 is the upper bound of the absolute value of the first derivative, M_2 is the upper bound for the second derivative, and $(b - a)$ is the range. The ratio of the two errors is $\frac{1}{6}\left(\frac{M_2}{M_1}\right)\Delta x$. Since Δx is some small fraction, the reduction in the truncation error by the use of trapezoidal integration can be considerable.

A second source of error is encountered in using a finite R register. Since the register is broken off and Δz pulses are transmitted instead of accumulating the sum in an infinite register, the quantity Δz (when accumulated in another integrator) is always in error by the remainder left in the R register. This error makes up part of the round-off error.

Another source contributing to round-off error is the system of intercommunication between integrators. In a system where the communication consists of either one pulse or no pulse, representing a +1 or -1 increment, respectively, what is known as binary intercommunication exists. Here, since the output must be either a plus one or a minus one, an error ϵ is introduced where

$$-\Delta z \leq \epsilon \leq +\Delta z \qquad (5\text{-}48)$$

This can be appreciated by considering an integrator that should have a zero output. Since only +1 or -1 are available, the error at any instant is a full unit of Δz. To find the true value, the average output must be considered. This source of error can be reduced by a factor of two by using what is known as ternary intercommunication. In this system, the output of an integrator can be +1, 0, or -1 and the error is

$$-\frac{1}{2}\Delta z \leq \epsilon \leq +\frac{1}{2}\Delta z \qquad (5\text{-}49)$$

Note that, now, two bits of information are required for intercommunication. That is, the amount of information to which there must be random access has doubled or, in other words, the Δz line of the machine is doubled.

It is also instructive to consider the effect of a loss of higher-order terms in, for example, the generation of sin 8. By taking a Taylor's series expansion of $\Delta \sin 8$, it is found that

$$\Delta \sin\theta = \cos\theta\Delta\theta - \frac{1}{2}\sin\theta\,(\Delta\theta)^2 - \frac{1}{6}\cos\theta\,(\Delta\theta)^3 + \ldots \qquad (5\text{-}50)$$

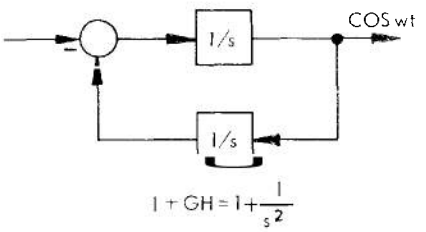
Yet, with the arrangement shown in Fig. 5-3, it can be seen that

$$\Delta \sin\theta \cong \cos\theta\Delta\theta \qquad (5\text{-}51)$$

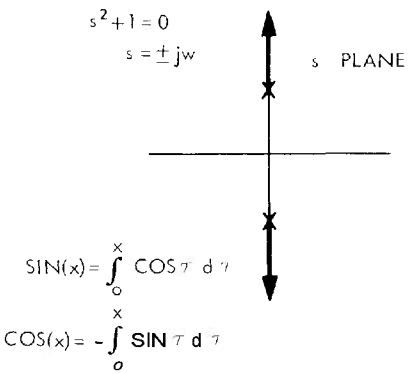
Thus, there is a first-order approximation where the higher-order terms have been neglected. In many problems where the range of θ is not too great, the error encountered here may be truly negligible. But in some applications, say a control application, where the computer may be required to run continuously for hours or even days, the drift in $\sin\theta$ due to these neglected terms would soon render the solution useless.

Also to be considered are the consequences of the effective time delays inherent in the DDA computational process. The most important delay is that between an overflow and the subsequent addition. The example which follows (adapted from Ref. 20) should serve to illustrate the point. Other more complete analyses may be found in Ref. 21.

Figure 5-10(A) shows an analog feedback loop used to generate the sine and cosine waveforms that are the solution of Eqs. 5-10 and 5-11. The associated root locus of Fig. 5-10(B) shows that the loop is conditionally stable for all gains. Figure 5-11(A) shows the simplest DDA implementation of Eqs. 5-10 and 5-11. The difference equations of Fig. 5-11(B) can be related to the sampled-data feedback system shown in Fig. 5-11(A). Evaluation by the Z-transform technique yields the Z-plane root locus shown in Fig. 5-11(B).

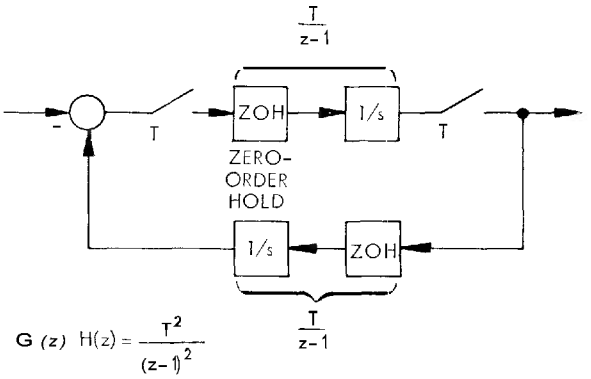


(A) Functional Diagram

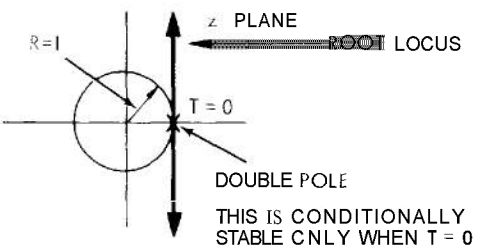


(B) The s-plane Root Locus

Figure 5-10. The stability of a continuous (analog) solution of the equations for sine and cosine functions.



(A) Functional Diagram



SIN (nil) = SIN (n) + COS (n) Δx
COS (nil) = COS (n) - SIN (n) Δx

(B) The z-plane Root Locus

Figure 5-11. A DDA-integrator solution of the sine and cosine equations; parallel implementation.

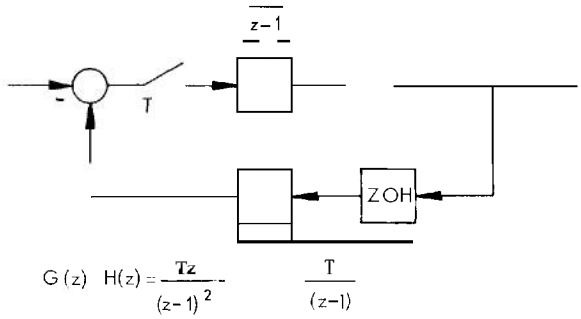
The root locus shows that the system is unstable (outside the unit circle) for all $\Delta x > 0$ values and is conditionally stable only for $\Delta x = 0$. For finite Δx values, the amplitude of the sine wave will grow exponentially with increasing x. To offset this error, the integrators must be continually initialized.

The configuration of Fig. 5-12 (A) has only one delay in the loop, which corresponds to a serial implementation. Note the quantity (n + 1) that appears on the right-hand side of the difference equations of Fig. 5-12(B), as compared with the difference equations of Fig. 5-11(B). This difference stems from the additional time delay associated with the additional zero-order hold shown in Fig. 5-11(A). The root locus of Fig. 5-12(B) shows that the system depicted in Fig. 5-12(A) is conditionally stable for $T \leq 4$, where T is the period of the iteration.

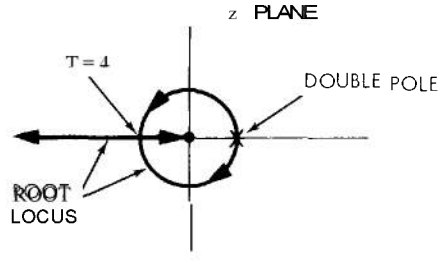
5-6 DDA COMPONENTS, CIRCUITS, AND HARDWARE

The conception of the DDA is generally attributed to Steele, a mathematician. His contribution was to show how one could realize digital accuracy in the time-honored differential-analyzer method of machine computation; how the process could be mechanized digitally: and how the time-sharing capabilities of a digital computer could be used to produce a machine that is smaller, simpler, and cheaper than an analog type, and yet have the inherent accuracy of a digital machine.

The components, circuits, and input-output peripheral equipment associated with DDA designs are similar to those used in standard digital computers, and are chosen to provide the required logical characteris-



(A) Functional Diagram



SIN (n+1) = SIN (n) + COS (n) Δx
COS (n+1) = COS (n) - SIN (n+1) Δx

(B) The z-plane Root Locus

Figure 5-12. A DDA-integrator solution of the sine and cosine equations; serial implementation.

tics and operating speed for any particular application. With the advent of the integrated circuit, functional elements such as arrays of logic gates, flip-flops, etc. and the LSI (large-scale integrated) circuits, many new mechanization possibilities exist. For example, Ref. 20 details a complete DDA adder-integrator integrated circuit that, when combined with a single-chip shift register, is a complete DDA element. Because of the extremely small size of these types of semiconductor elements -- MOS (metal-oxide-silicon) as well as bipolar--many new machine organizations are feasible from a hardware standpoint, As an example of a parallel DDA for the implementation of the equation.

$$y = \ell n(x) \tag{5-52}$$

by means of the relationship

$$dy = d[\ell n(x)] = \frac{dx}{x} \tag{5-53}$$

using a typical arrangement of these new LSI semiconductor elements, consider the system shown in Fig. 5-13. This circuit uses three DDA integrator elements (each in a flat-pack case approximately 0.5 in. X 0.3 in. X 0.1 in.) and five 20-bit shift-register elements (each in a TO-78 transistor-type case approximately 0.4 in. dia. X 0.2 in. high). These eight circuit elements require approximately the same size and number of interconnections that is required to construct a simple flip-flop of discrete transistors and diodes.

Because of the improved characteristics of glass and wire delay lines, the use of drum memories has diminished rapidly, with the great benefit of eliminating complex and unreliable electromechanical components. Similarly, the 16-, 25-, 50-, and 100-bit shift-register integrated-circuit chips that are starting to appear commercially should rapidly replace the delay-line memories (particularly in parallel DDA's), thereby providing another significant decrease in size, weight, and power -- and an increase in reliability.

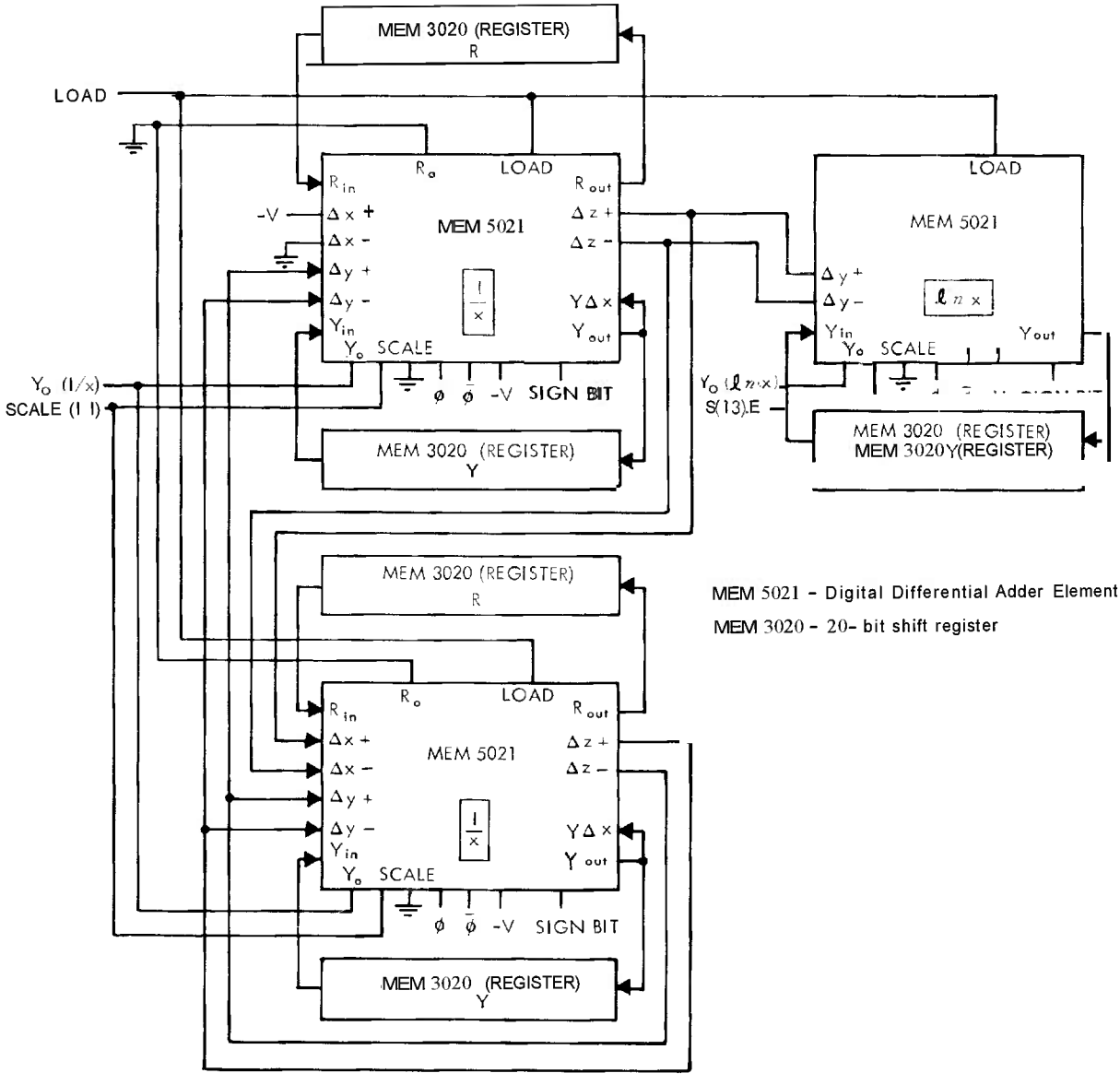


Figure 5-13. Interconnection diagram of DDA and shift-register integrated-circuit elements (MOS) to solve the equation $y = \ell n(x)$.

REFERENCES

1. G. F. Forbes, DDA, revised, Pacoima, California, Pacoima Press, 1957.
2. AMCP 706 - 293, Engineering Design Handbook, Surface-to-Air Missiles Series, Part Three, Computers.
3. R, S. Ledley, Digital Computer and Control Engineering, New York, McGraw-Hill, 1960.
4. R. K. Richards, Arithmetic Operation in Digital Computers, New York, D. Van Nostrand Co., 1955.
5. J. Prince, et al., Synthesis And Design Of A Variable Incremental Digital C o g puter For The Mauler Weapon System, General Electric TIS R58APS100.
6. J. Githens, Digital Differential Analyzers, Bell Laboratories report on the TRADIC Computer Research Program, April 1955.
7. E, M. Grabbe, S. Ramo, and D. E. Wooldridge, Editors, Handbook of Automation, Computation, and Control, Vol. 2, Computers and Data Processing, New York, John Wiley & Sons, Inc., 1959.
8. Vannevar Bush, "Differential Analyzer", J. Franklin Inst., **212,** 4, 447-448, (1931).
9. J. F. Donan, "The Serial-memory Digital Differential Analyzer", Mathematical Tables and Other Aids to Computation, 6, 38, 102-112 (1952).
10. Douglas R. Hartree, Calculating Instruments and Machines, University of Illinois Press, Urbana, Ill., 1949.
11. Myron J. Mendelson, "The Decimal Digital Differential Analyzer", Aeronaut. Eng. Rev., pp. 42-54, February 1954.
12. M. Palevsky, "The Design of the Bendix Digital Differential Analyzer", Proc. IRE, **41,** 1352-1356 (1953).
13. R. E. Sprague, "Fundamental Concepts of the Digital Differential Analyzer," Mathematical Tables and Other Aids tc Computation, 6, 41-48 (1952).
14. E, Weiss, "Applications of the CRC 105 Decimal Digital Differential Analyzer," IRE Trans. on Electronic Computers, EC-1, 19-24 (1952).
15. D. J. Winslow, "Incremental Computers in Simulation", Meeting of South East Simulation Council, Huntsville, Ala., Oct. 30, 1958.
16. B. M. Gordon, "Adapting Digital Techniques for Automatic Controls", Electrical Manufacturing, pp. 136-143, Nov. 1954; and pp. 120-126, Dec. 1954.
17. B. M. Gordon, "Digital Differential Analyzers", Instruments and Automation, pp. 1105-1109, June 1957.
18. H. W. Gschwind, "Digital Differential Analyzers", Electronic Computers, P. Von Handel, Ed., Prentice-Hall, 1961.
19. M. W. Goldman, "Design of a High-Speed DDA", Proceedings of the 1965 Fall Joint Computer Conference, pp. 926-949.
20. J. D. Callan, "MTOS Integrated Digital Differential Analyzer", Microelectronics Application Notes, General Instrument Co., Hicksville, New York, March 1967.
21. A. P. Sage and R. W. Burt, "Optimum Design and Error Analysis of Digital Integrators for Discrete System Simulation", Proceedings of the 1965 Fall Joint Computer Conference, pp. 903-914.
22. A. Gill, "Systematic Scaling for Digital Differential Analyzers", IRE PGEC, pp. 486-489, December 1959.

CHAPTER 6
ANALOG COMPUTERS*

6-1 INTRODUCTION

The definition of "analog computer" is not simple and clear-cut because the term embraces several distinguishing characteristics, a broad range of components, and various methods of problem solution. First among the characteristics is that problem variables are generally represented as continuously variable physical quantities that may take the form of mechanical, electrical, hydraulic, pneumatic, or magnetic quantities. An analog computer represents one physical form of the mathematical model of the system under consideration. The variables in the analog model may take the same physical form as in the original system, but more often the analogy is one of mathematical equivalence, because, as is illustrated in Chapter 1 of this handbook, many different physical systems obey mathematical laws of identical form.

Analog-computer components perform basic mathematical operations such as addition, multiplication, division, integration or function generation, and may be of a mechanical, electrical, electromechanical or electronic type. Certain general advantages in speed, accuracy, or reliability accrue to each type, and some types are better suited to performing specified mathematical operations. The most advantageous mathematical modeling of a given physical system may dictate the use of more than one type of element -- electromechanical for one operation, electronic for another, and so forth.

A classification of analog devices under three headings of direct analogies, indirect analogies, and simulators is shown in Fig. 6-1. The usefulness of a scaled replica or Direct Analog is apparent for obtaining valuable information in a study of the effectiveness of control of water-shed runoff, or in the collection of aerodynamic data. The power-system network analyzer is a form of direct analog consisting of both lumped-parameter and distributed-parameter portions. Voltages representing the generators are impressed on the analyze-, and currents and voltages are measured at distribution points and load points in the system. The network analyzer serves to emphasize a general characteristic of the analog computer -- the variables are customarily measured rather than counted. The measuring instruments typically used for recording variables in an analog computer are ammeters, voltmeters, oscillographs, magnetic and optical recorders, and plotting boards.

The second heading in Fig. 6-1, Indirect Analog, includes mechanical and electrical types. The slide rule, devised in the seventeenth century, needs no amplification as a basic engineering tool. Mechanical linkages are discussed in par. 6-4 through par. 6-4.15. The mechanical differential analyzer for solving ordinary differential equations is also reviewed in par. 6-4 through par. 6-4.15. Under Electrical Indirect Analogs, the items are self-explanatory with one exception. An example of the use of electrolytic tanks is in the determination of the trajectories of electrons in a cathode-ray tube.

One point that should be mentioned under Simulators is that this class of analog computer is usually constrained to operate in "real time", whereas analog devices in themselves may often operate in extended, or slow, time or in compressed, or fast, time.

Analog methods have two chief advan-

*By W. W. Seifert

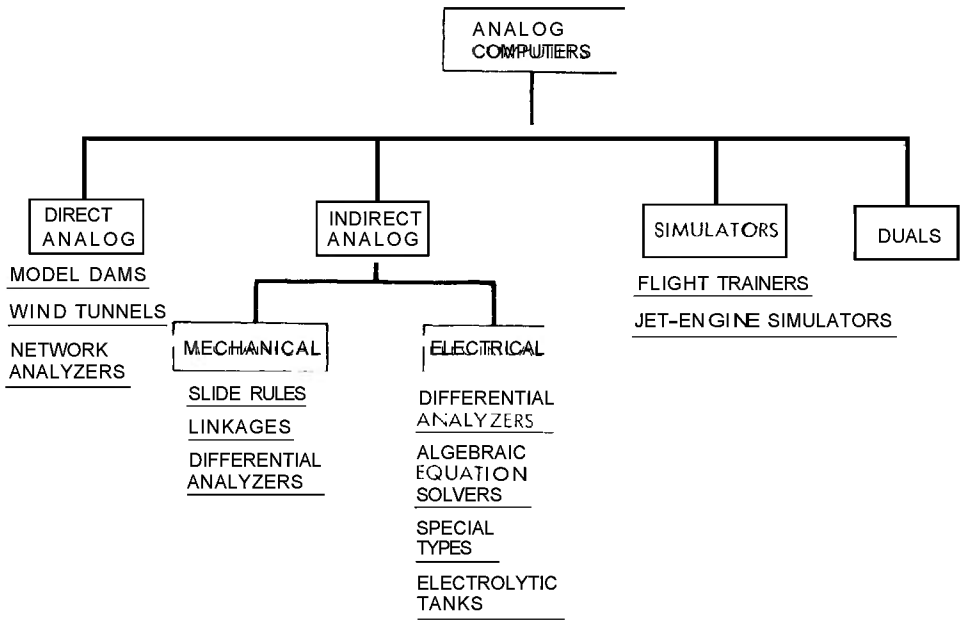


Figure 6-1. Classification of various analog devices.

ages: (1) the time required to solve a problem is short, even for complex sets of differential equations, and (2) once an analog computer has been set up to solve a problem, it can generate solutions for a wide range of system parameters in a very short time.

6-1.1 SOLUTION OF EQUATIONS BY ANALOG MEANS

6-1.2 Common Mechanical and Electrical Analogs

Each computing element in an analog device performs its mathematical operation on a physical quantity (such as an electrical voltage or a shaft rotation) where the physical quantity is equivalent to a variable in the mathematical model for the system. As is discussed in Chapter 1 of this handbook, knowledge of the equivalence on analogies relating different types of physical systems is a valuable asset in the solution of many problems relating to dynamic systems. Table 1-2 summarizes some of the analogies existing between electrical, mechanical, hydraulic, pneumatic, and thermodynamic systems. It is immediately apparent that the same mathematical form describes the dynamic performance of a single-degree-of-freedom system whether it be composed of electrical, mechanical, or fluid elements.

The concept of analogies is strengthened further by comparison of the laws that form the basis for electrical network analysis with the corresponding laws for mechanical systems. Kirchhoff's laws for electrical systems can be stated as:

(1) In any electrical network, the algebraic sum of all currents flowing toward any point is zero at all times, i.e.,

$$\sum i = 0 \qquad (6\text{-}1)$$

(2) The algebraic sum of all voltage drops around any closed circuit is zero at all times, i.e.,

$$\sum e = 0 \qquad (6\text{-}2)$$

where one form or the other is employed depending on the details of the particular system. Analogously, Newton's Third Law for mechanical systems takes one of the following forms:

(1) In translational systems, the algebraic sum of the forces acting at a

point on a body in equilibrium is zero, i.e.,

$$\sum F = 0 \qquad (6-3)$$

(2) In rotational systems, the algebraic sum of tlie torques acting on a body in equilibrium is zero, i.e.,

$$\sum T = 0 \qquad (6-4)$$

The analogous statements for steady-state conditions in other systems follow:

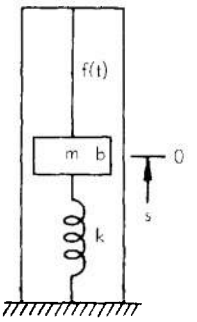
(1) In magnetic circuits, all flux lines must be continuous closed paths.

(2) In hydraulic systems, the law of conservation of mass allows the calculation of velocity distributions.

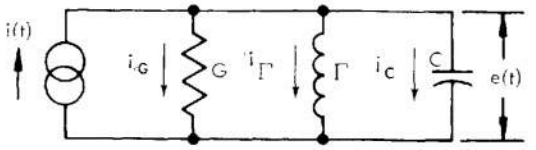
(3) In thermal systems, the law of conservation of energy permits the tal-culation of temperature distributions.

The concept of direct-analog computation can be illustrated by consideration of the example depicted in Fig. 6-2. A spring-supported mass m is constrained to move in a vertical direction (see Fig. 6-2(A)). At time t = 0, the mass is at rest in an equilibrium position, where the displacement s = 0. A force f(t) is applied between the frame of reference and the mass. Viscous friction exists between the mass and its guides. Tlic differential equation describing the system is

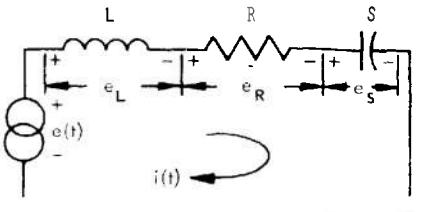
$$f(t) - f_k(t) - f_b(t) = m\frac{d^2s}{dt^2} \qquad (6-5)$$



(A) Spring-supported mass.



(B) Parallel RLC circuit in which current is analagous to force in (A).



(C) Series RLC circuit in which voltage is analogous to force in (A).

Figure 6-2. Analogous mechanical and electrical systems.

where the subscripts k and b identify the force contributed by the spring and the friction in the system, respectively. Substitution of the appropriate expressions for these forces yields, for the simple case of viscous friction and a linear spring,

$$f(t) = m\frac{d^2s}{dt^2} + b\frac{ds}{dt} + ks \qquad (6\text{-}6)$$

If Ey. 6-6 is rewritten in terms of velocity, $v = ds/dt$, it takes the form

$$f(t) = m\frac{dv}{dt} + bv + k\int v\,dt \qquad (6\text{-}7)$$

Application of Kirchhoff's first law to an electrical circuit consisting of a current generator i(t) driving a resistance R, an inductance L, and a capacitance C connected in parallel (see Fig. 6-2(B)) yields, as discussed in Chapter 1,

$$i(t) = i_G(t) + i_\Gamma(t) + i_C(t) \qquad (6\text{-}8)$$

where the subscript G refers to reciprocal resistance, or conductance, and the subscript Γ refers to reciprocal inductance. Substitution of the appropriate expressions relating currents to voltage drops shows that

$$i(t) = C\frac{de}{dt} + Ge + \Gamma\int e\,dt \qquad (6\text{-}9)$$

Comparison of Eqs. 6-7 and 6-9 shows that they are identical in form and, consequently, will have identical mathematical solutions. Therefore, if current is made analogous to force, it follows that voltage is analogous to velocity, capacitance is analogous to mass, conductance is analogous to viscous friction, and reciprocal inductance is analogous to spring stiffness.

Examination of the circuit of Fig. 6-2(C) shows that it also is analogous to the mechanical system of Fig. 6-2(A) if, in this case, voltage is made analogous to force. Application of Kirchhoff's second law to this circuit yields

$$e(t) = e_L(t) + e_R(t) + e_S(t) \qquad (6\text{-}10)$$

Substitution of the expressions for the voltage drops in terms of the current gives

$$e(t) = L\frac{di}{dt} + Ri + S\int i\,dt \qquad (6\text{-}11)$$

where the elastance S is the reciprocal of capacitance. Eq. 6-11 is identical in form with Eqs. 6-7 and 6-9. Consequently, if voltage is made analogous to force, current is analogous to velocity, inductance is analogous to mass, resistance is analogous to viscous friction, and elastance (reciprocal capacitance) is analogous to stiffness.

An awareness of these and other analogies is important to the designer because use of them may allow him to translate a given problem in one physical system, where modeling would be difficult, into terms of another physical system that is more readily adaptable to the construction and testing of low-cost models with variable parameters. Thus, simple electrical networks often can be used to reproduce the dynamic performance of mechanical, acoustical, hydraulic, magnetic, and thermal systems, as well as that of complex systems containing components of several different types.

6-1.3 Block Diagrams

The first step in describing a physical system in a manner suitable for analog computation consists of formulating a block diagram for the system. Initially, the information necessary to specify all the blocks in precise mathematical form may not be available, but such specification must be achieved before it is possible to carry out any type of computer studies of system performance. In a general block-diagram model, mathematical operations or operators are indicated by appropriately labeled boxes or blocks, while connecting lines denote quantities or signals to be acted upon or produced by such operations. The block represents merely the fact that the signal flowing into it is operated on in some fashion to yield the output quantity. The specific operation is indicated by the symbols entered in the block. This method of representation means, fundamentally, that a functional relationship exists between the output and the input quantities. The fact that the operation may not be defined exactly does not invalidate the block diagram as a very powerful tool in system analysis.

Block diagrams have become a very widely used tool in both the analysis and synthesis of engineering systems, Consequently, a commonly agreed-upon symbolic language for depicting block diagrams has evolved as a means of assisting engineers in using this tool as a precise and powerful means of describing system performance'.

A basic rule of block-diagram representation is that all signal flows are unidirectional, as signified by the arrows. This rule can be illustrated by the simple diagrams of Fig. 6-3. The left-hand figure represents the fact that the voltage across a resistance R is equal to the current flowing into it multiplied by the value of the resistance. An attempt to interpret a diagram by considering the flow in a direction opposite to the arrows obviously leads to incorrect results since voltage times resistance does not yield current. While in this simple case an obvious reciprocal relation exists, interpretation of diagrams in this way can lead to incorrect results and in general should be avoided.

Block diagrams can be formulated on the basis of variables expressed in either the time domain or the frequency domain. While frequency-domain notation is somewhat more convenient, it should be recognized that normally used instruments enable one to observe the variables as functions of time rather than frequency. As a result, while it is not correct, it is not uncommon for one to see block diagrams, especially if formulated for study on an analog computer, in which transform notation is employed within the blocks while the signals are indicated as time functions. Fig. 6-4 illustrates both the time-domain and tlic frequency-domain block-diagram notation for several basic operations.

Summation is represented by a circle with an inscribed "X" as shown in Fig. 6-5. If one of the inputs to the summing point is to be subtracted, this is indicated by a darkening of the appropriate quadrant of the circle.

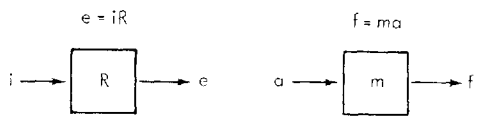


Figure 6-3. Block diagrams of Ohm's law and Newton's second law.

Thus, Fig. 6-5 indicates that $i_3 = i_1 - i_,$. Some workers indicate that a signal is to be subtracted by placing a minus sign beside the arrowhead on that variable.

The electrical schematic of Fig. 6-6 and the corresponding block diagram illustrate the formulation of a block diagram for a simple system. The block diagram can be formulated in a step-by-step fashion from the schematic. Note first that the voltage e_1 is to be considered as the system input and e_2 as the system output. Then note that the voltage e_3, which appears across the resistor R_1, is the difference between e_1 and e_2. This fact is represented in the block diagram by the summing circuit shown on the left-hand side. Next note that the current i is found by multiplying the voltage e3 by $1/R_1$. Then note that i_2 is given by $i - i_1$, as indicated by the second summing circuit. The output voltage e_2 is 1/C times the integral of the current flow into the capacitor C and thus is found by operating on the current i_2 by $\frac{1}{C}\int dt$. The next step is to find the current i_1, which combines in the second summing circuit with the current i to yield i_2. The current i_1 can be found as the voltage across the resistor R_2 multiplied by $1/R_2$. This voltage, in turn, is given as the voltage e, minus the voltage across the inductance L. This latter quantity is given by $L\frac{di_1}{dt}$. All of the currents and voltages in the circuit are now specified in the block diagram and it is complete.

A simple change of the variables in Fig. 6-6 to functions in which time is replaced by the Laplace transform variable s, and a corresponding change is made in the notation employed to indicate differentiation and integration, according to Fig. 6-4, enables one to convert the block diagram of Fig. 6-6 to the alternate form shown in Fig. 6-7.

While diagrams of these types permit one to see easily the interrelationships existing in a system and are useful as a first step in developing an analog-computer set-up for a system, it is desirable for mathematical analysis to reduce a complex diagram of this type into one containing only a single block. This block then represents the transfer function for the system. Inversely, a block diagram may have been drawn initially in terms

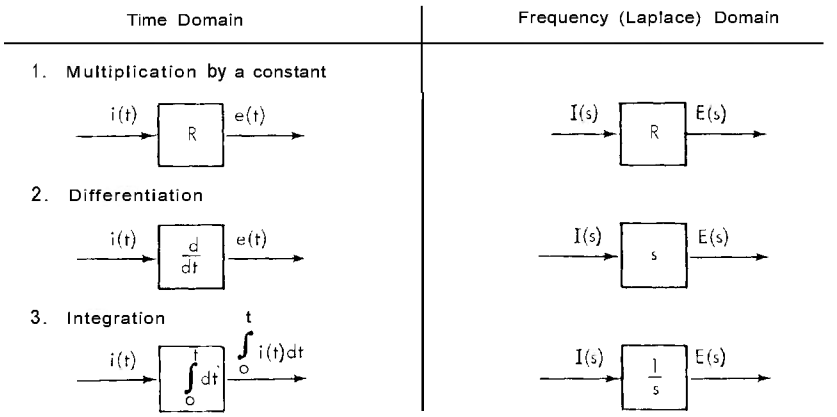


Figure 6-4. Block-diagram operations.

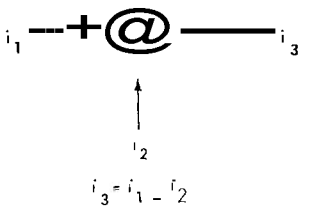


Figure 6-5. Symbol for a summing point.

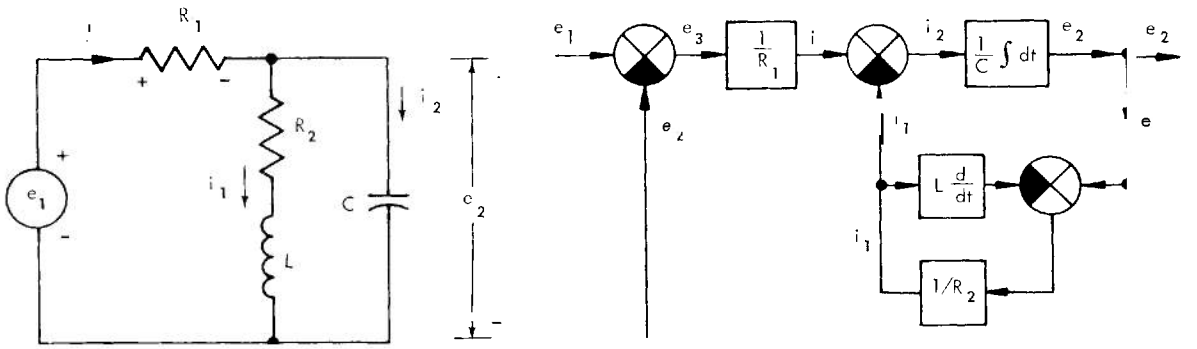


Figure 6-6. Series-parallel circuit and its block diagram.

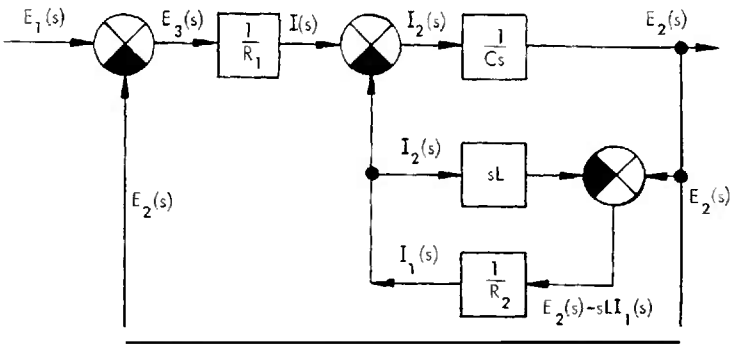


Figure 6-7. Block diagram of the system of Fig. 6-6 given in the Laplace domain.

of complex transfer functions and it is desired to recast it in terms of the basic operations performed by individual analog computing elements in preparation for study of the system on a computer. For these purposes, a group of rules have been developed for manipulating block diagrams. Some of the more important of these rules are illustrated in the paragraphs which follow.

Rule 1. Superposition.

The principle of superposition, which applies only to linear systems, states that the response of a system to several inputs applied simultaneously is equal to the sum of the responses to the inputs applied individually. Consequently, the response of an element to an individual forcing function can be found by considering all other inputs zero. On this basis, the two diagrams of Fig. 6-8, in which E_1 is the part of E due to I_1 and E_2 is part of E due to I_2, are equivalent.

Rule 2. Cascaded elements.

The order of linear cascaded elements may be interchanged or they may be combined by multiplying the functions of the independent elements, as indicated in Fig. 6-9.

Rule 3. Moving an element forward or backward past a summation point.

An element may be moved forward past a summation point if its reciprocal is inserted in each leg of the other inputs to the summation (see Fig. 6-10) or backward past a summation point, against the direction of flow, provided it is inserted in every leg that represents an input to the summation. (See Fig. 6-11.)

Rule 4. Moving an element forward or backward past a pickoff point.

An element may be moved forward past a pickoff point provided it is placed in each branch leading away from the pickoff point (see Fig. 6-12). Conversely, an element may be moved backward past a pickoff point provided its reciprocal is inserted in all branches other than the one in which it was originally located (see Fig. 6-13).

Rule 5. Combination of parallel paths.

Parallel paths lying between a pickoff point and a summation point may be combined into a single element, provided that there are no additional pickoff or summation points in either path. The resulting single element is represented by the sum of the elements in the individual paths. (See Fig. 6-14.)

Rule 6. Removal of a feedback loop.

A feedback loop with a forward transfer function $F_1(s)$ and a feedback transfer function $F_2(s)$ can be replaced by a single element equal to $F_1(s)/[1 \pm F_1(s)F_2(s)]$. The minus sign is used when the feedback is additive, the plus sign when the feedback is subtractive. (See Fig. 6-15.)

The application of these rules is illustrated by a reduction of the block diagram of Fig. 6-7 into one containing a single element. As a first step, the feedback path containing sL and $1/R_2$ is reduced to a single element by application of Rule 6. Here, $F_1(s)$ corresponds to $1/R_2$ and $F_2(s)$ to sL. As a next step, the feedback path containing the new element $\frac{1}{R_2 + sL}$ in the feedback path and the element 1/Cs in the forward path is reduced

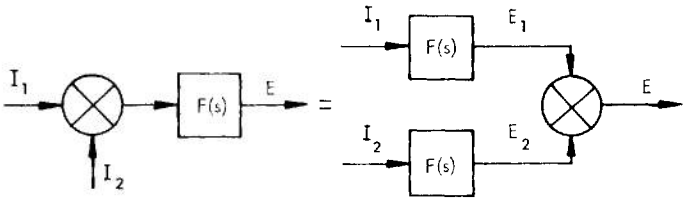


Figure 6-8. Equivalent configurations based upon superposition.

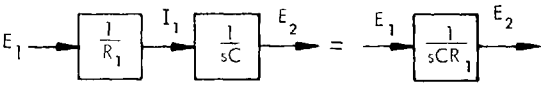


Figure 6-9. Combination of cascaded elements.

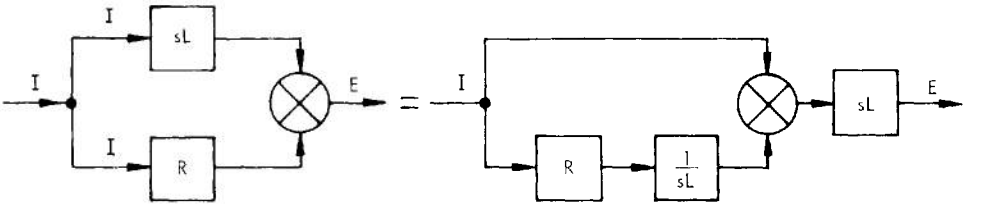


Figure 6-10. Movement of an element forward past a summation point.

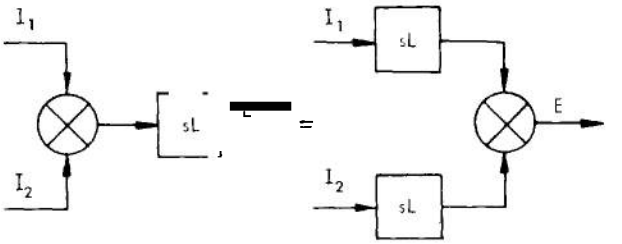


Figure 6-11. Movement of an element backward past a summation point.

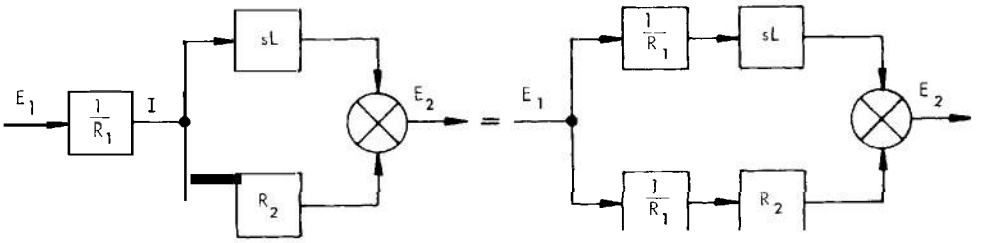


Figure 6-12. Movement of an element forward past a pickoff point.

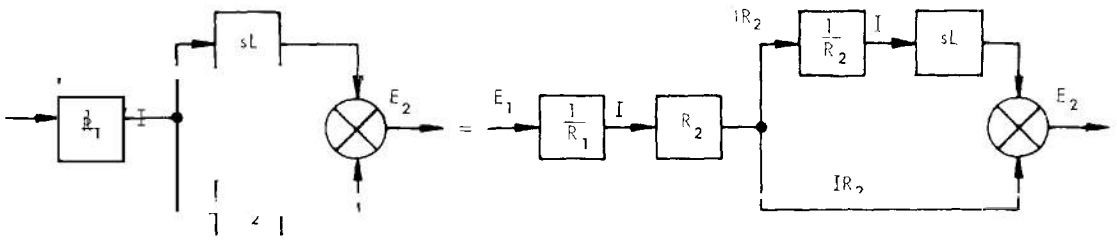


Figure 6-13. Movement of an element backward past a pickoff point.

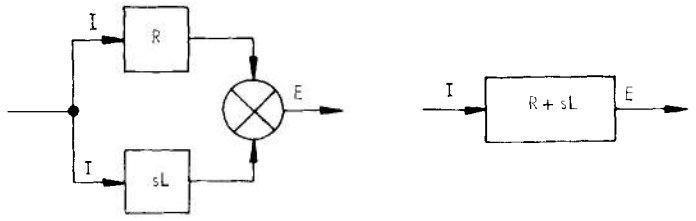


Figure 6-14. Combination of parallel paths.

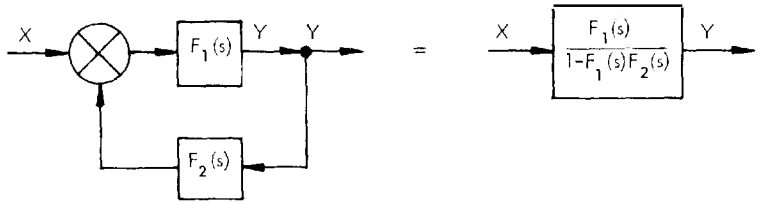


Figure 6-15. Removal of a feedback loop.

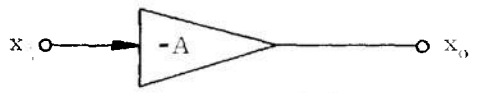
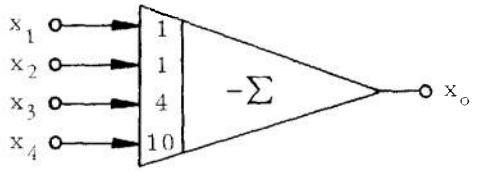
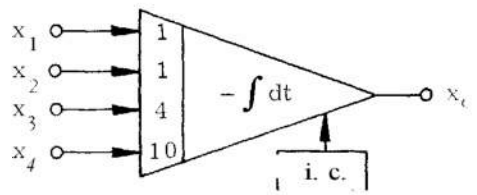
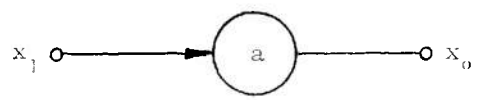
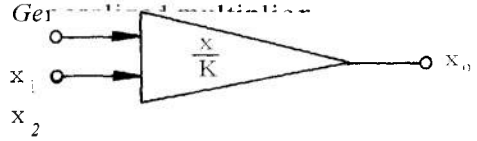
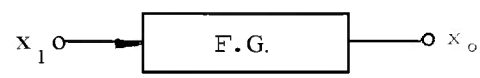
to a single element. The cascaded elements in the forward path of Fig. 6-16(B) are then combined according to Rule 2. Finally, the feedback system of Fig. 6-16(C) is reduced in accordance with Rule 6 to a system containing a single element. The function in the block of this final diagram is the transfer function for the system.

6-1.4 Analog Computer Diagrams

Diagrams drawn to indicate how an analog computer should be set up to solve a particular equation are closely related to the general block diagrams discussed in the preceding paragraph. However, a specialized set of symbols has been developed to conform with the operations performed by the actual elements of the computer. Although these symbols are not completely standardized, those shown in Table 6-1 are representative. It should be noted that this listing does not contain a symbol for differentiation. At first, this might appear to be a serious omission inasmuch as analog computers are used extensively in obtaining the solution to differential equations. In practice, however, for reasons that will be discussed later, it is more feasible to employ the process of integration than the process of differentiation.

Although important applications of analog computation have been made in the solution of algebraic equations and partial differential equations, the engineer concerned with the design of fire control systems is interested in analog techniques chiefly as they apply in the study of systems that can be described in

TABLE 6-1. SYMBOLS FOR ELECTRONIC ANALOG COMPUTING ELEMENTS.

Operation and symbol	Remarks
High-gain inverting amplifier  $x_o = -Ax_i$ where $\lvert A \rvert \rightarrow \infty$	The high-gain amplifier represents the basic building block of the electronic differential analyzer
Summing amplifier  $x_o = -(x_1 + x_2 + 4x_3 + 10x_4)$	The constants by which the various inputs are multiplied are typical of those normally provided in a summer or integrator
Summing integrator  $x_o = -\int (x_1 + x_2 + 4x_3 + 10x_4)\, dt + i.c.$	Each of these units provides a sign reversal. The initial condition is indicated in the box labeled i.c.
Coefficient multiplier  $x_o = ax_1$ where $0 \leq a \leq 1$	The coefficient a is manually set before a solution is run
Ger[illegible] multiplier  $x_o = x_1 x_2 / K$	This unit provides for multiplication of one dependent variable by another
Function generator  	The abbreviation F.G. may be replaced by a simple graph of the function

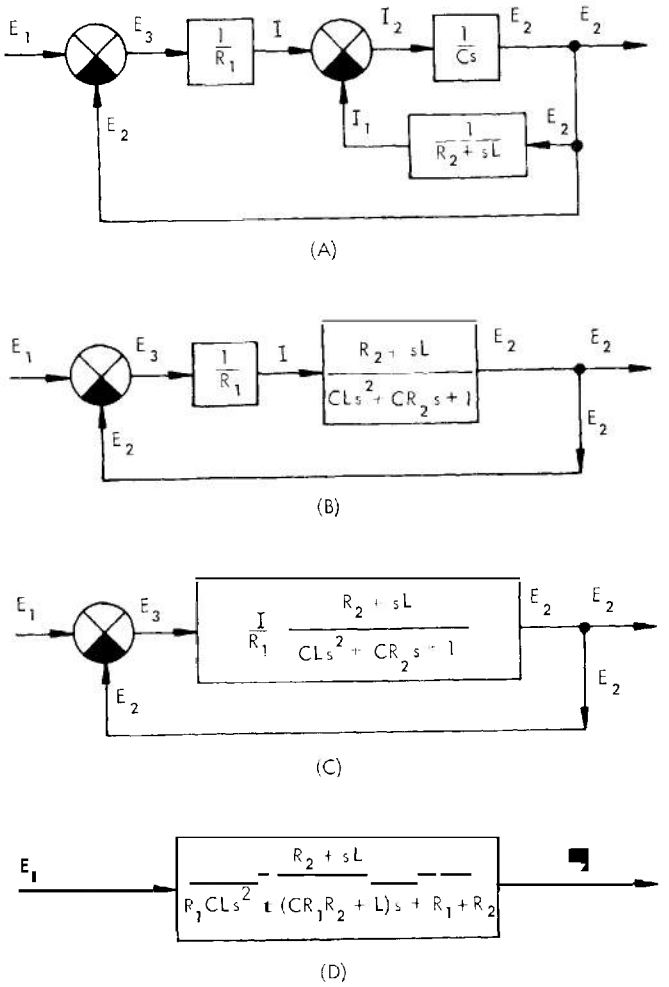


Figure 6-16. Steps in the reduction of the block diagram of Fig. 6-7.

terms of ordinary differential equations and in the mechanization of equations as an operational part of the fire control computer. A discussion of the basic technique for solving ordinary differential equations by analog means will serve to clarify further the use of block diagrams and to introduce the elements used in analog computers.

6-1.5 Analog Solution of Differential Equations

In order to introduce the techniques used for solving ordinary differential equations, consider as a simple example the differential equation

$$\frac{dx}{dt} + x^2 = 1 \qquad (6\text{-}12)$$

with the initial condition specified that $x = 0$ at $t = 0$. If the possibility of integration is presupposed, and if the derivative dx/dt is assumed to be known, the function x can be obtained, as indicated symbolically in Fig. 6-17(A) where use has been made of the symbols defined in Table 6-1. In an attempt to solve Eq. 6-12 the difficulty is encountered that the function dx/dt is not given explicitly. What is given is a relation involving dx/dt and x. If, for the moment, the assumption is made that the function x is known, the differential equation can be solved for dx/dt to yield

$$\frac{dx}{dt} = 1 - x^2 \qquad (6\text{-}13)$$

The operations on the right-hand side of

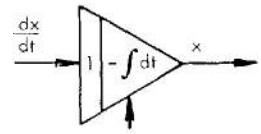
Eq. 6-13 are indicated symbolically by Fig. 6-17(B). The two symbolic representations given in Figs. 6-17(A) and 6-17(B) can be combined to yield the diagram shown in Fig. 6-18. The -1 represents a fixed voltage obtained from a reference source. The diagram given in Fig. 6-18 represents a closed-loop system that by its nature is forced to produce the desired solution, provided that the operations are performed in an ideal manner. Inasmuch as an integrator also serves as a summing unit, it is unnecessary to provide a summing unit as an individual component in the complete system. Fig. 6-19 incorporates this simplification.

The technique for solving a first-order equation is readily extended for the solution of an nth-order, linear, constant-coefficient differential equation of the generalized form

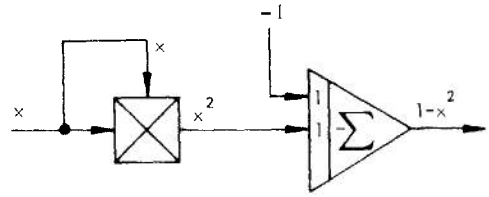
$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} \qquad + a_1 \frac{dy}{dt} + a_o y = f(t) \quad (6\text{-}14)$$

First, the highest derivative is separated by putting the equation in the form

$$\frac{d^n y}{dt^n} = -\frac{1}{a_n}\left[a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_o y - f(t)\right] \quad (6\text{-}15)$$



(A) Diagram representing the integration of dx/dt



(B) Diagiam representing the operations in the expression $1-x^2$

Figure 6-17. Basic diagrams associated with the analog solution of the differential equation $(dx/dt) + x^2 = 1$.

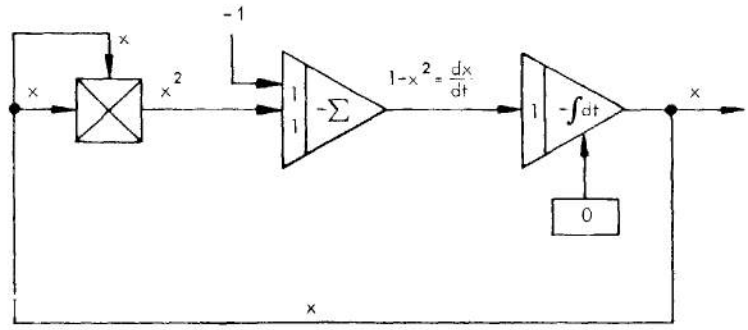


Figure 6-18. Diagram combining Figs. 6-17(A) and 6-17(B) to give the solution of the differential equation $(dx/dt) + x^2 = 1$

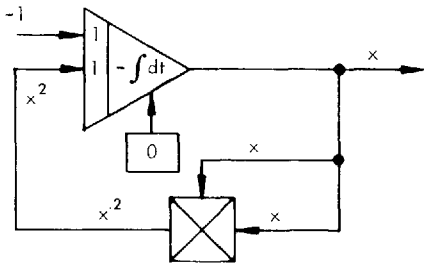


Figure 6-19. Simplification of Fig. 6-18.

Then, the availability of this highest derivative is assumed, and it is integrated n times to yield y. The various derivatives are multiplied by the appropriate coefficients, summed, added to f(t), and finally multiplied by $-1/a_n$ to give the highest derivative d^ny/dt^n. Because the signs of outputs of successive integrators alternate, care must be taken to see that each term is added with the correct sign. This technique can be illustrated by reference to the setup diagram of Fig. 6-20 for solving the third-order equation

$$a_3 \frac{d^3y}{dt^3} + a_2 \frac{d^2y}{dt^2} - a_1 \frac{dy}{dt} + a_0 y = f(t) \qquad (6\text{-}16)$$

In this setup, the assumption has been made that all the coefficients in the equation are positive and less than unity. The occurrence of negative coefficients would require the addition or removal of inverting amplifiers, and coefficients larger than unity would require the insertion of amplifiers with gains greater than unity.

6-1.6 TYPES OF ANALOG COMPUTERS

If a physical system is to be useful as an analog, its performance must be analogous to that of the mathematical equations it is to simulate and it must be possible to measure the performance of the physical system accurately and conveniently. Although a great variety of physical systems ranging from rubber membranes to large assemblages of sophisticated electronic units have found use as analog computers, the types of most importance as components of fire control systems or as design aids in the development of fire control systems are either mechanical, electromechanical, or electronic computers. Each type possesses certain general advantages in speed, accuracy, or reliability, and a specific type may be best suited to performing a specified mathematical operation. Consequently, it is common to see more than one type of computing element in a single complete computer. Although four- to five-place resolution and three- to four-place accuracy are typical of analog-computer performance in most simple operations of algebra and the calculus, the error resulting in the solution of an overall closed-loop system of medium complexity may be closer to 1 percent in a typical situation.

6-1.7 Electromechanical and Electronic Analog Computers

Any analog computer employing voltages and mechanical shaft angles as analog quantities falls into the electromechanical category. On this basis, nearly all computers, except the high-speed repetitive electronic type, fall into this category. In a somewhat more restrictive sense, the term electromechanical computer applies to computers in which a relatively large number of instrument servos is used to perform such operations as multiplication and function generation. Although high accuracy can be achieved with properly designed electromechanical computing elements, these units have a restricted speed of response and require somewhat more specialized maintenance than purely electronic elements. As a result, the tendency in the design of general-purpose analog computers is away from the use of servos and other electromechanical units. However, where a large number of nonlinear functions must be generated and a large number of multiplications performed, such as in flight trainers, servos may offer the best overall solution. Servo multipliers, dividers, and function generators are discussed in par. 6-4 through par. 6-4.15.

The term electronic analog computer generally refers to an analog computer for solving ordinary differential equations in which most, if not all, of the computation is done by purely electronic means. Such computers offer the advantage of much greater

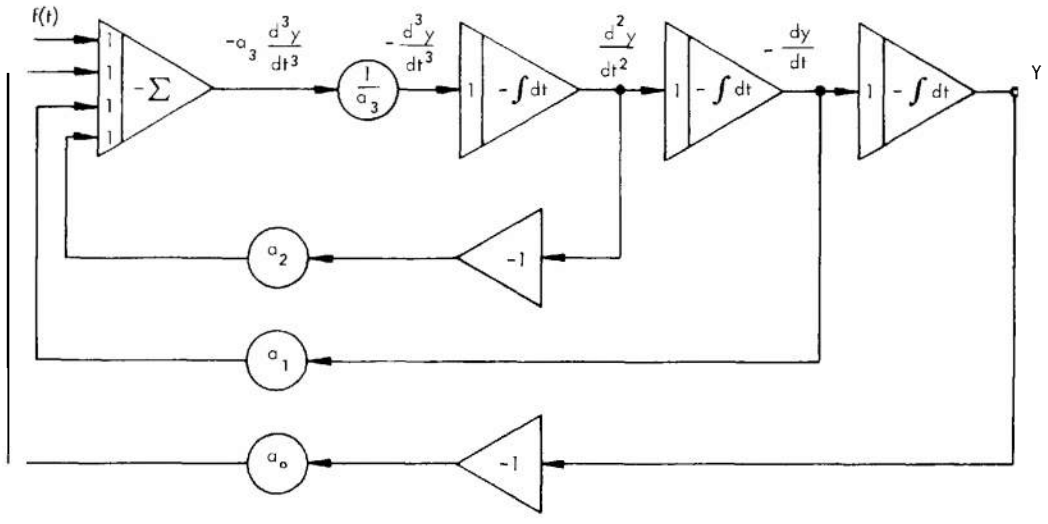


Figure 6-20. Setup for the solution of a linear third-order differential equation.

speed than a mechanical or electromechanical computer. In fact, all-electronic machines are sometimes designed to permit repeating the solution to a problem 10-60 times per second. Specific computer components are described under par. 6-3 through par. 6-3.9.

6-1.8 A-C Type

In an a-c computer a carrier voltage, usually 400 cps, is used throughout the machine. Here, the amplitude of an analog quantity is represented by the amplitude of the a-c voltage and the sign of the quantity by whether this voltage is in phase or 180 degrees out of phase with a reference voltage. This a-c suppressed-carrier technique is advantageous when data are transmitted over great distances and when vector transformations are required. The a-c computer can make use of some of the same components as used in d-c computer, such as summing circuits and coefficient potentiometers, provided careful attention is given to phase shift between computer components. For example, the addition of two a-c voltages that are slightlyout of phase can lead to considerable error. However, integration cannot be performed with a high-gain amplifier and an RC feedback network. Instead, a velocity servo is usually employed as an integrator in a-c computers.

A-c signals can be used to drive two-phase servomotors directly, to excite synchros, and to excite induction resolvers employed to perform trigonometric functions. The accuracy of an induction resolver is limited by magnetic uniformity and residual voltages, as well as by the difficulties of residual noise components and the problems of phase shiftthat are common to all inductive components. Nonetheless, the accuracy achieveable in a well-designed machine can be comparable with that achieved in a d-c computer.

In computers employing an alternating-current voltage as the analog quantity, transformers can be used for performing addition (see Fig. 6-21). Ifaccurate results are to be obtained, the transformers must be nearly ideal.

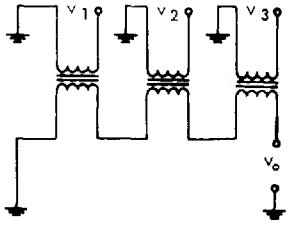


Figure 6-21. Transformer summing circuit.

6-1.9 D-C Type

In a d-c computer, all calculations are carried out with direct currents or voltages representing the analog quantities. Shielding is used extensively to help minimize noise and interference troubles in the handling of low-level signals, with the shielding carefully designed to avoid circulating ground currents. The amplifiers used in a direct-current computer are perforce direct-coupled, leading to the requirements for response from zero cps to several thousand cps for real-time computers (Ref. par. 6-3.1) and to several hundred thousand cps for compressed-time applications. The drift characteristics of the amplifiers must also be carefully controlled. Drift (change in the zero-signal-reference level of the output) arises in d-c amplifiers because of changes in power-supply voltages and heater voltages, changes in the characteristics of vacuum tubes or transistors, and changes in component values resulting from variations in temperature or humidity. Ideally, the input impedance of a computer amplifier should be infinite, its output impedance zero, and its gain infinite. For most applications, these characteristics are essentially achieved in modern vacuum-tube computer amplifiers and rapid strides have been made in recent years toward obtaining nearly as good characteristics from transistorized amplifiers. The strong appeal of transistorized design is offset by the inherently low input impedance and high output impedance characteristics of transistors, plus their susceptibility to temperature variations. Fortunately for the designer, so much work has been done in developing d-c amplifiers that an ample choice from existing designs is available in military quality.

The d-c analog computer is basically a real-time device, but can be used on a compressed or extended time scale.

6-1.10 Electrical Analog Computers

The use of electrical networks as analog computers has found wide application. Frequently, after experience with the technique has been gained, it is possible to arrange the electrical elements of resistance, inductance, and capacitance to mechanize complicated physical systems without the intermediate step of formulating equations for the given system. The flow of electricity in the network is a useful analog in structural design problems, in establishing fluid flow in pipes, in predicting neutron densities in a reactor, and in a range of similar functional applications.

Configurations of an electrolytic tank or a conducting sheet have currents that satisfy, under suitable conditions, various forms of the Laplace and Poisson partial differential equations. By these equations, it is possible to describe a great number of physical phenomena in the fields of electrodynamics, fluid dynamics, thermodynamics, and related problems.

Voltages can be summed electrically in the simple resistance network shown in Fig. 6-22. The output voltage v_o is given by the expression

$$v_o = \frac{R_2}{R_1 + R_2} v_1 + \frac{R_1}{R_1 + R_2} v_2 \qquad (6\text{-}17)$$

Although this circuit can be extended to permit summing n voltages, it has the disadvantage that the resistance across which the output voltage v_o is developed influences the result obtained.

To perform electrical differentiation, a voltage proportional to the derivative of a second voltage can be generated by a resistance-capacitance (RC) circuit or a resistance-inductance (RL) circuit (see Fig. 6-23). For many applications, a relatively crude approximation of the derivative is sufficient

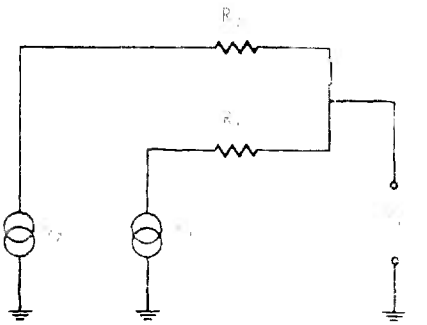


Figure 6-22. Simple resistive summing circuit.

and these simple circuits suffice. The accuracy of the RC differentiator is improved by using small values of R and C (short time constant), but this leads to a small output voltage. Likewise, the RL differentiator requires a large R and a small L for high accuracy.

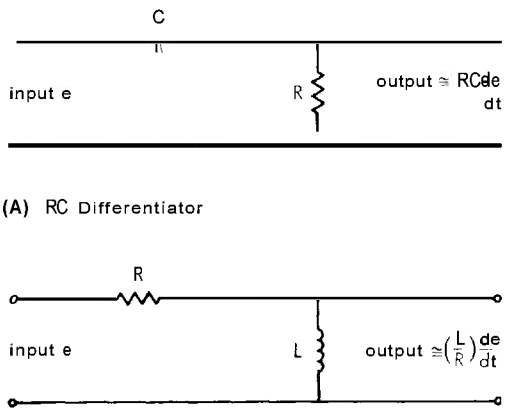


Figure 6-23. Differentiating circuits.

6-1.11 Mechanical Analog Computers

Mechanical analog computers generate problem solutions primarily by mechanical means. Although they have been displaced almost completely by electronic or electromechanical computers for general-purpose applications, they are still widely used as special-purpose computers. The accuracy achievable with the best mechanical computing elements exceeds that obtainable with electronic elements, and a mechanical computer can be reliable even when operated in an unfavorable environment.

Specific mechanical computing elements -- based upon the use of cams, linkages, and gears -- are described in par. 6-4 and include summation devices, integrators, multipliers and dividers, resolvers, and function generators.

In mechanical computers, three types of error contribute inaccuracies to the mechanization of mathematical relationships:

(1) Theoretical errors due to the inherent approximations of the geometry.

(2) Fabrication errors due to manufacturing tolerances and necessary clearances.

(3) Slip errors where friction drives or belt-connected units are required.

6-2 ANALOG SOLUTION OF EQUATIONS

6-2.1 BASIC SOLUTION METHODS

6-2.2 Ordinary Differential Equations[2]

The analog technique for the solution of a simple, first-order differential equation is presented in par. 6-1.5. This technique is thenextended to the solution of a generalized nth-order, linear, constant-coefficient differential equation. Solution of such an equation (Ref. Eq. 6-14) requires only integration, generator of f(t), and the operations of summation and multiplication by constant coefficients because the function defining the highest derivative of the dependent variable is a linear function of f(t) and the derivatives of y.

Analog computers are of special importance in solving ordinary differential equations. General-purpose computers of this type are called differential analyzers. The solution of ordinary differential equations by analog means is presented in the paragraphs which follow. The solution of other types of equations by analog methods is covered in succeeding paragraphs. These include the solution of simultaneous linear algebraic equations, nonlinear algebraic equations, and partial differential equations.

As a specific example of the solution of an ordinary differential equation by analog methods, consider the equation

$$69.4\frac{d^2y}{dt^2} + 9.17\frac{dy}{dt} + y = 20 \qquad (6\text{-}18)$$

This leads to the analog-computer setup shown in Fig. 6-24. However, such a setup is not unique because Eq. 6-18 can be rewritten in the form

$$\frac{d^2y}{dt^2} + \frac{9.17}{69.4}\frac{dy}{dt} + \frac{1}{69.4}y = \frac{20}{69.4} \qquad (6\text{-}19)$$

Eq. 6-19 could be solved with the alternative computer arrangement shown in Fig. 6-25.

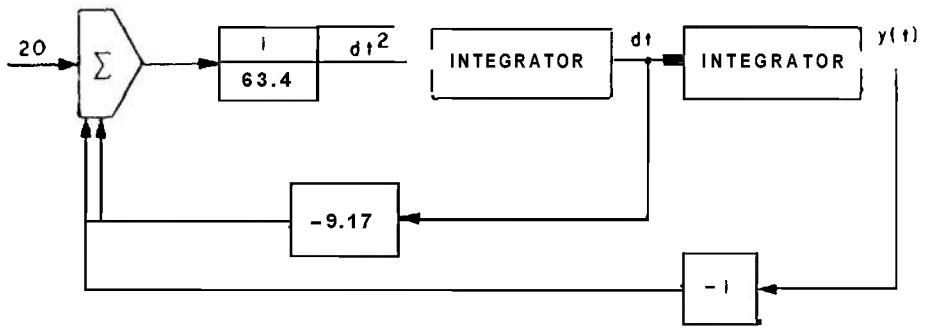


Figure 6-24. Analog-computer setup for a simple linear differential equation.

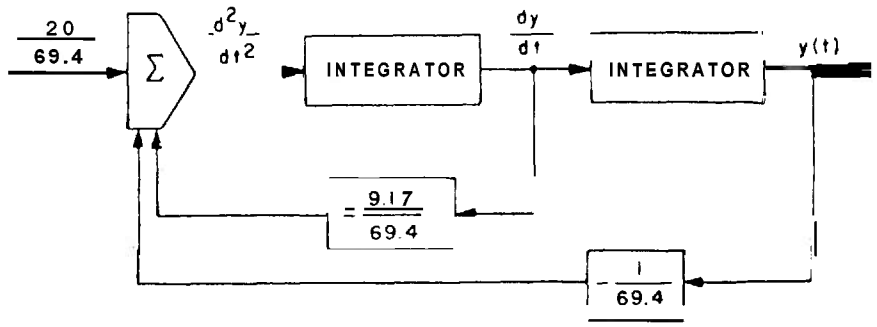


Figure 6-25. Alternative analog-computer setup for a simple linear differential equation.

The solution of equations outside this special class requires the use of nonlinear operations. Of these operations, the most important are the multiplication of two variables and the generation of functions of one variable. Some problems may require generation of functions of two or more variables, but often these are built up, at least approximately, from simpler operations. Fortunately, most of the problems encountered in practice can be handled in this way because, as discussed in par. 6-2.17, generation of functions of two or more variables is difficult.

Equations that contain time-varying coefficients may be considered one step more complex than linear, constant-coefficient, ordinary, differential equations. The equation

$$\frac{d^2y}{dt^2} + C\frac{dy}{dt} + f(t)y = 1 \qquad (6\text{-}20)$$

where C is a constant, is a simple example of this type and can be solved with the analog-computer setup illustrated in Fig. 6-26. Nonlineal-. differential equations, of which the following equation is a simple illustration, represent a still more complex type.

$$\frac{d^2y}{dt^2} + f(y)\frac{dy}{dt} + y = 1 \qquad (6\text{-}21)$$

Although the mathematical structure of Eqs. 6-20 and 6-21 is quite different, essentially the same computer operations are required in both cases, as can be seen by comparison of Fig. 6-26 with Fig. 6-27, which is an analog-computer setup diagram for solution of Eq. 6-21.

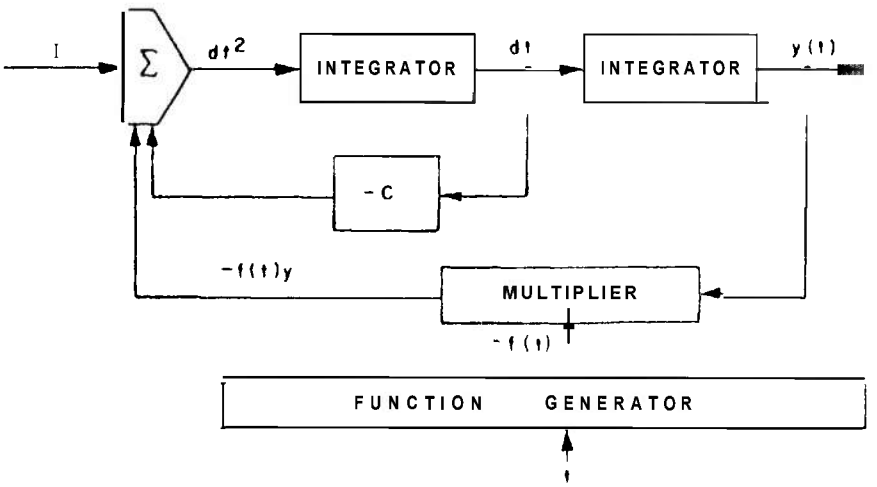


Figure 6-26. Analog-computer setup for a simple linear, time-varying differential equation.

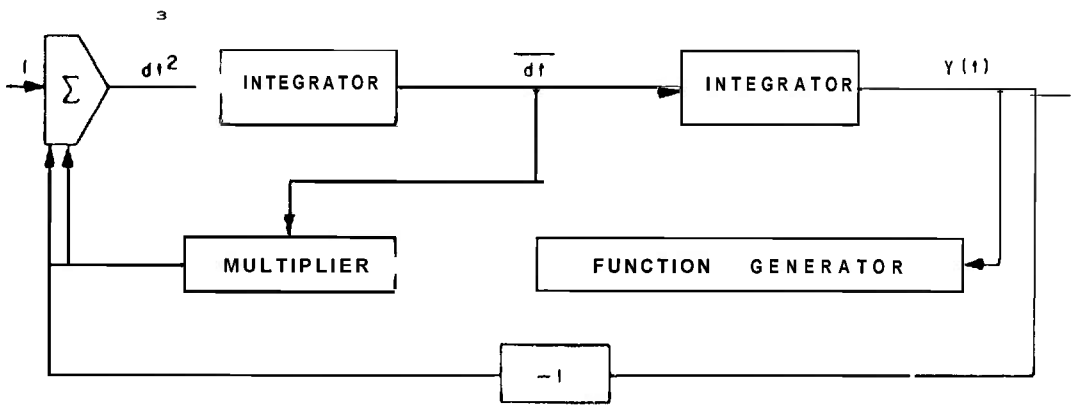


Figure 6-27. Analog-computer setup for a simple nonlinear differential equation.

6-2.3 Simultaneous Linear Equations[3]

The problem of finding the unknown x's that satisfy a set of simultaneous equations of the generalized form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_1 &= 0 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_2 &= 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_n &= 0 \end{aligned} \qquad (6\text{-}22)$$

where the a's and b's are known constants, and the equivalent problem of inverting matrices, arises frequently in engineering and science. In 1878, Lord Kelvin proposed a machine for solving such equations, but apparently he never built it. J. B. Wilbur, at the Massachusetts Institute of Technology, built several improved versions of Kelvin's machine in the 1930's. With the increase in interest in electrical analog computers in the 1940's, attention turned to electrical analog methods. Still more recently, with the widespread use of digital machines and the demand for techniques capable of handling

several hundred equztions, most problems of tliis class now are solved digitally. In certain applications, however, that involve a maximum of 12 to 15 simultaneous equations, some convenience may be gained by use of' analog techniques.

Two basically different methods exist for tlie analog solution of a set of linear simultaneous algebraic equations:

(1) Iterative, or successive-approximation, methods.
(2) Closed-loop or direct-solution methods.

In schemes employing the iterative method, the a and b coefficients are represented on groups of potentiometers. With the potentiometers representing the coefficients a_{11} through a_{1n} and b_1 switched into the circuit, the potentiometer representing x_1 is adjusted to give a null on an indicator. The second bank of potentiometers is then switched into the circuit and the potentiometer representing x_2 is adjusted to satisfy the second equation. This process is continued until each equation in turn has been switched into tlie circuit ancl tlie corresponding x potentiometer lias been adjusted for a null. The process is then repeated until a set of x's is obtained that yields a balance for each equation of tlic set. Tlie values of the x's are then read directly from the potentiometers.

Direct solution of a set of simultaneous algebraic equations by analog methods can be accomplished by employing feedback across high-gain computing amplifiers. The arrangement shown in Fig. 6-28 is for solving only two equations, but may be extended directly for solving systems of more equations. The desired a's and b's are set into the appropriate potentiometers, antl the b potentiometers are excited from a fixed voltage E. The output voltages e_1 and e_2 of the amplifiers represent the values of the unknowns x_1 and x_2. Because the gain characteristic of the amplifiers is a function of frequency, the use of' feedback around the amplifiers may lead to instability in circuits of the type shown. However, straightforward means exist for circumventing this difficulty.

6-2.4 Nonlinear Algebraic Equations[3]

A nonlinear algebraic equation that occurs frequently in scientific work has the generalized form

$$a_n z^n + a_{n-1} z^{n-1} + \cdots + a_2 z^2 + a_1 z + a_0 = 0 \quad (6\text{-}23)$$

where $a_n, a_1, a_2, \ldots, a_{n-1}$, antl a_0 are constants. General solutions to higher than fourth-degree polynomial equations cannot be obtained and considerable effort has been

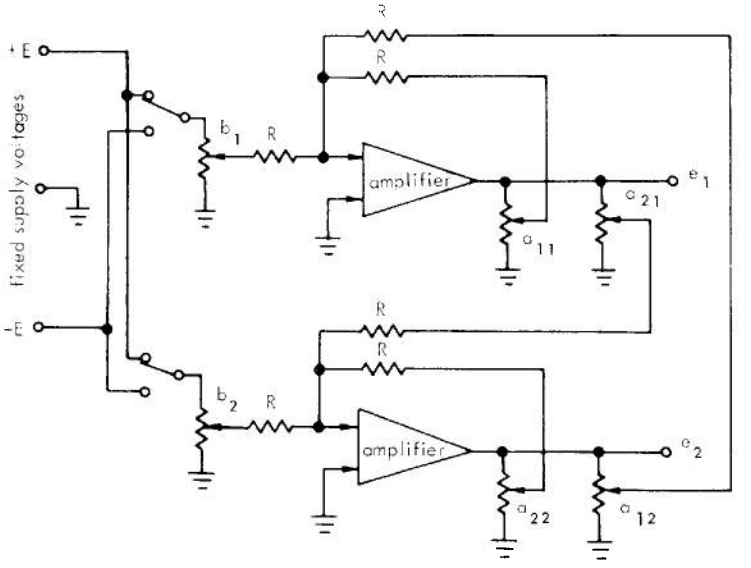


Figure 6-28. Circuit for a closed-loop solution of a pair of simultaneous equations.

devoted to developing machine methods for solving these equations. Mechanical, hydraulic, and electrical analog schemes have been used to a limited extent. With the development of digital techniques, however, the worker desiring to solve any number of polynomial equations usually uses a digital computer.

Basically, the solution of a polynomial equation requires generation of the required powers of the variable z, multiplication of these quantities by coefficients, and summation of the resultant terms with the constant a_n. The variable z is swept through a range of values, and a root occurs whenever the sum is zero. If the coefficients and the roots are all real, potentiometers and simple summing circuits are sufficient to perform the required operations. Complex roots can be handled by converting the original equation in z into a pair of simultaneous equations by the substitution of $z = x + iy$ or by conversion of the equation into trigonometric form by the substitution of $z = r(\cos\theta + i \sin\theta)$.

Numerous variations of these techniques -- as well as a number of other schemes -- have been proposed, but they have received little attention since digital methods have become widely available.

6-2.5 Partial Differential Equations[3]

The solution of partial differential equations by analog means is based upon the same concept as the solution of ordinary differential equations; namely, that the behavior of a variety of physical systems can be expressed by mathematical equations of the same form. Partial differential equations are generally more complex and difficult to solve analytically than ordinary differential equations, and a great deal of attention has been given to the development of analog methods of solution. However, generalized partial-differential-equation computers do not exist. Equipment must be tailored to a specific problem or narrow class of problems, and the tendency has been for each group of analysts to construct its own analogs.

Partial differential equations that are encountered frequently in scientific work and that have been investigated by analog techniques include the following:

a. Laplace's equation:

$$\nabla^2\phi = 0 \tag{6-24}$$

where

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} \tag{6-25}$$

b. Diffusion equation:

$$\nabla^2 f = K\frac{\partial\phi}{\partial t} \tag{6-26}$$

c. Wave equation:

$$\nabla^2 f = K\frac{\partial^2 f}{\partial t^2} \tag{6-27}$$

d. Poisson's equation:

$$\nabla^2\phi = f(x, y, z) \tag{6-28}$$

e. Wave equation with damping:

$$\nabla^2 f = K_1\frac{\partial^2\phi}{\partial t^2} + K_2\frac{\partial f}{\partial t} + K_3 f \tag{6-29}$$

f. Equations from theory of elasticity:

$$\nabla^4 f = 0 \tag{6-30}$$

$$\nabla^4 f = K\frac{\partial^2 f}{\partial t^2} \tag{6-31}$$

and

$$\nabla^4 f = K_1\frac{\partial^2\phi}{\partial t^2} + K_2\frac{\partial f}{\partial t} \tag{6-32}$$

where

$$\nabla^4 f = \frac{\partial^4 f}{\partial x^4} + 2\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} \tag{6-33}$$

Conductive solids, conductive liquids, resistance networks, resistance-reactance networks, electronic analog computers of the type used to solve ordinary differential equations, and nonelectric schemes, such as hydrodynamic analogs, elastic-sheet analogs, and soap films, have been used for the analog solution of partial differential equations.

6-2.6 SCALE FACTORS AND TIME SCALES[2]

After the basic block diagram for the representation of a physical system has been determined, scale factors must be assigned that relate (1) the amplitudes of voltages within the computer to the magnitudes of the corresponding mathematical variables in the differential equation to be solved and (2) the time required for an event to take place in the computer (real time) to the time required for it to occur in the problem being investigated (problem time).

In an electronic computer, the relationship between an equation variable y and its corresponding voltage v_1 can be written

$$v_1 = a_1 y_1 \tag{6-34}$$

In general, the scale factor a_1 is a dimensional constant since y_1 and v_1 usually have different dimensions. For example, if y_1 is a distance measured in feet, and 5 volts of v_1 correspond to 1 foot of y_1, Eq. 6-34 becomes

$$v_1 = (5v\ ft)y_1 \tag{6-35}$$

Because a computer contains more than one variable, the relationships between scale factors of different variables must be taken into account in the operation of a computer.

Consideration of the block diagram for the completely generalized analog-computing component shown in Fig. 6-29 leads to a method for handling scale factors that is applicable to any analog element the analyst may encounter. The output v_o of this component, as a function of the inputs v_i and machine time τ, can be expressed as

$$v_o = g(v_1, v_2, \ldots, v_n, \tau) \tag{6-36}$$

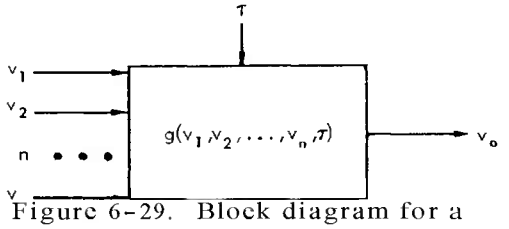


Figure 6-29. Block diagram for a generalized computing component.

The corresponding relationship in the physical situation can be expressed as

$$y_o = f(y_1, y_2, \ldots, y_n, t) \tag{6-37}$$

Computer variables and the corresponding problem variables can be related by a group of equations each of which has the form of Eq. 6-34, i.e.,

$$\left.\begin{aligned} v_o &= a_o y_o \\ v_1 &= a_1 y_1 \\ &\vdots \\ v_n &= a_n y_n \\ \tau &= a_t t \end{aligned}\right\} \tag{6-38}$$

Substitution of the relationships of Eq. 6-38 into Eq. 6-36 yields

$$a_o y_o = g(a_1 y_1, a_2 y_2, \ldots, a_n y_n, a_t t) \tag{6-39}$$

or

$$y_o = \frac{1}{a_o}\, g(a_1 y_1, a_2 y_2, \ldots, a_n y_n, a_t t) \tag{6-40}$$

For any particular element, the scale factors can be evaluated by comparison of Eq. 6-40 with Eq. 6-37. Multiplication of a variable y_1 by a dimensionless constant k, as expressed by

$$y_o = k y_1 \tag{6-41}$$

represents the simplest situation involved. The analog equivalent of this process is shown in Fig. 6-30, where the symbols above the lines denote equation variables and those below the lines denote computer variables. The analog component introduces a fixed gain c and gives an output

$$v_o = c v_1 \tag{6-42}$$

Substitution of the appropriate relationships of Eq. 6-38 into Eq. 6-42 yields

A change in the time scale on which a computer is operating can be effected by changing only those components performing operations inherently dependent on time. For example, the solution time for a third-order linear differential equation with the setup described could be doubled merely by halving the gains of each of the three integrators. No change in the initial conditions, the summing circuit, or the coefficient potentiometers would be required.

As a practical matter, it is desirable to arrange an analog computer (unless it is of the high-speed repetitive type) for a solution time in the range of 30 seconds to 2 minutes. A lower limit is set by the speed of response of mechanical elements, such as servo units or recorders, while an upper limit is set by integrator drift.

6-2.7 LINEAR OPERATIONS[2]

Pars. 6-1 and 6-2.2 show that ordinary differential equations can be solved by the instrumentation of various mathematical operations. The present discussion summarizes briefly techniques used to instrument linear operations and provides appropriate references to the more detailed discussions of specific devices that are given later in the chapter.

6-2.8 Scale Changing

The simplest operation performed in an electronic analog computer is scale changing, i.e., multiplying by a fixed Coefficient. This is accomplished by means of a high-gain amplifier with resistive feedback, as shown in Fig. 6-34.

If negligible current flows into the amplifier, the error voltage can be written directly as

$$v_e = \frac{R_f}{R_i + R_f} v_i + \frac{R_i}{R_i + R_f} v_o \qquad (6\text{-}57)$$

Furthermore, the output of the amplifier is related to its input by the relationship

$$v_o = -A v_e \qquad (6\text{-}58)$$

Where the amplifier gain is very high, the combination of Eqs. 6-57 and 6-58 shows that

$$v_o = -\frac{R_f}{R_i} v_i \qquad (6\text{-}59)$$

Examination of Eq. 6-59 shows that the gain of the circuit in Fig. 6-34 can be adjusted by a change in the value of either the input resistor or the feedback resistor. Because practical difficulties associated with the closed-loop stability of the amplifier may be encountered if an attempt is made to vary the feedback resistor over a wide range, the usual practice is to employ one fixed value of R_f (1 megohm is the usual value) and to vary R_i to change the overall gain. Although continuous adjustment of the gain over a range of 10 to 1 is easily achieved by variation of R_i, this method of setting arbitrary gains is not the one most frequently employed because, as shown by Ey. 6-59, the gain varies inversely with R_i and, consequently, setting R_i is somewhat inconvenient. The more usual practice is to permit adjustment of the overall gain in steps -- such as 1, 2, 4, and 10 -- by selection of the appropriate input resistance, and to provide continuous gain adjustment, when required, by the use of a potentiometer connected as shown in Fig. 6-35. In many applications, the error caused by loading the potentiometer with the resistance R_i is negligible, since typical

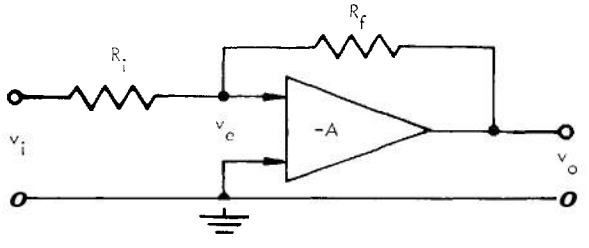


Figure 6-34. Amplifier with resistive feedback,

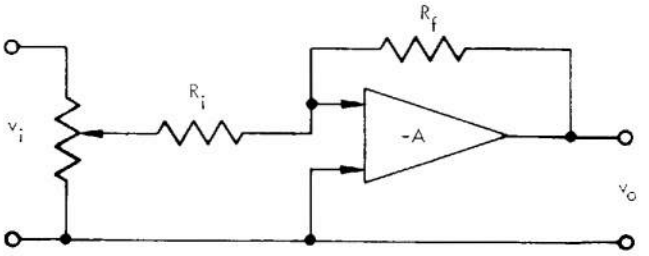


Figure 6-35. Use of a potentiometer for continuous gain adjustment.

values for the total resistance of the potentiometer range from 10,000 to 30,000 ohms, whereas R_i may range from 100,000 to 1,000,000 ohms. When increased accuracy is required, the potentiometer setting can be made with the aid of a digital voltmeter or a precision attenuator after the particular resistance R_i to be used is connected.

6-2.9 Summation

The circuit used for scale changing, and incidentally for providing sign reversals, is readily extended, as shown in Fig. 6-36, for a summation of voltages. If the error voltage in the circuit of Fig. 6-36 is negligible (that is, the amplifier gain is very large), the output voltage can be shown by simple circuit theory to be given by the equation

$$v_o = -\left(\frac{R_f}{R_1} v_1 + \frac{R_f}{R_2} v_2 + \cdots + \frac{R_f}{R_n} v_n\right) \qquad (6\text{-}60)$$

One summing-circuit arrangement used commercially provides seven inputs with respective gains of 1, 1, 1, 4, 4, 10, and 10. By connecting an input signal to the proper combination of input terminals, any integral value of amplifier gain from 1 to 31 may be obtained with this arrangement.

Since subtraction is the same process as addition, except that the sign is reversed, subtraction is not treated separately.

Electronic techniques for addition are discussed in par. 6-3.3, while mechanical techniques are discussed in par. 6-4.1.

6-2.10 Integration

The analog solution of ordinary differential equations is based on the use of integrators. Integration can be performed mechanically with ball-and-disk or disk-disk mechanisms (see par. 6-4.2), electromechanically with a rate servomechanism (see par. 6-4.2), or electrically with an RC

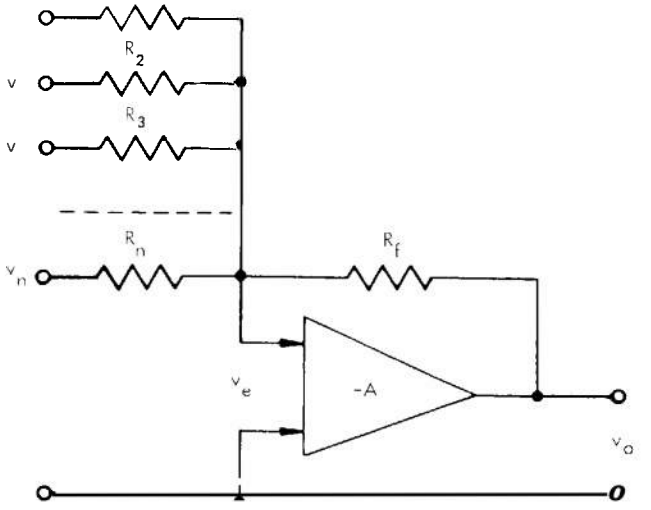


Figure 6-36. Representative circuit for the summation of n voltages.

feedback network around a high-gain amplifier (see par. 6-3.1). Pneumatic integrators, in which a gas is passed through an orifice into a tank, are also used.

Probably the first mechanical integrating device was the planimeter invented in 1814 by J. M. Hermann. Over the next 40 years, various planimeters were proposed, but this work did little to introduce integrating devices into mathematical analysis. In the early 1860's, James Thomson proposed a disk-sphere cylinder integrator, and about 10 years later William Thomson (who later became Lord Kelvin) conceived the basic idea of interconnecting integrators to obtain analog solutions to ordinary differential equations. The use of electronic integrators originated during World War II.

The basic circuit for performing integration in electronic analog computers is similar to that employed for scale changing but, as shown in Fig. 6-37, employs capacitive, ratlier than resistive, feedback. If, as before, the ideal-amplifier situation is analyzed, the input current i_i can be written as

$$i_i = \frac{v_i}{R_i} \qquad (6\text{-}61)$$

Because the input circuit of the amplifier draws negligible current, the feedback current i_f, as defined in Fig. 6-37, is the negative of i_i or

$$i_f = -\frac{v_i}{R_i} \qquad (6\text{-}62)$$

If the error voltage is negligible, the voltage across the capacitor equals the output voltage and, consequently, can be written as

$$v_o = -\frac{1}{C}\int_0^{t_1} i_f\, d\tau + v_o(0) \qquad (6\text{-}63)$$

where τ is computer time (real time) and t_1 is the time for which v_o is determined.

Substitution of Eq. 6-62 into Ey. 6-63 yields

$$v_o = \frac{1}{C}\int_0^{t_1} v_i\, d\tau + v_o(0) \qquad (6\text{-}64)$$

Eq. 6-64 shows that the gain factor of the integrator is determined by the product R_iC of the input resistor and the feedback capacitor. With the integrator, as with the scale changer, the error resulting from noninfinite amplifier gain is negligible in practical applications. However, the feedback capacitor used in an electronic integrator must have a very high leakage resistance or the performance of the integrator deteriorates. Again for the case of an infinite-gain amplifier, analysis of the circuit of Fig. 6-37 with the addition of a leakage resistance R_L in parallel with the capacitance C yields the relationship

$$v_o = -\frac{1}{R_iCs + \frac{R_i}{R_L}}\, v_i \qquad (6\text{-}65)$$

which has been transformed into the frequency domain, with s as the complex-frequency variable. Thus, a noninfinite R_L determines the frequency at which the operation of this circuit departs by a specified amount from that of an ideal integrator defined by the equivalent relationship

$$v_o = -\frac{1}{R_iCs}\, v_i \qquad (6\text{-}66)$$

The leakage resistance of a good 1-μf integrator capacitor, which usually employs polystyrene as the dielectric, may have a typical value of 1,000,000 megohms. This value of leakage resistance causes the performance to depart from that of an ideal integrator by the introduction of a phase error of 1 milliradian at a frequency of 1 millirad/sec. Consequently, very long solution times must be involved before leakage resistance introduces appreciable errors. Grid current in the input stage of the amplifier in Fig. 6-37 presents another limit on integrator performance. Here, the extraneous output resulting from grid current is given by

$$e_o = -\frac{1}{C}\int_0^t i_g\, d\tau \qquad (6\text{-}67)$$

where i_g is the grid current. Grid current produces an offset that increases with time and thus sets a limit on the maximum solution time that can be used before a specified error builds up.

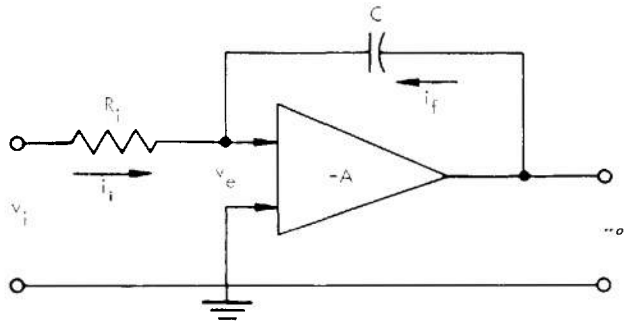


Figure 6-37. Basic circuit for integration.

In order to use an integrator in an electronic differential analyzer, means must be provided to set the initial value of the integral at any arbitrary value. Because the value of the integral in the circuit of Fig. 6-37 is proportional to the voltage across the capacitor, the direct way to set the initial value is to place a charge on the capacitor prior to the start of the solution. Fig. 6-38 illustrates a basic circuit often used. In the INITIAL CONDITION position, the amplifier input is switched to a resistive network. The potentiometer setting determines the voltage to which the capacitor is charged. When the switch is placed in the COMPUTE position, the capacitor initially retains its charge, but the rircuit begins to function as an integrator. In the solution of a set of equations, a number of such integrator circuits must be switched simultaneously. Consequently, the switching usually is accomplished with a number of relays with their coils connected in parallel and energized through a common switch. Closing a single switch then activates all the relay coils simultaneously, and the solution begins.

6-2.11 Synthesis of Rational Transfer Functions

Analog-computer studies often involve transfer functions of the form

$$H(s) = \frac{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_o}{s^n + b_{n-1} s^{n-1} + \cdots + b_o} \qquad (6\text{-}68)$$

where s is the complex-frequency variable and the a's and b's are real constants. Such functions can be instrumented by an appro-

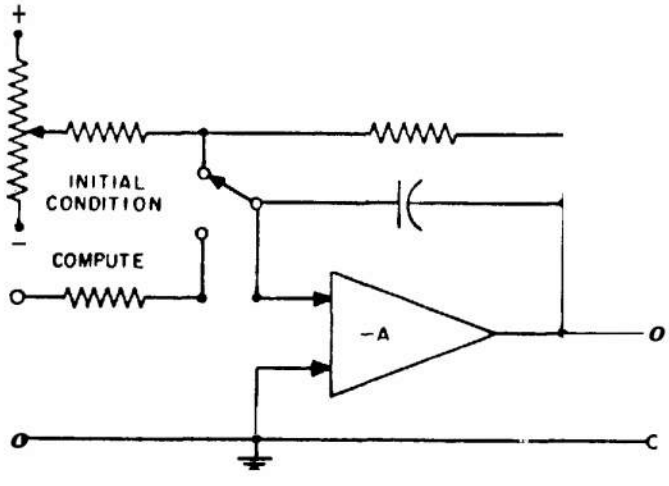


Figure 6-38. Electronic integrator with initial-condition circuit.

priate combination of the basic operations of integration, summation, and multiplication by a constant coefficient, as shown in Fig. 6-33.

This method is particularly suitable for realizing transfer functions having a small number of poles, particularly if the coefficients in the function require frequent change. Since the coefficients appear directly as the gains of amplifiers, almost no calculations are required in the synthesis. However, if high-order functions are to be synthesized, an excessively large number of active units is required by this method, and the equipment reduction effected by using the methods discussed in the remainder of this section may assume practical significance.

The simplest generalization of the basic integrator circuit is shown in Fig. 6-40. If an ideal amplifier with infinite gain is assumed, analysis of the circuit of Fig. 6-40 in terms of admittances leads to the relationship

$$\frac{e_o}{e_i} = -\frac{Y_A}{Y_B} \qquad (6\text{-}69)$$

If Y_A and Y_B are two-terminal RC networks, all their poles and zeros must alternate along the negative real axis of the complex frequency plane and the lowest critical frequencies must be zero. Consequently, the poles and *zeros* of the transfer function also must lie on this axis, but two poles *or* two zeros may occur together, and the lowest critical frequency may be a pole. Any transfer functions meeting these conditions can be written in the form

$$\frac{e_o}{e_i} = -\frac{\frac{N(s)}{G(s)}}{\frac{D(s)}{G(s)}} \qquad (6\text{-}70)$$

where N(s) and D(s) are polynomials having the forms, respectively, of the numerator and denominator of Eq. 6-68 and where G(s) can be selected so that N(s)/G(s) and D(s)/G(s) can be realized as two-terminal RC networks.

The synthesis of Y_A can be carried out in several ways, one of the simplest being to expand N(s)/G(s) in the form

$$\frac{N(s)}{G(s)} = s\left(C_1 + \sum_{i=2}^{Q} \frac{\frac{1}{R_i}}{s + \frac{1}{R_i C_i}}\right) \qquad (6\text{-}71)$$

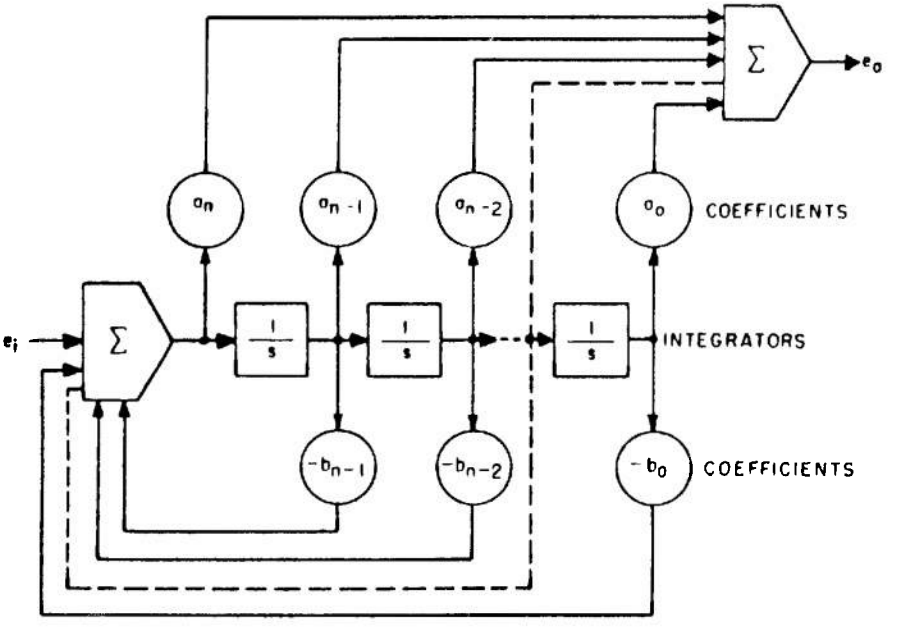


Figure 6-39. Integrator realization.

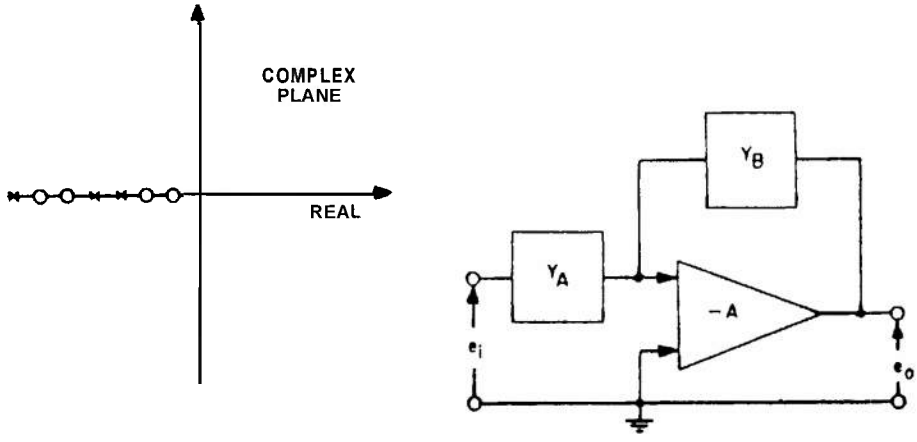


Figure 6-40. Block diagram for one-amplifier realization with two-terminal networks.

where

$$C_1 = \lim_{s \to \infty} \left[\frac{1}{s} \frac{N(s)}{G(s)} \right] \qquad (6\text{-}72)$$

The sum

$$\sum_{i=2}^{Q} \frac{\frac{1}{R_i}}{s + \frac{1}{R_i C_i}} \qquad (6\text{-}73)$$

is obtained by making a partial-fraction expansion of

$$\frac{1}{s}\left[\frac{N(s)}{G(s)} - C_1 s\right] \qquad (6\text{-}74)$$

The resulting network is shown in Fig. 6-41, where the values of R_i and C_i are in ohms and farads.

Substitution of three-terminal networks in place of the two-terminal networks of Fig. 6-40 yields a useful generalization of this method of synthesis. The resulting circuit, illustrated in Fig. 6-42, can be analyzed in terms of the input, output, and transfer admittances of the network. These admittances are defined for the A network by the relationships

$$i_1 = Y_{A11} e_1 - Y_{A12} e_2 \qquad (6\text{-}75)$$

and

$$i_2 = -Y_{A12} e_1 + Y_{A22} e_2 \qquad (6\text{-}76)$$

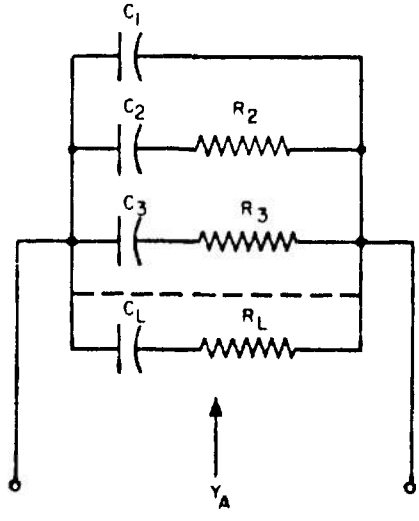


Figure 6-41. Resulting form of synthesis network.

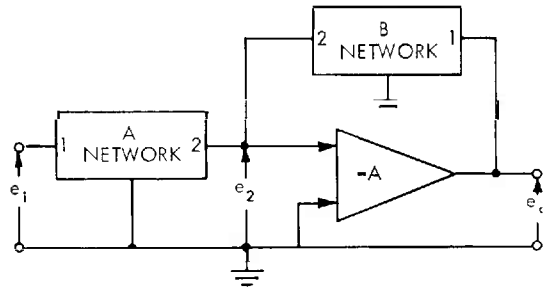


Figure 6-42. Block diagram for one-amplifier realization with three-terminal networks.

where the currents and voltages are shown in Fig. 6-43. A similar definition applies to the B network. The voltages in the circuit of Fig. 6-42 are related by the equation

$$e_2(Y_{A22} + Y_{B22}) - e_i(Y_{A12}) + e_o(Y_{B12}) \quad (6\text{-}77)$$

Furthermore,

$$e_1 = -Ae_2 \quad (6\text{-}78)$$

Solution of Eq. 6-77 and Eq. 6-78 yields the following relationship for the transfer function e_o/e_i:

$$\frac{Y_{A12}}{Y_{B12} + \dfrac{Y_{A22} + Y_{B22}}{A}} \quad (6\text{-}79)$$

As A becomes infinite, e_1/e_i approaches the negative of the ratio of the transfer admittances, that is,

$$\frac{e_o}{e_i} = -\frac{Y_{A12}}{Y_{B12}} \quad (6\text{-}80)$$

The error caused by a finite value of A can be evaluated from Eq. 6-79 which is the exact expression for the realized transfer function. The errors can be determined either as the displacements of the poles of the realized transfer function from the desired poles or as the error in the amplitude and phase of the realized transfer function at real frequencies. To keep the error small, it is necessary that the following relationship hold at all frequencies:

$$Y_{B12} \gg \frac{Y_{A22} + Y_{B22}}{A} \quad (6\text{-}81)$$

At high frequencies, either or both of the output admittances Y_{A22} and Y_{B22} may tend to become infinite. If such is the case, the B network should be so designed that Y_{B12} also goes to infinity at high frequencies.

The transfer admittance of a three-terminal network formed entirely of resistances and capacitances can have only simple poles that must lie on the negative real axis of the complex-frequency plane, but may have zeros that lie anywhere in the complex-frequency plane except on the positive real axis and that need not be simple. The poles of e_1/e_i in Eq. 6-80 follow from the poles in Y_{A12} or from the zeros of Y_{B12}, while the zeros of e_o/e_i follow from the zeros of Y_{A12} or from the poles of Y_{B12}. Consequently, little theoretical restriction is placed on the type of transfer impedance that can be formed by using a circuit of the type shown in Fig. 6-42.

Several general procedures for synthesizing three-terminal RC networks have been given in the literature[4,5]. These procedures are too lengthy to include here, but are relatively straightforward.

The principal restrictions imposed on this realization are the complexity of the synthesis calculations, the large number of elements, and the great range of element values that may be required.

Although few theoretical limitations are imposed on the type of transfer function realizable with the single-feedback-amplifier method just described, the use of additional amplifiers permits increased flexibility in the realization of the transfer function. This

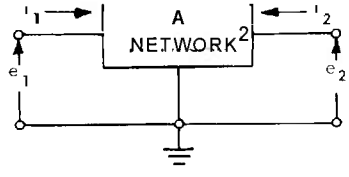


Figure 6-43. Definition of admittances.

flexibility can reduce the number of passive elements required to obtain a given function, decrease the spread of element values, and simplify the synthesis calculations. Such expedients are particularly important when complicated functions with many poles must be realized.

One synthesis metliod using three amplifiers is developed to demonstrate that any transfer function can be realized in this way. Once the particular method is understood, many possible variations become obvious.

The circuit for the three-amplifier realization is shown in Fig. 6-44. This circuit differs from the one-amplifier realization shown in Fig. 6-42 only by the addition of the C and D networks and the inverting amplifiers driving these networks. As is brought out in the following discussion, two-terminal networks are sufficient to realize any transfer function; hence, this case is considered.

The voltages in the system obey the relationships

$$e_2(Y_A + Y_B + Y_C - Y_D) - e_i(Y_A - Y_C) + e_o(Y_B - Y_D) \qquad (6\text{-}82)$$

and

$$e_o = -Ae_i \qquad (6\text{-}83)$$

Solution of Eq. 6-82 and Ey. 6-83 yields the following relationship forthe transfer function from e_i to e_o:

$$\frac{e_o}{} \quad \frac{Y_A - Y_C}{Y_B - Y_D + \left(\frac{Y_A + Y_B - Y_C + Y_D}{A} \right)} \qquad (6\text{-}84)$$

As A becomes large, Eq. 6-84 assumes the limiting form

$$\frac{e_o}{e_i} = -\frac{Y_A - Y_C}{Y_B - Y_D} \qquad (6\text{-}85)$$

If the desired transfer function is expressed as a ratio of two polynomials N(s)/D(s), the admittances must satisfy the relationship

$$\frac{Y_A - Y_C}{Y_B - Y_D} = \frac{N(s)}{D(s)} \qquad (6\text{-}86)$$

In order to realize the admittances as RC networks, Eq. 6-86 is separated to give

$$Y_A - Y_C = \frac{N(s)}{G(s)} \qquad (6\text{-}87)$$

and

$$Y_B - Y_D = \frac{D(s)}{G(s)} \qquad (6\text{-}88)$$

where G(s) is an arbitrary polynomial which does not alter the realized transfer function.

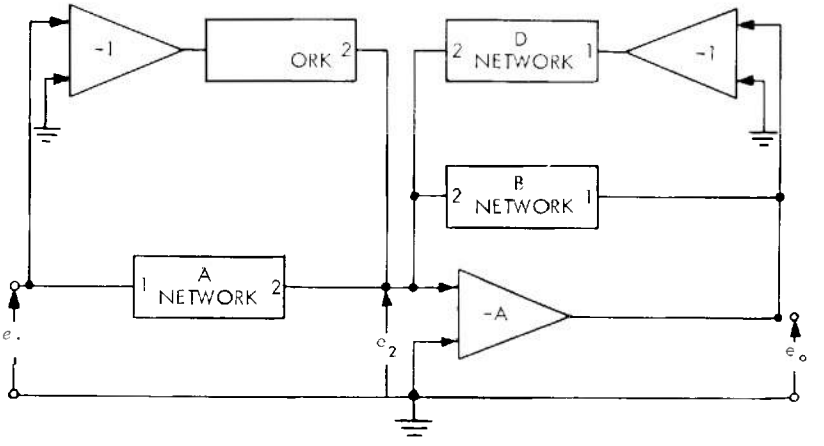


Figure 6-44. Block diagram for three-amplifier realization.

The realization follows the method used to obtain Y_A and Y_B in Fig. 6-40. The fraction N(s)/G(s) is expanded in the series given by Eq. 6-71, and the terms in the resulting expansion are divided between the A and C networks in such a way that all the elements have positive values. The additional freedom gained from allowing negative terms in the expansion makes possible the realization of any ratio N(s)/G(s) with two-terminal RC networks, provided that the two following conditions are met: (1) the zeros of G(s) lie on the negative real axis; (2) the ratio N(s)/G(s) goes to infinity no faster than s as s becomes infinite. An identical procedure is used to realize the ratio D(s)/G(s).

The error introduced by a finite gain A can be evaluated from Eq. 6-84 which is the exact expression for the realized transfer function. The method is the same as the method already described for evaluating the errors in the one-amplifier realization. However, in Eq. 6-84, the possibility exists of changing the synthesis procedure slightly to realize exactly the desired transfer function with a finite value of A. Eq. 6-84 can be rewritten in the form

$$\frac{e_o}{e_i} = -\frac{Y_A - Y_C}{Y_B\left(1 + \frac{1}{A}\right) - Y_D\left(1 - \frac{1}{A}\right) + \frac{1}{A}\left(Y_A + Y_C\right)} \tag{6-89}$$

If the desired transfer function is again designated by the ratio N(s)/D(s), it can be realized by making

$$Y_A - Y_C = \frac{N(s)}{G(s)} \tag{6-90}$$

and

$$Y_B\left(1 + \frac{1}{A}\right) - Y_D\left(1 - \frac{1}{A}\right) = \frac{D(s)}{G(s)} - \frac{1}{A}\left(Y_A + Y_C\right) \tag{6-91}$$

If $A > 1$, the ,A, B, C, and D networks can always be realized as two-terminal RC networks by using expansions of the form given in Eq. 6-71. The exact realization, obtained at the expense of including additional elements in the B and D networks, is justified only in special instances because the errors caused by a finite value of A in the approximate realization are usually less than the errors due to parasitic behavior of the elements.

A major advantage of the three-amplifier synthesis procedure is the simplicity of the calculations required to obtain the element values. The spread of element values is determined by the spread of the terms in the expansion of the ratio N(s)/G(s) that is given in Eq. 6-71 and the corresponding expansion of the ratio D(s)/G(s). The arbitrary zeros of G(s) can be chosen by a trial-and-error approach to control this spread.

6-2.12 NONLINEAR OPERATIONS[2,3]

In the solution of nonlinear ordinary differential equations, the need often arises for means that permit multiplication of two computer variables and the introduction of arbitrary functions of one or two variables. Because these operations are more difficult to perform than the linear operations, a great deal of effort has been spent by workers in the field of analog computation in the development of multipliers and function generators. The principal methods that have been developed for performing these nonlinear operations are discussed briefly in the immediately following paragraph. Where applicable, references are made to the descriptions of various devices appearing later in the chapter.

6-2.13 Multiplication and Division

Two types of multiplication arise in computer work: (1) multiplication of a computer variable by a constant and (2) multiplication of one computer variable by another. The first type is simple; the second is difficult. Multiplication can be performed mechanically, electromechanically, or electronically, as described in pars. 6-3.2 and 6-4.3.

The chief requirements for a multiplier to be used in a general-purpose analog computer are speed, accuracy, and relative simplicity. Servomultipliers of the type described in par. 6-4.3 can be built to meet the last two requirements, but their speed of response is inherently limited. Many attempts have been made to build all-electronic multipliers (Ref. par. 6-3.2) that meet all three requirements. Only since the late 1950's

has it been possible to reduce the errors in these multipliers to a degree comparable with that achieved in the linear computing components. However, the all-electronic schemes for multiplication remain complex in comparison with the means for performing linear operations.

The high accuracy achieved in such analog-computing components as integrators and coefficient multipliers results from the use of feedback in such a way that the stability and linearity of the units are determined by the characteristics of passive elements rather than those of vacuum tubes. A high-performance multiplier is difficult to design because the product cannot be compared directly with either of the input signals for the purpose of obtaining an error signal to be used in the feedback loop.

In the multipliers described, either one or the other of the two following schemes has been used to achieve high accuracy:

(1) An indirect type of feedback control
(2) A circuit in which vacuum tubes act merely as switches.

In a conventional servomultiplier, the indirect control of the feedback loop employs a reference voltage and a feedback potentiometer. The effectiveness of this method depends upon the constancy of the reference signal and upon the similarity of the control and multiplying potentiometers.

In the time-division multipliers[6,7] and in the quarter-square multiplier[8] using a segmented-straight-line representation of the square-law function, vacuum tubes are used merely as switches. The use of tubes in this manner offers great possibilities in the design of precision computing components, as demonstrated by performance that approaches that achieved in linear computing components.

Although division is in many ways similar to multiplication, some division schemes utilize special techniques and introduce additional problems.

At first glance, it might appear possible to perform mechanical division by interchanging the output and one of the inputs of a multiplier, such as one of those that are discussed in par. 6-4.3. The practical difficulty with this approach is that the quotient approaches infinity as the divisor approaches zero. This requirement exceeds the capacity of any physical device. Furthermore, even within the capacity of the device, a high input torque is required when the divisor is small and friction may make the device completely inoperative.

The circuitry of several electronic multipliers is discussed in par. 6-3.2, while mechanical and electromeclianical multipliers are described in par. 6-4.3.

6-2.14 Vector Resolution

Problems requiring the resolution of a vector into components in a particular coordinate system and the transformation of vector quantities from one coordinate system to another arise in the study of systems involving the determination of trajectories from component velocities and forces. The problem of representing the trajectory of an aircraft subject to forces of drag, thrust, gravity, etc., is typical of this class. These forces and the resulting trajectory can be described in terms of a set of axes fixed to the aircraft, in terms of axes fixed with respect to the earth, or in terms of a set of axes one of which is aligned with respect to the relative wind. Each of these axis sets offers advantages for some calculations and disadvantages for others. Consequently, in the study of an overall system it is usually advantageous to employ two or more coordinate systems and make appropriate transformations between them.

Vectors can be described in either a rectangular or a polar coordinate system. Figure 6-45 illustrates the representation, for a simple two-dimensional case, of a single vector quantity in both a rectangular and a polar coordinate system. In the rectangular X-Y system, the vector is described in terms of its components along the orthogonal X- and Y-axes. These two components are designated x and y. In the polar system, the vector is described by its magnitude and by the angle it makes with respect to a fixed reference axis. These are designated in the figure as r and θ.

If the vector is expressed in polar coordinates (r, θ), its components in a rectangular system are given by the equations

and

$$\left.\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \end{aligned}\right\} \qquad (6\text{-}92)$$

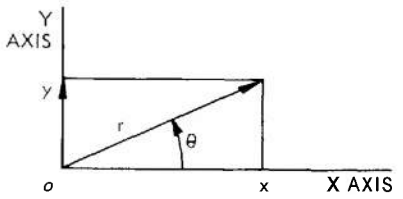


Figure 6-45. Representation of a vector in a rectangular coordinate system and in a polar coordinate system.

The alternate transformation, rectangular to polar, or as it is usually termed, simply the "polar" transformation, is accomplished in accordance with the relationships

$$\left.\begin{aligned} \theta &= \tan^{-1}\frac{y}{x} \\ r &= \sqrt{x^2 + y^2} \end{aligned}\right\} \qquad (6\text{-}93)$$

Fig. 6-46 illustrates the related problem of expressing a vector that is initially specified in one rectangular coordinate system in a second rectangular system having its origin common with the first but with the axes rotated through the angle 8. The components u and v in the second system are related to the components x and y in the first system and the angle of rotation 8 by the equations

$$\left.\begin{aligned} u &= x\cos\theta - y\sin\theta \\ v &= x\sin\theta + y\cos\theta \end{aligned}\right\} \qquad (6\text{-}94)$$

and

The direction of a vector in a three-dimensional coordinate system can be specified either in terms of the direction cosines of the vector or in terms of a set of Euler angles. The basic mathematical relationships involved with each of these techniques is discussed here and computer techniques for performing vector transformations by the two schemes are discussed in par. 6-4.4 through par. 6-4.7.

6-2.15 Direction Cosines

In this discussion, it will be assumed that the axis system under consideration is a right-handed one having axes X, Y, and Z. In such a system, a right-handed screw directed along the positive Z-axis will advance in the positive direction when it is rotated from the positive X-axis toward the positive Y-axis through the smaller (here 90°) angle.

Any vector $\vec{A}$ in such a system can be represented uniquely in the form

$$\vec{A} = A_x\vec{i} + A_y\vec{j} + A_z\vec{k} \qquad (6\text{-}95)$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are unit vectors along the x-, y-, and z-axes respectively, and A_x, A_y and A_z are the coordinates of the terminal point of the vector $\vec{A}$. The length of the vector $\vec{A}$ is then given by

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2} \qquad (6\text{-}96)$$

The direction of the vector can be specified by a set of direction angles, i.e., the angles that the vector makes with the three coordinate axes. The angles between the vector A and the positive X-, Y-, and Z-axes are denoted symbolically by (A,x), (A,y) and (A,z), respectively. The components of the vector are then given by the equation

$$\left.\begin{aligned} A_x &= A\cos(A,x) \\ A_y &= A\cos(A,y) \\ A_z &= A\cos(A,z) \end{aligned}\right\} \qquad (6\text{-}97)$$

Use of Eqs. 6-96 and 6-97 shows that

$$\cos^2(A,x) + \cos^2(A,y) + \cos^2(A,z) = 1 \qquad (6\text{-}98)$$

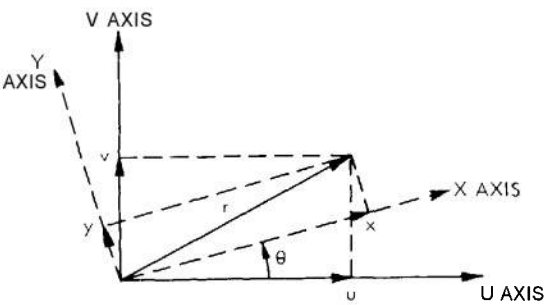


Figure 6-46. Rotation of a rectangular coordinate system.

Consequently, tlie direction angles are not independent and, if any two of them are specified, the third must satisfy Ey. 6-98. The cosines of these direction angles are called the direction cosines.

In developing methods for specifying the orientation of one coordinate system with respect to another having its origin common with the first, it is convenient to have an application in mind. The problem of specifying the orientation of a set of right-handed orthogonal axes fixed in an aircraft with a second axis system fixed in inertial space arises frequently and, therefore, provides a good example. The first axis system is called the body-axis system, and the second the inertial system. The origin of the body-axis system is fixed at the nominal center of gravity of tlie aircraft and the three body axes are fixed with respect to the aircraft. Unit vectors along the X-, Y-, and Z-axes in this system are designated $\vec{i}_b$, $\vec{j}_b$ and $\vec{k}_b$. The exact alignment of the X body-axis is somewhat arbitrary but here it shall be considered to be aligned with the principal axis of tlie aircraft. The Y- and Z-axes then form a right-handed system as shown in Fig. 6-47.

The inertial system is a right-handed triad of mutually perpendicular axes fixed in inertial space. It is assumed that the earth is an adequate local reference. Consider the X-Y inertial plane as being taken perpendicular to the gravity vector. The X inertial axis is usually fixed in the direction of true north. In this system, the unit vectors are designated $\vec{i}_i$, $\vec{j}_i$ and $\vec{k}_i$.

The orientation of the body-axis system with respect to the inertial system is illustrated in Fig. 6-48. The direction of each of the body axes can be specified with respect to the inertial axes by three direction cosines as shown in Fig. 6-49. To locate the three axes of a coordinate system, a total of nine direction cosines is needed, but when systems employing mutually perpendicular axes are used, six of these direction cosines are actually redundant.

6-2.16 Euler Angles

It can readily be visualized that a set of axes, such as the body axes of Fig. 6-47, could be oriented in any arbitrary way with respect to a set of inertial axes having the same origin by three successive angular rotations as defined in Fig. 6-50. It should be noted that the final orientation of a body following several rotations in space is dependent on the

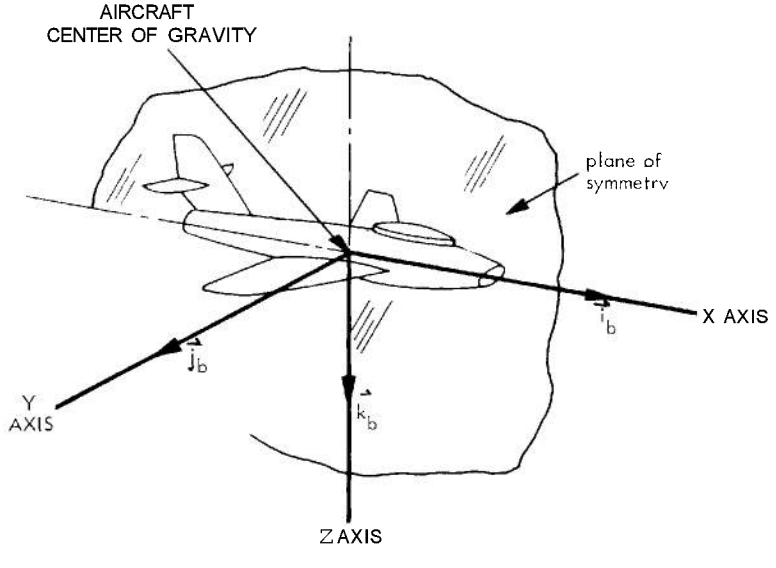


Figure 6-47. Example of a body-axis system.

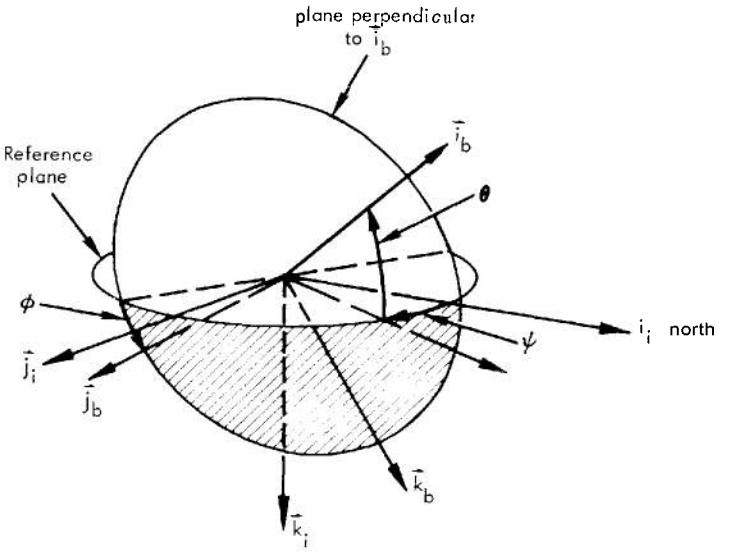


Figure 6-48. Orientation of a body-axis system with respect to an inertial system.

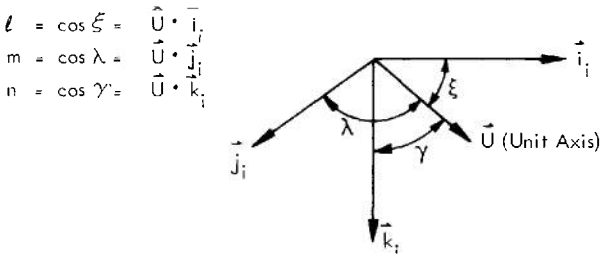


Figure 6-49. Direction cosines defining the orientation of an axis in inertial space.

order in which the rotations are made. Consequently, a convention in this regard must be set up and followed or erroneous results will be obtained. The convention indicated below is widely used, but is not the only one.

Assume that the two axis systems are initially coincident and it is desired to specify *a series* of three angular rotations that will define the final orientation of an axis system, for example a body-axis system, with respect to the inertial system. First, the azimuth ψ of the vertical plane containing i_b, $\vec{k}_i$ and the intersection of this plane with a reference plane defined by $\vec{i}_i$ and $\vec{j}_i$ is defined (see Fig. 6-48). This is achieved by a rotation about the $\vec{k}$ axis. A new axis system designated by the subscript 1 is then related to the original set by the first set of equations in Fig. 6-50.

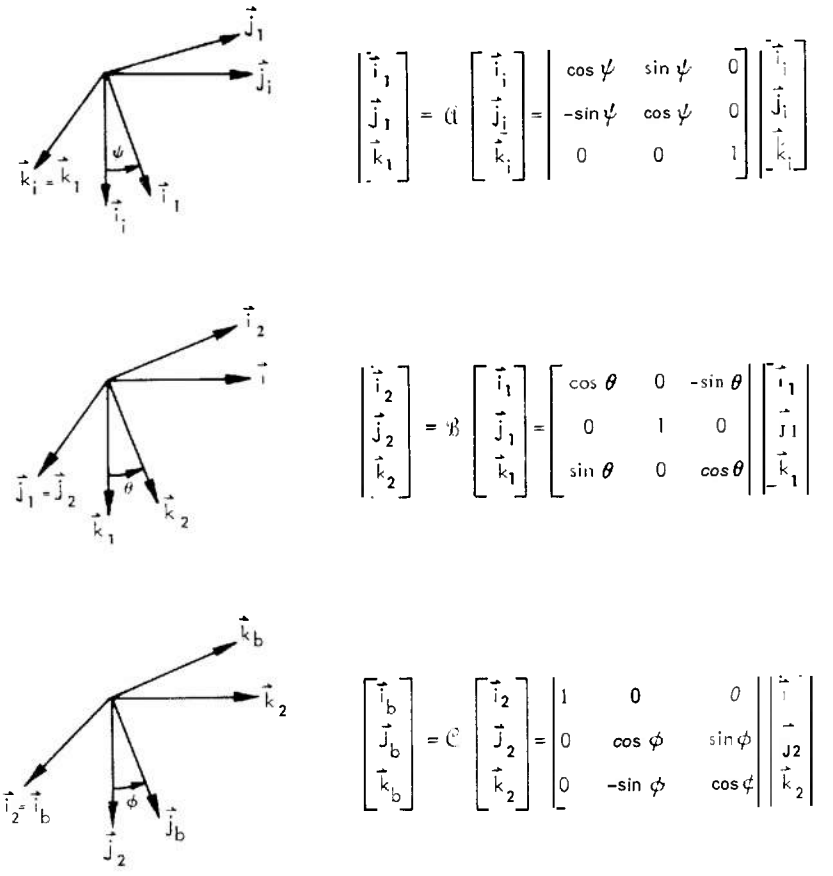


Figure 6-50. Euler-angle definitions.

The elevation angle θ of $\vec{i}_b$ above the reference plane is tlien defined. This is achieved by rotation about tlie $\vec{j}_1$ axis defined in the previous step anti leads to the second axis system specified in Fig. 6-50. The third and final rotation defines the roll angle ϕ and is achieved by a rotation about the i_2 axis defined by the previous step.

In matrix notation, the axes are related as follows (see the appendix to this chapter):

$$\begin{bmatrix} \vec{i}_b \\ \vec{j}_b \\ \vec{k}_b \end{bmatrix} = \mathcal{C}\mathcal{B}\mathcal{A} \begin{bmatrix} \vec{i}_i \\ \vec{j}_i \\ \vec{k}_i \end{bmatrix} = \mathcal{T} \begin{bmatrix} \vec{i}_i \\ \vec{j}_i \\ \vec{k}_i \end{bmatrix} \qquad (6\text{-}99)$$

where

$$\mathcal{T} = \begin{vmatrix} \ell_x & m_x & n_x \\ \ell_y & m_y & n_y \\ \ell_z & m_z & n_z \end{vmatrix}$$

Transformation from body axes to inertial axes is given in matrix form by

$$\begin{bmatrix} \vec{i}_i \\ \vec{j}_i \\ \vec{k}_i \end{bmatrix} = \mathcal{A}^{-1}\mathcal{B}^{-1}\mathcal{C}^{-1} \begin{bmatrix} \vec{i}_b \\ \vec{j}_b \\ \vec{k}_b \end{bmatrix} = \mathcal{T}^{-1} \begin{bmatrix} \vec{i}_b \\ \vec{j}_b \\ \vec{k}_b \end{bmatrix} \qquad (6\text{-}100)$$

where the notation $\mathcal{A}^{-1}$, etc. refers to the inverse matrix (see the appendix to this chapter) and

$$\mathfrak{D}^{-1} = \begin{bmatrix} \ell_x & \ell_y & \ell_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \qquad (6\text{-}101)$$

The nine direction cosines are equivalent to the three Euler angles and related to them by the following expressions:

$$\left.\begin{aligned} \ell_x &= \cos\theta\cos\psi \\ m_x &= \cos\theta\sin\psi \\ n_x &= -\sin\theta \\ \ell_y &= \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi \\ m_y &= \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi \\ n_y &= \sin\phi\cos\theta \\ \ell_z &= \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ m_z &= \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ n_z &= \cos\phi\cos\theta \end{aligned}\right\} \qquad (6\text{-}102)$$

Conversely, the three Euler angles can be found from the following expressions:

$$\left.\begin{aligned} \sin\theta &= -n_x \\ \tan\psi &= m_x/\ell_x \\ \tan\phi &= n_y/n_z \end{aligned}\right\} \qquad (6\text{-}103)$$

To transform a vector $\vec{v} = x'\vec{i}' + y'\vec{j}' + z'\vec{k}'$ from any axis system $(\vec{i}', \vec{j}', \vec{k}')$ to a reference axis system $(\vec{i}, \vec{j}, \vec{k})$, the following equation must be satisifed:

$$\vec{v} = x'\vec{i}' + y'\vec{j}' + z'\vec{k}' = x\vec{i} + y\vec{j} + z\vec{k} \qquad (6\text{-}104)$$

In terms of the nine direction cosines and expressed in matrix form, the required transformation is specified by the relationship

$$\begin{bmatrix} \ell_x & \ell_y & \ell_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (6\text{-}105)$$

In terms of Euler angles, the coordinate transformations take the form

$$\begin{aligned} x &= \ell_x x' + \ell_y y' + \ell_z z' \\ &= x'\cos\theta\cos\psi \\ &\quad + y'(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\ &\quad + z'(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \end{aligned} \qquad (6\text{-}106)$$

$$\begin{aligned} y &= m_x x' + m_y y' + m_z z' \\ &= x'\cos\theta\sin\psi \\ &\quad + y'(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) \\ &\quad + z'(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \end{aligned} \qquad (6\text{-}107)$$

$$\begin{aligned} z &= n_x x' + n_y y' + n_z z' \\ &= -x'\sin\theta + y'\sin\phi\cos\theta + z'\cos\phi\cos\theta \end{aligned} \qquad (6\text{-}108)$$

Conversely, a vector $\vec{v} = x\vec{i} + y\vec{j} + z\vec{k}$ given in terms of a reference axis system $(\vec{i}, \vec{j}, \vec{k})$ can be converted to an axis system $(\vec{i}', \vec{j}', \vec{k}')$ by means of the same family of direction cosines using the inverse matrix transformation

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \ell_x & m_x & n_x \\ \ell_y & m_y & n_y \\ \ell_z & m_z & n_z \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (6\text{-}109)$$

The fact that the transpose (see appendix) of the direction-cosine matrix is also the inverse of the matrix follows in this special case where the axes of each axis system are orthogonal or mutually perpendicular.

Written in terms of the Euler angles describing the orientation of the given axis system with respect to the reference axis system, the transformation equations are

$$\begin{aligned} x' &= \ell_x x + m_x y + n_x z \\ &= x\cos\theta\cos\psi + y\cos\theta\sin\psi - z\sin\theta \end{aligned} \qquad (6\text{-}110)$$

$$\begin{aligned} y' &= \ell_y x + m_y y + n_y z \\ &= x(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\ &\quad y(\cos\phi\cos\theta + \sin\phi\sin\theta\sin\psi) \\ &\quad + z(\sin\phi\cos\theta) \end{aligned} \qquad (6\text{-}111)$$

$$z' = l'_z x + m_z y + n_z z$$

$$x(\sin \quad \sin \quad \cdot \cos \quad \sin \quad \cos \quad)$$

$$+ y(\cos \cdot \sin \quad \sin \cdot - \sin \cdot \cos \cdot \quad (6\text{-}112)$$

$$+ z \cos \quad \cos$$

Techniques for instrumenting these transformations on an analog computer are described in par. 6-4.4 through par. 6-4.7.

Example 6-1 is a numerical illustration of the coordinate transformations expressed by Eqs. 6-110 through 6-112.

Example 6-1. Numerical illustration of a coordinate transformation

Let the Euler angles defining the coordinate system x', y', z' in terms of the reference set x, y, z be the following:

15

5

- 35

Then, from Eqs. 6-102,

l_x cos cos - 0 996 0 966 0 962

m_x cos sin - 0 996 0 259 - 0 258

n_x - sin -0 087

sin sin cos - cos sin

0 574 0 087 0 966 - 0 819 0 259 -0 164

n_y cos cos sin sin sin

0 819 0 966 0 574 0 087 0 259 0 804

n_z sin cos 0 574 0 996 0 572

l_z sin sin cos sin cos

0.574 0.259 0.819 0.087 0.966 0.218

m_z - cos sin sin - sin cos

- 0 819 0 087 0 259 - 0.574 0 966 - -0 536

n_z - cos cos = 0.819 0 996 0816

If the components of the vector, in the original axis system are x = 1, y = 2, z = 4 (in any convenient set of units), the components in the new axis system are, from Eqs. 6-110, 6-111, and 6-112,

$$x = l'_x x + m_x y + n_x z$$

0 962 1 0 258 2 - 0 087 4

- 1.130

$$y = l_y x + m_y y \quad n_y z$$

0 164 1 0 804 2 0 572 4

3 732

$$z = l'_z x + m_z y + n_z z$$

0.218 1 - 0.536 2 0.816 4

2 410

6-2.17 Generation of Arbitrary Nonlinear Functions

Until recently, attention was directed primarily on the development of techniques for generating functions of a single arbitrary variable. However, in the past 10 years, as a result of increasing interest in the study of systems that can be described adequately only by the use of functions of two or more arbitrary variables, much more attention has been given to this broader problem. The first portion of this discussion deals with the problem of generating functions of a single variable, while the latter portion explores the more general problem.

Two distinctly different methods are used for representing arbitrary nonlinear functions. The first scheme approximates the desired function by a continuous function that, depending on the purposes at hand, may be a function such as a polynomial in the independent variable or may be generated as a continuous physical variable as is done in mechanical-linkage computers. With the second method, the desired function is stored or otherwise represented at a finite number of discrete values of the independent variable and intermediate values are obtained by interpolation.

As an example of the first technique, a simple power series of the form

$$y_p = C_0 + C_1x + C_2x^2 + C_3x^3 + \ldots C_nx^n \qquad (6\text{-}113)$$

can be fitted to an arbitrary function by appropriate selection of the Coefficients C_0, $C_1, C_2, \ldots, C_n$. If a mathematical expression for the desired function is available, it is possible, in principle at least, to derive an expression for the error between the desired function and the function generated by the power series, in terms of the independent variable and the coefficients C_0, C_1, . . . , C_n. One basis on which these coefficients can be evaluated entails minimization of the square of this error over the range of interest of the independent variable.

If the desired function is designated

$$y = f(x) \qquad (6\text{-}114)$$

and the approximation to y is given by Eq. 6-113, then the error is given by the expression

$$e = y - y_p = f(x) - (C_0 + C_1x + C_2x^2 + \ldots) \qquad (6\text{-}115)$$

Then

$$e^2 = (y - y_p)^2 = |f(x) - (C_0 + C_1x + C_2x^2 + \ldots)|^2 \qquad (6\text{-}116)$$

It is desired to minimize the integral of e^2 over the range of interest of the independent variable, x_0 to x_m. This integral will be designated by the symbol J and is given by

$$J = \int_{x_0}^{x_m} e^2(x)dx = \int_{x}^{x_m} |f(x) - (C_0 + C_1x + C_2x^2 + \ldots)|^2 dx \qquad (6\text{-}117)$$

It is desired to minimize J by appropriate selection of the coefficients. To do this, the partial derivatives of J with respect to each of the coefficients are set, in turn, equal to zero. Thus,

$$\left.\begin{aligned} \frac{\partial J}{\partial C_0} &= \int_{x_0}^{x_m} 2e(x)\frac{\partial e(x)}{\partial C_0}dx = 0 \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \frac{\partial J}{\partial C_n} &= \int_{x_0}^{x_m} 2e(x)\frac{\partial e(x)}{\partial C_n}dx = 0 \end{aligned}\right\} \qquad (6\text{-}118)$$

Simultaneous solution of the resultant set of equations yields the desired C's. Actually, it is possible that these calculations will lead to a maximum for J rather than a minimum. Usually, it is not difficult to verify whether a maximum or a minimum is involved. If any doubt exists, the ultimate test for a minimum is that the second derivative of J be positive, when the first derivative vanishes.

As an illustration of this technique, consider the evaluation of the coefficients for a power series to approximate the function e^x over the range of x from 0 to 1. For simplicity in this illustration, the series will be terminated after the third term. Thus,

$$y_p = C_0 + C_1x + C_2x^2 \qquad (6\text{-}119)$$

Then

$$J = \int_0^1 \left[e^x - (C_0 + C_1x + C_2x^2)\right]^2 dx \qquad (6\text{-}120)$$

antl

$$\frac{J}{C_0} = \int_0^1 2\left[e^x - (C_0 + C_1x + C_2x^2)\right]dx = 0 \qquad (6\text{-}121)$$

$$\frac{J}{C_1} = \int_0^1 2\left[e^x - (C_0 + C_1x + C_2x^2)\right]xdx = 0 \qquad (6\text{-}122)$$

$$\frac{J}{C_2} = 2\left[e^x - (C_0 + C_1x + C_2x^2)\right]x^2dx = 0 \qquad (6\text{-}123)$$

Evaluation of these integrals and substitution of tlie numerical value for e gives

$$C_0 + \frac{C_1}{2} + \frac{C_2}{3} = 1.718 \qquad (6\text{-}124)$$

$$\frac{C_0}{2} + \frac{C_1}{3} + \frac{C_2}{4} = 1.000 \qquad (6\text{-}125)$$

$$\frac{C_0}{3} + \frac{C_1}{4} + \frac{C_2}{5} = 0.718 \qquad (6\text{-}126)$$

Solution of these equations yields

$$C_0 = 0.979; \quad C_1 = 1.048; \quad C_2 = 0.645 \qquad (6\text{-}127)$$

The corresponding coefficients for the case where only two terms are carried are

$$C_0 = 0.872; \quad C_1 = 1.692 \qquad (6\text{-}128)$$

The corresponding approximations for e^x are then

$$f_2(x) = 0.872 + 1.692x \qquad (6\text{-}129)$$

antl

$$f_3(x) = 0.979 + 1.048x + 0.645x^2 \qquad (6\text{-}130)$$

These results and the function e^x are plotted in Fig. 6-51.

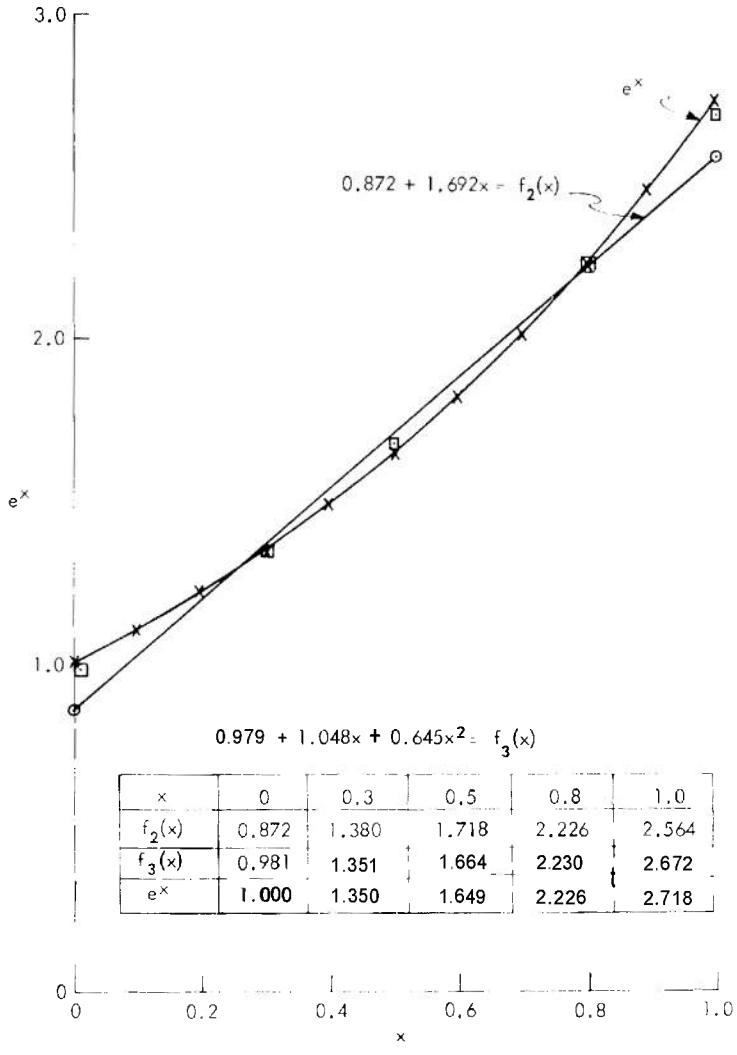
Unfortunately, the problem of carrying out the integrations required to evaluate the coefficients by this means is frequently so difficult, even for simple analytic functions, as to necessitate the use of numerical methods. Furthermore, the problem of solving the resultant set of simultaneous equations, if many coefficients are to be found, is also such as to require the use of machine methods.

Fortunately, if the accuracy desired is not extremely high, it is usually possible to arrive at a satisfactory set of coefficients by trial-and-error plotting of the power series in comparison with the desired function. Furthermore, one is forced to employ this method if no analytic representation of the desired function is available.

In addition to tlie power-series representation just discussed, expansions based upon trigonometric functions, Legendre polynomials, or Tchebycheff polynomials also find use in representing arbitrary functions.

The alternative method of generating arbitrary functions, whereby values of the function are stored for a firite number of discrete values of tlie independent variable and intermediate values are then found by interpolation, also finds extensive use in studies employing analog computers. For convenience, function generators based on this technique are here designated as interpolation-type, function generators.

In tlie simplest and most widely employed schemes, straight-line or linear interpolation is employed to obtain approximate value of the function for values intermediate to those stored. Fig. 6-52 illustrates the method. Here, it is readily seen that for a given arbitrary function, tlie error of approximation depends upon the number of line segments used and their distribution. The segments could be selected on the basis of equal



x	0	0.3	0.5	0.8	1.0
$f_2(x)$	0.872	1.380	1.718	2.226	2.564
$f_3(x)$	0.981	1.351	1.664	2.230	2.672
e^x	1.000	1.350	1.649	2.226	2.718

Figure 6-51. Plots of e^x and its approximations, where e is the error in the function y = f(x).

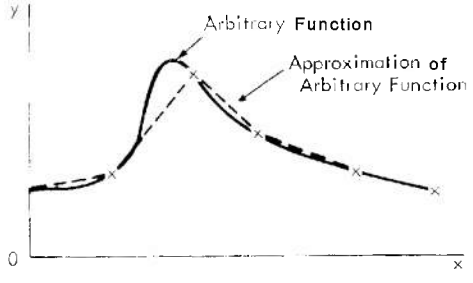


Figure 6-52. Straight-line approximation of an arbitrary function.

increments of the independent variable, or of the dependent variable, or on the basis of the relative curvature of the function at different points. Analytic determination of the optimum scheme to employ is tedious at best and the usual approach is to utilize a sufficiently large number of segments that the errors appear to be negligible for the purposes at hand. In analog studies, the maximum practical number of discrete points utilized is usually set by the design of the function generator itself. However, from a theoretical standpoint, there is no point in utilizing so many steps that the error in representing the discrete values of the function exceeds the errors of approximation between points. This is the round-off problem that arises in numerical calculations.

The interpolation errors could be reduced if a power-series method of approximating the desired function between the stored values were to be employed. However, for analog purposes, the amount of equipment required under this approach is usually so great as to preclude its use.

The representation of functions of two independent variables is inherently much more difficult than is the case of functions of a single variable. When relatively simple bivariable functions are to be represented, it may be possible to approximate them in terms of products or sums of single-variable functions. However, for many physical phenomena, this is impossible and a method designed specifically for generating functions of two independent variables must be employed. A variety of schemes, based on the use of three-dimensional cams or on the storage of values of the dependent variable for a number of values of each of the independent variables and use of interpolation techniques to find intermediate values, have been developed. Some of these are discussed later in this chapter (see par. 6-3.5) in connection with a description of the specific equipment utilized to generate such functions.

6-3 ELECTRONIC DIFFERENTIAL ANALYZERS

General-purpose electronic differential analyzers are now produced by a number of companies. Computing errors of individual elements in these machines vary between 0.02 and 3 percent of full scale, depending upon the mathematical operation involved and the quality of the component, but determination of the overall accuracy to be expected in a specific solution is difficult. Solution time is essentially independent of the problem being solved, but the number of computing elements used increases more or less directly with the complexity of the problem.

For many special-purpose applications, such as ground-based fire control, commercially available computer components can be used. If the signals handled by the control computer ever reach zero frequency, then the amplifiers used must be direct-coupled or d-c amplifiers. RC-coupled or a-c amplifiers have been successfully used for repetitive differential analyzers and simulators, but d-c amplifiers will be required for most real-time control systems. For this reason, the discussion on amplifiers is limited to the d-c operational amplifier.

6-3.1 OPERATIONAL AMPLIFIERS

The design of a d-c amplifier imposes a number of problems that are not encountered in the design of an ordinary a-c amplifier. One problem is that of bias, since each amplifier stage is coupled by resistance networks to the input of the following stage. The voltage level at the output of each stage must either be compatible with the grid voltage at the input of the following stage or else it must be introduced to the grid through an appropriate resistance network fed from a bias voltage supply.

A more serious problem is that of drift. Variations in the supply voltage to the amplifier (including heater supply voltages) cause the output level to change independently of the signal at the input. Changes in tube characteristics resulting from temperature changes or age likewise affect the level of the output voltage. Changes in passive circuit components as a result of temperature, humidity, or age produce the same effect. Considerable attention must therefore be given to the selection of well-regulated power supplies, high-quality vacuum tubes, including an input tube that exhibits very low grid current, and passive circuit elements that are stable in value over the temperature range and humidity conditions under which the

amplifier must operate. Burn-in of all components, including vacuum tubes, is often desirable. The use of differential-type circuits to reduce drift is a standard procedure as discussed in Ref. 9. Another solution to the problem of drift in the first stage of the amplifier is to provide a high-gain, drift-free circuit for the first stage by using a modulator and an a-c amplifier as discussed later. Grid-current effects can be minimized by operating the first-stage plate and screen grid at a low potential and operating the heaters at less than rated voltage. A differential input stage is usually used to compensate for changes in cathode emission, as well as to provide a summing function for the feed-forward loop. The amount of compensation is correct when the transconductance of the tube is equal to tlie reciprocal of the common cathode resistor.

Most amplifier designs that make use of the sampler-a-c amplifier-filter combination use a feedforward loop around tlie a-c amplifier section in order to bypass tlie high-frequency components of tlic signal (and, in most cases, tlie d-c component of the signal as well) to the d-c coupled section of the amplifier. A partial schematic for such an amplifier is shown in Fig. 6-53, from which the compensating networlis have been omitted for tlic sake of clarity. This is the general form of most d-c operational amplifiers having a gain of 10^6 or greater. In Fig. 6-53? it is seen that the grid voltage is sampled and amplified by two stages of amplification in tlic a-c section. The output of the demodulator has the opposite polarity of the grid voltage and is subtracted from the grid voltage itself (in the unity-feed-forward-loop case) at tlie input of the first stage of the d-c section of the amplifier through the action of the differential stage. Capacitance-diode coupling is sometimes used in the feedforward loop, as indicated, in order to reduce the effect of grid current.

The a-c section consists of an electromechanical vibrator or chopper, a two-stage a-c amplifier, a demodulating circuit, and an RC filter network having a large time constant. The d-c section consists of three stages of direct-coupled circuitry. The chopper is driven from an a-c voltage source of 60 to 400 cps. The voltage e_1 is grounded during a portion of each cycle of tile driving voltage of the chopper. If the voltage e_g is different from 0, the voltage at the grid of the first tube V_{1a} is a series of pulses that can be amplified by the a-c amplifier. The output of the second stage of the a-c amplifier is coupled to a diode-type demodulating circuit that is driven from the same voltage source that drives the chopper. The demodulated voltage is then filtered by the filter consisting of resistance R and capacitance C, which results in a slowly varying d-c voltage e_2 at the grid of V_{2a}. This comprises the entire a-c section and is quite similar in most commercial designs. This section is usually designed for a d-c gain of 1,000 to 3,000. The output of V_{1b} is also used to drive an overload circuit that provides a warning when the voltage of the a-c amplifier section exceeds a fixed value. This provides a convenient but not absolute indication of overloading of the d-c section.

The input stage to the cl-c section consists of a differential-type circuit that uses two halves of a twin triode and a common cathode resistor. This stage amplifies the sum of three voltages: (1) the output voltage of the a-c section filter, (2) a bias voltage obtained from a balanced potentiometer (both being summed at the grid of V_{2a}), and *(3)* the voltage e_3 from tlie feedforward loop. The second stage of the d-c section consists of a cathode follower that drives an amplifier stage. The cathode follower is often omitted and a high-gain pentode used instead of a twin triode for the second stage. The last stage consists of two tubes (usually two halves of separate twin triodes) used in a cascade-type circuit. From Fig. 6-53, one might conclude that tlie high-gain operational amplifier circuit is rather simple; two such circuits could be built on a single chassis and would require only eight vacuum tubes and one chopper between the two amplifiers. Sote that only one pole of the chopper contactor is used for the amplifier shown; tlie other pole could be used for a second identical amplifier. on the same chassis. The cascade circuit requires different halves of two different tubes because the filament supply of V_4 usually must be biased with a negative voltage.

Table 6-2 gives specifications which are typical of amplifiers such as the one indicated in Fig. 6-53.

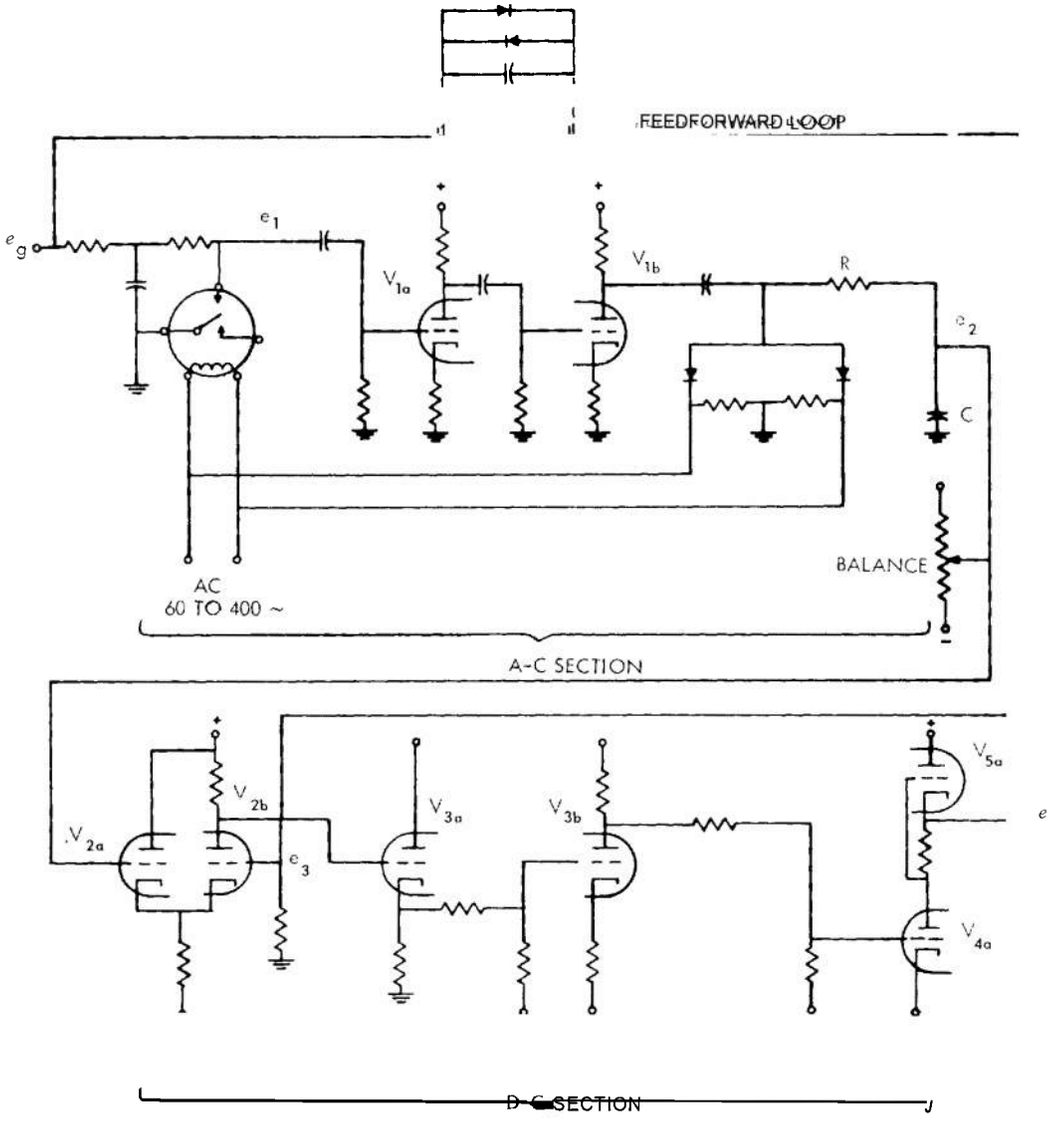


Figure 6-53. A typical operational amplifier,

If an operational amplifier meets the specifications of Table 6-2, the errors discussed earlier will usually be less than 0.1 percent over an operating range of 0 to 100 cps. This statement assumes, of course, that the amplifier is not misused; for example. the output load must be kept within the specified limits. The computing elements that are associated with the amplifiers in a high-quality computer are usually matched to tolerances of ± 0.1 percent or better.

6-3.2 MULTIPLIERS

Although a great many schemes have been prepared for performing multiplication in

TABLE 6-2. TYPICAL OPERATIONAL-AMPLIFIER SPECIFICATIONS.

GAIN	a-c section: 1,000 to 3,000 d-c section: 50,000 to 100,000 overall: greater than 10^8
D-C DRIFT	Referred to summing junction: < 0.25 mv/day Integrator (with 1.0 μf capacitor) Standby state -- < 100 mv/15 minutes Operate state -- < 100 mv/90 minutes
V_{2a} GRID CURRENT	maxinium: < 100 $\mu\mu$a average: < 30 $\mu\mu$a
FREQUENCY RESPONSE	open loop: flat to 0.005 cps, -6 db slope to 50 kc unity inverter (with 1 M resistors): bandwidth 10 kc to 30 kc with little or no resonant peak max. phase shift at 100 cps: 0.15° unity inverter (with 0.1 M resistors): bandwidth 30 kc to 100 kc max. phase shift at 100 cps: 0.1°
BANDWIDTH	

Gain	Resistors	Capacitor	Summer	Integrator
1	1 M	1.0 μf	10-15 kc	10 kc
4	0.25 M	1.0 μf	8-10 kc	10 kc
10	0.1 M	1.0 μf	8 kc	9 kc
0.1	1 M	1.0 μf	100 kc	130 kc

OUTPUT VOLTAGE RANGE	±125 volts
MASIMTJM CURRENT OUTPUT	20 ma
MINIMUM LOAD IMPEDANCE	5,000 ohms
SATURATION 0.01%	

to CPS	at Load Current
100	5 ma
70	10 ma
25	20 ma

OUTPUT IMPEDANCE	open loop, 500-1,000 ohms
NOISE LEVEL	referred to summing junction, 5.0 mv max., peak-to-peak

electronic analog computers*, only a few of these have found any real application. Two of these schemes are discussed here; namely:

(1) Time-division multiplier.
(2) Quarter-square multiplier.

6-3.3 Time-division Multiplier

The time-division multiplier is a pulse-width, pulse-amplitude multiplier, also called time-division. It operates on the principle that the average value E_a of a train of rectangular pulses (see Fig. 6-54) can be expressed as

$$E_a = \frac{t_1 - t_2}{T} E \qquad (6\text{-}131)$$

where t_1 is the width of the positive portion of each pulse cycle, t_2 is the width of the negative portion of each pulse cycle, and $T = t_1 + t_2$ (see Fig. 6-54). Through use of appropriate circuitry, the times t_1 and t_2 are controlled in such a manner that the average voltage E_a is given by the relationship

$$E = -\frac{v_1}{v_3} E \qquad (6\text{-}132)$$

where v_1 and v_3 are two voltage inputs to the multiplier. If the pulse-train amplitude E is made proportional to the multiplier input v_2 and the multiplier output v_o is made proportional to E_a, the operation of the multiplier can be expressed in the form

$$v_o = k \frac{v_1 v_2}{v_3} \qquad (6\text{-}133)$$

where k is a design constant.

The principle of this type of multiplier is relatively simple, and conventional pulse-circuit techniques initially were utilized in their design. However, the accuracy was limited by difficulties associated with accurate time division of the waveform and control of the amplitude. These difficulties are minimized by the use of a feedback system, to establish the proper timing; and a high-precision feedback-type electronic switch, to make the characteristics of the multiplier essentially independent of the tube characteristics.

To minimize the multiplier errors resulting from a finite switching time, the waveforms must have extremely steep sides. Lowering the basic repetition frequency alleviates this problem but increases the problem of filtering. Consequently, the choice of repetition frequency is a compromise.

For favorable combinations of inputs, the errors in a time-division multiplier can be held below 0.1 percent of full scale. As a result, time-division multipliers are used widely in analog-computer work.

Actually, several types of electronic multipliers develop an output of the form $v_1 v_2 / v_3$ (see Eq. 6-133), with the result that either multiplication or division is possible -- depending on which inputs are employed. Because division by a small number yields a large output, care must be exercised that the divisor does not become too small.

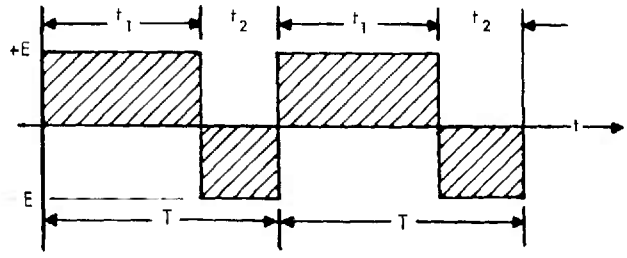


Figure 6-54. Basic waveform of a time-division multiplier.

* *See*, for example, pages *302-306* of Ref. 10.

6-3.4 Quarter-square Multiplier

Probably the most accurate and widely used electronic multiplier is the electronic quarter-square multiplier. Mathematically, an electronic quarter-square multiplier is the same as its mechanical equivalent (see par. 6-4.3). The only real difficulty encountered in building an accurate electronic multiplier is associated with the generation of the squares. The approximate square-law characteristics that can be generated directly with vacuum tubes or thyrite resistance elements give minimum errors that exceed 2 percent. A resistance network and a group of diodes can generate a straight-line-segment representation of a square-law function as described under function generation in par. 6-3.3. The errors in quarter-square multipliers of this design can be held to 1 percent or less.

6-3.5 FUNCTION GENERATORS

Currently, the most widely used means for generating arbitrary functions electronically is the diode function generator. Functions that are adequately represented by two or three straight-line segments can be generated using very simple diode-resistance networks such as shown in Fig. 6-55. Frequently, these networks are associated with an operational amplifier, as shown in Fig. 6-55(C). These function generators utilize the characteristic that an ideal diode offers no resistance to current flow in one direction but offers infinite resistance to current flow in the opposite direction.

This basic scheme can be generalized to permit generation of arbitrary functions as shown in Fig. 6-56(A). For the generation of functions of a single variable, the voltage v_2 is fixed with the polarity and amplitude re-

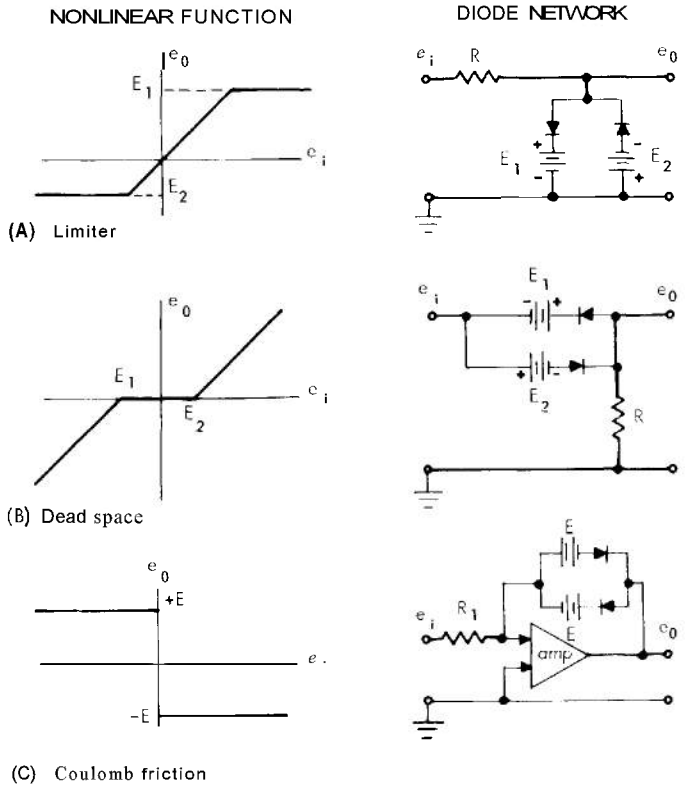


Figure 6-55. Diode networks used for generating three simple nonlinear functions.

yuired to give the desired output when the independent variable v_1 is zero. With voltage v_2 a negative constant, the circuit is suitable for generation of functions of the type illustrated in Fig. 6-56(B). When $v_1 = 0$, no diode conducts, and the function intercepts the $f(v_1)$ axis at $f(0) = -(R_f/R_{v2})\, v_2$ and has a slope of $m_0 = -R_f/R_{v1}$. As the input v_1 is increased in the positive direction, one diode after another begins to conduct in accordance with the settings s_i of the potentiometers P_1 through P_n. These break points occur at $v_{IBi} = -s_i/[(1-s_i)v_2]$. As each diode conducts, it connects a new input to the summing circuit ancl contributes a slope increment, $(R_f/R_i)(1-s_i)$. The figures shown apply specifically to a function having a positive value for f(0) and a negative slope that increases constantly as v_1 increases. However, by proper selection of the polarities of the voltages v_1 ancl v_2 and the diode connections, this scheme can be extended for the generation of functions lying in any of the four quadrants. By suitable combination of several of these basic circuits, it is possible to represent a function whose slope changes sign.

Because a great variety of functions can be setup in a straightforward manner on this type of generator, it is being uscd widely. The accuracy achieved depends on the particular function being generated, on the number of line segments used to represent the function, antl on the overall stability of the circuit. Commercially available generators utilize 8 to 20 diodes, with the possibility of coupling two units for the generation of a single function antl thereby providing up to 40 segments. With such units, a very wide variety of functions can be generated with errors of less than one percent.

Examination of the equations presented along with the discussion of the operation of tlie generator of Fig. 6-56 shows that both the point at which tlic generated function intercepts the z-axis and the break points of tlic function are directly proportional to the bias voltagc v_2. Consequently, a family of curves of the class shown in Fig. 6-57 can be generated directly by tliis method if the bias v_2 is macle proportional to the second variable. Other bivariable functions in which tlie intercepts on the $f(v_1)$ axis are not spaced uniformly with the variable v_2 can be generated with this unit if the bias voltage is made proportional to a function of v_2 rather than to v_2 itself. However, the added restriction is imposed that the breakpoints must follow the same functional relationship in order that the boundary curves ℓ_i remain straight lines. For functions having break points distributed in the same proportion on all the boundary curves, a relatively simple extension of the same method can be used. In the more general cases where the boundary curves of the function do not possess this simple property, the method is still theoretically possible but usually not practicable because an excessive amount of equipment is required.

A function generator that is basically an all-electronic curve follower employing a cathode-ray tube, an opaque mask, and a photocell received considerable attention ten years ago but has now been largely replaced by the diode function generator. In units of this type, generally referred to as a photoformer, a mask, with its edge cut in the shape of the desired function, is mounted close to the face of a cathode-ray tube, and a photocell is mounted in the front of the tube face in such a way as to pick up light from the fluorescent spot (see Fig. 6-58). The output of the cell is amplified and used to control the y position of the spot in such a way that as the spot is moved in the x direction it is made to ride along the edge of the curve with approximately one-half its area hidden by the mask. The y deflection voltage required to maintain this condition is thus proportionnlto the ordinate of the curve and can be taken as the output of the Function generator.

Although photoformers have been used extensively, they have several disadvantages. In particular, precise initial calibration of the unit is relatively difficult, and operation is highly subject to drift. Consequently, it is difficult to hold the errors in such a function generator below 3 percent for periods of more than a few hours. With the development of other types of function generators, the use of pliotoformers is decreasing.

6-3.6 DECISION UNITS

Frequently in the simulation of a physical system or in a computer used as part of a complex system, the need arises to perform logical operations based upon a timing se-

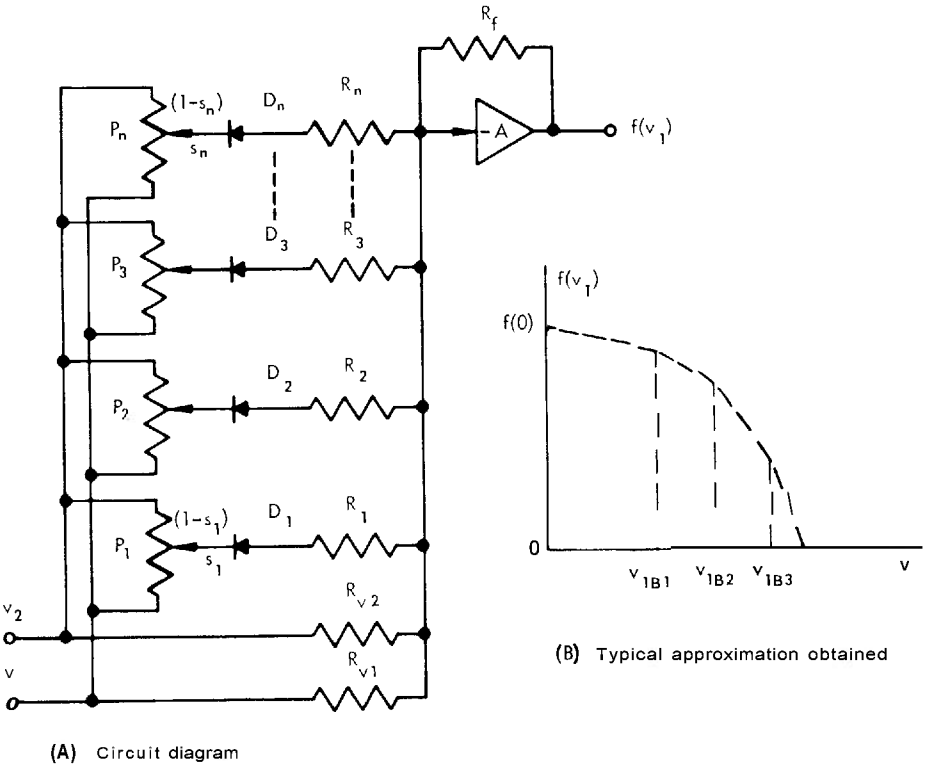


Figure 6-56. Approximation of an arbitrary function by means of a diode function generator.

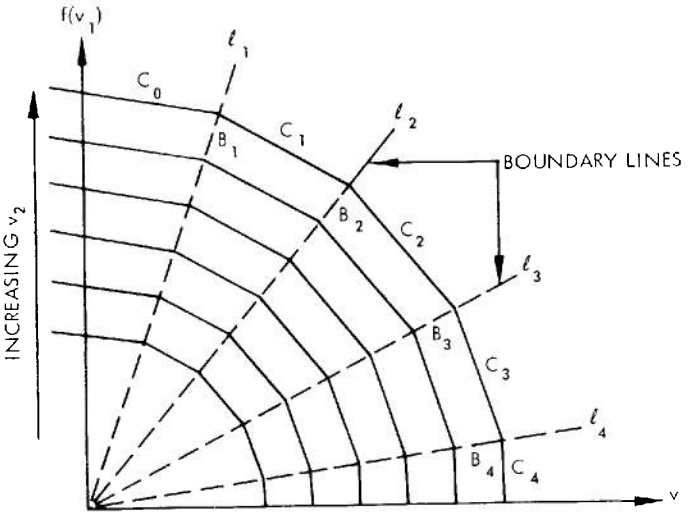


Figure 6-57. Simple bivariable function.

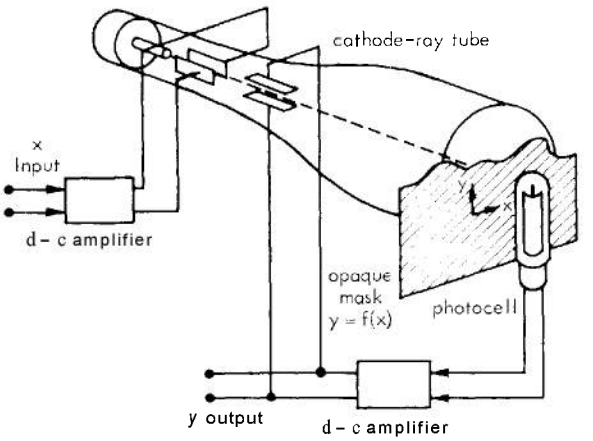


Figure 6-58. Basic form of the photoformer function generator.

quence or upon the amplitudes of prescribed variables within the computer. A simple example arises in connection with the establishment of a sequence for automatically setting a computer into the "initial condition" state, turning on the recorders, running a solution, turning off the recorders, resetting the initial conditions and, if desired, repeating the sequence after the change of some prescribed parameter. These operations can be done with simple timers and relays.

Other decision elements now widely used in analog computers are "voltage comparators" and "sample-and-hold" units. A voltage comparator is a unit that provides for the opening or closing of a switch (this may be a set of relay contacts or an electronic switch) when a computer voltage signal becomes just greater than or just less than a reference voltage, which itself may be fixed or may be another computer variable.

This operation is performed by supplying the two voltages to a summing circuit that feeds a high-gain amplifier, the output of which is arranged to drive the control coil of a relay or to close an electronic switch. Diodes are connected to the relay in such a manner that the relay is actuated by voltages of only a prescribed polarity. Additional diodes in the amplifier insure that it responds very quickly after being saturated. The high gain provided by the amplifier causes the relay to pull in or drop out with a very small change in the signal voltage above or below the comparison voltage.

Sample-and-hold units are now being used extensively in computations involving data that are available only at discrete points in time, and in carrying out mathematical operations that call for sampling a computer variable and storing its value for a prescribed time or until another variable has reached a predetermined value. Sampling can be carried out with a high-gain amplifier similar to that used in a voltage comparator and driving a high-speed relay or an electronic switch. For a sampling operation, the input to this amplifier takes the form of a short voltage pulse that closes the relay at the moment the sample is to be taken. The closure of the switch applies the voltage to be sampled to a buffer amplifier having unity gain and a very low output impedance. This amplifier charges the capacitor of an integrator in a manner similar to that used in establishing initial conditions on an integrator. At the end of the sampling pulse, the capacitor is disconnected from the charging (amplifier and becomes the feedback capacitor of an integrator. If the integrator input is zero and the capacitor charge does not leak off, the voltage output of the integrator will maintain the sampled value until a new sample is taken. A sample-and-hold circuit must be designed in such a way that the sampling pulse is of sufficiently short duration that

the voltage being sampled does not change value appreciably during the sampling time, but is sufficiently long that the holding capacitor can be charged very nearly to the sampled value during the sampling interval.

6-3.7 INPUT-OUTPUT EQUIPMENT

6-3.8 Input Equipment

Inasmuch as no input, other than the initial conditions, is involved in obtaining the homogeneous solution to a differential equation, and a simple switch-closure provides the forcing function for obtaining the step response of a system, a great deal of work is done on analog computers without any requirement for generating complex input signals. For computers utilized as part of an operating system, the inputs may be derived from a tracking unit (for example, a radar tracking antenna), or from temperature, pressure, or mechanical-position transducers. However, two types of input equipment that deserve brief comment are (1) the reference-voltage supplies used in establishing the initial-condition voltages on integrators and comparators, and (2) the noise generators used in carrying out studies of the performance of systems when subject to random inputs or random disturbances.

6-3.9 Reference Voltage Supplies

The normal practice is to provide both positive and negative reference voltages equal to the maximum voltage at which the majority of the computing components are designed to operate; common reference voltages are +100 and -100 volts. When computers are used to study linear constant-coefficient differential equations, the accuracy of the solution does not depend* on the exact value of the reference voltage or even on the constancy of the reference if all reference voltages used in the computer are of equal amplitude and vary in the same way. If, however, a computer is to be employed to study differential equations that contain variable coefficients or nonlinear terms, appreciable errors may be introduced unless the reference voltages are well regulated. Consequently, a reference supply that is well regulated and has a low internal impedance is generally essential. Furthermore, the wiring system used to distribute the reference voltage should have a very low impedance if the reference voltage is to be the same in all parts of the computer.

The requirements for the reference-voltage supply are very similar to those for the regulated power supplies used elsewhere in the computer (see par. 6-5.1 through par. 6-5.6) except that the voltage is lower. Either a battery or a special regulated power supply may be used, but a regulated supply is generally preferable. However, several modifications of the usual regulator circuit are found in reference-voltage supplies. First, gas-discharge tubes make satisfactory voltage standards in regulated power supplies, provided relatively slow variations in the output voltage can be tolerated. If the output is to be maintained at an absolute voltage, however, some standard other than a gas tube must be employed. In spite of its low voltage, a standard cell has been one of the most satisfactory voltage standards, but Zener diodes are now being used extensively for this purpose. Second, since the positive and negative reference voltages should be of exactly the same magnitude, both voltages should be derived from the same reference supply. This type of operation can be achieved by regulating one supply from the voltage standard and the second supply from the output of the first.

6-3.10 Noise Generators[10]

In computer studies, the usual requirement for a noise generator is that the random-signal output have a Gaussian amplitude probability distribution and that its power spectrum extend from essentially zero frequency to a maximum frequency of 30 to 40 cps. The scheme generally used for the generation of such signals is shown in block-diagram form in Fig. 6-59. The output of the gas tube extends from zero frequency up to a maximum determined by the bandwidth of the circuitry used to amplify the signal. However, because the relative power in the very-low-frequency components of the output

* Provided the sensitivity of the unit on which the solution is recorded is also controlled by the reference voltage.

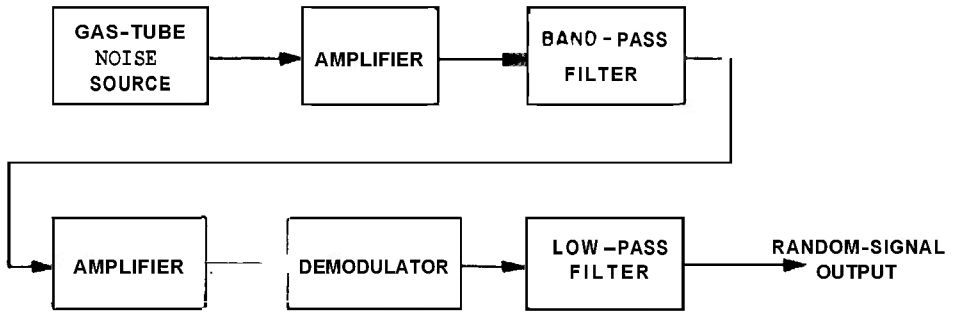


Figure 6-59. Block diagram of a random-signal generator,

changes randomly with time and because a d-c voltage exists across the tube, the scheme shown in Fig. 6-59 offers a better method for developing the desired signal than if conventional d-c amplification were employed. Here, the output of the gas tube is amplified in an a-c coupled amplifier with a band pass somewhat wider than that of the filter that follows it. The signal then is passed through the filter, amplified some more, and finally demodulated in a keyed demodulator operating at the center frequency of the filter and passed through a low-pass filter to eliminate the upper sideband of the demodulator output. If an output signal with a power spectrum that is flat in the range from 0 to 100 cps or less is desired, the band-pass filter can be centered at 400 cps, and either a mechanical chopper or a vacuum-tube demodulator keyed at 400 cps can be used for demodulation.

Unfortunately, the amplitude of the gas-tube output changes erratically from time to time by amounts of 10 percent or more. This characteristic makes the generator represented in Fig. 6-59 unsuitable as a random-signal source if accurate data on system performance are to be collected. The difficulty can be eliminated if the output of the second amplifier is fed also to an averaging detector whose output, after being passed through a filter with a long time constant, is applied to the first amplifier as an automatic-gain-control voltage.

If random-signal generation equipment is to be used in computer studies, means must be provided for shaping the power spectrum of the random signal injected into the setup, as required for the particular study and for monitoring random signals in the system.

The basic mathematical expression used in malting random-signal calculations for linear systems relates the power spectra at the input and output of the system. If the system function is specified as $H(j\omega)$, the power spectra of the input $\Phi_{in}(\omega)$ and of the output $\Phi_{out}(\omega)$ are related by the expression

$$\Phi_{out}(\omega) = |H(j\omega)|^2 \Phi_{in}(\omega) \qquad (6\text{-}134)$$

Furthermore, if the mean-square value of the output, whether it be a mechanical motion or a voltage, is denoted by $\overline{E^2_{out}}$, then

$$\overline{E^2_{out}} = \int_{-\infty}^{\infty} \Phi_{out}(\omega) d\omega \qquad (6\text{-}135)$$

Consequently,

$$\overline{E^2_{out}} = \int_{-\infty}^{\infty} |H(j\omega)|^2 \Phi_{in}(\omega) d\omega \qquad (6\text{-}136)$$

A proper choice of the function $H(j\omega)$ allows assignment of various meanings to the quantity E_{out}. Once $H(j\omega)$ and $\Phi_{in}(\omega)$ are known, $\overline{E^2_{out}}$ can be calculated conveniently with the aid of a table::: that allows evaluation

* *See* table of integrals in the Appendix of Ref. 10.

of integrals of the form appearing in Eq. 6-135 by purely algebraic means.

Determination of the function $H(j\omega)$ that is required to produce a desired shape of power spectrum is simplest if $\Phi_{in}\ \omega)$ can be considered a constant over the frequency band of interest. This condition is equivalent to the statement that the output of the random-signal generator must be essentially white noise for the frequencies involved. consequently, the output filters provided with noise generators are designed to give an output spectrum that is essentially flat up to a frequency higher than will be used in computer studies. Final shaping of the power spectrum can then be accomplished with filters employing standard computing elements.

The problem of measuring or monitoring random signals is basically one of determining the mean-square value of a random signal whose power spectrum is confined to frequencies of a few cycles per second. This operation can be instrumented, as shown in Fig. 6-60, by passing the signal first through a full-wave rectifier; then through a squaring unit, which may be a diode function generator of the type described in par. 6-3.5; then through a filter with a long time constant; and finally displaying the result on a d-c meter. Because a statistical signal is being measured, the meter reading will show variations about the true output E^2_{out} . The amplitude of these variations depends upon the type of filter inserted between the squaring unit and the meter. An approximate value of the expected error can be obtained if a signal having a rectangular power spectrum flat in the range from $-\omega_c$ to $+\omega_c$, as shown in Fig. 6-61, is taken as the input to the monitor. If a simple-lag, low-pass filter having a transferfunction $1/(\tau_F\ s+1)$ is used (in which s is the complex frequency variable), then the probable error in any observation of the mean square of the applied signal is given by

$$\text{Error} = \sqrt{\frac{\pi}{\tau_F \omega_c}} \qquad (6\text{-}137)$$

where

τ_F = time constant of the filter

Consequently, if signals having a cutoff frequency as low as 12 rad/sec are to be observed with an error of 5 percent or less, the filter must have a time constant of at least 120 seconds. If greater accuracy is desired or if a filter with a shorter time constant is to be used, a series of uncorrelated measurements could be taken, and the results averaged. Then, the error would be approximately $1/\sqrt{n}$ times the error in a single observation.

The random-signal-generating equipment discussed thus far provides signals with a Gaussian amplitude probability distribution. Two other types of signals used to a lesser degree in control-systems studies are square waves with a fixed 'amplitude but random zero-crossing times, as shown in Fig. 6-62(A), and signals that change amplitude at equal time intervals but assume any arbitrary amplitude, as shown in Fig. 6-62(B).

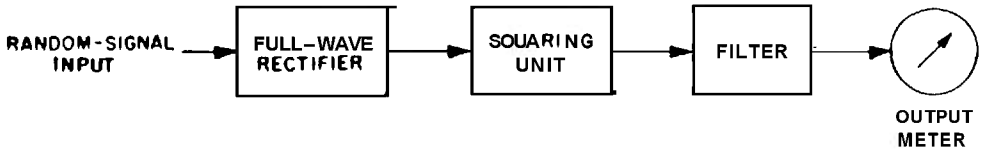


Figure 6-60. Block diagram of a noise monitor.

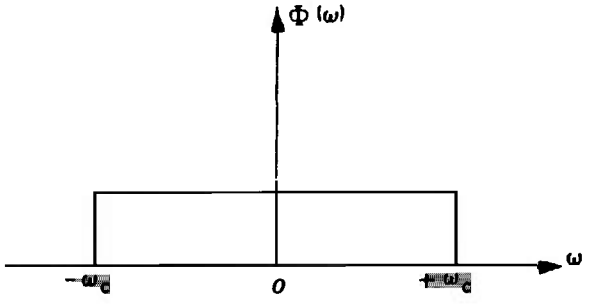


Figure 6-61. Rectangular power spectrum.

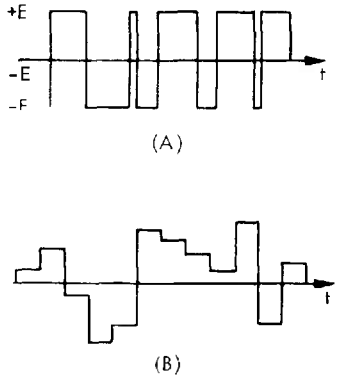


Figure 6-62. Non-Gaussian random signals.

6-3.11 Output Equipment

The output equipment used with analog computers consist of stripchart recorders, plotting boards, digital voltmeters, and oscilloscopes. These units are discussed in pars. 6-5.10 through 6-5.14.

6-4 MECHANICAL AND ELECTROMECHANICAL DIFFERENTIAL ANALYZERS [2]

6-4.1 SUMMATION DEVICES

Addition can be performed mechanically with differentials made up of linkages, racks, or gears as illustrated in Fig. 6-63. The form of device used to perform mechanical addition is dictated largely by whether translational or rotational mechanical motions are to be added. The linkage differential sums two linear motions and gives a linear output, as does the rack and gear unit also. Two rotary motions are summed to give a linear output in a screw differential, while two rotary inputs yield a rotary output in a gear differential. Each type of mechanical differential is subject to fabrication errors which lead to backlash. Therefore, larger elements can be made with smaller errors. Instrument gear differentials, using gears approximately one inch in diameter, are available with backlashes as low as 5 minutes of arc.

6-4.2 INTEGRATORS

The geometry of a classical Kelvin disk-disk mechanical integrator is shown in Fig. 6-64. Integrators of this type were used in the early differential analyzers built at the Massachusetts Institute of Technology by Rush and in many fire-control computers built during World War II. In this integrator:

- x = angular position of large input disk
- y = radial position of small disk as measured from center of large disk
- r = radius of small disk
- g = scale factor relating angular rotation of the output shaft to that of small disk
- z = angular position of output shaft

An expression relating a differential rotation dz of the output shaft to a differential rotation dx of the input shaft can be written directly from the geometry of the system as follows:

$$2\pi y\, dx = 2\pi gr\, dz \qquad (6\text{-}138)$$

or

$$z = \frac{1}{gr}\int y\, dx \qquad (6\text{-}139)$$

Mechanical integrators are also widely used as continuously variable speed changers. In commercially available units, the size of the large disk ranges from 1.5 to 5 inches. These units utilize hardened steel alloys. The typical accuracy specification for a 1.5-inch unit is 0.5 percent for loads up to 1 inch-ounce.

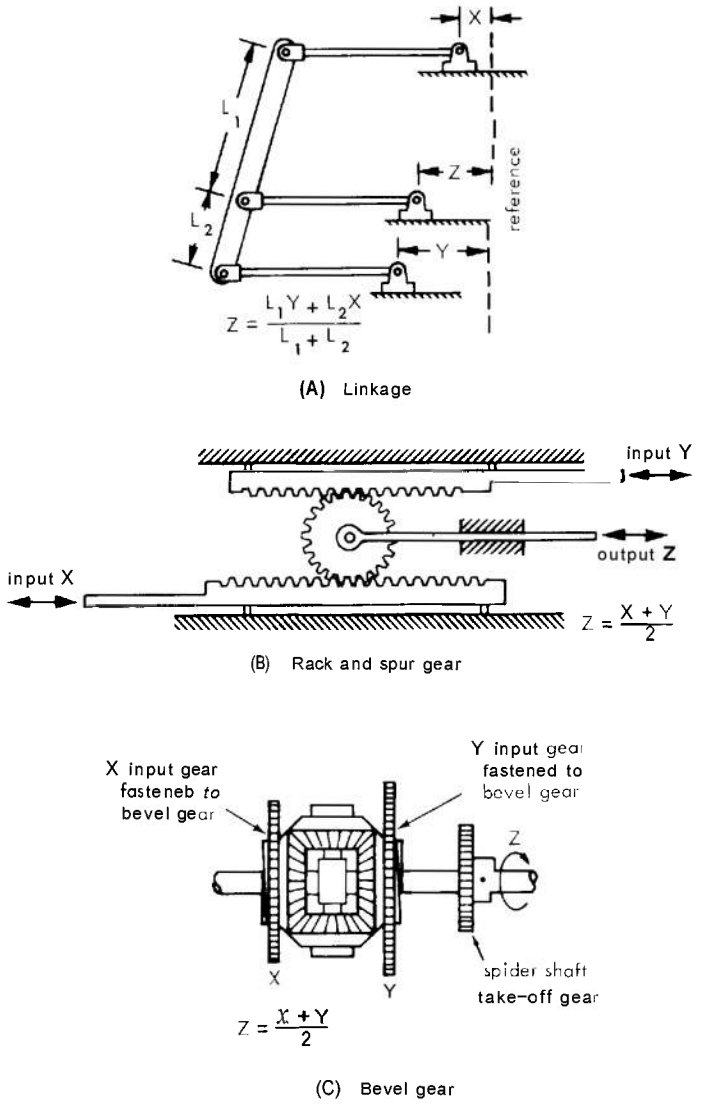


Figure 6-63. Typical mechanical differentials,

Accuracy improves as the disk size is increased.

The primary deficiency of the disk-disk integratoris that it can supply only a limited load torque. An increase in the output-torque capabilities requires an increase in the compressive load between the disks, but this in turn increases the force required for sliding the small disk. If the solution of a problem requires that the output of one integrator drive the input of another, a difficult compromise results. This limitation on the disk-disk integrator is partly obviated in the ball-and-disk integrator, where rolling friction replaces sliding friction.

In the M.I.T. differential analyzers, torque amplifiers were used to obtain increased output torque. The original units employed mechanical torque amplification; the later machine used a servo followup system.

A significant advantage offered by a mechanical integrator is that each of its inputs can be a function of any arbitrary variable.

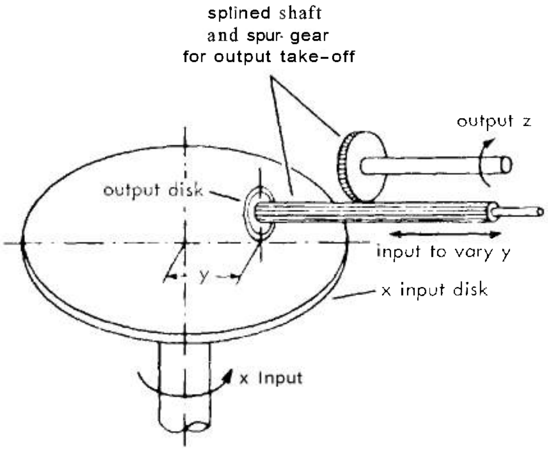


Figure 6-64. Geometry of the disk-disk integrator.

Consequently, a mechanical integrator is not restricted to performing integration with respect to time as is an electronic integrator. Because of this feature along with their reliability and relative simplicity, mechanical integrators find considerable use in special purpose computers.

Integration is accomplished electromechanically with a rate servomechanism (see Fig. 6-65). Servo action causes the voltage developed by the tachometer v_t to equal the input voltage v_1. If k_t is the gain of the tachometer in volts/(rad/(sec) for the particular reference employed and if the tachometer turns through an angle θ, the motor runs at a speed such that

$$v_t = v_1 = k_t(d\theta/dt) \qquad (6\text{-}140)$$

If the potentiometer is geared 1:G to the tachometer and supplied with a voltage v_2, the output voltage v_o is given by the relationship

$$v_o = \frac{v_2}{G k_t \theta_F}\int v_1\,dt \qquad (6\text{-}141)$$

where θ_F is the angular rotation of the potentiometer, corresponding to the voltage v_2. A servo of this type employing a drag-cup tachometer represents the most satisfactory technique available for integrating a signal in the form of a suppressed-carrier a-c voltage. An accuracy of better than 1 part in 1000 can be achieved, but the frequency response is limited.

6-4.3 MULTIPLIERS AND DIVIDERS

6-4.4 Mechanical Multipliers

The operation of multiplying a computer variable by a constant can be achieved mechanically by either a simple gear ratio or a lever system.

Multiplication of one computer variable by another can be performed by (1) interconnection of a pair of integrators, (2) linkage mechanisms based upon similar triangles, or (3) square-law or logarithmic gears or cams.

Mechanization of the mathematical relationship

$$z = xy = \int_{y_o}^{y} x\,dy + \int_{x_o}^{x} y\,dx \qquad (6\text{-}142)$$

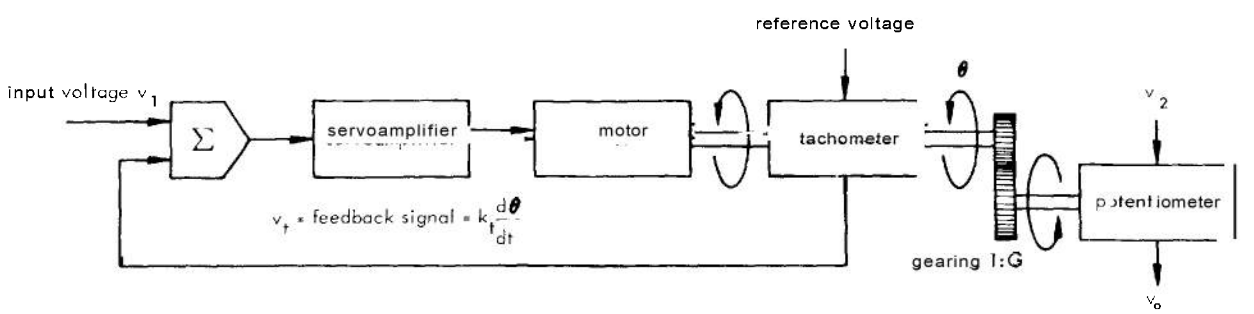


Figure 6-65. Block diagram of a rate-servo integrator

indicates how multiplication canbe performed with a pair of generalized integrators plus a differential for summing their outputs. The interconnections required are shown schematically in Fig. 6-66.

Linkage multipliers based upon similar triangles can take several forms; Fig. 6-67 shows the basic idea involved. The T-shaped member is free to rotate about the axis O. Provision is made for positioning the member B along the fixed slide C and for positioning the pin P_2 in slide A, which is parallel to C. By similar triangles, it is evident that

$$\frac{z}{y} = \frac{x}{k} \quad \text{or} \quad z = \frac{xy}{k} \qquad (6\text{-}143)$$

Thus, the product is obtained as the distance z of the pin P_1 from the slide C.

A mechanical quarter-squares multiplier based upon the identity

$$xy = \frac{1}{4}\left[(x+y)^2 - (x-y)^2\right] \qquad (6\text{-}144)$$

can be mechanized with squaring cams or square-law spiral-face gears, each of which is discussed under function generation (see par. 6-4.16). Fig. 6-68 is a schematic representation of such a multiplier.

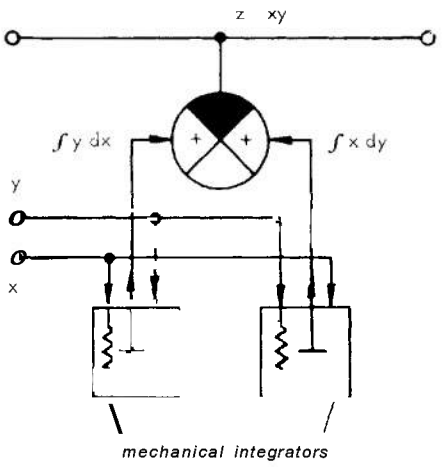


Figure 6-66. Schematic representation of multiplication by means of a pair of integrators plus a differential.

6-4.5 Servomultipliers

A servomultiplier includes a control potentiometer and a multiplying potentiometer, which are mechanically coupled and driven by a servomotor (see Fig. 6-69). The control potentiometer is excited from the multiplicand voltage v_2. The servo zeroes its error voltage by rotating the arm of the control potentiometer to a position such that the voltage at

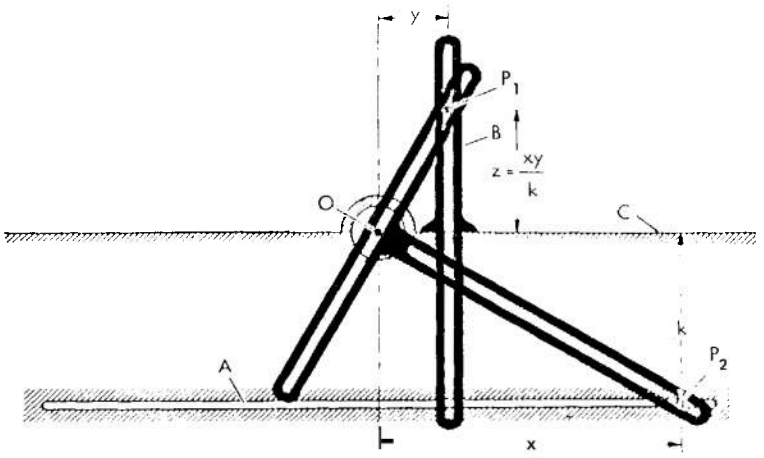


Figure 6-67. Linkage multiplier.

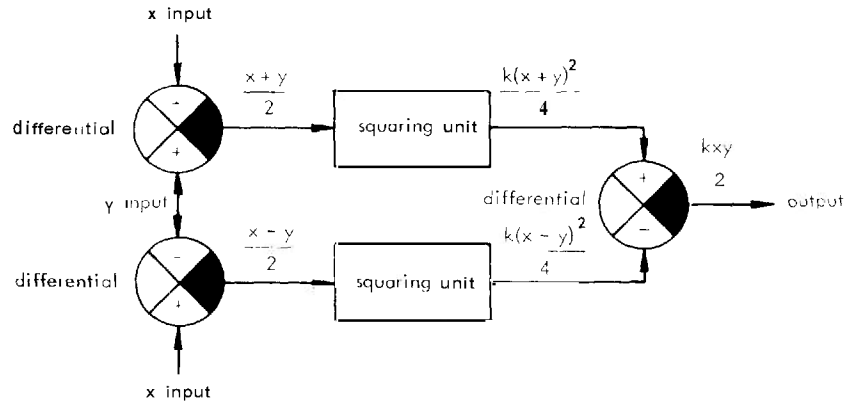


Figure 6-68. Schematic representation of a quarter-squares multiplier.

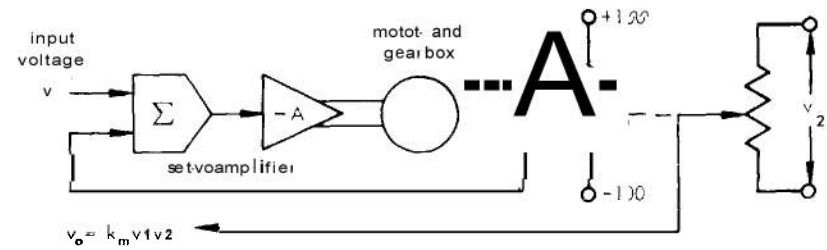


Figure 6-69. Schematic representation of a servomultiplier,

the potentiometer slider equals some fixed constant times v_1. The common position of the two potentiometer sliders is, therefore, proportional to v_1. If the two potentiometers are both linear and if they track exactly, the voltage at the slider of the multiplying potentiometer can be expressed as

$$v_o = k_m v_1 v_2 \qquad (6\text{-}145)$$

where k_m is the gain factor of the multiplier.

If both potentiometers are excited with voltages that are balanced to ground, true four-quadrant multiplication is achieved. Several multiplying potentiometers can be ganged with a single reference potentiometer, so that one voltage v_1 can be multiplied by several other voltages with a single servo unit. Nonlinear multiplying potentiometers can be used if a multiplication of the form

$$v_o = k_m v_1 f(v_2) \qquad (6\text{-}146)$$

is desired.

Accuracies of the order of 0.05 percent can be achieved with servomultipliers. For low-frequency applications, where their limited bandwidth and acceleration capabilities are adequate, servomultipliers find wide application.

6-4.6 Mechanical Dividers

The practical difficulty of performing division by interchanging the output and one of the inputs of a multiplier is discussed in par. 6-2.13. Specifically, the quotient approaches infinity as the divisor approaches zero, which is an operation exceeding the capacity of any physical device. Furthermore, even within the capacity of the device,

when the divisor is small, a high input torque is required and friction may make the device completely inoperable. This latter difficulty can be avoided by interconnecting a multiplier, a differential, and a motor in the form of a servo loop, as shown in Fig. 6-70.

If the assumption is made that the quotient z is within physical limitations, z can be multiplied by the divisor y to give zy. Subtraction of zy from the dividend x yields an error signal e that can be transformed into an electrical signal, amplified, and used to drive the z input of the multiplier. If the gain of the servo loop is high, the servo will cause the error to approach zero. Thus,

$$x - yz = 0 \quad \text{or} \quad z = x/y \qquad (6\text{-}147)$$

In an alternate approach, the reciprocal of the divisor is obtained from a function cam (see par. 6-4.16) and this result is multiplied by the dividend in a conventional multiplier.

6-4.7 Electromechanical Dividers

The position-servo scheme used for multiplication can be rearranged as shown in Fig. 6-71 to permit division. The assumption may be made that the motor velocity is related to the error voltage v_e by the constant k. The output of the control potentiometer is represented as its input voltage times the factor θ/θ_F, where θ is the angle the slider has been moved from the zero output position and θ_F is the full-scale rotation. The error voltage can be written as

$$v_e = v_2 - v_3 \frac{\theta}{\theta_F} \qquad (6\text{-}148)$$

In the steady state, θ equals $\theta_F v_2/v_3$, and the output of the potentiometer excited from v_1 is $v_1\theta/\theta_F$, which equals $v_1 v_2/v_3$. Because the loop gain (kv_3/θ_F) of this system varies directly with the input v_3, the system is sluggish for small values of v_3, but may oscillate for large values of v_3. This difficulty is overcome by passing the error signal through a third potentiometer, as shown in Fig. 6-72.

Division can also be performed with a single, linear, tapped potentiometer, as shown in Fig. 6-73, provided that the divisor is never less than a prescribed value. This scheme is particularly useful in cases where the divisor is in the form of a shaft angle Q. If operation is restricted to the section of the potentiometer below the tap point, i.e., where $\theta > 0$, the ratio of output voltage to input voltage can be written as

$$\frac{e_o}{e_i} = \frac{R_L}{\rho}\left(\frac{1}{\phi}\right) \quad \text{for} \quad \frac{R_L}{\rho} < \phi < \theta_{max} + \frac{R_L}{\rho} \qquad (6\text{-}149)$$

where ρ is the resistance of the potentiometer in ohms/rad and ϕ is defined such that

$$\phi = \theta + \frac{R_L}{\rho} \qquad (6\text{-}150)$$

Thus, division is achieved with respect to the variable ϕ.

6-4.8 COORDINATE-SYSTEM CONVERTERS

A coordinate converter transforms a set of quantities in a Cartesian or rectangular coordinate system into an equivalent set of quantities in a polar coordinate system, or

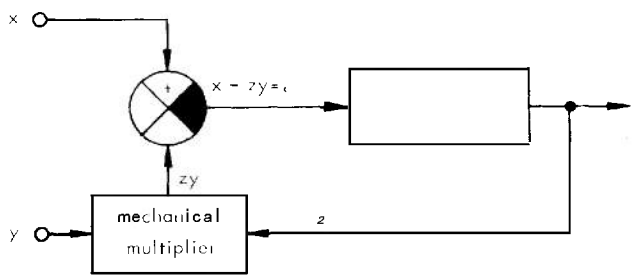


Figure 6-70. Block diagram of a divider employing a servo-driven multiplier.

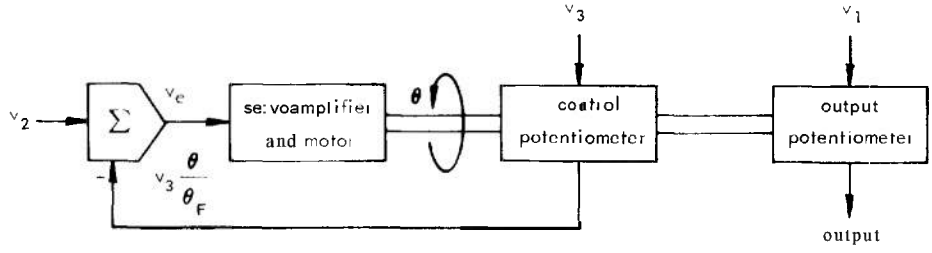
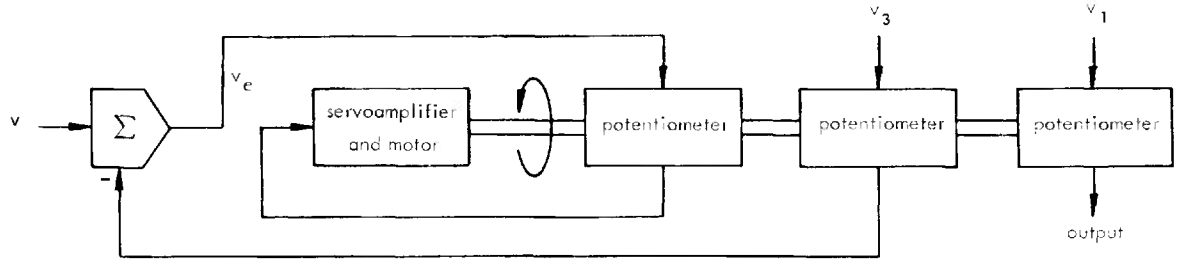


Figure 6-71. Block diagram of a position servo used for division.



F'igure 6-72. Block diagram of a gain-compensated divider servo.

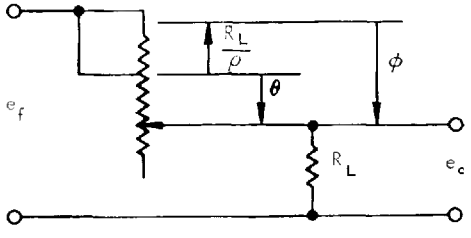


Figure 6-73. Division circuit based on a single, tapped, linear potentiometer.

vice versa. Vector resolution is essentially the same operation (see par. 6-2.14). The schemes discussed here canbe extended from two to three dimensions by the use of additional components similarto those described.

6-4.9 Mechanical Converters

Coordinate conversion can be performed mechanicallyby using the Scotch yoke mechanism. Fig. 6-74 indicates an arrangement that would convert wind velocity and heading into north-south and east-west components of wind velocity or perform any equivalent polar-to-rectangular coordinate conversion. Both the magnitude and direction of the wind are variable and must enter the computation. The crank of the Scotch yoke is positioned in accordance with the wind direction, and the angular motions derived by tlie gear picltoffs represent the sine and cosine of the crank angle or, in this case, the north-south east-west components of a unit velocity wind. Multiplication of actual wind velocity by these components yields the desired components. The utilization of this scheme for conversion from rectangular topolar coordinates is usually not practical unless servos are added.

6-4.10 Electromechanical Converters

The induction resolver, discussed in par. 6-4.14, under function generators, is designed specifically for coordinate conversion. If conversion is to be made from polar to rectangular coordinates, the rotor of the resolver is positioned to the required angle and the magnitude is introduced as a voltage applied

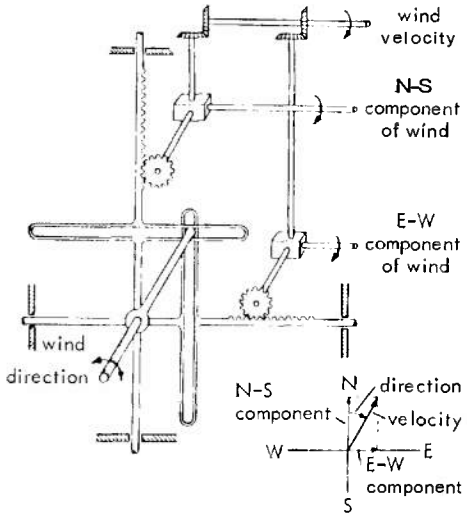


Figure 6-74. Mechanical coordinate converter.

to one of the stator windings. The voltages induced in the two rotor windings are then the rectangular components of the input signal. Since one input of the resolver is mechanical and the other electrical, and the components of the input vector may be available only as electrical or as mechanical quantities, a preliminary conversion must frequently be made.

The determination of the polar coordinates of a vector from its rectangular components can be performed using an induction resolver driven by a position servo (see Fig. 6-75). If the two input voltages to the resolver are x and y and the resolver shaft angle is θ, the outputs of the resolver can be expressed as

$$v_{R1} = y \cos \theta - x \sin \theta \qquad (6\text{-}151)$$

and

$$v_{R2} = x \cos \theta + y \sin \theta \qquad (6\text{-}152)$$

As shown in Fig. 6-76, the magnitude of the vector in polar form is given by x cos 6 + y sin θ and is thus v_{R2}. The resolver output v_{R1} is used as the error signal for a servo that positions the resolver. When v_{R1} is zero, the resolver is positioned to the proper angle θ so as to satisfy the geometric requirements of the coordinate conversion. An electrical signal corresponding to the shaft angle can be obtained from a potentiometer.

An equivalent coordinate conversion system can be built by using sine-cosine potentiometers. Two potentiometers must be used and their output summed to obtain voltages equivalent to those derived from a single resolver. In some applications, the sine-cosine potentiometers can be replaced with linear potentiometers driven from a Scotch-yoke mechanism.

6-4.11 Three-dimensional Vector Resolution by Computers

This paragraph shows the techniques for utilizing a computer to carry out a full three-dimensional vector resolution as discussed in par. 6-2.14 through par. 6-2.16. Since it is frequently convenient to carry out this computation in terms of the rotational velocity of the coordinate system rather than merely in terms of its angular orientation, several additional concepts are introduced at this point.

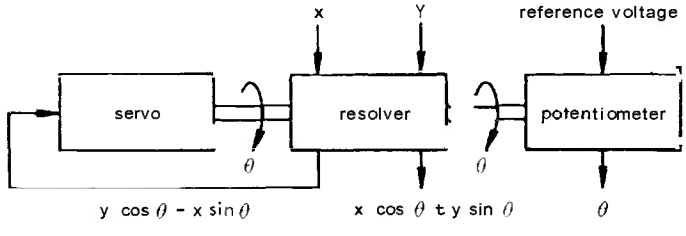


Figure 6-75. Simplified diagram of a rectangular-to-polar converter.

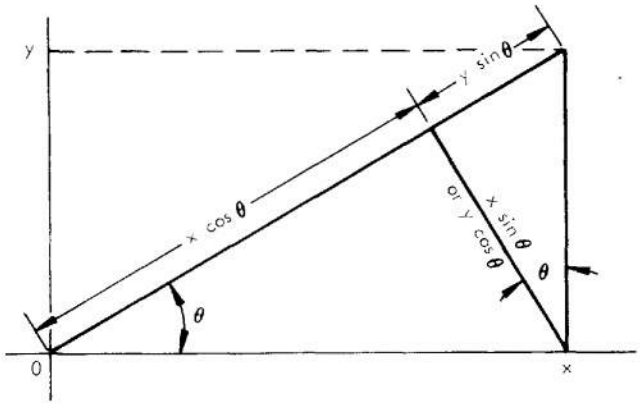


Figure 6-76. Geometry of the eoordinate-conversion system.

The rotational velocity vector can be written in terms of the body-axis system as

$$\vec{\omega}_b = \omega_{xb}\vec{i}_b + \omega_{yb}\vec{j}_b + \omega_{zb}\vec{k}_b \qquad (6\text{-}153)$$

where ω_{xb}, ω_{yb}, and ω_{zb} are the rates about the three body axes oriented along $\vec{i}_b$, $\vec{j}_b$ and $\vec{k}_b$, respeetively.

The rate at which the body-axis set is rotating is given b

$$\begin{bmatrix} \frac{d\vec{i}_b}{dt} \\ \frac{d\vec{j}_b}{dt} \\ \frac{d\vec{k}_b}{dt} \end{bmatrix} = \begin{bmatrix} \vec{\omega}_b \times \vec{i}_b \\ \vec{\omega}_b \times \vec{j}_b \\ \vec{\omega}_b \times \vec{k}_b \end{bmatrix} \qquad (6\text{-}154)$$

where $\times$ denotes a vector cross-produet. Insertion of Eq. 6-153 in Eq. 6-154 yields, after slight rearrangement,

$$\begin{bmatrix} \frac{d\vec{i}_b}{dt} \\ \frac{d\vec{j}_b}{dt} \\ \frac{d\vec{k}_b}{dt} \end{bmatrix} = \begin{bmatrix} 0 & \omega_{zb} & -\omega_{yb} \\ -\omega_{zb} & 0 & \omega_{xb} \\ \omega_{yb} & -\omega_{xb} & 0 \end{bmatrix} \begin{bmatrix} \vec{i}_b \\ \vec{j}_b \\ \vec{k}_b \end{bmatrix} \qquad (6\text{-}155)$$

Differentiation of eaeh side of Eq. 6-100 yields the following relationship after insertion of Eq. 6-155. Here it is to be noted that the inertial reference frame is fixed and therefore its derivative must be zero.

$$\begin{bmatrix} \dot{\ell}_x & \dot{\ell}_y & \dot{\ell}_z \\ \dot{m}_x & \dot{m}_y & \dot{m}_z \\ \dot{n}_x & \dot{n}_y & \dot{n}_z \end{bmatrix} \begin{bmatrix} \vec{i}_b \\ \vec{j}_b \\ \vec{k}_b \end{bmatrix} + \begin{bmatrix} \ell_x & \ell_y & \ell_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \begin{bmatrix} 0 & -\omega_{zb} & -\omega_{yb} \\ -\omega_{zb} & 0 & -\omega_{xb} \\ \omega_{yb} & -\omega_{xb} & 0 \end{bmatrix} \begin{bmatrix} \vec{i}_b \\ \vec{j}_b \\ \vec{k}_b \end{bmatrix} = 0 \qquad (6\text{-}156)$$

When all components of the veetor resulting from Eq. 6-156 are individually equated to zero, the following nine equations for the derivatives of the direction cosines result:

$$\left.\begin{aligned} \dot{\ell}_x &= \omega_{zb}\ell_y - \omega_{yb}\ell_z \\ \dot{m}_x &= \omega_{zb}m_y - \omega_{yb}m_z \\ \dot{n}_x &= \omega_{zb}n_y - \omega_{yb}n_z \\ \dot{\ell}_y &= \omega_{xb}\ell_z - \omega_{zb}\ell_x \\ \dot{m}_y &= \omega_{xb}m_z - \omega_{zb}m_x \\ \dot{n}_y &= \omega_{xb}n_z - \omega_{zb}n_x \\ \dot{\ell}_z &= \omega_{yb}\ell_x - \omega_{xb}\ell_y \\ \dot{m}_z &= \omega_{yb}m_x - \omega_{xb}m_y \\ n_z &= \omega_{yb}n_x - \omega_{xb}n_y \end{aligned}\right| \qquad (6\text{-}157)$$

Fig. 6-77 shows how the direction cosines can be generated from the three body rates ω_{xb}, ω_{yb}, and ω_{zb}. The blocks identified by the symbol 1/s represent integrator servos having a voltage input and a shaft-angle output, while the boxes identified by the symbol p represent potentiometers. If this same setup were to be instrumented on an all-electronic computer, the electromechanical integrators could be replaced by electronic integrators and the potentiometers by electronic multipliers.

Once the direction cosines are available as shaft angles, the resolution of a vector from a body-axis coordinate system to an inertial system is accomplished readily by the arrangement shown in Fig. 6-78.

An equivalent computation based upon use of Euler angles can also be instrumented. The equations for the derivatives of the Euler angles can be derived from a combination of the direction-cosine definitions given in Eqs. 6-102 and the equations for the derivatives of the direction cosines given by Eqs. 6-157. Differentiation of the expression for the direction cosine n_x in Eqs. 6-102 yields

$$\dot{n}_x = (-\cos\theta)\dot{\theta} \qquad (6\text{-}158)$$

while insertion of the expressions for n_y and n_z from Eqs. 6-102 in the expression for $\dot{n}_x$ of Eqs. 6-157 yields

$$\dot{n}_x = \omega_{zb}\cos\phi\sin\psi - \omega_{yb}\cos\phi\cos\psi \qquad (6\text{-}159)$$

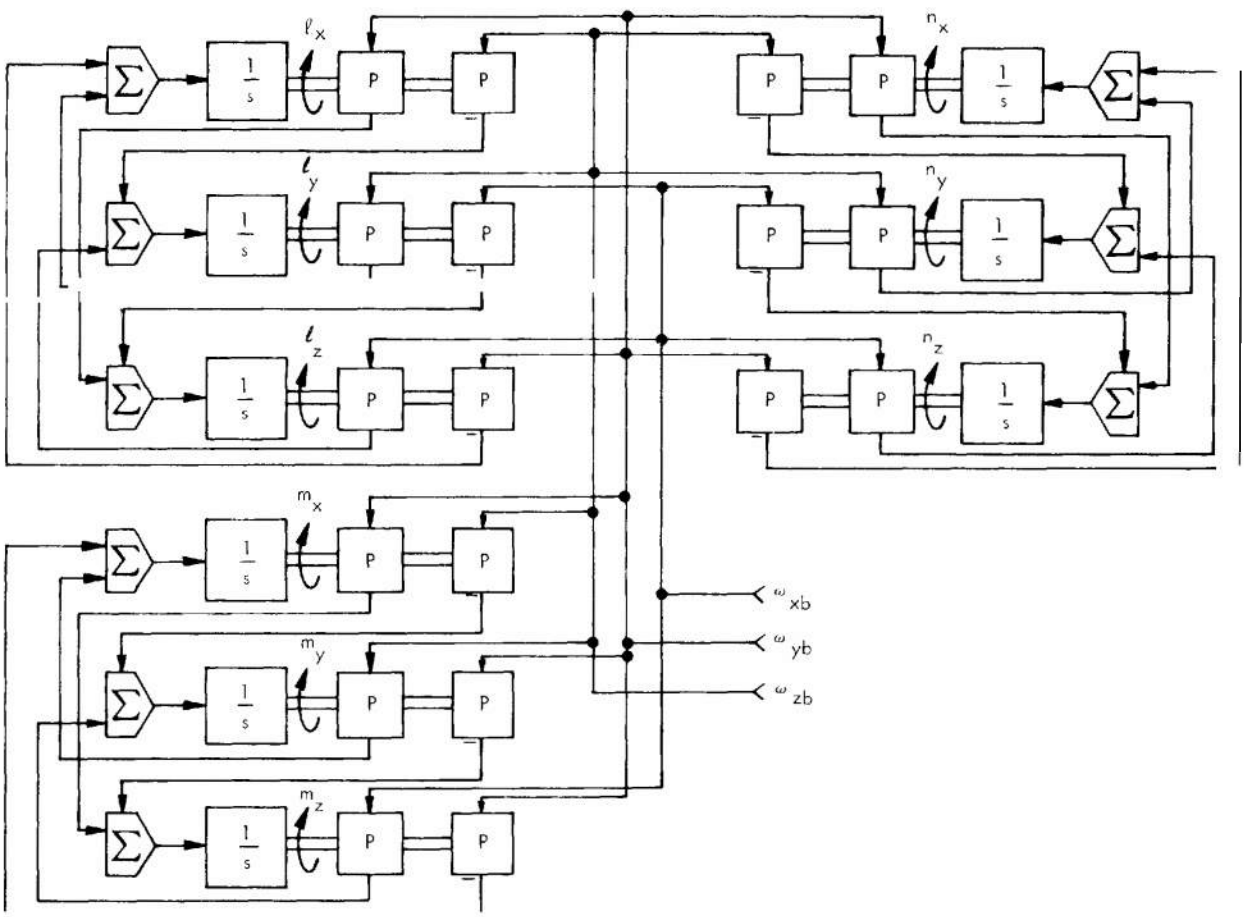


Figure 6-77. Block diagram of a system for generating direction cosines.

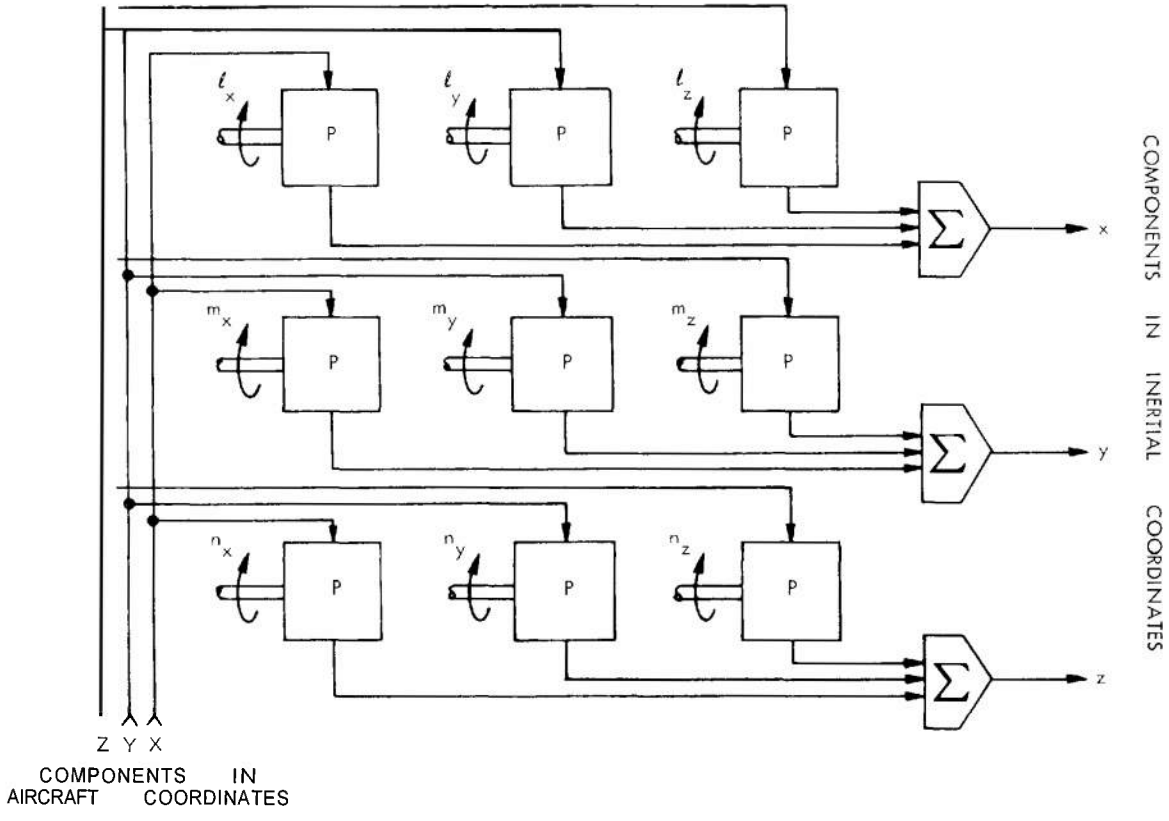


Figure 6-78. Block diagram of a system for converting from aircraft coordinates to inertial coordinates.

When these two expressions are equated and solved for θ, the result is

$$\dot{\theta} = \omega_{yb}\cos\phi - \omega_{zb}\sin\phi \qquad (6\text{-}160)$$

In a similar manner, differentiation of the expression for n_Y in Eqs. 6-102 yields

$$\dot{n} = \cos\theta(\cos\phi)\dot{\phi} - (\sin\theta)\dot{\theta}\sin\phi \qquad (6\text{-}161)$$

while insertion of the expression of n_x and n_z in the equation for $\dot{n}_y$ of Eqs. 6-157 yields

$$\dot{n} = \omega_{zb}\sin\theta + \omega_{xb}\cos\theta\cos\phi \qquad (6\text{-}162)$$

Insertion of the expression for $\dot{\theta}$ from Eq. 6-160 and solution of Eqs. 6-161 and 6-162 for $\dot{\phi}$ yields

$$\dot{\phi} = \omega_{xb} + \tan\theta(\omega_{yb}\sin\phi + \omega_{zb}\cos\phi) \qquad (6\text{-}163)$$

In a similar fashion, it can be shown that

$$\dot{\psi} = \sec\theta(\omega_{yb}\sin\phi + \omega_{zb}\cos\phi) \qquad (6\text{-}164)$$

Fig. 6-79 shows a computer setup for generating the Euler angles as shaft angles from body rates ω_{xb}, ω_{yb}, and ω_{zb}, while Fig. 6-80 shows how a vector can be resolved from a body-axis system to an inertial-axis system or vice versa using these Euler angles to position electromechanical resolvers of the type described in par. 6-4.14. In an all-electronic computer, diode function generators set up to generate the necessary trigonometric functions could be used in place of the resolvers.

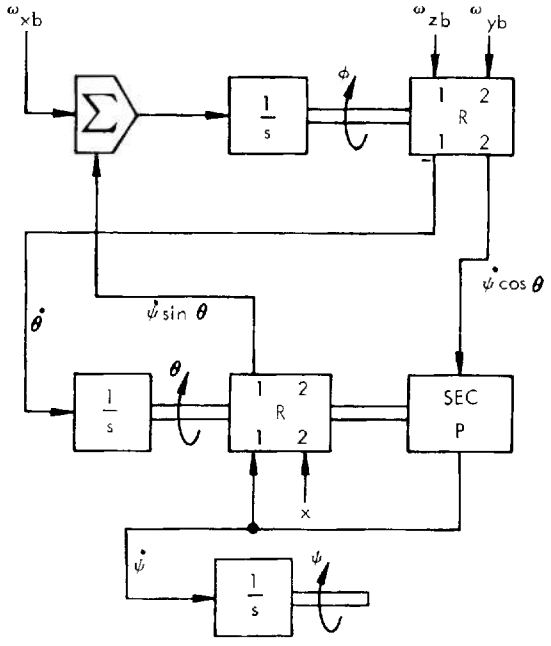


Figure 6-79. Block diagram of a system for the direct solution of Euler angles.

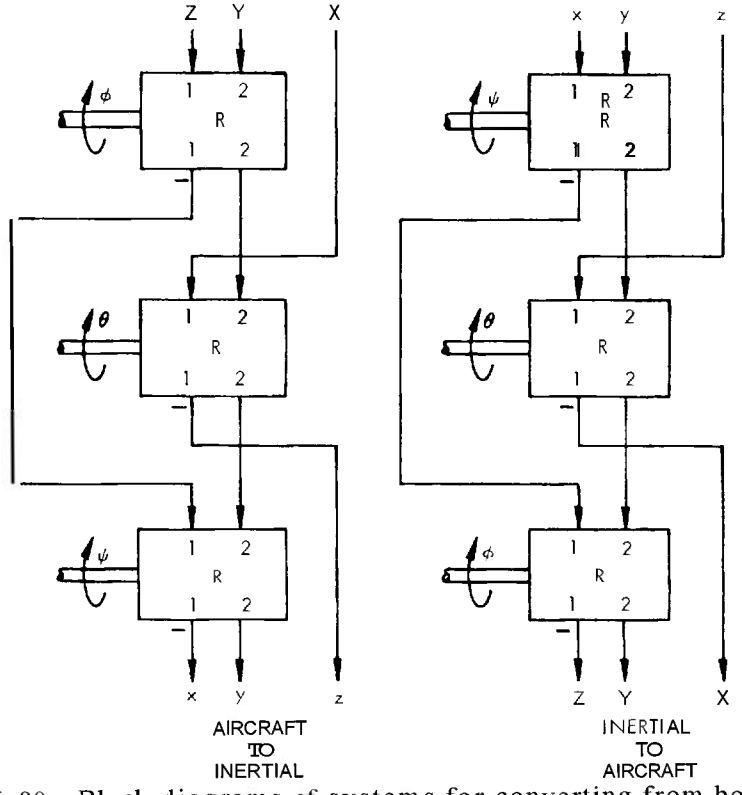


Figure 6-80. Block diagrams of systems for converting from body-axis coordinates to inertial-axis coordinates and vice versa by the use of resolvers.

6-4.12 FUNCTION GENERATORS

6-4.13 Mechanical Trigonometric Generators

The double Scotch yoke mechanism (see Fig. 6-81) is one of the most frequently used mechanical devices for generating sines and cosines. A crank pin C rotates about a pivot P at a fixed distance r. The pin fits snugly into a pair of slotted members that are mounted at right angles to each other and are free to slide in fixed supports. As the crank is angularly positioned to an input angle θ, the horizontal extension executes a motion r *cos* θ and the vertical extension executes a motion r sin θ.

The gear mechanism shown in Fig. 6-82 is another means for generating sine and cosine functions mechanically. In this device, the diameter of the large internal gear is twice that of the planet gear that is arranged to roll inside it. As the larger gear rotates about its axis, the small gear rotates inside and its center describes a circle. Because of the geometry of the system, the pin P moves along the line A-A', and its distance from the center of the large circle is 2r sin θ or 2r *cos* θ, depending on the reference taken for θ. Since these units involve rolling rather than sliding motion, they have low friction.

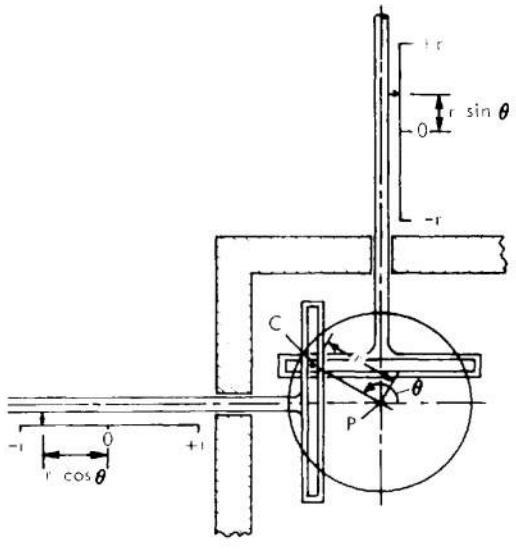


Figure 6-81. Double Scotch yoke mechanism.

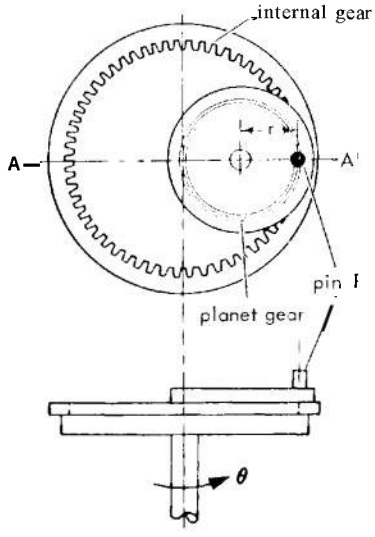


Figure 6-82. Gear-type sine-cosine generator,

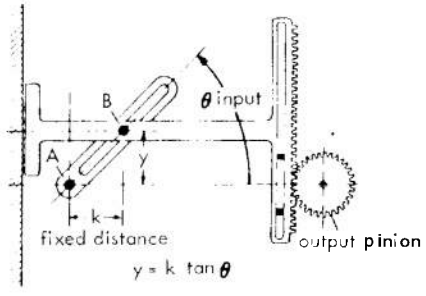


Figure 6-83. Modification of the Scotch yoke for generating a tangent function.

The modification of the Scotch yoke shown in Fig. 6-83 can be used to generate a tangent function over a limited range of the argument. A somewhat similar mechanism can be employed for generating the secant function.

6-4.14 Electrical Trigonometric Generators

Sines and cosines can be generated electromechanically with either a sine-cosine potentiometer or an induction resolver. A shaped-card potentiometer is shown pictorially and schematically in Fig., 6-84. The resistance is a complete 360" element with

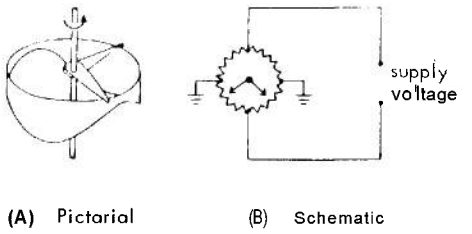


Figure 6-84. Shaped-card sine-cosine potentiometer.

four taps spaced at 90°. A balanced supply voltage is applied to one pair of diametrically opposite taps; the other pair of taps is grounded. Each quadrant of the resistance element is tapered to give a sinusoidal output when a specified load is connected between the slider and ground. Voltages proportional to the sine and cosine of the shaft angle are developed between ground and each of a pair of sliders mounted 90° apart. Precision units of this type are built with diameters of 10-20 inches. In these units, the maximum voltage error can be held below 0.15 percent of the maximum output, and a mechanical resolution of approximately 0.02° can be attained. For certain disadvantages associated with this type of unit, see par. 6-4.18.

Special circuits employing linear potentiometers can be used to generate tangent and secant functions. For generation of the tangent function (see Fig. 6-85), a linear potentiometer with a total resistance $2R_o$ is supplied with voltages $+v_i/2$ and $-v_i/2$ through the resistors R_2 and the potentiometer is loaded between the slider and ground with a resistance R_1. The transfer gain of this circuit can be expressed in the form

$$\frac{v_o}{v_i} = K \frac{\phi}{1 - \gamma\phi^2} \qquad (6\text{-}165)$$

where the constants K and γ depend on the circuit parameters. With a proper choice of these constants, this circuit approximates a tangent function to within 1 percent over the range of ϕ from 0° to 60°.

The transfer gain of the circuit for approximating the secant function (see Fig. 6-86) can be written as

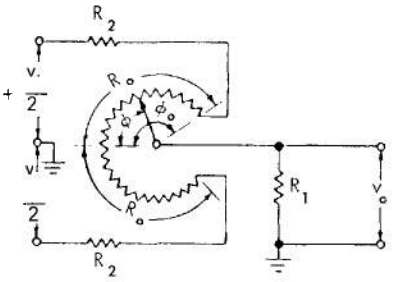


Figure 6-85. Circuit for the generation of the tangent function.

$$\frac{v_o}{v_i} \cong \frac{2R_1\phi_o^2}{R_o} \sec\phi \qquad (6\text{-}166)$$

where the parameters are adjusted to make

$$\phi_o^2\left(4\frac{R_1}{R_o} + 1\right) \cong 2 \qquad (6\text{-}167)$$

With this scheme, the approximation to the secant is in error by approximately $\phi^4/24$.

A second type of sine-cosine generator is the induction resolver, which may be considered to be a particular type of synchro generator. It consists of a cylindrical rotor, carrying two distributed windings with their axes in space quadrature, and a cylindrical stator, also with two distributed windings with axes in space quadrature. Each of the primary windings, which are normally on the stator, develops in the annular air gap a flux that ideally goes through one cycle of sinusoidal variation in the circumference of the air gap. In turn, the voltage induced in each output winding varies with the sine (or cosine) of the rotor angle. Connections to the rotor are made through slip rings. Precision resolvers for operation in the frequency range of 60-1000 cps are available from a number of instrument manufacturers.

Fig. 6-87 is a basic schematic representation of an induction resolver. Voltages v_{s1} and v_{s2} of the same phase are applied to the two stator windings, and the voltages v_{R1} and v_{R2} are induced in the rotor windings. These voltages are related by the equations

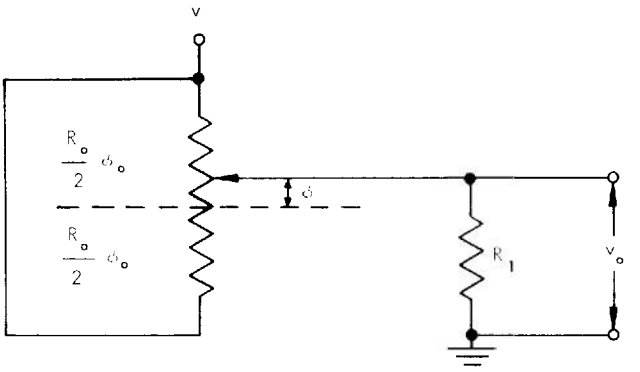


Figure 6-86. Circuit for approximating the secant function.

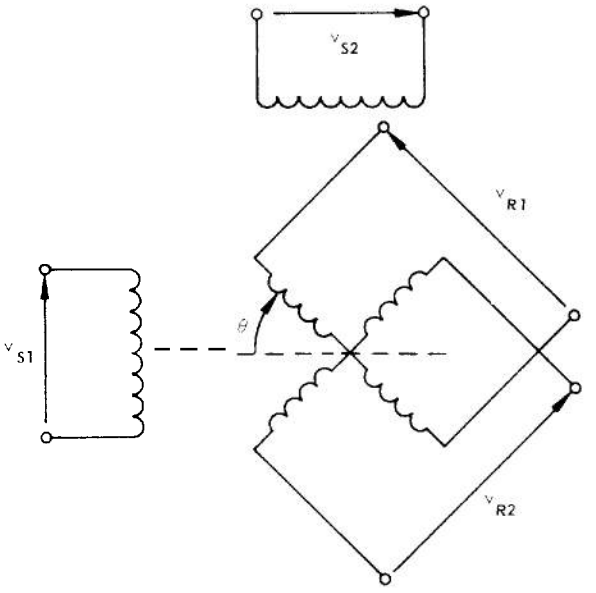


Figure 6-87. Schematic diagram of an induction resolver

$$v_{R1} = v_{S1} \sin \theta - v_{S2} \cos \theta \qquad (6\text{-}168)$$

and

$$v_{R2} = v_{S1} \cos \theta + v_{S2} \sin \theta \qquad (6\text{-}169)$$

where θ is the angle defining the position of the rotor with respect to the stator.

6-4.15 Arbitrary Function Generators

There are various mechanical and electromechanical devices capable of generating more than one function. Mechanical devices in this category are cams, noncircular gears, and linkage mechanisms. The principal electronic/mechanical methods use nonlinear or

tapped potentiometers and electromechanical curve followers.

Mechanical function generators are difficult to design and expensive to build, but they are more accurate and more reliable than electronic and electromechanical units. In addition, they can be used in environments unsuited to electrical equipment.

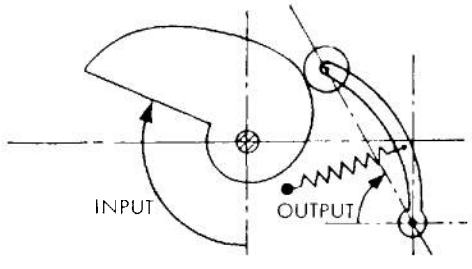
6-4.16 Cams and Noncircular Gears

A cam is basically a physical replica of the function to be generated. These units are designed in a variety of forms, the plane cam with a spring-loaded follower [see Fig. 6-88(A)] being one of the simplest and easiest to make. A more positive action than afforded by the spring-loaded follower can be achieved by milling a groove of the desired shape in a metal disk, as depicted in Fig. 6-88(B). A pin or roller inserted in the slot serves as a follower and generates a linear output motion. Cams are also made in the form of cylinders with a groove milled in the surface and a roller arranged to slide along a slot as the cylinder is rotated.

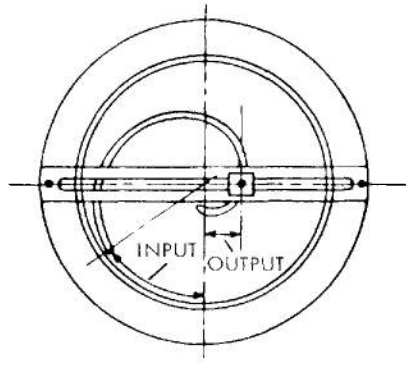
Fig. 6-89 shows a radial function gear and a spiral-face function gear. The use of function gears has been limited because of the difficulty of design and fabrication. However, with proper design and manufacture, high precision can be achieved, and such gears have found important uses in special-purpose computers, such as in a mechanical quarter-square multiplier.

6-4.17 Linkage Mechanisms

Linkage mechanisms consist of rigid elements moving in a plane and pivoted to each other, to a fixed base, or to slides. Linkage computers can be designed to perform a number of functions -- including addition, multiplication, and squaring. Unfortunately, few standard bar-linkage function generators exist and one must usually design a linkage suitable for a particular purpose. Although linkage devices are reliable, economical to construct, and frequently smaller than other

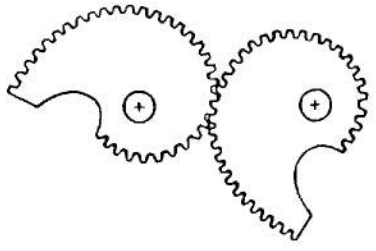


(A) Plane cam with spring-loaded follower

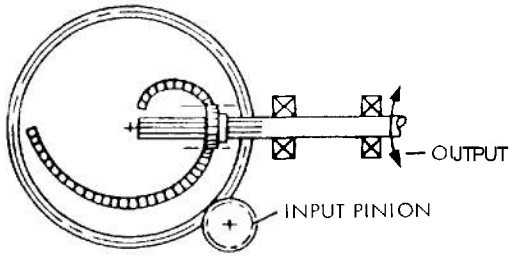


(B) Cam with groove contact

Figure 6-88. Typical cams.



(A) Radial function gear



(B) Spiral-face function gear

Figure 6-89. Typical function gears.

types of computers for the same purposes, they have not been used widely because they are relatively difficult to design and the field of mechanizable functions is somewhat restricted.

6-4.18 Special Potentiometers

Several methods of generating nonlinear functions with potentiometers are available. In one type of nonlinear potentiometer, such as the sine-cosine potentiometer of Fig. 6-84, resistance wire is wound on a tapered card. The shape of the card determines the functional relationship between mechanical motion and resistance change. This method has several disadvantages. Accurate machiningof the shaped card is difficult and the ratio of maximum to minimum card width should be less than 10 to 1 to avoid a fragile card. High card slopes also must be avoided because it is impossible to make the wire stay in place in such regions of a card. A combination of several wire sizes and a tapered card can be used to accommodate a greater range of slopes. Another method for producing a nonlinear element is to wind the resistance element with a variable wire spacing. However, the resolution becomes poorer as the wire spacing is increased. Potentiometers that will generate nonlinear functionsto accuracies of the order of one or two percent are used frequently in spite of these limitations.

An entirely different means for generating nonlinear functions is to provide a number of taps along a linear resistance element. External resistors are used to make the parallel combination match the desired resistance-versus-shaft-angle curve at the tap points and the resistance element in the potentiometer provides a means of interpolating between points. The various function-generation schemes based upon this type of unit differ principally in the manner in which the voltages at the taps are established. For monotonic functions, simple resistive loading of the type shown in Fig. 6-90(A) suffices. However, if the derivative of the desired function is not of the same sign over the entire function, it becomes necessary to inject currents at intermediate taps. This canbe done with the type of generalization of the simple loading scheme shown in Fig. 6-90(B). Alternatively, the voltage at each tap can be established either from a low-impedance source or by an iterative-adjustment procedure if the source impedance cannot be neglected.

With schemes of this type, the accuracy of the approximation to the desired function improves as the number of taps on the potentiometer is increased, but the amount of setup effort required also increases. For many applications, a potentiorneter with 8-10 taps provides an adequate approximation, but potentiometers with 25-30 taps are available if a more accurate representation is required. One major limitation on this system is that the potentiometer must be driven mechanically and, therefore, the speed of response is severely limited.

6-4.19 Electromechanical Curve Readers

In one of the most successful of the automatic curve readers, the curve is drawn with conducting paint on a flat piece of rectangular-coordinate graph paper. By means of a pair of servo drives, a reading head is positioned along one axis in accordance with the independent variable and along a perpendicular axis in accordance with thefunction. Positioning in the direction of the independent variable is controlled with a linear potentiometer. A radio-frequency current is passed through the conducting paint., and the field produced by this current induces voltages in an electromagnetic pickup mounted on the carriage. The pickup and its associated detector give a zero output signal when the head is exactly centered over the curve. The signal increases, with a sign dependent on the direction of motion, as the head is positioned over the curve by a servo that uses the output of the reading head as its error signal. A linear potentiometer mounted parallel to the axis of the function delivers an electrical output proportional to the position of the head and, thus, to the desired function.

6-5 COMPLETE COMPUTERS

In previous chapters, the computing units that are the principal building blocks in any analog computer have been described. However, a complete computer must include a large amount of equipment that is not used

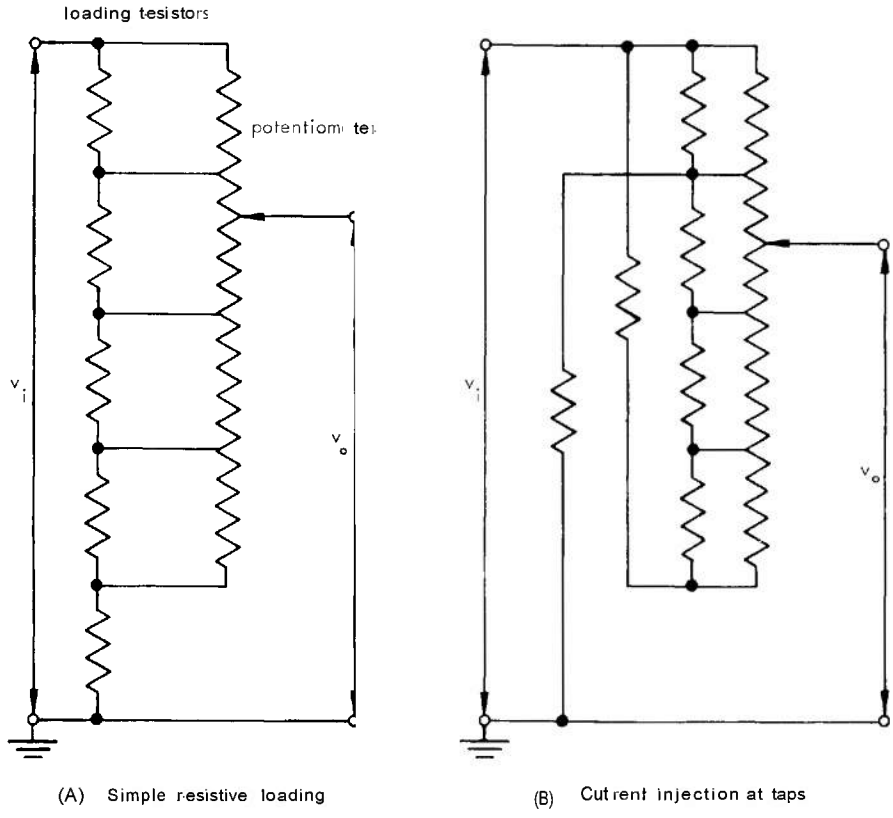


Figure 6-90. Function generation with a tapped potentiometer,

directly in the computations, but without which the computing elements are not usable. Into this class fall power-supply equipment, means for readily interconnecting the computing elements, sequencing and overload equipment, recording equipment, and test equipment. Some of the main features of these equipments are described in this chapter.

6-5.1 POWER SUPPLIES

A complete analog computer, particularly one of large scale, requires a variety of power supplies for its operation. The principal power supplies used are the following:

(1) Filament power supplies, both a-c and d-c.
(2) Unregulated high-voltage d-c supplies, both positive and negative.
(3) Regulated high-voltage d-c supplies, both positive and negative.
(4) Reference-voltage supplies.
(5) D-c supply for relays, clutches, and associated equipment.
(6) A-c supply for servomotors.
(7) A-c supply to drive choppers for d-c amplifier stabilization.

The characteristics of the various supplies, the features of the supplies that are different from those required in other applications, and the means by which some of these special characteristics are obtained are discussed in the paragraphs which follow.

6-5.2 Filament Power Supplies

If sufficient forethought is given to the problem of supplying filament power for the vacuum tubes in a computer, no particular

difficulties arise in providing a satisfactory supply. Tubes employing 6.3-volt heaters are used most commonly, although 12.6-volt versions of many of the common 6.3-volt tubes are available. In a small or medium-size computer, the current requirements for 6.3-volt tubes are not excessive. However, in a large-scale computer, considerable saving in copper can be realized by doubling the supply voltage anti either running pairs of 6.3-volt tubes in series or using 12.6-volt tubes. In at least one commercial computer, the heater in the first tube in each amplifier is operated at one half tlic rated voltage in order to reduce grid current. Practically all the commercial electronic differential analyzers use a-c filament supplies, but d-c supplies are used in some of the large-scale custom-built computers.

In the power-output stages of servo amplifiers and other high-level units, hum pickup from tlie heater supply is no problem, and an unregulated a-c supply is preferable because of its simplicity. If an a-c filament supply is to be used in a computer that does not employ chopper-stabilized amplifiers, drift can be reduced if tlie filament supply is derived from an a-c constant-voltage regulator. Commercial constant-voltage transformers are available in a range of volt-ampere capacities. One class of constant-voltage a-c transformers depends on the saturation of a magnetic material to produce the necessary nonlinear impedance. Control is achieved by the use of a resonant circuit in conjunction with the saturable element. This type of regulator is economical antl requires essentially no maintenance, but tlie output waveform is necessarily distorted by core saturation. This distortion may be serious in tlie operation of some types of equipment. A slightly different type of a-c regulator utilizes a temperature-limited diode as one element in a bridge circuit. An error voltage is derived from tlie bridge and, after amplification, is used to control the direct current in a saturable reactor. Good output waveform is achieved by a filter inserted between the saturable reactor antl the load.

Electromechanical regulators also can be used and are very satisfactory if given adequate periodic maintenance.

The use of a d-c filament supply in tlie low-level stages of computer amplifiers offers some advantage because the use of direct current prevents hum pickup from the filaments. In some installations, the difficulty associated with the use of two types of heattr supplies is avoided by supplying all the vacuum tubes with direct current. Several types of d-c filament supplies are available as commercial units. A selenium rectifier can be used with an appropriate transformer and an LC filter to reduce the ripple to less than one volt. A simple supply of this type has poor regulation and may lead to serious drift unless chopper-stabilized amplifiers are employed. Regulated, electronic d-c filament supplies are commercially available with a variety of current capacities. These units include a degenerative control loop that employs a saturable reactor as a nonlinear impedance in series with the primary of the transformer that supplies tlie rectifier.

In an installation where hundreds of amperes of heater current are required, a motor-generator set may prove to be the most suitable type of supply.

The ripple in ttie voltage from a generator is at a considerably higher frequency and a much lower amplitude than that in the output of a full-wave rectifier driven from a single-pliase 60-cps supply. The filter required to remove this higher frequency can use a much smaller LC combination tlian is required with ttie rectifier. Because inductances of even a small fraction of a henry are large and expensive if they have a direct-current-carrying capacity in the range of hundreds of amperes, and because extremely large values of capacitance may be required, tlie filter is an expensive part of a rectifier-type filanient supply.

6-5.3 High-voltage D-C Supplies

Both positive and negative iiigti-voltage d-c supplies operating at sex-eral different voltages are required in a complete analog-

*See Chapter 16 of Ref. 11.

computer installation. Although an unregulated supply can be used for high-power-level stages, such as the output stage used to drive a servomotor, regulated supplies are used to supply the direct current for other portions of almost all analog computers.

The practice commonly followed in commercial analog computers is to provide separate power supplies, each including a rectifier and a regulator, for each group of equipment. For example, a power supply for the required positive and negative voltages is assigned to a group of computing amplifiers, another supply is assigned to a function-generator group, and still another to a group of servomultipliers. Power-supply voltages commonly used in analog computers are +300 volts, -300 volts, and +500 volts. The 300-volt supplies are designed with a current capacity of 0.5 to 1.5 amperes. A supply may be used to provide either a positive or a negative output with respect to ground, depending on which output terminal is grounded. The +500-volt supply is used to provide bias voltages and, consequently, is required to supply only a few milliamperes of current. A -500-volt supply is not necessary since the -500 volts can be obtained by connecting a low-capacity -200-volt supply in series with the -300-volt supply.

A typical power-supply unit consists of a transformer and a full-wave rectifier section followed by an electronic voltage regulator. Fig. 6-91 shows a schematic diagram of the rectifier section of a typical power supply.

The output of the rectifier is not suitable, for several reasons, for use as the plate supply of the computing components of an analog computer. First, the internal impedance of a 1-ampere supply of this type may be 10 to 20 ohms at zero frequency. Unless the internal impedance of the power supplies feeding the computing components is held to a few tenths of an ohm, undesirable cross coupling of signals occurs between components connected to a common supply. Second, changes in the output voltage of the rectifier with changes in the a-c line voltage and in the load result in excessive drift in the output of Computing components. Even if chopper stabilization is employed in the computing amplifiers, the principal causes of drift should be reduced. Third, unless a large amount of filtering is provided following the rectifier, a large ripple component (possibly 10 volts or more) appears in the d-c output voltage. If the plate-supply voltage to the computing amplifiers contains an appreciable ripple voltage, some of this ripple will appear at the output of the amplifier and may increase the noise to such an extent that the useful operating range of the amplifier is seriously restricted.

By the use of an electronic voltage regulator,* each of these difficulties associated with the basic rectifier can be eliminated. Fig. 6-92 shows the basic form of the series regulators employed in most electronic voltage-regulator units. A VR tube is frequently used as the voltage standard, and a resistive voltage divider connected across the regulator output is adjusted to give approximately the same voltage at the tap point as that across the VR tube. In the simplest regulators, a single tube serves as both the voltage-comparison unit and the amplifier. In this instance, the voltage at the cathode of the tube is established by the VR tube, and the tap point on the voltage divider is connected to the grid of the tube.

The fact that the plate current from the control amplifier flows through the reference-voltage source impairs the performance of this simple circuit. Since the plate current varies and the voltage across a VR tube is not entirely independent of the current flow through the tube, this circuit is not used in high-quality voltage regulators. Instead, the reference voltage is applied to one grid of a differential amplifier, and the voltage from the voltage divider is applied to the other grid. Increased gain, and consequently better regulation, is achieved by adding a separate stage of amplification. The series-control element consists of one tube or as many tubes in parallel as are needed to carry the required output current.

Fig. 6-93 shows the schematic diagram of a complete electronic voltage regulator. This unit is supplied with unregulated direct

* See Part III of Ref. 11.

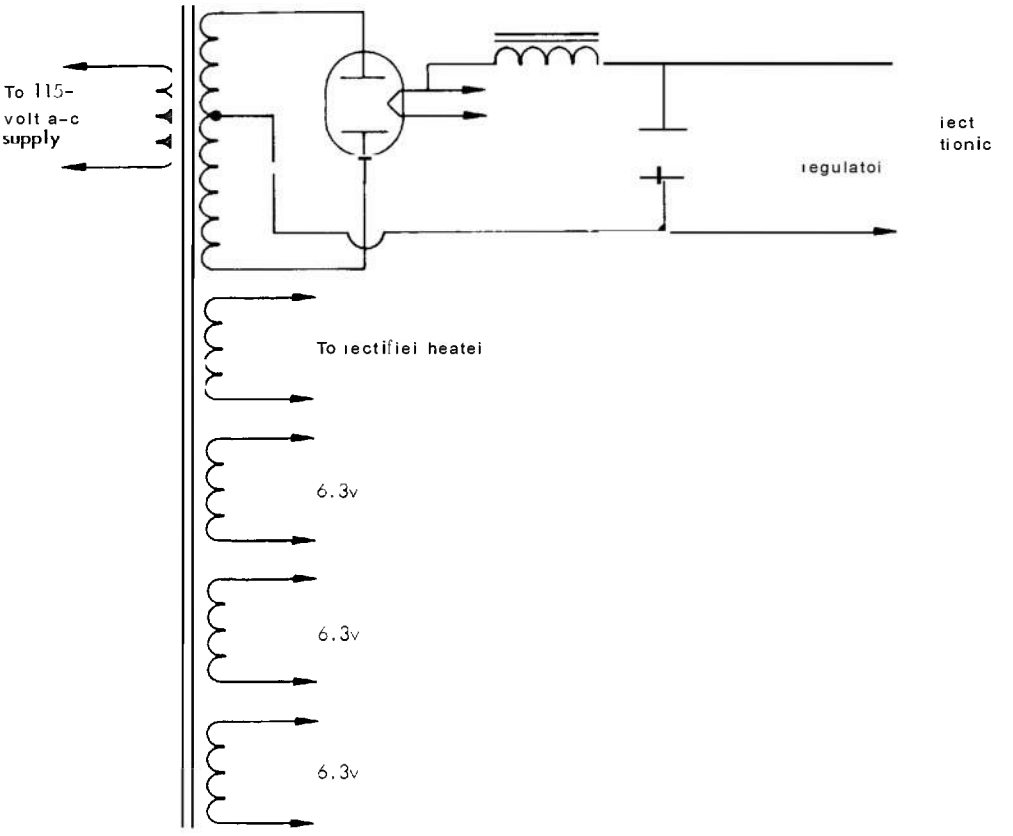


Figure 6-91. Rectifier section of a typical power supply

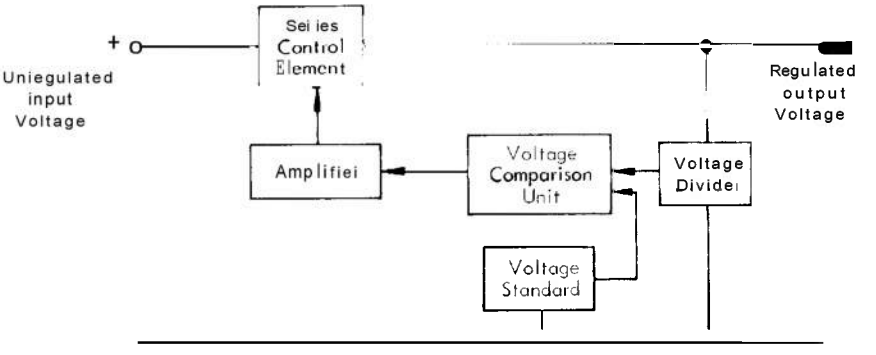


Figure 6-92. Block diagram showing the basic form of the series regulators employed in most electronic voltage-regulator units.

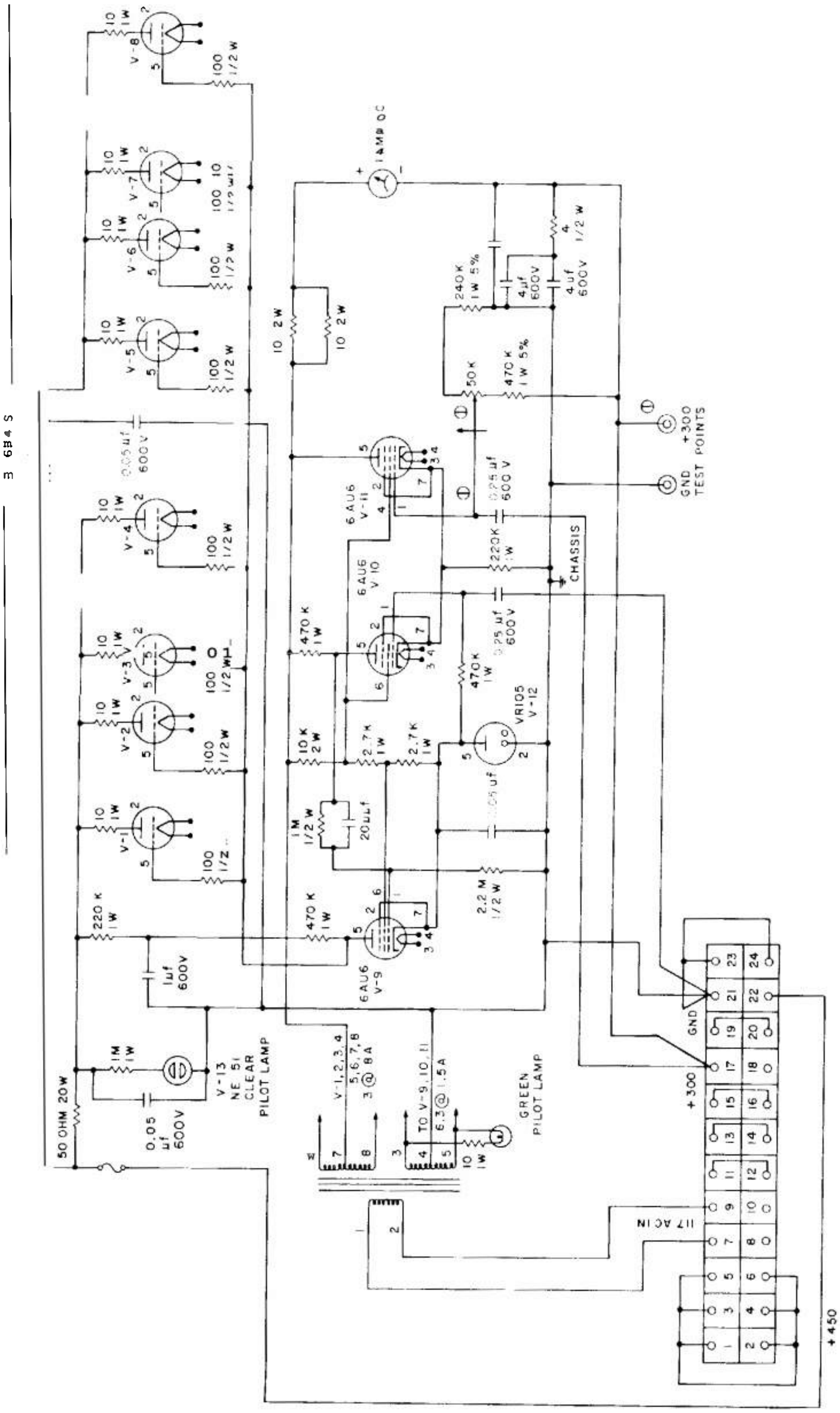


Figure 6-03 Schematic diagram of a typical +300-v 800-ma voltage regulator.

current at approximately 460 volts and supplies 300 bolts regulated direct current. The current capacity 'is approximately 800 ma. With a ripple component of 10 volts on the input voltage, the ripple on the output is approximately 0.25 millivolt. The internal impedance of tlie supply is less than 0.1 ohm. Tn this supply, tlie desired current capacity was obtained by placing eight 6R4 tubes in parallel. Several high-current-capacity tubes have been developed for regulator service. Tlie type 6AS7, for example, is a double triode each section of which is designed to pass a current of 130 ma at a voltage drop of 100 bolts. The newer 6336 tube has approximately double the rating of the 6AS7.

In a small- or medium-size coniputer installation, as many as six rectifier-regulator units of each polarity may be required to supply the necessary high-voltage direct current. The flexibility derived froin the use of a few sniall supplies outweighs the saving in space ancl maintenance that results if a single unit is used to supply +300 volts ancl another to supply -300 volts. A large-scale installation, on the other hand, may require 25 amperes or more of direct current at each polarity. This current could be obtained either from a number of supplies, each with a capacity of approximately 1 ampere, or from a centralized supply. If a high-current-capacity central rectifier is employed, some advantage can be gained from the use of a scheme somewhat niore complex than tliat shown in Fig. 6-91. The filtering problem can be reduced considerably if a polyphase rather tlian a single-phase rectifier is used. With a polyphase rectifier, the amplitude of the ripple voltage is reduced, and its frequency is increased. Both of these factors ease tlie filtering probleni. Use of a half-wave, three-phase rectifier offers considerable advantage over a full-nave, single-phase unit at relatively little increase in complexity. If the current required is of the order of 15 amperes or more, the further. reduction in ripple that can be achieved with a full-wave, three-phase unit may offset its increased complexity.

Even if a centralized rectifier is used in a large installation, the use of individual regulator units for different racks or groups of equipment is recommended. No particular advantage is gained by the use of a single high-capacity regulator in place of a group of smaller units. The latter essentially eliminate the problem of cross coupling between different computing units and, if correctly located, do not require long, low-resistance feeders. A number of series-type regulator units of the type represented in Fig. 6-92 can be fed from a common rectifier if a positive output voltage is desired. However. regulators of this type cannot be used in parallel froni a common supply if a negative output is desired. Tn this case, the positive output terminal would he grounded. Since the positive input terminals of all the units would be fed froin the same point, the control elements of all the regulators would lie in parallel and, consequently, could not operate satisfactorily. This difficulty can be circumvented in several ways, but probably tlie most satisfactory is the use of a shunt rather than a series regulator. The basic shunt regulator is represented in Fig. 6-94. With this type of regulator, the output voltage is the difference between the input voltage and the drop across the series resistor. This drop, in turn, is determined by the sum of the current drawn by the load and that drawn by the shunt regulator tubes. The current drawn by the shunt element is controlled in sucli a way that a fixed output voltage is niaintained, regardless of changes in load current or input voltage. Fig. 6-95 shows the complete schematic diagram of a shunt regulator designed for negative operation, A regulator of this type draws constant current ;rom the supply rectifier, regardless of tlie load current drawn from the regulator. Consequently, as a means of reducing both tlie current drain on the rectifier and the heat dissipated in the regulator, from one to five shunt tubes can be switched into tlic circuit and the series resistor can be changed simultaneously for applications where only a fraction of the full output is required.

Tlie grids of the shunt tubes must operate at a voltage below that of their cathodes and, in turn, must be driven by a tube that operates with its cathode and grid at still lower voltages. Consequently, in addition to tlie -450-vol? unregulated input from which this regulator operates, tno other voltages, -400 volts antl -475 volts, are supplied to the amplifier tubes in the regulator. The regu-

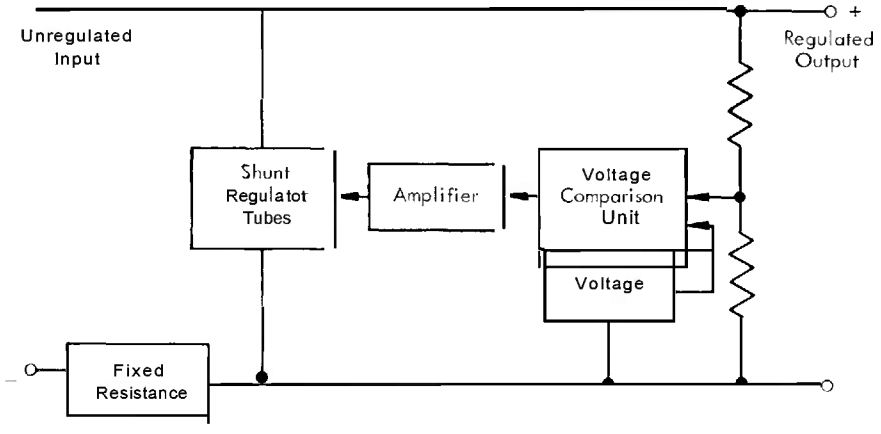


Figure 6-94. Block diagram of the basic shunt regulator.

lator of Fig. 6-95 has an internal impedance of approximately 0.06 ohm and a ripple output of approximately 0.3 millivolt.

Both the positive and the negative regulators described have a very low internal impedance and provide an output with a very low ripple content. However, the voltage across a gas tube, even when it is operated at constant current, varies somewhat with time and temperature; therefore, the output of any regulator unit that utilizes a gas tube as a voltage standard cannot be expected lo maintain an absolute level of output voltage. For most computer applications, variations of a few volts in the supply voltages can be tolerated, provided a fixed ratio between supply voltages is maintained. Interregulation of a positive and a negative supply can be achieved by using one voltage standard and by deriving the standard voltage for one of the regulators from the unit with which it is paired.

6-5.4 Relay Supplies

In a computer, the relays, stepping switches, and clutches frequently are designed to operate on 24 to 28 volts direct current. Several types of power supplies can be used for this purpose. In an installation where this type of load is fixed and relatively small, a power supply consisting of a transformer, a selenium rectifier, and an RC filter provides a simple yet satisfactory solution. If, however, this type of load varies widely from problem to problem, the operation is improved considerably if a supply with better regulation than that provided by the simple rectifier is used. Either one of the electronically regulated d-c supplies described for supplying filament voltage or a small motor-generator set provides a satisfactory supply. In a small installation, a relay supply with a capacity of a few hundred milliamperes may suffice. In a large installation, a supply of 25 to 50 amperes may be required.

6-5.5 A-C Supplies

In the majority of the electronic differential analyzers now available commercially, no special a-c supplies are required. However, several generalized computers have been built that employ a suppressed-carrier-modulated signal as the data carrier. Furthermore, an a-c data carrier is used in a number of special-purpose computers.

The requirements for the a-c supply in such applications are somewhat similar to those for d-c reference supplies, as discussed in par. 6-3.9. The supply should be well regulated and have a low internal impedance. Furthermore, the harmonic content should be kept very low. If appreciable phase shift of the harmonics occurs as they are transmitted through the computer, the harmonics will not add in the same way as the fundamental components. As a result, excessive

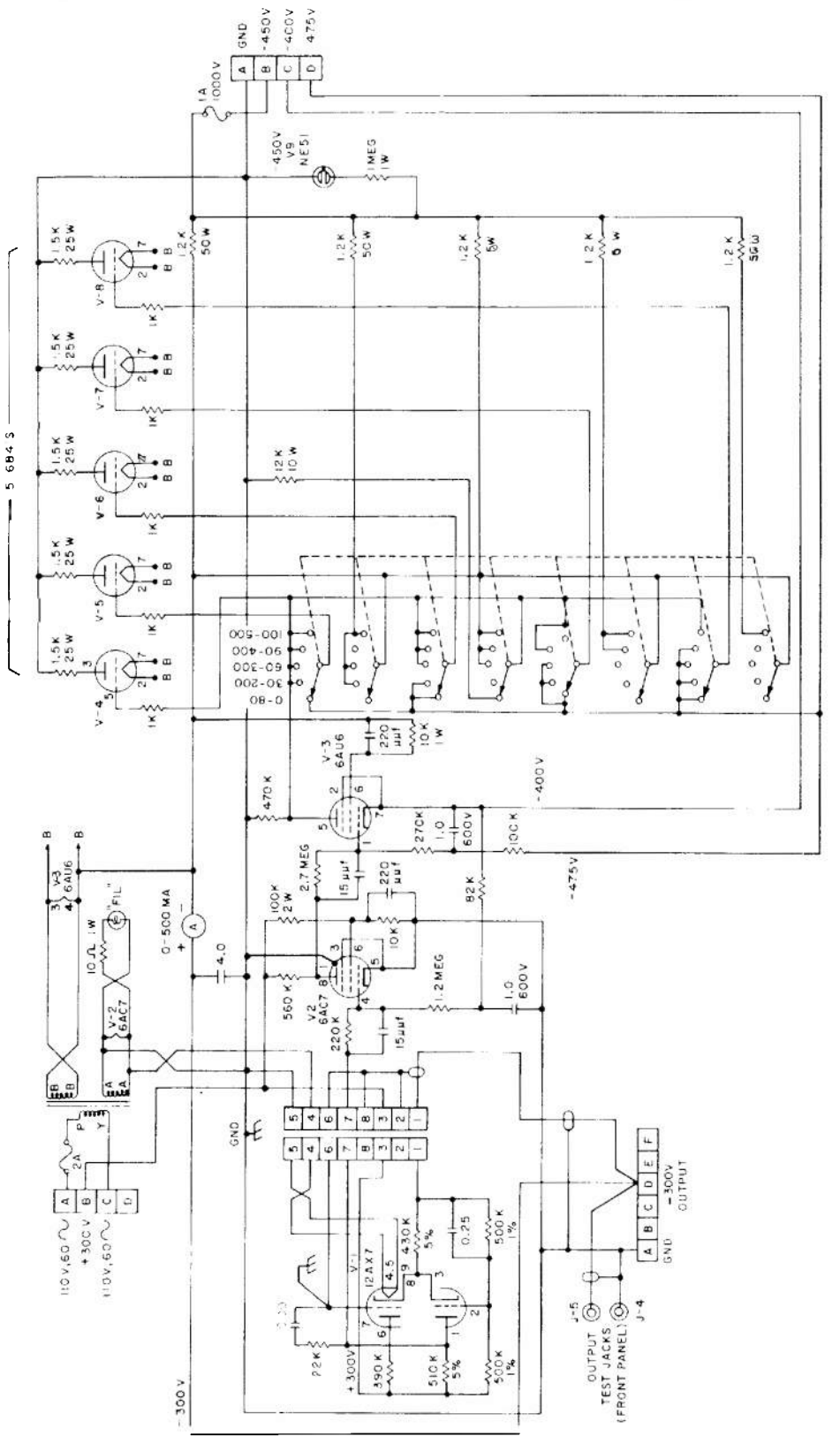


Figure 6-95. Schematic diagram of a typical shunt regulator designed for negative operation.

harmonic voltages may appear at summing points and cause overloading of tlie servos or other computing elements. For a fixed computer that operates on a 60-cps carrier, the required voltages could be supplied directly from the a-c mains. In the case of a mobile computer or one that uses other tlian a 60-cps carrier, the required a-c voltages could be supplied either by a rotating machine or by an electronic generator. Hecause a-c computing equipment is sensitive to the frequency of the voltage used as the data carrier, both the frequency regulation and the amplitude regulation of the reference supply should be satisfactory.

If two-phase servomotors are employed in the computer, a supply also is required for the reference fields of the motors. Ttiis supply must be of the same frequency as the data carrier and must maintain a fixed pliase relationship (usually 90°) with respect to the carrier. This requirement can be met if a two-phase alternator is used or if the electronic generator includes two power units that derive their excitation from a common oscillator through appropriate phase-shifting networks. If very little 90° power is required in a computer, a phase shifter could be provided at each motor and the 90° central supply could be omitted. However, this solution quickly loses its advantage as the size and flexibility of the computer increase.

Another special a-c supply that is found in some computers employing chopper-stabilized d-c amplifiers is a supply to provide chopper excitation. Frequencies in tlie range from 20 to 200 cps but usually bearing no simple relationship to 60 cps are used to drive choppers. Such frequencies cannot be obtained directly from the power line. Since the power requirements for chopper supplies are relatively small, even in a large installation, a vacuum-tube oscillator followed by a power amplifier usually proves satisfactory.

6-5.6 Grounding Systems

Although tlie grounding system in any computer should receive careful consideration, the problems associated with grounding become more pronounced as the size of the computer installation increases. An ideal grounding system would be one with infinite conductivity, with the result that tlie potential drop between any two points in the ground system would be zero. If a truly infinite conductivity system were available, signal and power grounds could be connected in any convenient way without causing extraneous ground signals that might impair the accuracy of the computer. The amount of copper required to achieve this result is excessive. Consequently, some fundamental rules should be followed in planning the grounding system. Tlie grounding system for handling signal grounds should be independent of the system used as the return path for power connections in the computer, except €or a single interconnection at one point. If the cross coupling through the grounding system is to be held to a minimum in a large-scale installation employing a variety of power voltages, both alternating and direct, separate ground systems should be provided for each type of power supply; for example, vacuum-tube heater supplies, plate supplies, relay and clutch supplies, and a-c servomotor supplies. These separate ground systems should be interconnected at a single point.

Edwards[12] has indicated that a considerable saving in the size of conductor needed in the signal ground system of a large computer employing chopper-stabilized d-c amplifiers can be achieved if the ground points against which the choppers compare the error voltages in the various amplifiers are connected to a ground buss separate from that used for grounding potentiometer and other signal-carrying elements. By the use of this technique in an installation that includes ten computing cabinets, the offset of the chopper ground as measured between any two amplifiers in the installation has been held to 0.1 millivolt, and a large saving in copper over tliat required by a brute-force method has been achieved.

6-5.7 PATCHING ANU PROGRAMMING EQUIPMENT

In any computer, means must be provided for interconnecting tlie computing elements in the form required to carry out the desired computation and for placing the computer in its various modes of operation, such as "INITIAL CONDITIONS", "RUN", "HOLD", and "RESET". These and related topics are usually grouped under the heading of patching

anti programming. The range in complexity of this type of equipment as found in different computers is extremely wide. At one end of tlie scale is the special-purpose coniputer designed to solve only one specific set of equations, sucli as a fire-control computer. Tlie interconnections in such a computer would be permanent, and tlie programming equipment might reduce merely to an Oh-OFF switch. At the other end of tlie scale is the large generalized computer that is capable of solving a great variety of problems, such as a simulator. The feasibility of patching and programming such a machine entirely from punched tape was demonstrated on the mechanical differential analyzer[13] built at M.I.T. in the late 1930's. Between these two extremes lies an enormous range of possibilities. For any particular installation, the decision as to the degree to which precabling, manual patching, and automatic patching and control should be eniployecl is basically one of economics rather than of teclinical feasibility. Ilowever, in addition to tlie initial cost of the installation, the computation of costs must include an appropriate weighing of set-up time, checking time, running time, and maintenance over the expected life of the computer.

Because patching and programming are separate operations in most analog computers, these functions are discussed separately in the paragraphs which follow.

6-5.13 Patching Equipment

In the simplest type of analog computer, tile input and output terminals of each coniputing component are made available, and the interconnections required for the solution of a problem are made by means of cables run directly between the components. Fig. 6-96 provides an example of a computer in which patching is done in this manner. This computer consists of a collection of computing elenients in small boxes. Tlie various boxes provide the basic functions of summation, coefficient setting, and integration, a few commonly required simple linear functions such as $1/(1 + as)$, and nonlinear elements such as limiters and dead-zone simulators. The boxes containing these elements can be arranged on a table or in a rack in a way that more or less duplicates tlic block diagram of tile system being studied. Patching, then, consists primarily of providing connections between tlie boxes. This set-up procedure has considerable educational value in enabling tlie operator to visualize the system being studied. However, it is effective only if tlie number of elements involved is relatively small; it becomes unmanageable for very large problems.

Another computer utilizes a different approach to the patching problem. In this computer, twenty-four computing amplifiers, eighteen potentiometers, twelve 1-μf capacitors, ten diodes, and three limiters are mounted in the computer cabinet. The terminals of these elements are wired to a patch bay into which may be plugged a removable probleni hoard. The patch-bay terminals are arranged in a 21-by-29 array. The wiring to the patch bay is arranged anc the problem boards inscribed to facilitate setting up the basic operations commonly required in analog-computer work. Groups of four resistors are associated with the input terminal of a number of the amplifiers. These resistors provide for summation gains of 1 anti 10. In

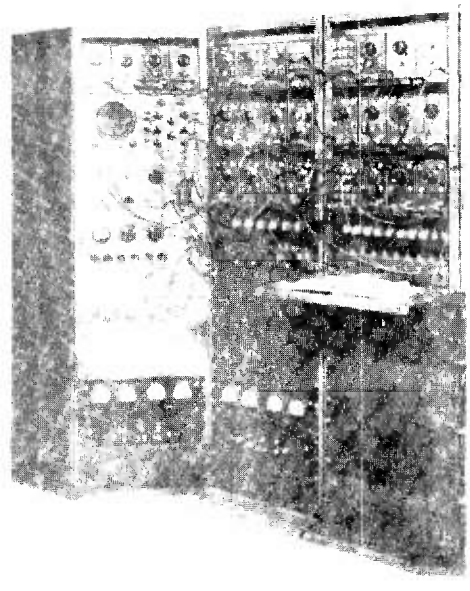


Figure 6-96. Typical analog-computer installation in which patching is accomplished by the use of cabling between components.

addition to the resistors pernianently connected to the patch bay, an assortment of additional resistors is provided. Each of these is arranged in a small housing provided with a plug and jack. The unit can be plugged into the appropriate terminal on the patch board, and in turn, a patch cable can be plugged into the jack. Decade resistance units that are direct-setting to within 1 percent also are available as plug-inunits. A problem set up on one of these removable boards can be taken off, if desired, and returned to the machine merely by replacing the board and setting the coefficient potentiometers to the required values. This system of patching provides considerable flexibility in setup, but if the problem utilizes most of the available equipment, the patch board becomes somewhat cluttered and difficult to follow. Because the board is large enough to allow for the addition of external resistors, it is rather difficult to plug into position.

Several other computer manufacturers have evolved a slightly different approach to the patching problem. In the computers developed by these groups, all the computing components are inside the machine, and the patching is accomplished by making the appropriate interconnections on the patch board with simple cables. Setup of a problem also requires adjustment of the coefficient potentiometers, but this is a simple operation. Because the patch board in these computers is used solely for the purpose of making interconnections, the terminals on the patch board can be placed closer together. One of these computers uses a metal patch board as shown in Fig. 6-97. A metal board is used in order to confine all leakage currents to ground paths and to prevent terminal-to-terminal leakage. Fig. 6-98 shows a front view of the computer with its metal patch board in place.

Mistakes made in patching represent one of the principal sources of errors in the solutions obtained from electronic differential analyzers. In any complex problem, the occurrence of patching errors can be reduced significantly if, after one operator sets up a patch board, a second operator checks each and every connection against the setup diagram. A group in the Aeronautical Hesearch Laboratory at the Wright Air Development Center has carried the process of patch-board checking one step farther with a unit that they have designated as a patch-board verifier. A prewired patch board is inserted in a standard receptacle in this device. The circuitry of the patch-board verifier is arranged in such a way that each terminal of the board is examined in turn and a record is printed in coded form of all the interconnections on the board. In addition to automatic preparation of a list of interconnections in a form that can be checked readily, this unit provides an electrical check on each patch cable and, thus, indicates open or short-circuited cables.

At the present time, small problems can be set up on a computer with very little trouble, but the difficulties associated with the initial setup and checking of patch boards and coefficients increase rapidly with an increase in the complexity of the problem studied.

6-5.9 Programming Equipment

In the simplest applications of differential analyzers, the programming equipment takes the form of a switch by means of which the computer can be put into a "RESET", "HOLD", or "OPERATE" condition. In the RESET position, the desired initial conditions are established in all the integrators and servos in the computer. Some studies are made merely to learn how a system responds as it comes to rest after being released with a given set of initial conditions. In such cases, a solution is obtained merely by switching the computer from the RESET to the OPERATE position. After steady-state conditions have been reached, the computer can be switched to the HOLD position, where the solution is "frozen".

Instead of obtaining a transient solution of this type, the analyst may wish to see how the system responds to a prescribed continuous forcing function, such as a sinusoid or a random signal. Programming in this case is only slightly more complicated than in the situation just described. In a problem where only tlie steady-state solution is required, the computer is placed in the OPERATE position, the forcing function is connected, and the response of the system is observed on a recorder until steady-state conditions are reached.

Simple manual operation of the control switches on tlie computer and recorder is

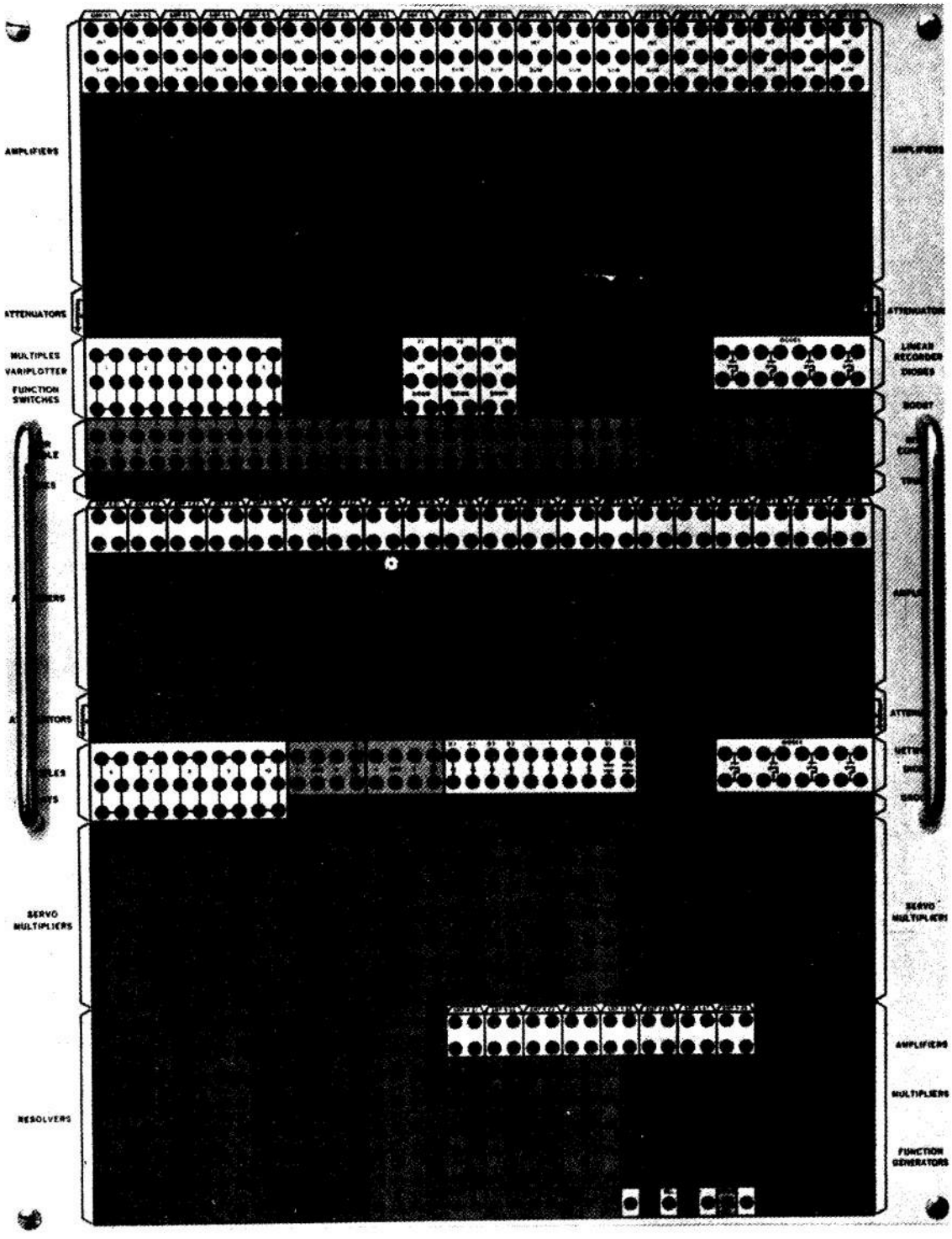


Figure 6-97. A removable metal patch board.

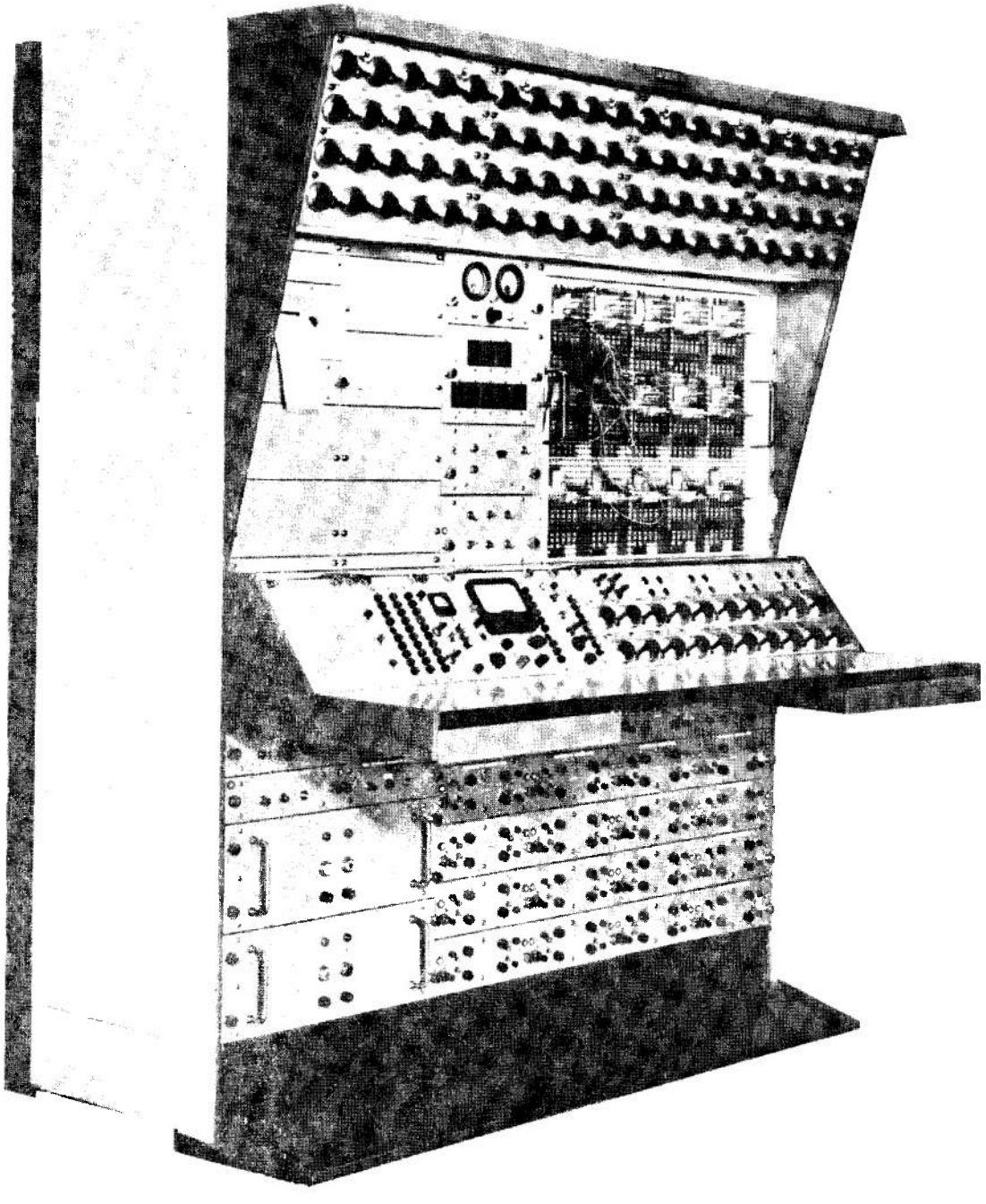


Figure 6-98. A general-purpose analog computer with its metal patch board in place.

entirely satisfactory when a small number of runs is desired. However, if many runs are required, as in a statistical study, or if electromechanical computing elements are being used under conditions where they may be driven into their limit stops at the end of the solution, strain on both the equipment and the operator can be reduced considerably if automatic-sequencing equipnient is installed in tlie computer.

The flexibility of this equipment should lie consistent with the variety of problems that niay be studied on the computer. For a computer that is to be employed in a variety of studies, one good solution is the provision of a collection of tiniers and switching amplifiers that can be patched together and into the coniputer setup in the same way as the computing elements. The timers should be of the adjustable-interval type and should be designed to start when an electrical signal is applied. At tlie end of a predetermined interval, which may be adjustable from 1 to 30 seconds, a pair of contacts in ttie timer is closed. These contacts can be used to initiate any of a variety of actions in tlie computer.

Automatic-sequencing equipment could be used with a coniputer to establish the following series of events during the solution of a typical missile-trajectory-type problem. First, tlie prescribed initial conditions are established throughout the computer. Second, in order to establish the zero for each channel of the recorder, the paper feed in tlie recorders is started and run a few inches with zero input into tlie recorder amplifiers. After a few inches of paper have been run, scale-marking pips are recorded to give a permanent record of the scale used in recording each variable. After this marker is recorded, a code nuniber is stamped automatically on each record for use in identifying that particular solution, and an inch or two of record is run to show ttie initial values of tlie recorded variables. Next, a very short niarker pip is recorded on each channel to indicate the exact start of the problem, and, simultaneously, the computer is switched into the operate condition. During the course of the computation, changes in the forni of the system may occur, either after a predetermined time interval or when some variable in the system passes through a prescribed value. Finally, when the range reaches some predetermined minimum value or begins to increase because a miss has occurred, this fact is detected by a switching amplifier, the computer is switched to the hold or reset position, and the recorders are stopped. The sequencing equipment can be arranged to run a single solution and stop, to repeat solutions with identical solutions until given a command to stop, or to sequence one or more parameters through a prescribed set of values and then stop.

Equipment of this type can reduce very significantly tlie burden of repetitious operations that, otherwise, the operator would be required to do. This saving is particularly important if the computer is being used in statistical studies such as those involved in an investigation of the effect of radar noise on missile trajectories. Without the automatic-sequencing equipment, the full attention of tlie operator would be required during tlie runs, and numerous solutions might be spoiled inadvertently owing to operator fatigue. With automatic equipment of ttie type described, the operator must be on the lookout only for machine malfunction.

6-5.10 OUTPUT AND OVERLOAD EQUIPMENT

If useful results are to be obtained from a computer, means must be provided for observing and recording computer variables. All of the modern general-purpose computers employ voltages as the analog quantity, and consequently, the problem of recording variables is exactly that of recording voltages. In some cases, a record of the way in which voltage varies as a function of time may be desired, while in others the requirement may be to record one voltage as a function of some arbitrary voltage generated as part of the coniputation. In a real-time computer, recording of tlie first type usually is performed with a strip recorder, while in tlie second case a plotting board is used. In repetitive computers, cathode-ray oscilloscopes are used.

6-5.11 Strip Recorders[14]

Fig. 6-99 shows the type of strip recorder that is used commonly as an output element for analog computers, and Fig. 6-100 shows a graph made with such a recorder.

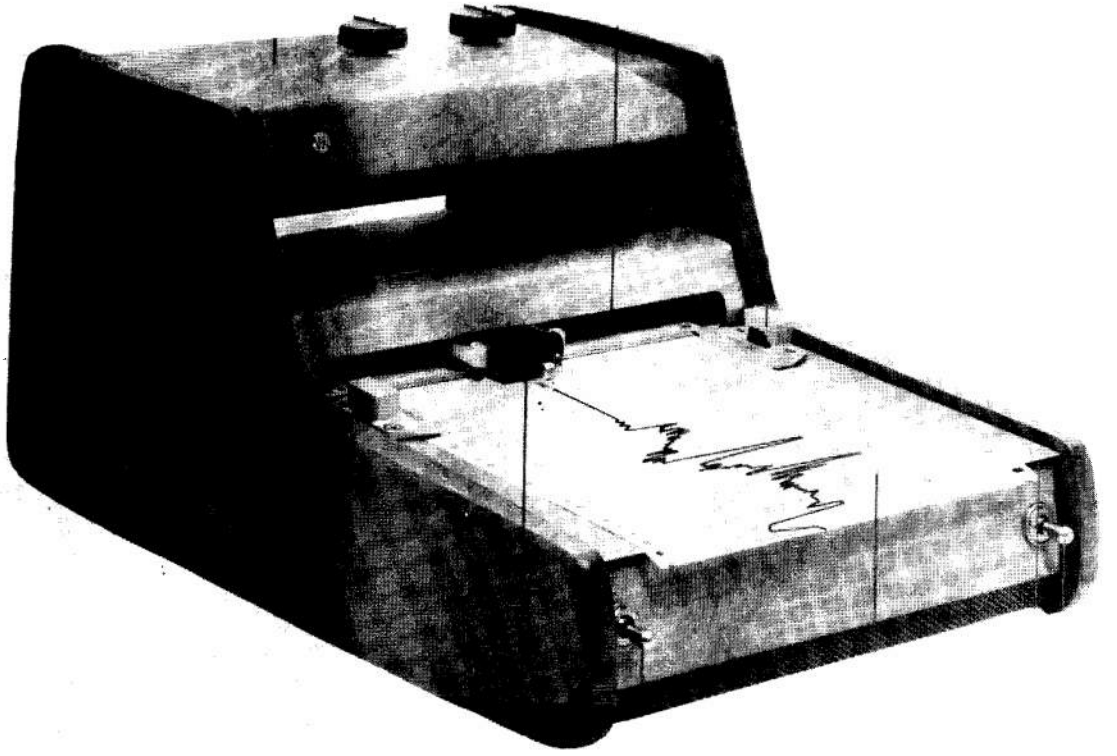


Figure 6-99. A typical strip recorder.

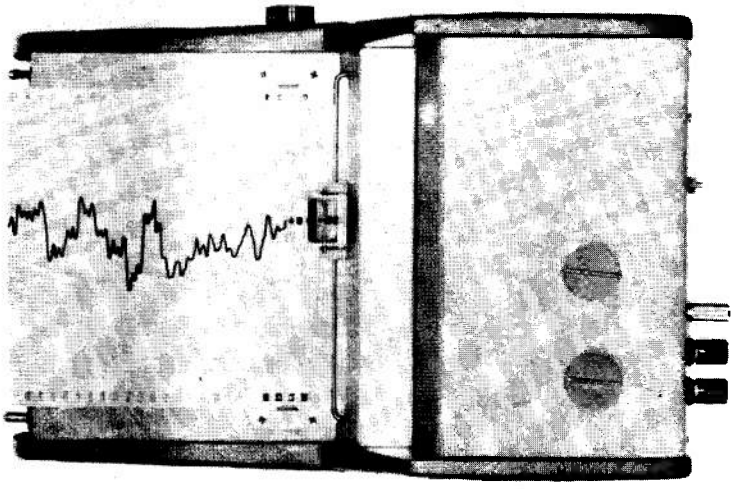


Figure 6-100. Graph made from a typical strip recorder.

Recorders of this type consist of a chart-drive mechanism and one or more pen actuators with their associated pens. The drive mechanism, which is actuated by an electric motor, is arranged so that any one of several chart speeds can be selected, either by the simple movement of a shift niechanism or by manually changing gears in the mechanism. Speeds in the range from 0.25 to 25 cm/sec are used in computer applications. The pen actuators usually employed in the United States are patterned after a D'Arsonval meter movement, and no attempt is made to improve their linearity or frequency response by closing a position-feedback loop around the unit. A servo recorder has been developed in England in which the feedback signal is derived from a linear potentiometer. The galvanometer unit supplies the torque to deflect the pen in the usual manner, but the spring that normally provides restoring torque is omitted.

With either type of pen actuator, a static linearity of approximately 2 to 3 percent is achieved. The amplitude response of the pen actuator with its amplifier is usually flat to 10 or 20 cps and can be considered usable, with reduced deflections, at frequencies as high as 60 cps.

In most. recorders ofthis type, the record is made with ink on untreated chart paper, but recorders also are available in which a hot wire produces a line by contact with specially treated paper. The pens of an ink recorder must be cleaned and filled periodically, but the cost of the paper for these recorders is much less than for a recorder that uses treated paper.

The need frequently arises to reproduce selected recordings made during a computer study. The paper normally used for ink recording is sufficiently transparent for reproduction by the usual semidry reproduction processes, but in most cases an exposure that produces a satisfactory image of the grid lines does not reproduce the graph well, and vice versa. The treated paper used with hot-wire recorders usually is not transparent. In either case, satisfactory reproductions can be made by xerographic reproduction*: or by photographing the record and reproducing it by offset printing.

6-5.12 Plotting Tables

Plotting tables are used for recording outputs from a computer when a plot of one variable as a function of some other arbitrary variable in the computation is desired or when greater recording accuracy than can be achieved with a strip recorder is needed. Fig. 6-101 shows a typical plotting table. The unit is arranged with an arm and a carriage that move at 90 degrees to each other in a system of rectangular coordinates. The motions of the arm and the carriage are controlled by position servos that derive their feedback information from rectilinear potentiometers. One potentiometer is mounted to the frame, and the wiper is carried by the arm, whereas the other is mounted on the arm, and the wiper is attached to the carriage. A typical plotting table may have a useful surface 30 inches square. The servos usually permit a maximum writing speed of approximately 8 in./sec with a maximum pen acceleration of 150 in./sec/sec. The static accuracy of the servos is limited by the linearity of the feedback potentiometers and can be made to approach 0.05 percent of full scale.

A plotting table usually is equipped also with a pen-lift circuit so that the pen can be lifted free of the plotting surface while it is being repositioned between solutions or while a new sheet of paper is being positioned. Another feature often provided is a timing device for making timing marks at predetermined intervals during the plotting. These marks can be made either with the pen used for recording the solution or with an auxiliary pen.

Tables can be obtained with either one or two writing pens. On tables with two pens, automatic pen switching is provided if the entire plotting surface is to be utilized simultaneously for both plots.

*In xerographic reproduction, graphic matter is copied by the action of light on an electrically charged photoconductive insulating surface. In this process, the latent image is usually developed with powders that adhere only to the areas that remain electrically charged. The image formed by the powders can then be transferred to a sheet of paper.

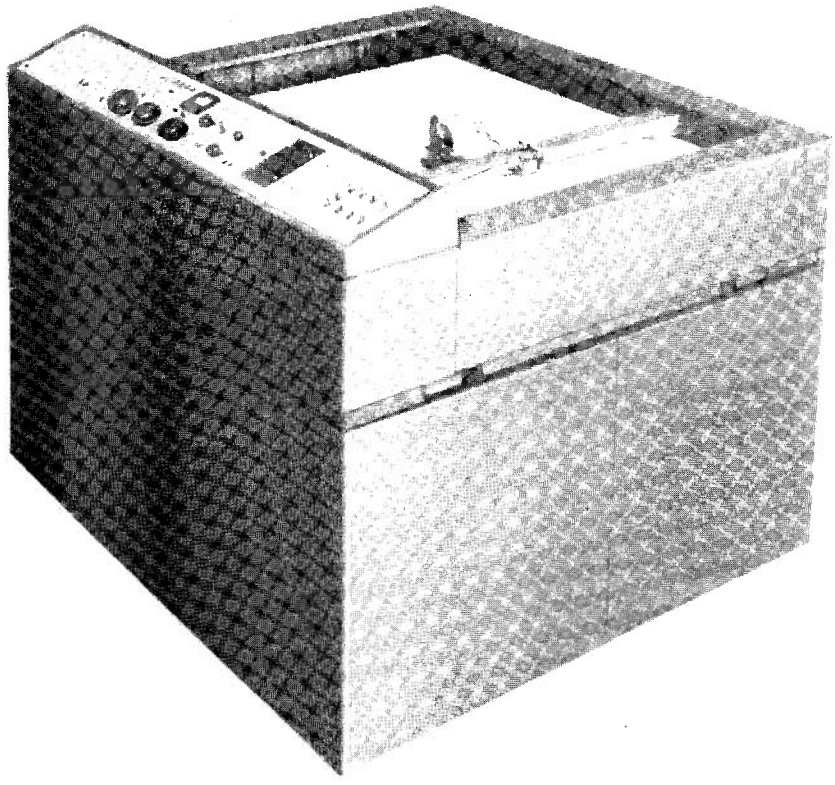


Figure 6-101. A typical plotting table,

A novel and very useful feature that is available on some plotting tables is a vacuum system for holding the paper. The recording paper is positioned on the writing surface, and a vacuum pump is started to hold the paper securely in position.

Plotting tables are available either with a horizontal plotting surface, as shown in Fig. 6-101, or a vertical surface for use where floor space is limited. Table-top models such as shown in Fig. 6-102 are also widely used.

6-5.1 3 Oscilloscopes

Although oscilloscopes seldom are used for recording the outputs of real-time computers, they serve as important monitoring devices, especially in a-c computers, and are the principal means for recording the outputs of repetitive computers. Most oscilloscopes with a frequency response that is essentially uniform down to frequencies somewhat below the computer repetition frequency can be employed satisfactorily; how-

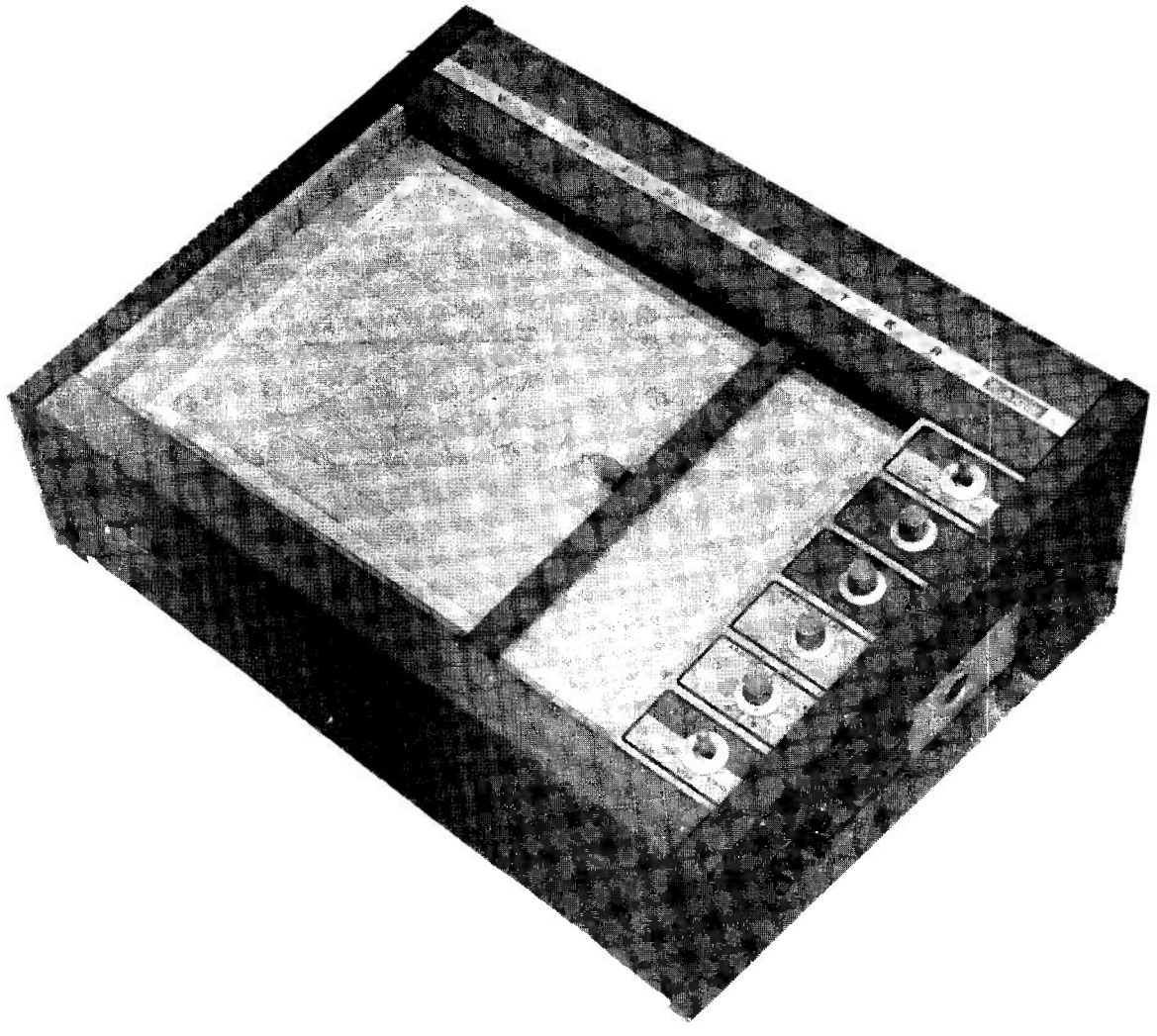


Figure 6-102. A typical table-top plotting table.

ever, a dual-beam oscilloscope enables direct comparison between two voltages in the computer. A single-beam oscilloscope with an electronic switch can be used for the same purpose, although the intensity of the traces is reduced somewhat. Four signals can be observed simultaneously if a dual-beam oscilloscope with two electronic switches is used. If quantitative results are to be obtained, a flat-faced cathode-ray tube should be used to minimize distortion. For direct viewing a tube with a persistence of 10 to 15 seconds will satisfactorily retain the trace of the complete solution.

6-5.14 Servo and Digital Voltmeters

Voltmeters in which the output appears directly as a series of digits rather than as the position of a pointer on a scale are becoming increasingly popular for monitoring the signals in real-time electronic differen-

tial analyzers. Some voltmeters of this type are simple position-servo devices in which a motor drives a linear potentiometer to produce a null between the potentiometer output voltageand the input voltage to be measured. The output is displayed by a counter that is geared to the potentiometer.

A second type of direct-reading voltmeter is designed on the principle of an electronic analog-to-digital conversion unit[15]. In one voltmeter of this type, the need for moving parts is completely avoided by the use of an output-displayunit that consists of a stack of ten plastic blocks, each of which has a numeral carved into it. A smalllamp bulb is arranged adjacent to each block. Fromthe front of the stack, only thenumeral next to an illuminated lamp is visible. Four- or five-digit registers of this type can be mounted in a cabinet with the conversion equipment or can be mounted at one or more convenient locations in the computer.

6-5.15 Overload Indication Circuits

Even operators who have had considerable skill in setting up computer studies experience difficulty in selecting scale factors that ensure signal voltages high enough to provide operation well above the noise level but low enough to prevent a loss in accuracy due to overloading of the computing components. Unless the computer is equipped with an overload-indication system, overloading of one or more of the computing units may occur during solution and pass unnoticed.

Overload-indication circuits for operational amplifiers can be designed on several different bases. In one scheme, the presence of anoverload is based directly upon the voltage developed by the output stage of the amplifier. The overload indication is provided by a gas-discharge tube that is connected to the output stage and biased to fire at the prescribed voltage. The voltage developed by the output stage, however, is not a true indication of linear operation of the amplifier. For example, the maximum voltage that can be developed by an amplifier within its linear range depends upon the load connected to the amplifier and the condition of the output tube.

In amplifiers that employ chopper stabilization, the voltage developed by the stabilization amplifier normally is low as long as the amplifier is operating linearly, but rises sharply if the amplifier is driven into the nonlinear region. Consequently, the voltage developed by the stabilization amplifier can be used to indicate overloads. This indication is not based upon a specified voltage at the amplifier output, but purely upon the occurrence of an overload. If an amplifier that has a nominal maximum output of 100 volts has to deliver 130 volts and does this without requiring an excessive voltage at its error point, it is presumably operating satisfactorily and no changes in scaling are necessary. On the other hand, if under a certain condition an amplifier becomes nonlinear at 90 volts, this overload system indicates the fact even though the amplifier still is operating below the nominal maximum output.

Indicator lights that are used to show that an overload has occurred can be located on each amplifier unit, or the overload indicators for all units can be grouped on a single panel for convenience.

6-5.16 CONSTRUCTION TECHNIQUES AND MAINTENANCE CONSIDERATIONS

If a computer is to perform in a satisfactory manner over a period of time, the following conditions must be met by the computer:

(1) It must be designed to meet the required specifications.
(2) It must operate reliably for extended periods of time, possibly under widely changing environmental conditions.
(3) It must have been designed in such a way that defective units can be located readily and replaced or repaired quickly.

A number of the problems associated with the design of individual computing elements have been discussed earlier in this chapter. However, the need to follow accepted procedures for the design and construction of electronic, and electromechanical equipment cannot be stressed too strongly. Maintenance-engineering principles are discussed in Ref. 20 and these principles generally apply to this specific problem of computer design. Some aspects of maintenance and checking as

applied specifically to computers are discussed in the paragraphs which follow.

6-5.17 Maintenance and Checking

The fact that a correct computer representation of a system has been arrived at and that the required computing equipment is available does not insure that correct solutions will be obtained from a computer. Correct results can be obtained only if the initial design of the components was satisfactory for the particular application, if the performance of the components has not deteriorated seriously, and if no patching errors have occurred. The computer operator usually is not responsible for the initial design of the equipment. However, if he is to obtain valid results, he must see that the equipment is maintained properly and he must employ operating procedures that enable him to detect errors in the computer solutions. These problems of maintenance and checking are discussed briefly in this section.

6-5.18 Maintenance

Two schools of thought exist with regard to the manner in which maintenance should be carried on. One group recommends that equipment be operated until its performance is no longer acceptable; this procedure is termed breakdown maintenance. The other group recommends that each piece of equipment be tested periodically and repaired if its performance has deteriorated appreciably; this procedure is termed periodic maintenance.

An attempt to arrive at an optimum maintenance program must begin with a statement of the optimization criterion. A criterion that is useful in the case of computers used for carrying out design studies is that the overall cost of maintenance should be minimized under the condition that cost includes both the direct expenses of maintenance and an appropriate charge for the time that the computer is out of service as a result of component failures[16]. If periodic maintenance is to be employed, careful consideration must be given to the procedures followed. One approach that might be taken specifies that, during maintenance, all vacuum tubes in a unit should be replaced each time the unit undergoes its periodic tests. Another procedure requires only that the performance of a unit meet a set of normal operating specifications. Neither of these procedures accomplishes what should be the real purpose of periodic testing of components. Such testing is effective only if, as a result of the tests, definite assurance is obtained that the tested component has a higher probability of continuing to give satisfactory performance for some specified period than if it had never been tested. This statement implies that the testing procedures used in a satisfactory periodic maintenance program should enable the operator to predict with some assurance that a unit that passes the tests should have a high probability of continuing to give satisfactory performance at least until the next maintenance check. The marginal-checking procedures widely used as an aid in improving the reliability of digital computers are designed to locate elements that probably would fail relatively soon with continued use. Although these procedures are not applied readily for checking a complete analog computer, they can be applied effectively during the testing of individual amplifiers at a test position.

A simple marginal check that can be made on an amplifier consists first of measuring the open-loop gain of the amplifier at one frequency, which may be any frequency near the center of the passband, using normal supply voltages. Then, the heater supply voltage is reduced by 10 to 15 percent and the open-loop gain is measured again. If the loop gain drops only slightly as a result of the change in heater voltage, the probability is high that the tubes will continue to give satisfactory operation for a number of additional hours. On the other hand, a marked drop in loop gain indicates that the emission characteristics of one or more tubes in the unit are deteriorating rapidly, and the unit may fail shortly. In chopper-stabilized amplifiers, this test can be carried out separately on the main sections and on the stabilizing sections of the amplifiers.

If a periodic maintenance program is to be really effective, a complete history should be kept on each unit to show when it was tested and what changes were required before the unit met test specifications. Analysis of such records aids in the determination of the optimum length of the maintenance cycle and

possibly suggests slight circuit modifications if certain components show high failure rates.

With small computer installations, the usual practice is to follow a breakdown-maintenance procedure because, in such installations, the failure of a component is noted relatively easily, and the cost of machine time lost while the component is being replaced is small. On the other hand, periodic maintenance has proved the more economical procedure in the operation of large installations. However, experience has indicated that if a computer is set up for the study of a large problem, which may be on the machine for a period of three to six weeks, computing components should not be removed from the setup merely so that they can be given their periodic inspection. Components in use should be maintained purely on a breakdown basis. In the case of equipment for field use, it is probably impractical to do much but perform breakdown maintenance, except when the computer is returned to the operating base for overhaul.

6-5.19 Checking

A good maintenance program does not guarantee that the results obtained from a computer will be correct. In addition, the operator must use thorough and systematic checking procedures. Theoretical error-analysis techniques have been investigated[17] but these are of relatively little aid in computer operation. The checking procedures that are used can be classified as methods that provide complete checking and those that provide partial checking.

The most conclusive check that can be made on a computer solution is obtained by comparison of the computer solution with an analytic or numerical solution of the identical problem. Usually, analytic solutions are unobtainable, and for a complicated problem considerable time is required both to program and to run a solution even on a large-scale digital computer. Consequently, the analog-computer operator usually resorts to partial checking, with possibly one overall numerical check solution if a particularly complicated system is being studied.

Regardless of what checking procedures are used, correct results cannot be obtained if the mathematical model of the system is not formulated correctly. If possible, several analysts should take part in the preparation of the mathematical model, and each step in the process -- particularly if approximations are involved -- should be examined carefully.

The next step involves translation of the mathematical statement of the problem into a computer setup diagram. This step can be checked effectively by rewriting the mathematical equations from the computer diagram alone.

After the computer diagram has been checked, the operator is ready to proceed to wire the patch boards, set coefficient potentiometers, and make what other setup adjustments are required. As a step in the check-out of a complete computer setup, static checking is a simple and effective procedure. For a static check, the output of each integrator is disconnected from its load, and in its place is substituted a fixed voltage. The voltages that should appear throughout the computer when the fixed test voltages are applied can be calculated independently from the setup diagram and compared with the voltages measured in the computer. Although this method is effective, it does not check any of the integrators and, furthermore, requires removing some patch cords and inserting test voltages. Whenever changes of this type are made, the chance exists of reinserting patch cords in the wrong position, and special care should be taken to avoid the introduction of new errors.

6-5.20 ENVIRONMENTAL EFFECTS

6-5.21 Size, Weight, and Power Considerations

The constraints on the design of computing equipment for use at fixed locations are relatively simple to meet compared with those imposed on equipment for field or airborne use. At fixed locations, size is not an especially critical factor, temperature can be controlled by air conditioning, and a well-regulated primary power supply can be provided. However, for field or airborne use, severe constraints are placed on the size and weight of equipment and also on the total power consumed. The use of transistors and semiconductor diodes in place of vacuum tubes offers very significant possibilities for re-

ducing tlie size, weight and power consumption of electronic coinputing equipment. Transistorized analog coniputing equipment is in widespread use and the performance achieved with transistorized operational amplifiers is rapidly approaching that achieved with the more-conventional vacuum-tube equipment.

The possihility of using purely mechanical elements in computers for field or airborne use should not be overlooked since such devices can be made quite small and reliable under widely varying environmental conditions.

The regulation of the primary power source used to feed a field or airborne coniputer is frequently rather poor and this factor must be taken into account in the design of a computing systeni. While a well-designed high-voltage regulator should maintain a constant output voltage in spite of significant changes in line voltage, the regulator will cease to function properly if the supply voltage drops too low. The range of line-voltage variations that can be tolerated by the voltage regulators can be extended, but only at the expense of increased size, weight, and power dissipation at normal supply voltage. Consequently, in designing a complete system, attention should be given to providing reasonably good regulation in the primary power supply in order to reduce the design problems associated with the individual computing elements. Furthermore, fluctuations in the voltage supplied to the heaters of the tubes in an operational amplifier can cause offsets and high integrator drift-rates. Consequently, it may be desirable to supply the heaters from a constant-voltage transformer.

6-5.22 Temperature, Humidity, Altitude, Shock, and Vibration

The general considerations to be followed in designing equipment to withstand temperature variations, high humidity, high altitude, and severe shock and vibration are outlined in Chapter 5 of Ref. 20. Accepted good design practices should be followed in the design of all computing equipment. Special precautions should be taken to avoid high-temperature regions in the equipment; adequate spacing should be provided between conductors so that arcing will not occur at high altitudes and so that leakage will not become significant under conditions of high humidity and varying temperature, when moisture may condense on the equipment. Furthermore, adequate mechanical strength must be provided at all points so that no damage will result from shocks and vibration to which tlie unit may be subjected. Even though every care is taken to follow good design practices, units intended for field or airborne use should be tested under cycles of temperature and humidity and should be subjected to shock and vibration tests before final acceptance of the design. Only in this way can possible weaknesses in design be detected before the units are subjected to actual field tests.

APPENDIX TO CHAPTER 6

THE BASIC OPERATIONS OF MATRIX ALGEBRA

Matrices are useful as a short notation for systems of simultaneous equations. The rules for matrix operations provide a convenient means of keeping track of the solutions of such systems of equations. The following material defines the various types of matrices employed in matrix algebra and summarizes the rules that govern the use of matrices.

A matrix is a rectangular array of mn quantities, called an "m by n matrix," arranged in m rows and n columns. These mn quantities are called the elements of the matrix. If m = n, i.e., number of rows equals the number of columns, the matrix is said to be a square matrix of order n. The element of a matrix that is in the ith row and jth column, where i may have any value from 1 to m and j may have any value from 1 to n, is called the general element of the matrix; a usual notation is a_{ij}.

Matrices, even though without numerical value, can be treated as entities and thus can be added, subtracted, multiplied, orhave other operations performed on them. Such arrays offer a particulary convenient method for calculating simultaneous changes in a series of related variables. The mechanics of the matrix algebra is illustrated below.

(1) Addition

Additionof any pair of matrices [A] and [B] is possible only if the number of rows and the number of columns respectively are equal. Addition is both associative and commutative, i.e.,

$$([A] + [B]) + [C] = [A] + ([B] + [C]) = [A] + [C] + [B] = [C] + [A] + [B] \text{ etc.}$$

Addition is performed row by row, each element in each column of the first matrix is added to the corresponding element in the corresponding column of the second matrix, thus forming one matrix. The process is illustrated:

$$[A] + [B] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & -1 & -2 & -3 \end{bmatrix} + \begin{bmatrix} 6 & 2 & 2 & 0 \\ -4 & -8 & 0 & -1 \\ -9 & 5 & 4 & 7 \end{bmatrix} = \begin{bmatrix} 7 & 4 & 5 & 4 \\ 1 & -2 & 7 & 7 \\ 0 & 4 & 2 & 4 \end{bmatrix}$$

(2) Subtraction

Subtraction is the same as addition except that the corresponding elements are subtracted from each other. The associative and commutative properties apply. The process is illustrated:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & -1 & -2 & -3 \end{bmatrix} - \begin{bmatrix} 0 & 2 & -6 & 0 \\ 7 & 5 & 3 & 4 \\ 6 & -3 & -9 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 9 & 4 \\ -2 & 1 & 4 & 4 \\ 3 & 2 & 7 & -4 \end{bmatrix}$$

(3) Multiplication

Multiplication is a more complex process in which the two matrices, [A] and [B], need not be the same size. However, the two matrices must be compatible, i.e., the number of columns of the left matrix must equal the number of rows of the right matrix. Thus the multiplication of the matrices [5 X 3] X [3 X 8] is possible. The multiplication would result in another matrix of size 5 X 8 with each element of the matrix consisting of the sum of three terms.

Multiplication is associative but, except for a special case, not commutative. It is emphasized that the matrix on the left multiplies the matrix on the right, i.e., premultiplication.

$$[A] \cdot ([B] \cdot [C]) = ([A] \cdot [B]) \cdot [C]$$

$$[A] \cdot [C] \cdot [B] \neq [A] \cdot [B] \cdot [C]$$

In multiplication each term in the upper row of the left matrix successively multiplies the corresponding term in the first column of the right matrix, the sum of the resulting products is the number entered into the position at column 1, row 1 of the product matrix. The upper row of the left matrix is now used in an identical manner with the second column of the right matrix to find a value for the position at column 2, row 1 of the product matrix. This operation is repeated with the first row of the left matrix multiplying every column of the right matrix. The entire operation is repeated with each row of the left matrix. The process is illustrated:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} \end{bmatrix}$$

$$\begin{bmatrix} 6 & 1 & 8 \\ 0 & -3 & 5 \\ 1 & 2 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 6+3+0 & 12+4+40 \\ 0-9+0 & 0-12+25 \\ 1+6+0 & 2+8+30 \end{bmatrix} = \begin{bmatrix} 9 & 56 \\ -9 & 13 \\ 7 & 40 \end{bmatrix}$$

(4) Multiplication by a Constant

If a matrix is multiplied by a constant, each element of the matrix is multiplied by the constant.

$$k[A] = \begin{bmatrix} ka_{11} & ka_{12} & ka_{13} \\ ka_{21} & ka_{22} & ka_{23} \end{bmatrix}$$

$$2\begin{bmatrix} 1 & 2 & 3 \\ 4 & -2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 8 & -4 & 0 \end{bmatrix}$$

(5) Transpose

The transposed matrix of $[A] = [a_{ij}]$, indicated by $[\tilde{A}]$ or $[A]^T = [a_{ji}]$, is formed from [A] by interchanging rows and columns.

$$\text{If } [A] = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad \text{then } [\tilde{A}] = \begin{vmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{vmatrix}$$

(6) Identity Matrix

The identity matrix or unit matrix has unity for elements along the main diagonal (the diagonal from the upper left to the lower right corners). All other elements are zero, i.e.:

$$a_{ij} = 1 \text{ if } i = j; \quad a_{ij} = 0 \text{ if } i \neq j$$

The notation for this matrix is [I].

$$III = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 1 \end{vmatrix}$$

For every matrix:

$$[A][I] = [I][A] = [A]$$

(7) Inverse Matrix

This operation is a time-consuming operation when performed by hand but is easily handled by the electronic computer. Only a square matrix has an inverse. It is assumed that the reader is familiar with the algebra of determinants. The notation for the inverse of [A] is $[A]^{-1}$.

The finding of the inverse of a matrix, or the "reciprocal" as it is sometimes called, involves four steps: (1) replace each element of the matrix by its cofactor, considering the

matrix now as a determinant; (2) transpose the newly formed matrix; (3) evaluate the original matrix, considering the matrix as a determinant; and (4) arrange the quantities resulting from the previous steps. The operation of finding the inverse of a matrix is illustrated:

$$A = \begin{vmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{vmatrix}$$

Step 1: Find matrix of the cofactors, remembering the rules of algebraic sign.

$$\begin{bmatrix} (1-2) & -(1-4) & (2-4) \\ -(1-0) & (2-0) & -(4-4) \\ (1-0) & -(2-0) & (2-1) \end{bmatrix} = \begin{vmatrix} -1 & 3 & -2 \\ -1 & 2 & 0 \\ 1 & -2 & 1 \end{vmatrix}$$

Step 2: Form transpose:

$$\begin{vmatrix} -1 & -1 & 1 \\ 3 & 2 & -2 \\ -2 & 0 & 1 \end{vmatrix}$$

(Steps 1 and 2 result in what is called the "adjoint" of matrix A.)

Step 3: Evaluate determinant of A:

$$[A] = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{bmatrix} = 2(1-2) - 1(1-4) + 0(2-4) = -2+3+0 = 1$$

Step 4: Form the inverse:

$$[A]^{-1} = \frac{[\text{adjoint}]}{[A]} = \frac{\begin{bmatrix} -1 & -1 & 1 \\ 3 & 2 & -2 \\ -2 & 0 & 1 \end{bmatrix}}{1} = \begin{bmatrix} -1 & -1 & 1 \\ 3 & 2 & -2 \\ -2 & 0 & 1 \end{bmatrix}$$

A unique property of the inverse is the relationship:

$$[A]^{-1}[A] = [A][A]^{-1} = [I]$$

REFERENCES

1. J. E. Alexander, and J. M. Bailey, Systems Engineering Mathematics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962, Chapter 5.
2. W. W. Seifert and C. W. Steeg, Jr., Control Systems Engineering, McGraw-Hill Book Company, Inc., New York, N. Y., 1960, Chapter 8.
3. W. W. Seifert, "Analog Computers", article in the Encyclopedia of Science and Technoloev. McGraw-Hill Book Company, Inc., New York, N. Y.
4. A. Fialkow, and I. Gerst, "The Transfer Function of General Two Terminal-Pair RC Networks", Quarterly Applied Mathematics 10, 113-127 (1952).
5. E. A. Guillemin, "Synthesis of RC-Networks", Journal of Mathematical Physics 28, 22-42 (1949).
6. E. A. Goldberg, "A High-Accuracy Time-Division Multiplier", RCA Review 13, 265-274 (1952).
7. C. D. Morrill and R. V. Baum, "A Stabilized Time-Division Multiplier", Electronics 25, No. 12, 139-141 (1952).
8. B. Chance, F. C. Williams, C. Yang, J. Busser, and J. Higgins, "A Quarter-Square Multiplier Using a Segmented Parabolic Characteristic", Review of Scientific Instruments 22, 683-688 (1951).
9. G. A. Korn, and T. M. Korn, Electronic Analog Computers, McGraw-Hill Book Company, Inc., New York, N. Y., 1956.
10. W. W. Seifert and C. W. Steeg, Jr., Control Systems Engineering, McGraw-Hill Book Company, Inc., New York, N. Y., 1960.
11. I. A. Greenwood, Jr., J. V. Holden, Jr., and D. MacRae, Jr., Electronic Instruments, M.I.T. Radiation Laboratory Series, Vol. 21, McGraw-Hill Book Company, Inc., New York, N. Y., 1948, Chapter 16.
12. C. M. Edwards, "A New Approach to Grounding in D-C Analog Computers", Proceedings of the Western Joint Computer Conference, March 1-3, 1955, pages 23-26.
13. V. Bush, and S. H. Caldwell, "A New Type of Differential Analyzer", Journal of the Franklin Institute 240, 255-326, (1945).
14. C. Kaizer, "Chart Recorders", Electromechanical Design 8, No. 1, 305-314.
15. P. Barr, "Voltage-to-Digital Converters and Digital Voltmeters", Electromechanical Design 8, No. 1, 147-159.
16. €1. Jacobs, Jr., "Equipment Reliability as Applied to Analogue Computers", Journal of the Association for Computing Machinery 1, No. 1, 21-26 (1954).
17. K. S. Miller and F. J. Murray, "A Mathematical Basis for an Error Analysis of Differential Analyzers", Journal of Mathematical Physics 32, Nos. 2-3, 136-163 (1953).
18. R. L. Halfman, Dynamics: Particles, Rigid bodies, and Systems, Vol. 1, Addison-Wesley Publishing Company, Lnc., Reading, Massachusetts, 1962.
19. M. E. Connelly, Simulation of Aircraft, Report 7591-R-1, Servomechanisms Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, February 15, 1958.
20. AMCP 706-327, Engineering Design Handbook, Fire Control Systems - General.

GENERAL BIBLIOGRAPHY FOR ANALOG COMPUTERS

1. S. M. Grabbe, S. Ramo, and D. E. Wooldridge, Editors, Handbook of Automation, Computation, and Control, Vol. 2, Computers and Data Processing, John Wiley & Sons, Tnc., New York, N. Y., 1959.
2. C. L. Johnson, Analog Computer Techniques, McGraw-Hill Book Company, Inc., New York, N. Y., 1956.
3. G. A. Korn and T. M. Korn, Electronic Analog Computers, McGraw-Hill Book Company, Inc., New York, N. Y., 1956.
4. W. W. Soroka, Analog Methods in Computation and Simulation, McGraw-Hill Book Company, Inc., New York, N. Y., 1954.
5. J. N. Warfield, Introduction to Electronic Analog Computers, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1959.
6. W. W. Seifert and C. W. Steeg, Jr., Control Systems Engineering. McGraw-Hill Book Company, Tnc., New York, N. Y., 1960.
7. AMCP 706-293, Engineering Design Handbook, Surface-to-Air Missile Series Part Three, Computers.
8. H. H. Goode and R. E. Machol, System Engineering, McGraw-Hill Book Company, Inc., New York, N. Y.
9. A. Svoboda, Computing Mechanisms and Linkages, M.I.T. Radiation Laboratory Series, McGraw-Hill Book Company, Inc., New York, N. Y.
10. I. A. Greenwood, J. V. Holdam, Jr., and D. MacRae, Jr., Electronic Instruments, Vol. 21 of M.I.T. Radiation Laboratory Series, McGraw-Hill Rook Company, Inc., New York, N. Y., 1948.
11. C. A. A. Wass, An Introduction to Electronic Analogue Computers, Pergamon Press, Ltd., London, 1955.
12. E. A. Goldberg, "Stabilization of Wide-Band Direct-Current Amplifiers for Zero and Gain", RCA Review 11, 296-300 (1950).
13. S. Fifer, Analogue Computation, 4 Vols., McGraw-Hill Book Company, Inc., New York, N. Y., 1961.

CHAPTER 7
ANALOG-DIGITAL CONVERSION TECHNIQUES

7-1 PURPOSE OF CONVERSIONS

Automatic read-in and read-out of data between analog elements and digital-computing elements in fire control systems may call upon a wide variety of input-output devices to perform the necessary analog-to-digital (A/D) and digital-to-analog (D/A) conversions. For read-in, in addition to voltage-to-digital conversions, the requirement may be to convert shaft rotation, time interval, frequency, or a position into digital form. For read-out, again in addition to digital-to-voltage conversion, it may become necessary to convert a digital signal to mechanical motion. The speed and the accuracy with which these conversions can be made cover a wide range, more or less at the choice of the designer.

An extensive survey of A/D and D/A conversion devices can be found in Refs. 3 through 7. This chapter will briefly summarize the four following types of conversion:

1. Analog voltage to a digital output
2. Mechanical motion to a digital output
3. Digital signal to an analog voltage
4. Digital signal to mechanical motion

7-2 CONVERSION OF AN ANALOG VOLTAGE TO A DIGITAL OUTPUT

7-2.1 COMPARISON CIRCUITS

Analog-to-digital converters consisting of comparison circuits are commonly used for the purpose of converting an analog voltage to a digital output. These voltage-to-digital encoders frequently consist of several pieces of equipment, and fall largely into one of the two following classifications:

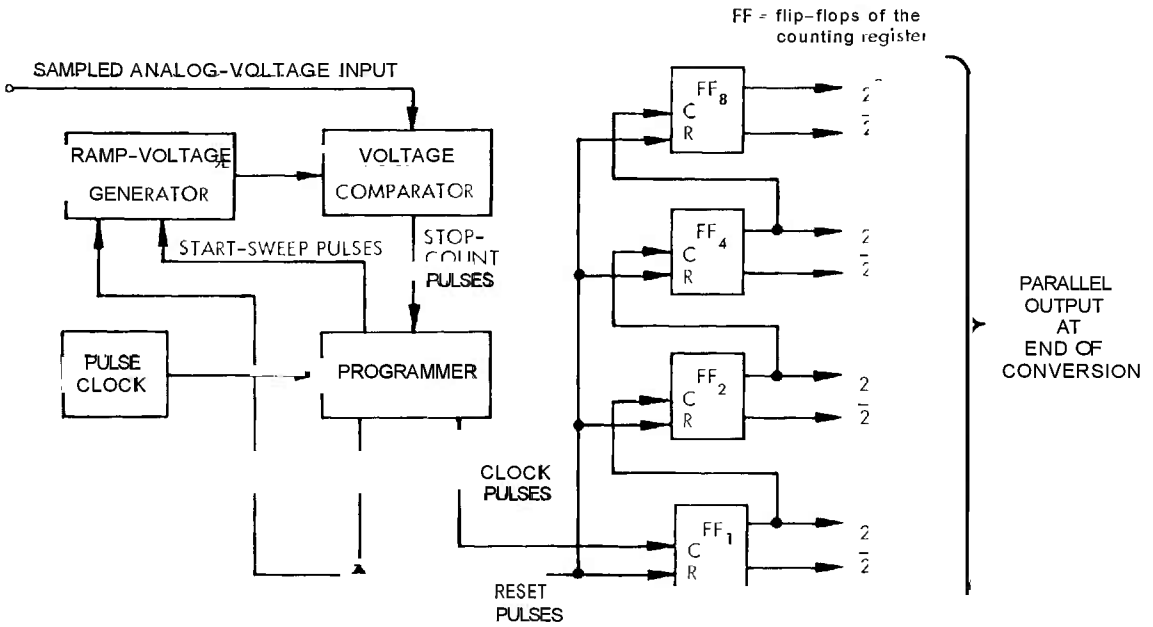
1. Level-at-a-time encoders (also called time-based encoders).
2. Digit-at-a-time encoders (also called feedback-voltage comparison encoders).

7-2.1.1 Level-at-a-time Voltage-to-digital Encoders

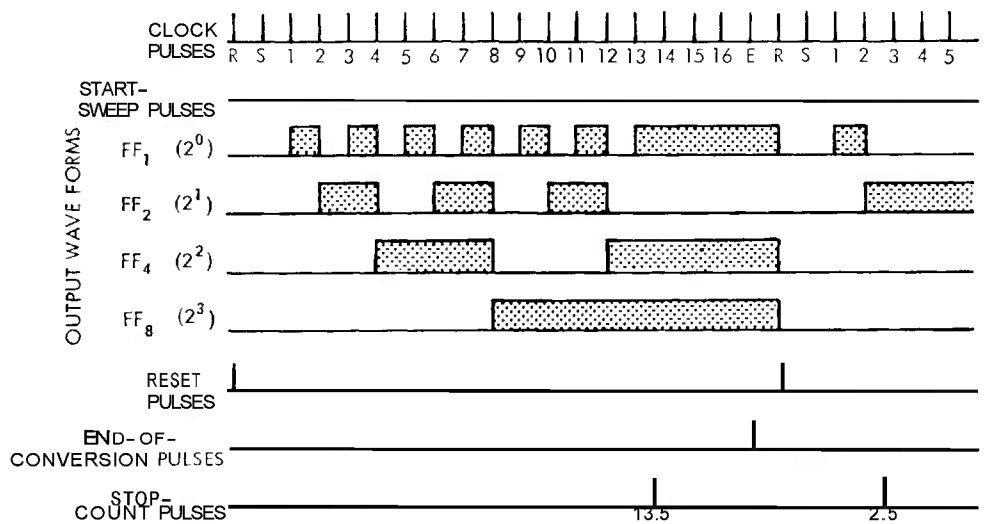
Fig. 7-1(A) gives the basic block diagram of a widely used type of analog-to-digital converter that compares a linearly rising ramp voltage with the input analog-voltage sample until they are equal. When the two voltages are equal, a binary counter is stopped at a count proportional to the analog-voltage input. This occurs because the binary counter is started at the same time as the ramp-voltage generator for each conversion, and because the counting register is designed to reach full count at the same time that the ramp voltage reaches full scale. (Full scale here corresponds to 16 counts.)

A digital programmer controls each conversion cycle, as shown in the waveform diagram of Fig. 7-1(B). The "start-sweep" pulse also controls a gate that steers clock pulses to the counting register. This register counts pulses until the voltage comparator detects equality between the input analog voltage and the ramp voltage, at which time a "stop count" signal is sent to the programmer to shut off clock pulses to the counter. At the end of 16 counts, an "end-of-conversion" pulse is sent out for use by associated equipment in transferring

* By E. St. George, Jr.



(A) Simplified Block Diagram



(B) Wave Forms

Figure 7-1. Simplified block diagram and associated wave forms for a level-at-a-time type of voltage-to-digital encoder.

the contents of the counting register to a storage area. A "reset" pulse then clears the counter in preparation for the next conversion, Although a parallel output is shown in Fig. 7-1(A), a parallel-to-serial converter may be used if serial read-out is desired—provided there is enough time allowed between the "end of conversion" pulse and the "reset" pulse for the register to be emptied in serial fashion.

Fig. 7-1(B) shows a complete cycle for the conversion of a 13.5-volt analog input to digital form (assuming a scale factor of 1 volt per count), followed by the beginning of a cycle for a 2.5-volt analog input. Even though there will be no further count accumulated for the 2.5-volt input after the last count shown, the full 16-count sequence must be completed before the "end of conversion" pulse will he generated. Although it may not be so apparent for a 4-bit counter, this type of conversion is very slow since clock pulses must be counted for every level that the counter is capable of storing, or 2^n clock pulses must be counted by an n-bit counter for every conversion. A 10-bit counter, for example, must count 1024 clock pulses since its resolution is 1 part in 1024. The accuracy of this type of converter is typically about plus or minus 0.1% of full scale, plus or minus 1/2 the least significant bit for a 10-bit counter plus sign bit. The maximum conversion rate for a 10-bit counter is approximately 100 conversions per second.

The ramp-voltage generator and the voltage comparator of Fig. 7-1(A) are both based on the use of operational amplifiers of the general type described in Chapter 6 (see par. 6-3.1). Because of the short cycle time, however, chopper stabilization is not usually necessary. On the other hand, a high gain-bandwidth product is required. The specifications for a typical transistor operational amplifier of a type suitable for this application are given in Table 7-1.

The ramp-voltage generator of Fig. 7-2 is derived from the electronic integrator of Fig. 6-38. As shown in Fig. 7-2, a solid-state switch (indicated by the mechanical-switch symbol) provides a means of initiating the ramp and of resetting to zero voltage. Initiation is accomplished by means of the "start-sweep" pulses shown in Figs. 7-1(A) and 7-1(B). Reset is obtained from the "reset" pulses that are also shown in these figures,

The principal errors in the ramp-voltage generator are given by the ideal input-output equation,

$$e_{o(ideal)} = -\frac{1}{RC}\int_o^T e_i\,dt \qquad (7\text{-}1)$$

and the corresponding equation with error terms included,

$$e_o = -\frac{1}{C}\int_o^T \left[\frac{e_i \pm E_{os}}{R} \pm I_{os}\right] dt \qquad (7\text{-}2)$$

where

E_{os} = the offset voltage of the operational amplifier

I_{os} = the offset current of the operational amplifier

and the other quantities are defined by Fig. 7-2.

As an example, assume a 10-bit counter with 100 conversions per second. Then T = 10 millisec. From Table 7-1, the offset voltage E_{os} can arise either from temperature variation or from voltage variation. (The long-term drift is unimportant because of the short cycle time.) For an assumed temperature variation ΔT of 60°C,

$$E_{os} \quad \left(30\ \frac{\mu volt}{^\circ C}\right)\Delta T \quad 1800\ \mu volt\ maximum \qquad (7\text{-}3)$$

For an assumed supply voltage variation ΔV_s of 10 millivolts,

$$E_{os} \quad \left(200\ \frac{\mu volt}{volt}\right)\Delta V_s \quad 2\ \mu volt\ typical \qquad (7\text{-}4)$$

The effect of temperature variation is by far the greater. From Table 7-1, the offset current I_{os} due to temperature variation is

$$I_{os} = 0.8\ \frac{nanoampere}{^\circ C}\ \Delta T$$

$$= 48\ nanoamperes\ maximum \qquad (7\text{-}5)$$

TABLE 7-1. VALUES FOR THE CHARACTERISTICS OF A TYPICAL HIGH-SPEED TRANSISTOR OPERATIONAL AMPLIFIER.

NOTE: VALUES GIVEN ARE FOR 25°C UNLESS OTHERWISE STATED.

CHARACTERISTIC	SYMBOL	JALUES	CONDITION	UNITS
Supply Voltage (3-wire D.C.)	V_s	±15	Design Center	Volts D.C.
Supply Current { Quiescent / Full output	I_s	±10 ±30	Max.	mA D.C.
Output-Voltage Range, Full Load	E_o	±10	Min.	Volts P-P
Output-Current Range	I_o	±20	Min.	mA P-P
Input Common-Mode Voltage Range	E_{cm}	± 3	Max.	Volts P-P
Voltage Offset Stability @ Const. Temp. (Long Term)	E_{os}	±100	Typical	Micro-volts
Offset Voltage Temperature Coeff. -25°C to +85°C	$\Delta E_{os}/\Delta T$	10 30	Typical Max.	μV/°C
Offset Voltage/Supply Voltage Stability Coefficient	$\Delta E_{os}/\Delta V_{cc}$	200	Typical	μV/V
Input Offset Current	I_{os}	±10 ±30	Typical Max.	Nano-amperes
Offset Current Temperature Coeff. 0 to + 55°C	$\Delta I_{os}/\Delta T$	0.2 0.5	Typical Max.	nA/°C
Offset Current Temperature Coeff. -25°C lo +85°C	$\Delta I_{os}/\Delta T$	0.8	Max.	nA/°C
Open-Loop Gain @D.C. R_L = 10K	A_o	$0.5x10^6$ —	'Typical Min.	—
Open-Loop Gain @ D.C. R_L = 500 ohms	A_o	$3.0x10^5$ $0.5x10^5$	Typical Min.	—
Unity-Gain Crossover Frequency	f_t	100	Typical	mC
Frequency Limit For Full Output (Unity-Gain Inverter)	f_p	800	Typical	KC
Differential Input Impedance @D.C.	Z_d	0.1	Typical	Meg-ohms
Common-Mode Input Impedance @ D.C.	Z_{cm}	—	Typical	Meg-ohtns

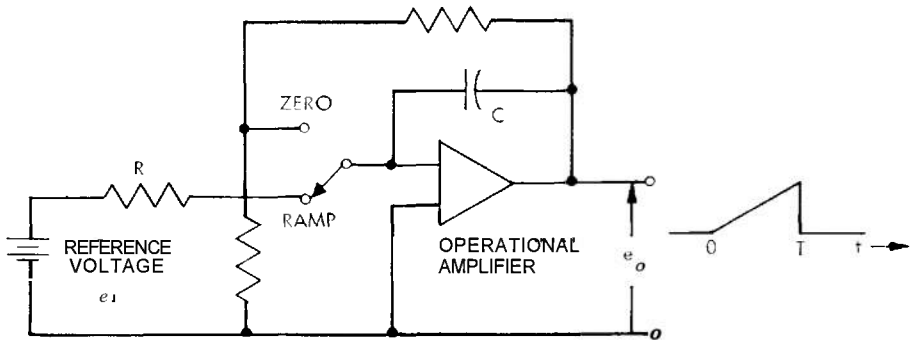


Figure 7-2. Schematic diagram of a typical ramp-voltage generator.

Eq. 7-2 can be rewritten to separate the ideal ($e_{o(ideal)}$) and error (Δe_o) terms as follows:

$$\frac{1}{RC}\int_0^T e_i\,d \quad \frac{1}{C}\left(1 \quad \frac{os}{R} + I_{os}\right)\int_0^T dt$$

$$_{o(ideal} \qquad \left[1\frac{E_{os}}{R} I\right]$$

$$e_{o(ideal)} \cdot e_o \qquad (7\text{-}6)$$

Since the offset voltage and the offset current may cause errors of the same sign, the worst-case error is

$$e_o \quad -\frac{1}{0.1\times10^{-6}}\left(\frac{1.8\times10^{-3}}{10^5} - 48\times10^{-9}\right)\left(10\times10^{-3}\right)$$

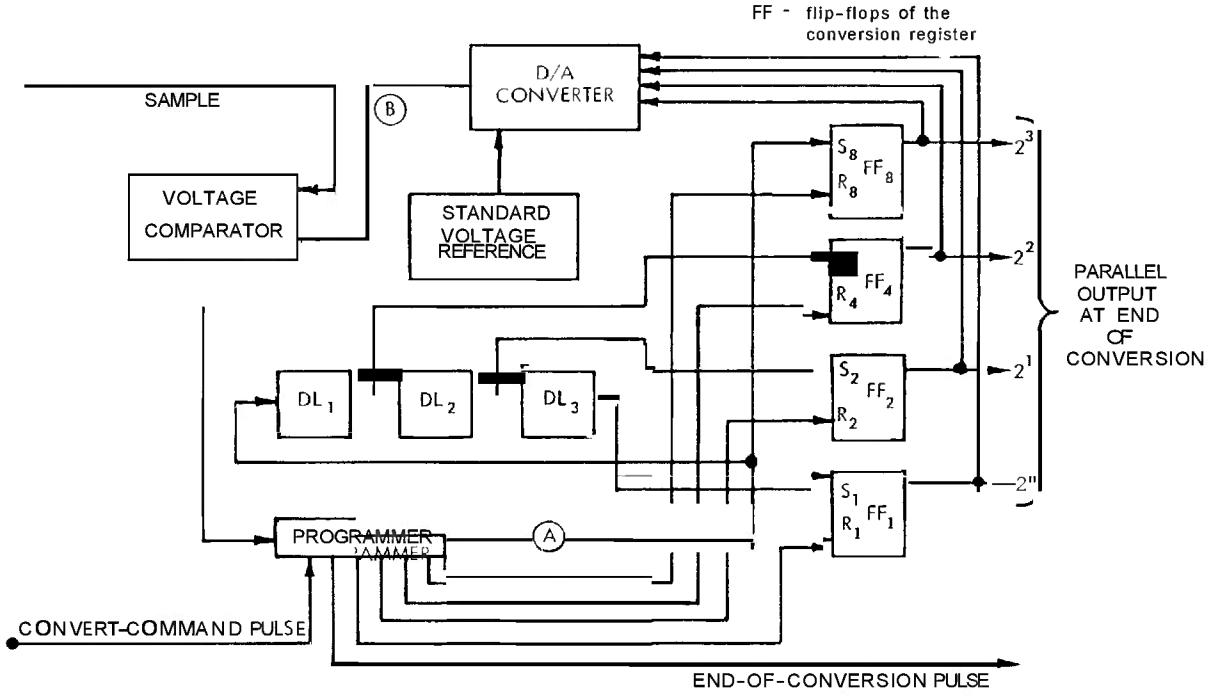
6.6 millivolts

where RC was chosen to be 0.01 sec, with C=0.1 μf and R=100K ohms, which are typical values for integrators of the type under consideration. For the maximum value of e_o (10 volts; see Table 7-1), the percentage error ($\Delta e_o / e_o$) × 100 is therefore equal to 0.066 percent maximum.

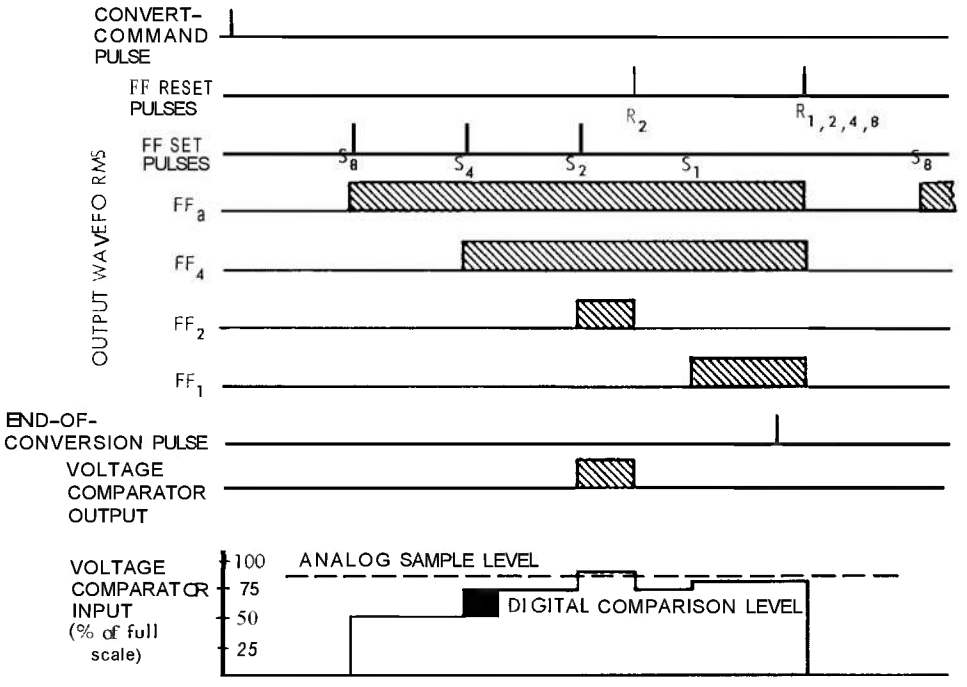
7-2.1.2 Digit-at-a-time Voltage-to-digital Encoders

Fig. 7-3(A) is the basic block diagram of a typical digit-at-a-time type of voltage-to-digital encoder. The significant wave forms are shown in Fig. 7-3(B). This type of converter is much faster than the time-base type of digitizer described in par. 7-2.1.1 since the basic conversion sub-cycle time is about equal to the clock-pulse interval of the previously described encoder, and the total number of sub-cycles is equal to the number of bits in the conversion register. Thus, for a 10-bit encoder, only 10 successive approximation sub-cycles must be performed to effect a complete analog-to-digital conversion. The conversion rate is about 2 microseconds per digit typically; therefore, a complete conversion takes about 25 microseconds, allowing time for logical control of the read-out, clearing of the register in preparation for another conversion, etc. Thus, the conversion rate for this type of converter is approximately 40,000 conversions per second, as compared with 100 conversions per second for the time base type of encoder.

As shown by Fig. 7-3(A), the convert-command pulse comes in from an esternal source, although it may be generated within the A/D converter if a constant conversion rate is desired. With the external command shown, the converter may be made to operate on a "demand" basis, thus eliminating redundant data for slowly varying functions, while sampling more rapidly upon command when the analog input varies at higher frequencies. The programmer, upon command to convert, sends a set pulse to the most significant flip-flop of the conversion register. The weighted analog value of this most



(A) Simplified block diagram



(B) Wave forms

Figure 7-3. Simplified block diagram arid associated wave forms for a digit-at-a-time type of voltage-to-digital encoder.

significant flip-flop, passed along to the voltage comparator at ◯ is compared with the analog input sample. A signal is then sent back to the programmer that indicates whether the half-scale analog output from the D/A converter;: is greater or less than the analog input sample, The absolute value of the voltage output of the D/A converter is calibrated accurately against a standard voltage reference, as indicated in Fig. 7-3(A). If the input sample is greater than half scale, the most significant flip-flop is left in the set position; if not, the flip-flop is reset.' Approximately 2 microseconds after the most significant flip-flop has been set, the same set pulse has traveled through the first section of a delay line DL_1 and sets the second most significant flip-flop FF_4. If FF_8 was left set, the output of the D/A converter will be 3/4 of full scale; if FF_8 was reset at the end of the first digit sub-cycle, the output of the D/A converter will be 1/4 of full scale, In the timing diagram of Fig. 7-3(B), the most significant flip-flop was not reset, so the output of the D/A converter immediately after the S_4 pulse is shown at 75% of full scale.

Further delayed pulses S_2 and S_1 act upon their respective flip-flops in like manner until the conversion is completed. When a suitable time interval has elapsed after the last sub-cycle, the programmer issues an end-of-conversion pulse to the external recording or computing equipment so that the contents of the conversion register may be logged, after which all flip-flops are cleared in readiness for the next conversion. In Fig. 7-3(B), the voltage converted is shown as a dashed line at about 84% of full scale. After the S_2 pulse, the output of the D/A converter was 87.5%--or higher than the input sample--so FF_2 was reset. After the S_1 pulse, the output of the D/A converter was 81.25%--or the nearest value to 84% that was attainable with a 4-bit system--since the resolution is only 6.25%. Thus, the number read-out of the conversion register at the end of conversion would be 81.25% of full scale. If a pulse is sent out to external equipment each time a reset pulse to one of the flip-flops is inhibited, a serial pulse train representing the digital (conversion is available for the operation of serial-type recording or computing equipment. Conversely, the reset pulses themselves may be used to represent the complement of a serial-output pulse train,

Note: At the present time, fire control systems use 11-bit A/D and D/A converters. These still produce considerable error; however, increasing the bit capacity to reduce the error will increase the equipment size unnecessarily.

7-2.2 THE LOGIC USED TO OPTIMIZE THE SPEED OF CONVERSION

A level-at-a-time voltage-to-digital encoder--the basic time-base A/D encoder described in par. 7-2.1.1--starts counting at zero time, representing zero voltage, and registers a steady stream of counts until stopped. Hence, an n-bit conversion takes 2^n pulse times, and the speed of conversion can be increased only by increasing the clock pulse rate and providing the necessary higher-speed circuits for encoding.

A digit-at-a-time voltage-to-digital encoder--a variation of the time-base encoder that uses feedback and a D/A converter and is described in par. 7-2.1.2--presets the counter to 2^{n-1} and, at the start of conversion, is directed by an error-sign circuit to count in the proper direction to meet and match the analog voltage. A substantial increase in speed is offered by the digit-at-a-time encoder because this method reduces the conversion time to n+1 pulse times.

For very-high-speed conversion, an amplified analog input sample can be used to deflect a cathode-ray-tube (CRT) electron beam across a coding mask. Inside the CRT, a system of electron collector wires is arranged behind the mask to detect the presence of electrons passing through the holes in the code mask. Using the electron-beam-deflection technique, television video signals have been encoded to 7-bit accuracy at

* *See* par. 7-4 for a description of D/A converters.

rates up to 10 megacycles per second. Since the deflection accuracy of the CRT system is not comparable with the comparison accuracy of the voltage-comparison (digit-at-a-time) encoder, however, the increase in speed of the character-at-a-time method employed by the CRT systeni is achieved at a sacrifice in accuracy.

7-2.3 THE USE OF SERVOS WITH SHAFT ENCODERS

Perhaps the slowest encoding method, but certainly a straight-forward method of voltage-to-digital conversion, is simply to attach a suitable coding disc to the shaft of an existing servo element in the systeni. If the final angular position of the shaft is proportional to the desired analog voltage "input", the digital equivalent can be read from the coding disc as discussed in par. 7-3. The response time of the servo is the major contributing factor to the time of conversion, and the maximum time could be an appreciable fraction of a second.

7-2.4 STEPPING SWITCHES, RELAYS, AND TRANSISTOR SWITCHES FOR A/D CONVERSION [2]

Rotary devices for direct conversion of angular shaft position to binary code are useful for manual or relatively slow introduction of data into a digital system. Electrically or mechanically driven stepping switches can be adapted for the slower automatic functions, producing direct digital signals. Since these will normally be in decimal form, either relays or solid-state OR gate logic can be used to encode them in natural binary form as shown in the logic diagram of Fig. 7-4(A) or the equivalent matrix of Fig. 7-4(B). These circuits require a voltage on one input line only, representing the decimal number to be encoded in natural binary coded decimal form.

7-3 CONVERSION OF MECHANICAL MOTION TO A DIGITAL OUTPUT

7-3.1 COMMUTATOR-TYPE ENCODING DISCS AND DRUMS

A commutator-type encoding disc (or drum) using brushes that press against coded rings with appropriate conducting and nonconducting sectors is one means that can be employed to represent a digital code. This arrangement does, however, possess limitations due to (a) surface wear and contamination, (b) physical restrictions of brush size or alignment, and (c) positions of uncertainty during transitions between conducting and nonconducting regions, Fig. 7-5 illustrates the technique of dividing one shaft revolution into 16 parts and reading out a 4-bit binary code through four brushes—B_0, B_1, B_2, and B_3.

The uncertainty of coding during transitions can be eliminated as discussed in par. 7-3.4. Coding discs, approximately 4 inches in diameter, are commercially available with as many as 1024 sectors (10 binary digits).

The equivalent drum technique is illustrated in Fig. 7-6. The concept of using brushes to contact conducting and nonconducting sectors that is employed here is the same as that used with a coding disc. Drum encoders are less common than disc encoders, however, because discs are simpler to construct and have fewer sources of error.

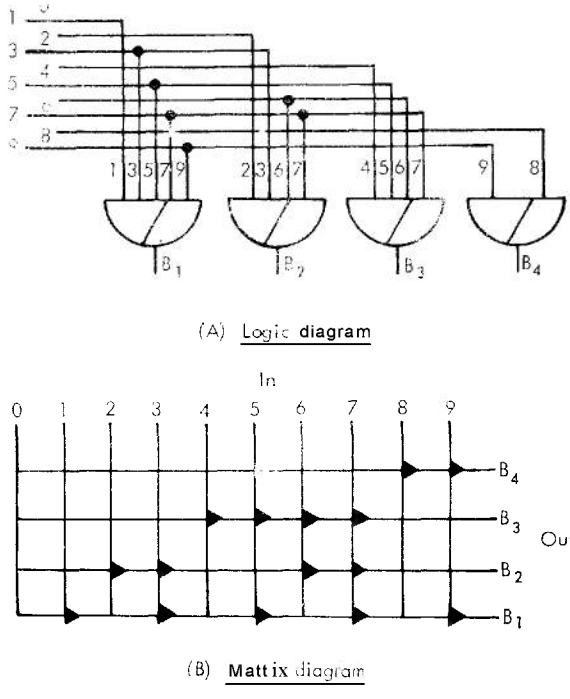


Figure 7-4. The logic diagram and matrix diagram for a natural binary coded decimal encoder.

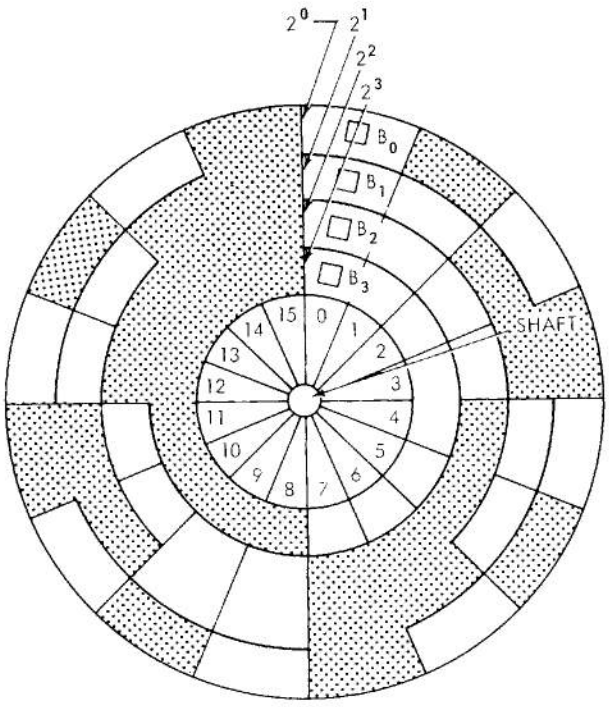


Figure 7-5. A typical binary coding disc.

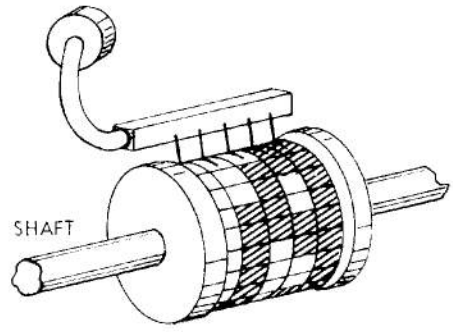
7-3.2 MAGNETIC ENCODERS

An example of the rotating-drum technique with magnetic encoding is discussed in Refs, 14 and 15. As shown in Fig. 7-7, the equipment consists of the following components:

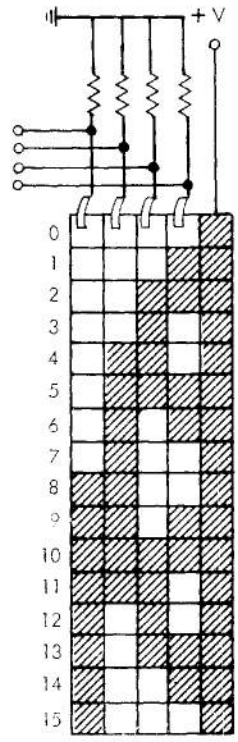
1. An index disc and a high-speed magnetic disc on one shaft rotating at a constant speed. The index disc generates a train of index pulses (nominally 500 per revolution).
2. An index-track reading head
3. A reference shaft for the analog input that is coupled to the input shaft of the device so as to position a pulse writing head at an appropriate angular displacement.
4. A reading head for the magnetic disc
5. An erase electric magnet
6. A converter consisting of amplifiers, pulse shapers, a frequency multiplier, gates, and a delay circuit to operate on the pulses.

The sequence of operation, illustrated in Fig. 7-8, is as follows:

1. At the instant of read-signal initiation, a single magnetic pulse mark is written on the high-speed disc.



(A) Drum with brushes



(B) Corresponding cyclic-code pattern wrapped around a drum

Figure 7-6. Direct-drive angular-shaft-position analog-to-digital converter.

2. Coincidentally, a train of index pulses is started at the output to an electronic counter.
3. The reading head detects the arrival of the magnetic mark on the high-speed disc, and stops the index pulse-train output. The count of the pulse train is the measure of the angular displacement of the analog-controlled writing head from the reference position of the reading head,

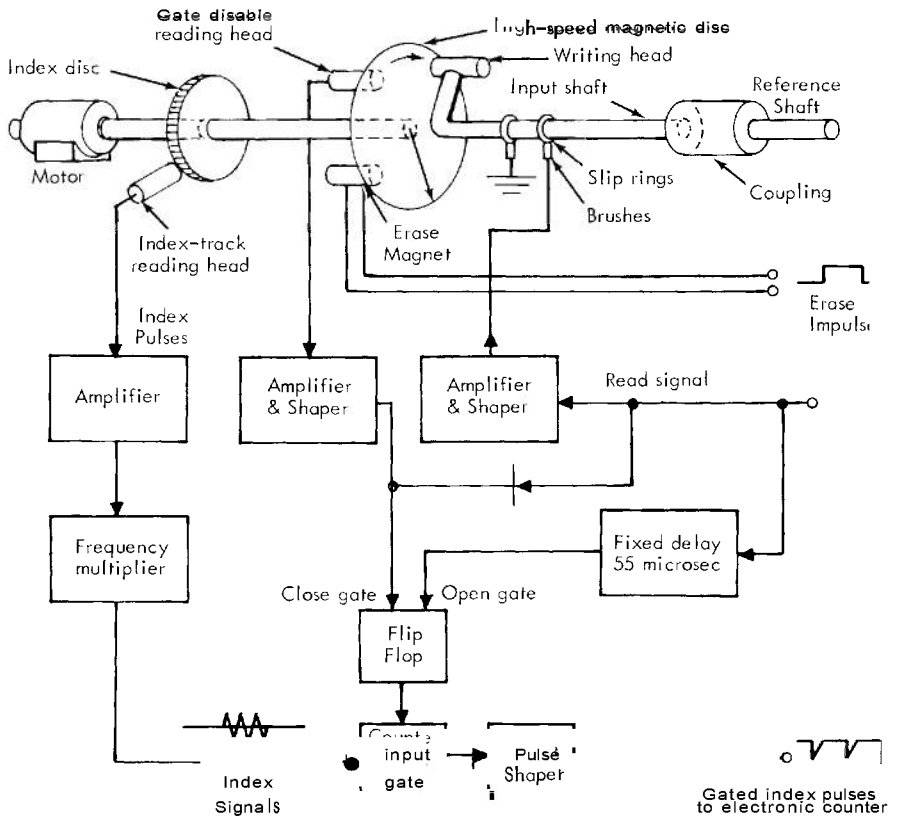
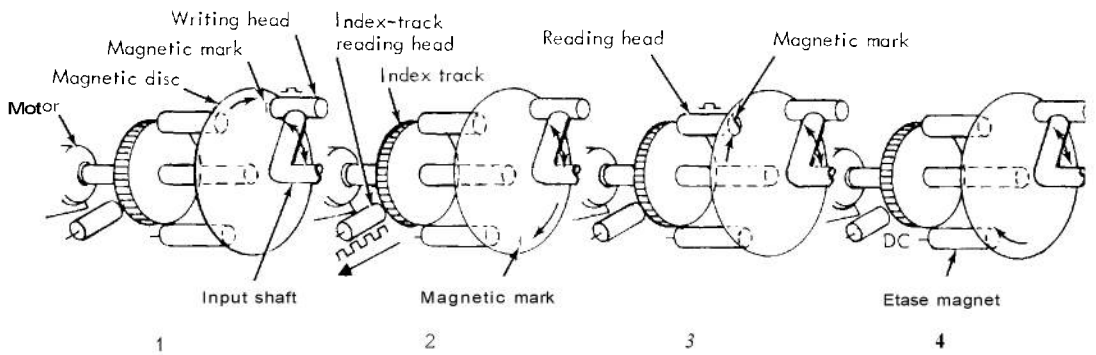


Figure 7-7. Block diagram of the Engineering Research Associates shaft monitor.



1. Magnetic mark is "written" on revolving magnetic disc at instant of read-signal initiation.
2. Index-track reading head generates train of index pulses. Converter triples frequency and shapes pulses which appear as output of Shaft Monitor.
3. Reading head detects magnetic mark and stops index pulse train, the length of which represents time interval between time magnet mark was written and lead.
4. Erase magnet is energized for one complete revolution of magnetic disc, prepating for next reading.

Figure 7-8. Operation sequence of the Engineering Research Associates shaft monitor.

4. The erase magnet is energized for one complete revolution of the magnetic disc in order to prepare this disc for the next reading,

The nominal performance characteristics of the system provide 20 readings per second to an accuracy of plus or minus 0.09 degree for an input speed of 120 rpm.

7-3.3 PIIOTOELECTRIC ENCODERS

For higher resolutions, the use of opaque and transparent areas to represent the code pattern on a disc and the use of a thin radial line of illumination to shine through the disc onto photocells have proved to be very effective. Photoelectric discs have been made with an accuracy of one part in 131,072 (17 binary digits). The art of fiber optics--where the diameter of the light-conducting fibers is measured in millionths of an inch--has progressed to where great optical flexibility, compression in size, permanency of alignment, and fine discrimination between sectors and adjacent channels orbits can be achieved. The methods of minimizing ambiguity at the boundary between two sectors are the same basically as for commutator discs, and are discussed in par. 7-3.4.

A simple, compact shaft-angle indicator has been contrived €or digital pickoff of velocity information (Ref. 13). Rased on the principle of interference patterns produced by two sectored discs, with one disc having one more opaque and transparent sector than the other, this device has achieved an accuracy of better than one minute of arc. In the coarse pattern shown in Fig. 7-9, the transmitted light (reflected light can be used) varies from zero to full to zero for the passage of each sector. Kith 512 sectors and four photo pickoffs in quadrature, digital logic can distinguish 1/2048 part of a revolution. It should be noted that this pickoff produces incremental rather than arithmetic data.

7-3.4 CODES AND BRUSH (READING HEAD) ARRANGEMENTS EMPLOYED

One technique devised to avoid errors due to imperfections and uncertainty in transitions from sector to sector in coded discs is to use a cyclic code in place of the binary code. Table 7-2 lists corresponding decimal, binary, and cyclic code numbers; and illustrates how in cyclic code the successive numbers differ from each other in only one digit column, It is important to note that in reading cyclic code (also called reflected code or Gray code) the sign of successive ONES alternates, starting with the most significant ONE as positive. It can be readily verified from Fig. 7-10 that in cyclic code small misalignments do not result in an error larger than one bit.

A second technique for avoiding reading errors is to use two brushes or heads for reading each binary digit. (Ref. 8, 9). The brushes are *so* positioned that if one brush is in transition, the mating brush is completely within either a conducting area or a nonconducting area. Suitable circuits must be provided to select the brush that is not over a transition. The amount of equipment required to make this selection is about the same as that required to convert cyclic code back into useful binary form (Ref. 10).

7-4 CONVERSION OF A DIGITAL SIGNAL TO AN ANALOG VOLTAGE

If a digital number is available in serial form, a remarkably simple scheme developed by Shannon and Rack can be made to convert accurately to seven bits (1 part in 128), although the switching and timing equipment are somewhat complex. In this scheme, the R-C circuit of tlie serial-to-voltage inverter shown in Fig. 7-11 is adjusted so that the stored charge decays to exactly half its value in one pulse time. The switch closes for one pulse time for a 1 digit and is open for *a* 0 digit. The numbers are read in serially, least significant digit first. If the first digit is a 1, the capacitor builds up one unit of charge which then starts to leak off, becoming one-half at the end of the second pulse, one-fourth at the end of the third, and *so* forth. Thus, regardless of successive openings and closings of the "switch", the first digit has made a contribution of 2-" times the basic voltage at the end of the last pulse if it were a one, and nothing if it were a zero. In the same fashion, the second digit makes a contribution of either 0 or $2^{-(n-1)}$, and so forth. The

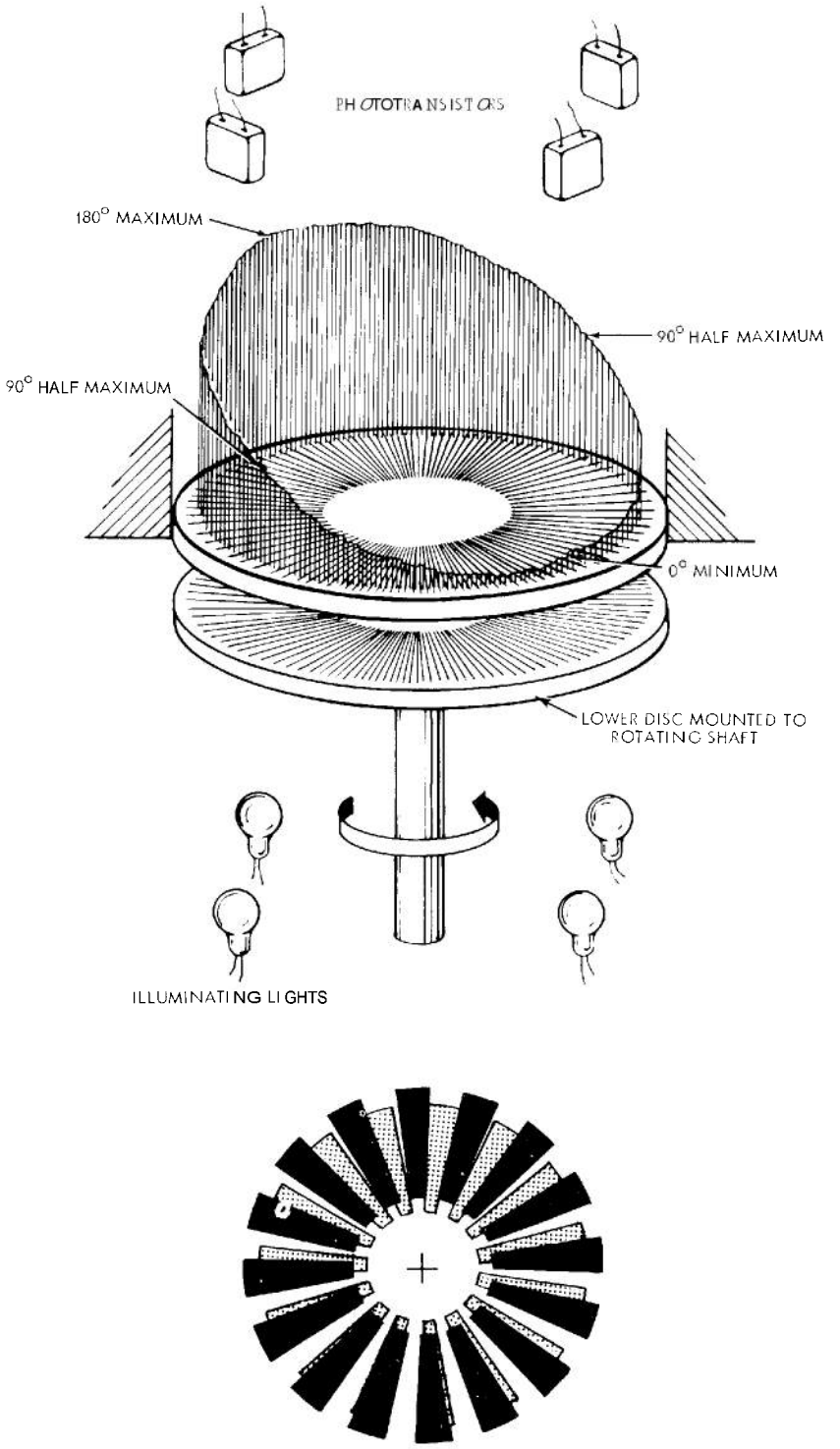


Figure 7-9. Arrangement of sectored discs and photodetectors for readout of shaft motion.

TABLE 7-2. A CYCLIC CODE AND ITS DECIMAL AND BINARY EQUIVALENTS.

Decimal Number	Binary Number	Cyclic Code	Decimal Number	Binary Number	Cyclic Code
	16 8 4 2 1	3 1 5 7 3 1		16 8 4 2 1	3 1 5 7 3 1
0	00000	00000	16	10000	11000
1	00001	00001	17	10001	11001
2	00010	00011	18	10010	11011
3	00011	00010	19	10011	11010
4	00100	00110	20	10100	11110
5	00101	00111	21	10101	11111
6	00110	00101	22	10110	11101
7	00111	00100	23	10111	11100
8	01000	01100	24	11000	10100
9	01001	01101	25	11001	10101
10	01010	01111	26	11010	10111
11	01011	01110	27	11011	10110
12	01100	01010	28	11100	10010
13	01101	01011	29	11101	10011
14	01110	01001	30	11110	10001
15	01111	01000	31	11111	10000

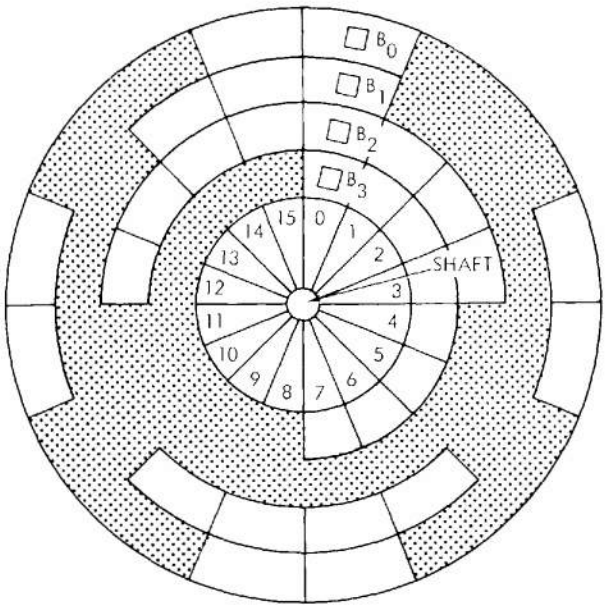


Figure 7-10. A typical cyclic coding disc.

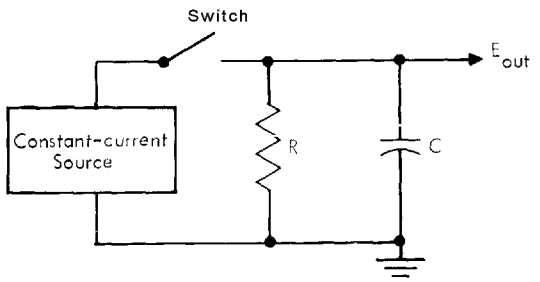


Figure 7-11. Schematic diagram of a serial-to-voltage converter.

contributions of each digit are all additive, and at the end of the last pulse the desired voltage is available at the output. The voltage must be read immediately since it continues to decay.

Other electrical networks for converting a digital signal to a voltage are shown in Fig. 7-12. The accuracy of these schemes is determined primarily by the stability of the sources and the precision of the resistors,

In Fig. 7-12(A) each current source is associated with a binary digit, least significant first, starting on the left. When the binary digit is a one, the related current source puts out a standard, regulated current I; for a zero bit, it has zero output. If the current sources are assumed to have infinite impedance, the output voltage is

$$e_o = \frac{IR}{2^{n+1}} p \qquad (7\text{-}7)$$

where

p = number to be converted

n + 1 = number of stages

In Fig. 7-12(B), standard voltage sources each put out a voltage E for a bit value of one, and no voltage for zero. In this case, if the voltage sources are assumed to have zero impedance, the output voltage is

$$e_o = \frac{E}{2^{n+1}} p \qquad (7\text{-}8)$$

Typical accuracy figures for the circuits shown in Figs. 7-12(A) and 7-12(B) are one part in 1000.

Figs. 7-12(C) and 7-12(D) show circuits that are particularly useful when the digital information is available through relays, In both cases, the relay positions shown in these illustrations are those for a zero bit; the relay pulls in when its associated bit is a one. The output voltage for the circuit in Fig. 7-12(C) is

$$e_o = \frac{E_1}{\frac{R}{R_c} t2 \quad -1} p \qquad (7\text{-}9)$$

and the expression for the output voltage of Fig. 7-12(D) is

$$e_o = \frac{E}{2^{n+1} - 1} p \qquad (7\text{-}10)$$

The circuits of Figs. 7-12(C) and 7-12(D) have the advantage over those of Figs. 7-12(A) antl 7-12(B) of requiring only one precision resistor per stage, and of avoiding sensitivity to source impedance and output impedance. In the circuits of Figs. 7-12(C) antl 7-12(D), accuracies of one part in 4000 ran be attained. Through the use of all-electronic switching and sampling of two or more digital inputs, a single D/A converter

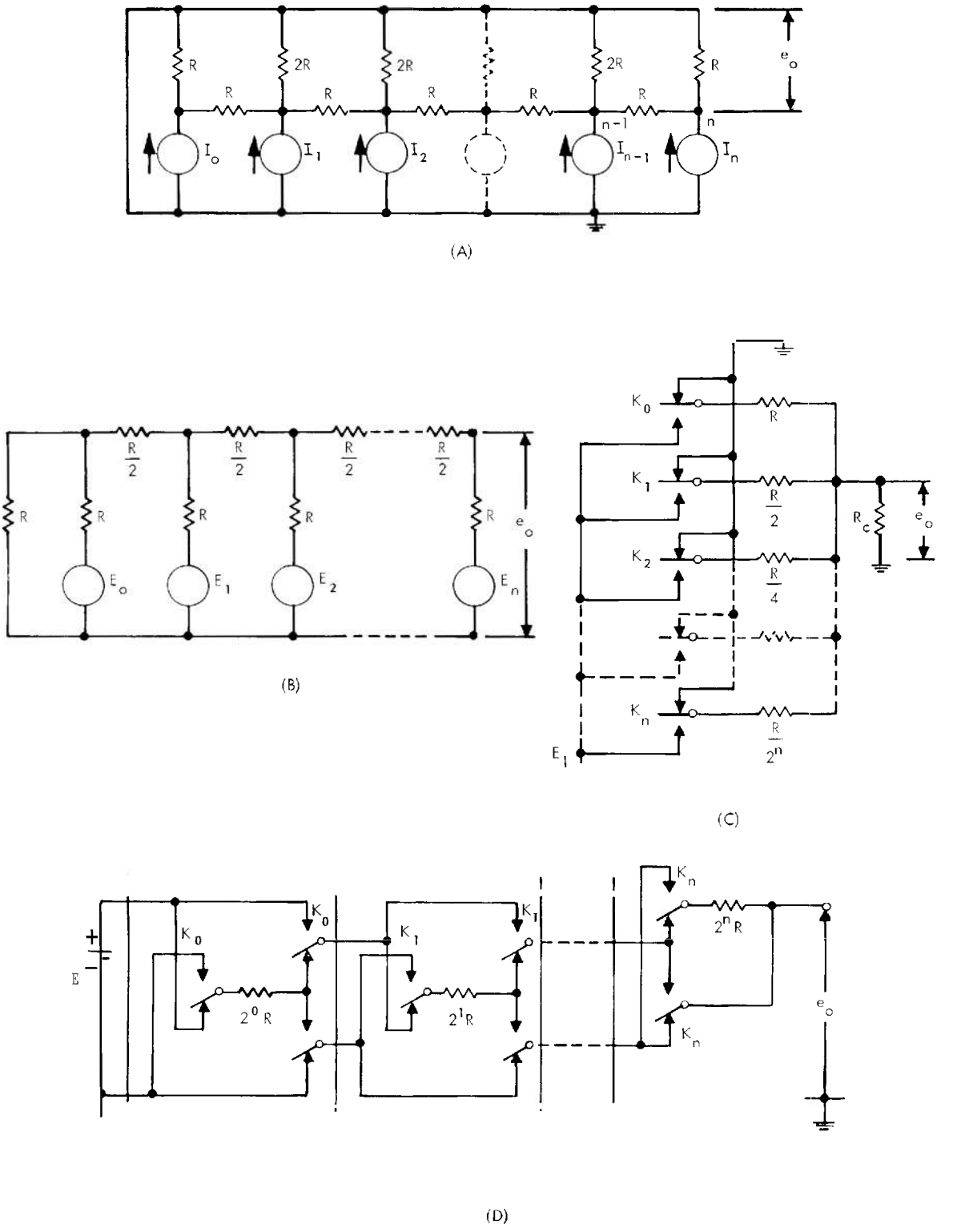


Figure 7-12. Schematic diagrams of typical digital-to-voltage converters.

can serve to convert more than one channel of data--typically handling as many as 100,000 bits per second.

7-5 CONVERSION OF A DIGITAL SIGNAL TO MECHANICAL MOTION

The requirement of conversion of parallel digital information to shaft angle can be satisfied with the servomechanism shown in Fig. 7-13. The position of the output shaft is digitized by means of a coding disc, The static conversion accuracy can be made to approach that of the coding disc--normally a maximum of one part in 131,072 (17 binary digits). In operation, the coded shaft position is subtracted from the desired shaft position and the difference number is converted into a voltage used to drive the output motor,

Incremental digital information can be converted to shaft position by means of shaft-angle feedback from an incremental magnetic encoder:: and a reversible counter as indicated by Fig. 7-14. In this application, the encoder provides an integral number of pulses per revolution of the output shaft. The anti-coincidence circuit resolves the coincident pulse problem so that the reversible counter does not receive both up and down counts simultaneously. The reversible counter and the digital-to-analog converter provide the error-detector function. If the pulse source introduces up counts and the encoder introduces down counts, the reversible counter holds the servo error at any instant. Inasmuch as the digital-to-analog converter transforms only the error, it can be a very simple device. In some cases, only the three states--up one or more counts, zero, and down one or more counts--are employed.

The servo system of Fig. 7-14 is useful in pulse-to-position or frequency-to-velocity conversions. A typical application is in the precise frequency regulation of a-c alternators where accuracy is limited only to that of the pulse source.

Stepping motors, through appropriate logic networks for either serial or parallel digital operation, can be used to convert to mechanical motion--at rates of up to 2400 steps per second. Since there is no feedback in such systems, the accuracy may be limited by backlash and other errors in gearing and mechanical components.

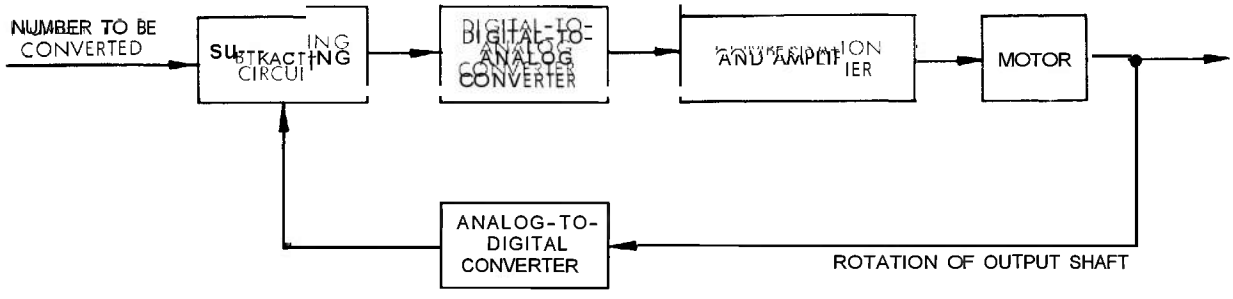


Figure 7-13. A servomechanism for digital-to-analog conversion.

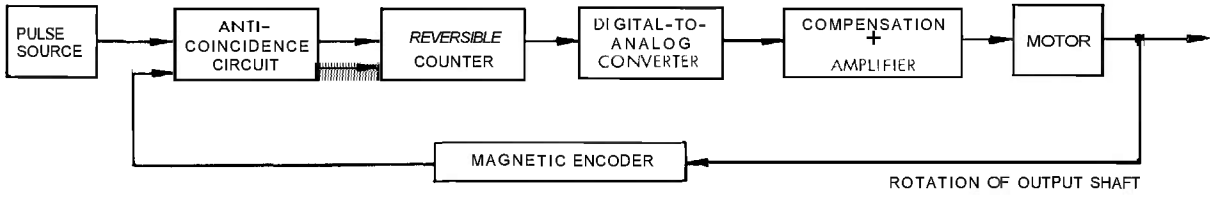


Figure 7-14. A servomechanism for incremental digital-to-analog conversion.

* Small stepping motors are often used as incremental magnetic encoders.

REFERENCES

1. A. K. Susskind. Notes on Analog-Digital Conversion Techniques, John Wiley, New York, 1957.
2. H. H. Goode, and K. E. Machol, System Engineering, McGraw-Hill, New York, 1957.
3. H. E. Burke, Jr., "A Survey of Analog-to-Digital Converters", Proc. IKE 41, Part 10, 1455-1462 (1953).
4. M. L. Klein, F. K. Williams, and H. C. Morgan, "Analog-to-Digital Conversion", Instruments and Automation 29, 911-917 (1956).
5. M. L. Klein, F. K. Williams, and H. C. Morgan, "Practical Analog-Digital Converters", Instruments and Automation 29, 1109-1117 (1956).
6. M. L. Klein, F. K. Williams, and H. C. Morgan, "High-speed Digital Conversion", Instruments and Automation 29, 1297-1302, (1956).
7. Numerous articles on digital voltmeters in the Harry Diamond Laboratory Journals, particularly Vol. 16, No. 12, August 1965, "A Fast-Reaching Digital Voltmeter with 0.005% Accuracy in Integrating Capability."
8. L. P. Retzinger, Jr., "An Input-Output System for a Digital Control Computer", Proc. of the Wescon Computer Sessions, 67-76, 1954.
9. J. B. Speller, "A Digital Converter", Proc, of the Wescon Computer Sessions, 29-31, 1954.
10. H. G. Follingstad, J. N. Shine, and R. E. Yaeger, "An Optical Position Encoder & Digit Register", Proc. IRE 40, Part II, 1573-1583, (1952).
11. M. Phister, Jr., Logical Design of Digital Computers, John Wiley, New York, 1958.
12. I. Flores, Computer Logic. The Functional Design of Digital Computers, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1960.
13. I. Flores, "Digital, Analog, Hybrid Shaft-Angle Indicator with Frictionless 'Optical Gearing'", Electromechanical Design, May 1958.
14. G. W. Lund, Shaft-Position Analog-to-Digital Converter, Presented at AIEE Summer General Meeting, 1953.
15. Stanley Fifer, Analogue Computation, McGraw-Hill, New York, 1961.

CHAPTER 8
ANALOG-DIGITAL COMPARISONS

8-1 BASIS OF COMPARISON

In the design of computing devices for fire control applications, many factors influence the choice between analog computation, digital computation, or a combination of both. When a combined system is selected, a choice must then be made as to the points in the computation at which the conversions from one type to the other will occur. Most of the factors on which analog-digital comparisons are based bear one upon the other., starting with what are often the most critical--speed and accuracy. These are followed by factors of complexity, of reliability, and of the effects of any special environmental conditions. Cost, site, weight, and power considerations complete the list of often interrelated factors on which the comparisons are based.

If one glances quickly through a computing system from input to output, a few of the salient decision points are immediately apparent. The input signals of a fire control system are basically analog in nature. If these signals are to be transmitted over any great distance, however, it will become either necessary or desirable to convert them to some form of pulse-code modulation. This means, in effect, converting them to digital form. Thus, the problem of conversion of signals between analog and digital form may be encountered in the use of either analog or digital computing equipment. Although analog instrumentation may efficiently provide fast dynamic response, coordinate transformation, and power gain; digital techniques may provide more accuracy in certain calculations, more convenient changeover between alternative modes of operation, and more economical storage of functions arid constants. An inherent disadvantage of a digital system is that a considerable amount of equipment is dictated for solution of even the simplest control problem. At the output of the system, digital techniques may be well adapted to displays, but actual control functions may require a conversion to analog circuitry and devices.

8-2 COMPARISONS BASED ON THE SPEED WITH WHICH SOLUTIONS ARE OBTAINED

One chief advantage of analog techniques is that operations such as integration are performed continuously and rapidly, and most of the simple operations of algebra and the calculus fall well within useful limits of effectiveness. Further, although time-sharing is theoretically possible uith analog computing elements, the convenient and most common approach is to assign a specific computing element for each individual operation to he performed, with the effect that overall solution time is essentially unaffected by the multiple computations. In the case of nonlinear operations, which are relatively difficult to perform by analog techniques, the errors introduced may be as much as an order of magnitude more than for linear operations, an effectthat could offset an advantage of speed. The dynamic performance of servos and many nonlinear devices in a larger-scale system may he marginal as the requirement for more nearly real-time operation inc ceases. 411-electronic analog computers achieve much greater speed than the mechanical or electromechanical analog computers, and are sometimes designed to permit repeating the

* By E St. George, Jr

solution of a problem 10 to 60 or more times per second. This contrasts uith typical servo performance wherein a unit might require a few tenths of a second to reach full scale at a maximum speed, and might require comparable time to accelerate from zero to full speed.

The introduction of a sampling device, such as a digital computer, in a closed-loop fire control system introduces a time lag related to the sampling interval. Input devices, such as certain types of radars, niay introduce such lags in an otherwise analog system. If the output of a system containing a sampling element is to be continuous and smooth, it must be filtered. In general, the speed of response of such a system will be at least two times the sampling interval. Fire control systems that must operate with very noisy input data may also be limited in response by the requirement for filtering.

If it is determined that a computation in a fire control system must be completed no more frequently than once every second and a digital-computer configuration under consideration could complete the computation in one-tenth of a second, then the digital computer, with perhaps little increase in hardware, could handle 10 different inputs on a time-shared basis. It is important to note that an almost inevitable but sometimes subtle demand is imposed on the basic speed of digital computation by the way in which requirements of speed and accuracy multiply together. For example, an apparently simple problem might involve using a tachometer pulse generator to convert shaft rotation to a digital input. If it is required that the output be read 50 times per second to an accuracy of 1 part in 1000, then the resulting requirement for a pulse-rate-handling capability is 50 X 1000 or 50 kc for this particular channel of input data. Increasing the computing rate of a digital device tends strongly to increase its size and cost. (The term "size" should be interpreted here as related to the number of components since techniques of miniaturization -- in themselves somewhat costly -- tend steadily to reduce sheer bulk.) The size of an analog computer is much less sensitive to increase with increasing speed requirements.

An approximate, but relatively correct, comparison of several digital machines--all solving the same problem--reveals how the basic speed of access to the working memory, the use of serial or parallel operation, and the clock rate combine to determine the speed of operation. The problem used is as follows:

Solve the following set of equations for J and Z.

$$J = \frac{(ZF - G)K}{H + L}$$

$$Z = \sqrt{LY}$$

where

$Y = AX + B$ for $D < Y < C$

X is the input variable

A, B, C, D, E, F, G, H, K, and L are constants

The method of solution chosen is as follows:

1. Read in new input X (exclusive of input switching time, input selection time, settling time, or A/D conversion)
2. Calculate (A) (X) + B = Y
3. Compare Y with C to insure that $Y < C$
4. Compare Y with D to insure that $Y > D$
5. Calculate $Z = \sqrt{(E)(Y)}$ (with an accuracy of 10 bits – 0.1%)
6. Calculate $J = (ZF - G)(K)/(H + L)$
7. Store J in bulk memory (use average access time)
8. Store Z in bulk memory (use average access time)

A comparison of seventypes of digital machines that could be used to solve this problem appears in Table 8-1.

Obviously, a computer with all-core storage, parallel logic, and a high clock-rate would be the fastest. Very rarely, however, is the designer permitted so simple a choice. Core memories are expensive, and with their drivers are bulky; therefore, they are usually restricted to the working memory, where their high-speed characteristics are of greatest value. Similarly, the use of parallel logic greatly increases the size and cost of equipment as compared with the use of serial logic, but does eliminate the necessity for parallel-series conversions at inputs and outputs. Finally, the cost of computers generally increases with clock frequency--

TABLE 8-1. A COMPARISON OF SEVEN TYPES OF DIGITAL MACHINES.

Operating Mode	Memory Type		Clock Frequency	Nominal Problem Time
	Bulk	Working		
Serial	Drum	Drum	100 kc	130 msec
Serial	Drum	Fast-Acess Registers	160 kc	80 msec
Serial	Disc	Core	170 kc	30 msec
Parallel	Drum	Core	50 kc	9 msec
Serial	Drum	Core	1 mc	7 msec
Parallel	Drum	Core	1.5 mc	2 msec
Parallel	Core	Core	1 mc	50 μsec

partly because of the greater cost of high-speed transistors and diodes, and partly because of the greater problems involved in the shielding and transmission of signals.

8-3 COMPARISONS BASED ON THE ACCURACY OF THE SOLUTIONS OBTAINED

The characteristics of materials and circuit elements establish the best basic accuracy obtainable in simple analog computations as roughly one part in 10^4. Noise usually limits the minimum discernible variation in a signal to aresolution of this same order. Errors arise from backlash in mechanical linkages, changing values in electrical components, and drift in amplifiers. These errors tend to add rather slowly, however, since most analog elements are used in feedbackloops and the effects of error increments are thereby minimized. The net result is that therelatively simpleinathematical operations of summation, multiplication by a constant, or integration can be accomplished with an accuracy of three significant figures. The more-difficult operations of multiplication (and division) of variables and generation of complex functions may display a marginal accuracy approaching two significant figures, or one percent of full scale.

Of the three types of analog computers (mechanical, electromechanical, and electronic), the accuracy achievable withthe best mechanical computing elements exceeds that obtainable with electronic elements. High accuracy can he achieved with properly designed electromechanical computing elements, but their speed of response is restricted and they require somewhat more specialized maintenance than electronic elements.

A digital device can be contrived to produce any desired degree of precision (e.g., the mathematical constant π can be obtained to 2,000 places). A digital computing machine has one intrinsic error, namely-, cumulative round-off, that may add up to serious significance in certain long computations. The digital technique has the inherent error of truncation, resulting fromthe Factthat digital computations are carried on in a step-by-step manner. Althoughthe errors introduced by round-off and step-wise approximation are quite difficult to compute, some attempt should be made to appraisetheir effects

because these effects can become substantial [1-4]. Particularly in adigital computer, the requirements of speed and accuracy are intimately interrelated. The computing circuitry, the logical arrangement, the computing interval, and the programming must be selected in such a manner as to permit real-time operation for a specific fire control system.

8-4 COMPARISONS BASED ON THE COMPLEXITY OF THE COMPUTING DEVICES INVOLVED

A comparison of analog and digital techniques on the basis of complexity must consider the capabilities of the operator, problems of programming and communications, and the ability to change over quickly and easily from one problem to another. If the system requirements are relatively simple, and the basic inputs and outputs are analog, an analog system will often be more satisfactory and less expensive than a digital one. Frequently, an operator will be employed in tracking or positioning operations, which are inherently analog in nature.

If input (or output) data are being transmitted in digital code, which permits freedom from errors over much greater distances than analog methods, the digital choice is obvious. The simplicity of the apparatus and circuitry associated with the transmission of a signal by a synchro loop, for example, is always attractive when the character of the signal is well-adapted to analog methods. As the system becomes more complex, however, there is a tendency to design for digital and automatic operation.

Analog machines have an inherent versatility in their ability to perform directly such varied functions as integration, sine generation, and multiplication. The digital machines must build up such functions out of simple numerical processes of addition, subtraction, and multiplication by the radix, Rut in contrast, the digital machine is more flexible than the analog since it is only necessary to insert a new program to start a new problem. Even when new interconnections for an analog machine are substituted by plug-in patch panels, and various calibration and initial-condition settings are provided by nearly automatic, remote controls; starting a new problem is time-consuming.

Establishing communication with the machine in the first place can give rise to considerable complexity of input-output equipment for digital machinery. Instructions for relatively simple operations consist of a large mass of details. Programming consists essentially of an exercise in numerical computation, with a limited choice of methods and language determined by specific design features of the computer. Rut with appropriate equipment and programming, for example, it can be arranged for the human operator to type out English-language instructions on a key-board and be limited only by easily comprehended rules of computer grammar and vocabulary.

One aspect of complexity that must often be considered in connection with a computing machine is the ease with which the machine can be expanded. In general, a digital machine must be designed from the start to encompass the maximum foreseeable demand. Although the addition of more input-output equipment is always a possibility, the machine must be designed with a capacity of controlling, reading from, and writing into the ultimate total number of peripheral devices that will be required. Analog equipment, on the other hand, is susceptible to the addition of components--one at a time if necessary--as the complexity of the problem increases.

8-5 COMPARISONS BASED ON THE RELIABILITY OBTAINABLE

In one sense of reliability--whether or not the device will operate when wanted, as distinguished from being broken down--both analog and digital devices can be designed to show a good record. Purely mechanical analog computing devices can be made, not unlike the one-horse shay, to work without repair until they fall apart--an exaggeration that is useful to illustrate one school of thought as to the way in which maintenance should be carried out. Operation of equipment until its performance is no longer acceptable is termed breakdown maintenance. In the area of passive electrical components --resistors, inductors, and capacitors--

breakdown maintenance is all that can be expected since tests on these components give little indication of future performance.

In the case of electromechanical components, either analog or digital, more specialized maintenance procedures are almost always required than with purely electronic devices. Wiping contacts, make-and-break contacts, motor-driven devices, pen or stylus recorders, and tape reading or punching units all present inherent problems of either wear, adjustment, cleanliness, corrosion, lubrication, or a combination of these that may definitely call for scheduled preventive maintenance--particularly if the operating environment is unfavorable.

The second school of thought on maintenance recommends that each piece of equipment be tested periodically and repaired if its performance has deteriorated appreciably. This procedure is called periodic maintenance. It is difficult to conceive of a piece of military equipment for which periodic maintenance is not specified, as an assurance that a high state of readiness is being maintained.

The designer of advanced computing equipment can and should so construct either analog or digital computing devices that repairable faults can be quickly remedied by removal and replacement of modular elements. In modern solid-state digital circuitry, the use of relatively few types of plug-in logic circuit cards can account for the vast majority of functions within the organization of the machine, thereby minimizing the spare-parts problem and greatly facilitating rapid substitution. The ability of test programs and routines to predict trouble, through marginal checking of components, plus the use of error-detecting logic that warns automatically while an actual problem is being run, gives an edge to digital computers in the dynamic detection of unreliable performance. This is particularly useful because a digital machine is equally prone to a large mistake (most significant digit) as to a small mistake (least significant digit). Beyond a highly unlikely catastrophic failure, an analog device is more likely to make a small error. The yes-no circuitry of digital devices makes them less susceptible to line-voltage variations and to leakage resistance that can bleed off voltages in analog equipment.

Just as the designer's last weapon in the struggle to maintain reliability is replacement, his first weapon is quality control. Bad solder joints or faulty- contacts anywhere in the system will plague analog or digital equipment equally. However rugged the electrical or electronic components themselves may be, necessary intercabling, plugs and connectors, or patch panels introduce elements of uncertainty.

Redundancy in cabling, to provide two or more alternate circuits so that at least one will be conducting if contact on another is interrupted, serves equally well for analog or digital devices. Redundancy in internal connections is particularly applicable in analog circuitry at points of wear such as in relay contacts, where the use of multiple contacts will not appreciably increase tie size of the equipment.

Providing redundant subsystems, as opposed to merely duplicating questionable interconnections, usually means increasing the amount of equipment by more than a factor of two for either analog or digital techniques. If the first unit fails, means must he provided to detect the malfunction, plus, means for switching in the second unit. Extending this concept one step further, one could duplicate the detecting and switching equipment because it, too, could fail. In extremely vital situations, triplicate or even quadruplicate systems have been provided. Digital computers are sometimes designed with redundant paths that permit a signal to be passed even though one of the paths is defective. Carrying this concept still further, the SAGE air-defense system uses two completely duplicate digital computers.

Error-detecting and correcting methods within a single nonduplicated system are particularly applicable to digital systems. Various methods can be applied to analog systems, but these are usually assoziated more with preventive maintenance. Modern digital computing equipment is usually designed with built-in circuitry that will automatically detect the majority of errors that occur during computing-system operation.

8-6 COMPARISONS BASED ON THE NATURE OF ENVIRONMENTAL EFFECTS

Before looking at some of the design problems associated with the operational environments that must be anticipated, it is worthwhile to examine a few of the preoperational environments. The problem of handling can compromise the integrity of many otherwise perfectly acceptable components. For example, a small, precision electromechanical device that is allowed to tip over on its side on a hard-surface work bench might sustain a shock of 500 g's, whereas after final assembly in a package with shock mounts it might never be expected to sustain more than 20 g's. Shipment of precision optical and electromechanical equipment in large assemblies by rail has had to be abandoned in many cases when it was not possible to insulate against shocks in transit exceeding 30 g's. Equally startling was the experience of 50% loss through damage of ocean freight consisting of automobile parts packaged for overseas shipment. In this case, it was found to be more economical to use air freight exclusively. These examples are cited to illustrate the staggering hazards that lurk beyond the normal scope of the designer.

Another area that has been found troublesome until suitable controls were established is one that can appropriately be called hidden testing. There have been cases where equipment has reached its final destination, presumably to be used for the first time, and has been found to have remarkably short life before requiring maintenance. The simple device of providing an accumulated running-time clock would reveal that this type of equipment presented an irresistible challenge to technical and operating personnel all along the line to turn it on and make sure it worked, or to demonstrate it to somebody, or to duplicate an acceptance test procedure.

Completely mechanical analog computing devices are, in many cases, best adapted to the most hostile environments. Repeated physical shock, thermal shock, radiation, or extremely high temperatures can be circumvented by mechanical analog techniques. For example, hydraulic analog computing devices perform remarkably in the extreme environment of aircraft jet engines, and attempts to perfect electromechanical substitutes have not shown promise.

The existence of make-and-break contacts, presure/sliding contacts, or patch connections in any computing equipment makes open season for dust, moisture, or vibration. Similarly, any form of vacuum tube, gas tube, lamp, or bulb invites failure from physical shock or, sometimes, thermal shock.

If all-electronic elements are being compared for both analog and digital devices, they are on very equal environmental ground, with one major exception -- field power. Naturally, power failure or sudden surges will induce malfunction or component failure regardless of computing technique. But a solid state digital system, with yes-no logic is not sensitive -- within limits -- to voltage changes.

A stored-program digital computer with a core memory may lose its program upon power failure, unless battery power is provided. Environmentally, the use of batteries can create a storage problem and a warm-up problem. Also, of course, the batteries are that much more component hardware.

In analog circuitry, wherever a voltage level represents an absolute value, such as across an integrating capacitor, any leakage due to moisture or other contamination will degrade the performance of the system.

It is recommended that the design process include appropriate heat-transfer calculations so that forced warm-up can be provided if necessary, and that all possible use of heat-sink properties be incorporated to allow for the extremes of artic- and desert-type environments. Design actions that can be taken to assure operable equipment under environmental extremes are discussed in Chapter 5 of Ref. 5.

8-7 COMPARISONS BASED ON COST, SIZE, WEIGHT, AND POWER CONSIDERATIONS

The cost of an analog computer is approximately proportional to its size. If it is necessary to expand the analog computer, components can be added one at a time, with obvious restrictions as to the capability of programming and suitably interconnecting any new elements. Naturally, any equipment

for field service can be flexible only within its packaging, arid considerable trading-off would have to transpire to equate future expandability against present physical size. In general, increasing the accuracy of an analog computer entails greater cost than providing greater digital accuracy if, indeed, it is even possible to increase the analog accuracy; however, the reverse may also be true. Any digital machine is likely to have a fairly large minimum cost, and increments of accuracy attainable are then relatively inexpensive.

If a digital computer is not originally built with expansion in mind, which would typically include adding blocks of memory and adding input-output equipment, it is not generally susceptible to expansion. The control capabilities must be provided in the original design, together with sufficient logic circuitry*

Since a wide variety of MIL-SPEC components for both analog and digital computers are available from existing manufacturers, a hidden cost of either type of design might lie in the time to develop and approve new components that had to run the full course from paper design to hardware. Balanced against the choice of the existing and proven is taking the calculated risk that, at the time it is needed for assembly, a clearly superior state-of-the-art development that has been promised may actually fail to materialize.

In any case where the relative merits of analog-versus-digital techniques do not emerge with reasonable clarity from a stud) of the situation, it is probably wise to carry forward preliminary designs on both types to obtain rough comparative costs.

As for the class of mathematical problem, more integration can be bought per dollar with analog computers and more arithmetic per dollar with digital computers.

As a general rule, it may be stated that -- for both analog and digital computers -- size, weight and cost are interrelated so that increases in size and weight lead directly to increased costs. Special miniaturization techniques undertaken to reduce size and weight may also increase unit costs by a large factor.

Comparing computer power sources from a cost viewpoint shous that the necessity for a super-regulated power supply for the analog computer results in a cost greatly exceeding that of the digital-computer power supply, whose sole requirement is that it be isolated from line transients,

REFERENCES

1. "Theoretical and Experimental Studies on the Accumulation of Error in the Numerical Solution of Initial-Value Problems for Systems of Ordinary Differential Equations", Proc. International Conf. Infor. Proc., UNESCO, Paris (June 1959), pp. 36-43.

2. D.J. Keil and R.E. Smith, "Maintaining Predetermined Accuracies in Digital Results", Datamation 8, No. 11, 73-76 (1962).

3. S.H. Crandall, Engineering Analysis, McGraw-Hill Book Company, Inc., New York, N.Y., 1956.

4. A.S. Ilouseholder, Principles of Numerital Analysis, McGraw-Hill Book Company, Inc., New York, N.Y., 1953.

5. AMCP 706-327, Engineering Design Handbook, Fire Control Series, Section 1, Fire Control Systems – General.

CHAPTER 9
RELIABILITY AND CHECK-OUT PROCEDURES*

9-1 INTRODUCTION

An important consideration in regard to the usefulness of fire control computers is their reliability, It goes without saying that equipment that is not reliable is less than useless. From a military point of view, a quantitative specification of reliability is desirable, and a convenient -- but by no means unique -- method of quantitatively defining reliability is to specify the average time between the failures of the equipment. Failure is defined as a condition in which the fire control system is rendered completely inoperable (catastrophic failure) or in which the fire control system is degraded in its performance to such a degree as to fail to meet acceptable limits.

Fire control equipment with a long mean-time-between-failures (MTBF) and consequently high reliability is the result of sound design, good quality control, and dependable maintenance. There is a general tendency to think of reliability in terms of quality control only. However, while quality control is one of the essential ingredients of reliability, design and maintenance are equally important. Design features that incorporate a quantitative approach to the selection of reliable components include elimination of unnecessary adjustments, the use of redundancy, the use of derating standards on components or subassemblies, and the full consideration of environmental factors.

These and other design features relating to reliability are discussed in general terms subsequently in this chapter. For specific design information concerning particular areas of reliability, however, the fire control system designer should consult the applicable extensive documentation that is available. For example, the U.S. Air Force Rome Air Development Center series of documents on reliability[1] constitutes an excellent source of detailed information concerning the reliability aspects of electronic design. Particularly valuable information concerning reliability and other design aspects of the specialized electronic field of integrated circuits (microelectronics) appears in a collection of documents published by the U.S. Army Electronics Command.: Included is information concerning the Department of Defense policy that has extended the throw-away concept (ref. par. 5-4.7.2 of Ref. 3) to integrated circuits.

The mean-time-between-failures (MTBF) has been mentioned as a measure of reliability. The relation of this quantity to probability considerations should be kept in mind when discussing reliability. The MTBF for a fire control system is determined primarily by the weakest link in the system's chain of components, i.e., the component most likely to fail in the shortest time in the particular environmental conditions to be encountered, Since fire control systems are made up of many components, statistical reasoning must be used to determine the aggregate effect of a multiplicity of components on the performance of the system. If the failure rate is defined as the reciprocal of the MTBF for a particular component, then in a system the aggregate mean-time-between-failures is equal to the reciprocal of the sum of the average failure rates of the individual components, This is for a system of the series type, namely a system in which the behavior of any single component has an effect on the overall system

* by E. St. George, Jr.

operation. In the case where redundancy* is used in the design, it is possible for a component to completely fail and have the system continue to operate. The calculation of failure rate in which redundance is used is determined by multiplying the product of the basic failure rate by a redundancy factor.[1]

Another quantitative way of looking at reliability and one helpful in intuitive reasoning is to think in terms of the probability of success of a particular mission. In a series system, the probability of success is equal to the product of the probability of each of the components. The necessity for high reliability of components in a complex system may be illustrated as follows. In a system made up of four components in series, each having a probability of success of 0.5, the probability of success of the system is equal to the product of each of the component probabilities, or about 0.06. In larger, more-complex systems, the effect is even more dramatic and in order to obtain good system reliabilities the individual components must achieve well over a 99 percent probability-of-success figure.

Another consideration affecting reliability is the general behavior of components (and consequently systems) in regard to a breaking-in period† and also in regard to wearing out. Many components show a strong statistical tendency to have high failure rates during the early part of the operating life. Another way of indicating this is that if 100 components are put on life test, a certain percentage will tend to fail during a distinct early-failure period. Those that survive this period tend to have much longer life spans until a wear-out period is reached, at which time the failure rate increases. These characteristics are illustrated in Fig. 9-1. Because of the fact that any previously undetected failure is likely to show up in the initial use of a component, placing a component in its normal operating condition for a brief period is an excellent means of checking the reliability of the component. If the component operates satisfactorily during this check-out period, a high probability exists that it will have a normal operating life.

9-2 EFFECT OF ENVIRONMENT ON RELIABILITY

Various environmental factors including temperature, moisture, shock, vibration, pressure (or lack of it), and contamination obviously have a marked effect on the mean life of a component and consequently on the system that is made up of components. Fire control computers are subject to a variety of environments, from a reasonably benign garrison installation, through groundborne and waterborne installations, to airborne equipment. Not to be discounted are the effects of shock and vibration in shipment, which in some cases exceed conditions to be encountered in the worst airborne applications (ref. par. 8-6 of Chapter 8).

In order to carry out quantitative studies on anticipated mean life, one must have statistical performance data on components under various environmental conditions. While a great deal of work has been done in this area, a staggering amount of statistical information is required to cover all possible components under all possible conditions. Designers, therefore, tend to use statistical mean-life data based on life performance of components under ordinary temperature and pressure with no shock or vibration, and to modify these values to take into consideration the effects of adverse environment. A great deal, of course, can be done to minimize the effects of shock and vibration by properly mounting or insulating components within the structure of the fire control assembly. The same is true of other environmental effects such as salt spray, contaminating atmospheres, and reduced pressure. With any of these, the use of encapsulation, hermetic sealing, and insulating coatings and platings are often effective in increasing the mean life of sensitive components under adverse conditions.

Here, a word of caution is introduced in regard to the concept of mean life. In the majority of reliability calculations, mean life has to do with mean operating life and, in general, under the environmental conditions anticipated, refers only to the useful.

* Redundancy, as the word implies, means a duplication of a particular subsystem function in a critical area so that, if the failure of one of the subsystems occurs, the other will be able to carry out the required function.

† Also commonly referred to as an aging period or a burn-in period.

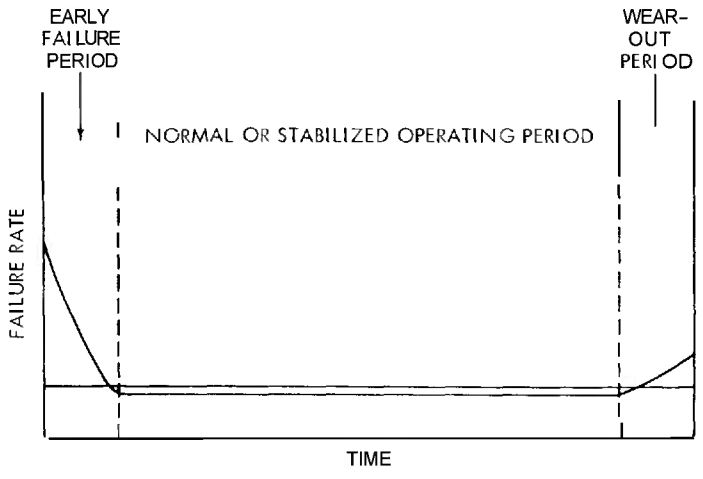


Figure 9-1. Equipment-life characteristics.

life of the equipment when in operation. Another factor that should be taken into account from the overall reliability point of view is the mean shelf-life where stored equipment tends to deteriorate, sometimes as a result of adverse environmental conditions such as excess moisture or micro-organisms. Even under benign environments, long periods of storage tend to deteriorate the operation of systems because of aging of the components within the system. Some data are available to designers on deterioration and change in properties of components with shelf-life; such information can be used in the design of fire control computers in much the same way that mean-operating-life data are used.

The behavior of a simple component in a system is, of course, strongly affected by manufacturing and assembly techniques, which in turn are reflected in performance under adverse environmental conditions. Environmental testing, particularly in regard to shock and vibration but also from the point of view of temperature, salt spray and moisture, is thus important as a final check on the overall system as well as the components making up the system. Many new techniques have been devised for connecting electrical components that are more rugged and less subject to failure than ordinary soldered joints. One example of this is in the use of welded electronic assemblies in which connections between electronic components are made with miniature spot-welded junctions. A high degree of art is required to satisfactorily accomplish this, although the process is now automated in such a way that electrical energy is carefully metered in order that the spot weld will have been sufficiently heated but not excessively so. Metallographic examination of sample welds is necessary for quality control. Advances in printed-circuit assembly techniques also have tended to increase reliability under adverse shock and vibration conditions. While some methods of construction to overcome adverse environment lend themselves better to digital units and others to analog, the general remarks made here apply equally well to either digital- or analog-type computers used in all types of fire control systems.

9-3 LOGICAL DESIGN OF COMPUTERS TO OBTAIN THE DESIRED DEGREE OF RELIABILITY

In the area of logical design, probably the most effective tool for increasing or controlling reliability is the use of self-checking and self-correcting codes in the logical design of the fire control computer. Another effective concept is that of utilizing redundant elements. Redundancy is an expensive, but often effective, method of increasing reliability. If it is practical to provide the extra space and weight, and if the additional cost of the duplicate subsystems can be tol-

erated, it is possible to appreciably increase the reliability of the fire control computer. Lest one conclude that the use of redundancy offers an inexpensive cure-all, it should be borne in mind that duplication of equipment requires an increase in the number of components by more than a factor of two, because not only must switching for the alternate equipment be provided but also some method of determining that the first subsystem has failed must be available. Furthermore, it must be taken into account that the gain in reliability from redundancy is not quite what might be expected at first because the switching and detecting equipment involved cannot be made one-hundred-percent reliable.

With regard to self-checking and self-correcting codes in the design of larger computing machines, an excellent example is represented by the test scheme provided for the FADAC computer. (The FADAC computer itself is described in Chapter 13.) The FADAC Automatic Logic Tester (FALT)[4] checks the logic of the FADAC computer and localizes any errors detected. Logic tests are read from a 5-channel, punched paper tape by a photo-electric tape reader. As this tape is read, corresponding logic tests are performed on the FADAC computer. When FALT detects an error, the reader halts and an ERROR light flashes. The point at which the error occurs is displayed on a visible numerical readout as a pair of numbers: a marker number and an index number. Reference to these two numbers in a test listing identifies the area in which the error has occurred.

FALT-test-tape programming is based directly on FADAC logic equations so that FADAC is tested at the level of the individual logic gate. Identification of logic gates associated with a malfunction are made in the test listing. Provisions have been made in the test listings to include not only the logic but also the board locations associated with it, in order to enable the technician to check the visible numerical readout against the test listing and immediately identify the boards on which the suspected logic is physically located.

Maintenance of the FADAC system requires that failures or malfunctions be rapidly located and removed so that the system can be kept in operation with a minimum of down-time. Although the repair of any individual logic circuit is relatively simple, the more rapidly the malfunction is removed at the field-maintenance level, the quicker the system can be returned to operation. FALT has been designed to locate FADAC logic malfunctions as rapidly as possible. When FRLT has localized the malfunction, the normal field check-out procedure would be to trace the suspected logic to its actual physical location on a plug-in board, and replace the board. Once the board has been replaced, the test tape associated with the suspected logic is re-run to determine whether the malfunction has been removed by board replacement.

FALT logic is mechanized to obey certain instructions or commands that are punched on paper tape or manually entered through the F4LT control panel. The nature of the command determines which outputs are transmitted to what elements of FADAC logic. To enable FALT to perform these functions, a temporary storage memory composed of several registers and counters is designed to control, hold, or count data read from the tape. These logic and control functions are performed by flip-flops, logic networks, and crosspoint networks.

The E'ADAC computer itself uses 19 identical crosspoint network boards. A crosspoint network board contains ten circuits divided into two types: four of one type and six of a second type. Each of the first type of circuit is confined to the selection and testing of flip-flops, including pseudo flip-flops. Each of the second type of circuit has an additional gate, thereby allowing the circuit to test single-ended outputs, such as primary logic gates or logic drivers.

Crosspoint network boards are located electronically between FALT and the computer. Each board is capable of linking FALT with the computer for automatically performing four separate functions. Ten of 190 possible addresses into the computer, as commanded by FALT, are provided by each crosspoint network board. The boards enable field-level personnel to isolate a circuit board in which a malfunction exists. The automatic location of a defective circuit board allows the computer to be restored to operation in minimum time. A crosspoint network board performs four principal functions:

(1) It selects an address in the computer on command from FALT.

(2) It orders a test on a flip-flop, and sends the resultant signal to FALT for examination.

(3) It orders a test, conducted similar to flip-flop testing, such that there are two gates, one gate per side. Because of the two sides, this gate test is called a "pseudo flip-flop".

(4) It orders a test on single-ended gates, such as primary gates or logic drivers, and sends the tested signal to FALT for examination.

Five-channel teletype coded paper tape is used as the input to FALT. The input device is the memory loading unit. The test-tape information is fed into FALT at the rate of 600 sprocket rows per second and FALT will process the test commands at this rate. Six tapes totaling approximately 2600 feet are required to perform a complete static check-out of FADAC logic. If this required 2600 feet of tape were not separated, it could be fed into FALT, and the results of F'ALT commands applied to FADAC in approximately 8.7 minutes. However, actual computer check-out time is largely a function of preliminary operational setup and manual setup, in which a technician must perform certain instructions manually.

Marker numbers are punched on each test tape so that the location of the malfunction can be found in relation to the information location on the tape. The marker number precedes each marker test group shown in the test listings, such as the negation logic test of a particular flip-flop. The marker number consists of four octal digits with a range of 1111 to 7777 (zeros are not used). When an error is detected during the running of a test tape, the marker numbers will be displayed in the visible numerical readout on the front panel of FALT. The technician will then find the marker number displayed on FALT in the test listings. FALT stores and displays the marker numbers through the marker register. As the marker numbers are read from the tape, they are shifted through the input register to the marker register, where they are stored and displayed in the visible-readout indicators. During the normal running of the tapes, the visible readouts are continually flashing off and on, but will display the marker number at the location in the test at which the tape halts.

Index numbers, which are three digits in length, are displayed in the readout but are not punched on the tape. The index number is made available by the counting of tape characters through the FALT index counter. The end of a tape character is identified by the presence of a hole in the fifth channel of the tape. With two exceptions (the delete and HRA* characters), the index counter increases by one count each time this hole is sensed. When an error is detected during the running of a test tape, the index numbers will be displayed in the readout on the front panel of FALT. The technician must then locate the marker number in the test listing before proceeding to the index number. The index number represents the actual name of the flip-flop within the marker test group.

Test listings are compiled for each test tape to enable the technician to (1) determine the overall test function of the tape, (2) give him a graphic view of the actual sequence, and (3) enable him to determine the command or flip-flop address from the surrounding listing at which the malfunction has caused a halt in the tape. Test listings are basically a tabular listing of the actual FALT commands and addresses programmed into the test tape. Each tape is separated into sections called marker test groups. Between each marker number on the tape, a series of commands and addresses are programmed to test a flip-flop, "AND" gate, logic driver, or write switch by applying certain inputs dependent on the nature of the command. This is the marker test group within the test listing. Each marker test group is headed by the marker number identifying it, and is further identified by a description of the test group. All test groups are sequential, i.e., the test listing is numbered sequentially from left to right for as many index counts as are required for the number of commands and addresses in the marker test group. When the readout for the index numbers is displayed after an error, the technician must check the index number displayed against the index count in the listing.

* Halt Reader and light "A" lamp; the HRA neon indicator lights when a programmed halt (HRA) instruction has been executed by FALT.

FALT commands and addresses (the particular FADAC flip-flop "addressed" by the tape characters following the command character) are not listed. A typical portion of a FALT tape listing is shown in Fig. 9-2.

FALT panel controls initiate and regulate the following functions (ref. Fig. 9-3):

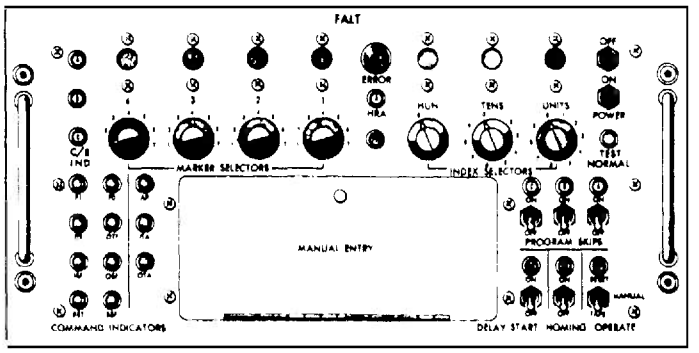
(1) The crosspoint boards enable the selection of a flip-flop in the computer by addressing that component (after commands, the address follows) so that a prescribed test may be ordered and executed by a FALT command. Some commands 1-set or 0-set a flip-flop; other commands determine which outputs are to be transmitted to flip-flops, primary "and" gates, and logic drivers.

(2) With respect to flip-flop testing, once the applicable crosspoint board has located an address in the computer, the FALT command "1TF" or "0TF" for that address determines whether that flip-flop is 1-*set* or 0-set. FALT receives back, via the crosspoint board, the flip-flop outputs on the FIT and FOT lines. For a 1-set condition, FIT will be true (-6 volts), which corresponds to the unprimed flip-flop output, and the FOT will be false (0 volts).

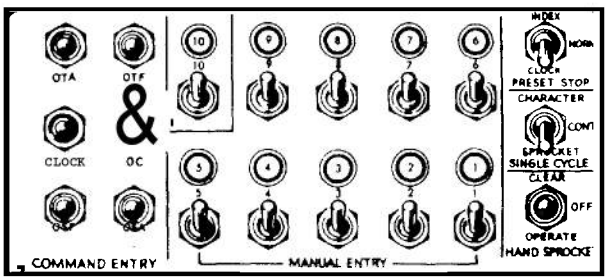
(3) For each of the "or" diodes, there is an "and" gate. The "and" gate must be set to the "true" state in order to test the "or" diode. Setting of the "and" gate is accomplished by causing each of the terms in the gate to be "true". FALT next commands the computer to generate a clock pulse. With the clock pulse, the flip-flop is set so that the "or" diode is an input term. That flip-flop is then addressed by the crosspoint and 1-tested. If

```
B       1       1       4       t
B
BPULL  HOARD #319 AND&OR BOARD #321 FOR ERRORS AT  1146-073
B               1146-085                                -094
B                     -104
BPULL  BOARD #235 AND&OR BOARD #243 FOR ERRORS AT  1146-114
B               1146-117                          1147-004
B       1       1       4       7
B
BPULL  BOARD #340 AND&OR BOARC #343 FOR ERRORS AT  1147-013
B               1147-025                                -034
B                     -044
BPULL  BOARD #234 AND&OR BOAR3 #243 FOR ERRORS AT  1147-053
B               1147-064                                -073
B       1       1       5       1
B
B               HRA
B
BADDRESSABLE PRIMARIES
B    SOME OF THE PRIMARY *AND* GATES IN FADAC ARE CONNECTED FOR TEST
BBY THE AP LINE OF FALT.  THIS TAPE TESTS THESE PRIMARIES.
B
B1. START THE TAPE READER.
BPULL BOARD #202 FOR ERRORS AT  1151-004   1151-015   1151-024
B                                 -034       -046
BPULL BOARD #205 FOR ERRORS AT  1151-054   1151-065   1151-074
B                                 -086       -094
BPULL BOARD #206 FOR ERRORS AT  1151-105   1151-114   1151-123
B                               1154-005   1154-015
B       1       1       5       4
B
BPULL BOARD #207 FOR ERRORS AT  1154-025   1154-034   1154-045
B                                 -953       -065
BPULL BOARD #208 FOR ERRORS AT  1154-075   1154-084   1154-095
B                                 -105       -113
BPULL BOARD #212 FOR ERRORS AT  1154-125   1155-005   1155-016
B                               1155-024
B       1       1       5       5
B
BPULL BOARD #143,234,243 FOR ERRORS AT     1155-033   1155-048
```

Figure 9-2. Typical portion of a FALT tape listing.



(A) Normal Controls



(B) Manual Controls

Figure 9-3. E'ALT operating controls.

the flip-flop is 1-set, the 'lor" diode is satisfactory. Each diode in the "and" gate is then tested in the same manner as primary "and" gate diodes. Each flip-flop has "gating" logic associated with both the 1-set and 0-set side. The gate on the 0-set side is next tested in the same manner as for 1-set, with the exception that the flip-flop is 0-set, then addressed, anti 0-tested.

(4) To insure that a primary "and" gate is functioning properly, two conditions must be satisfied. First, each term in the gate must be set to its true condition, and the output of that gate tested to be at a true level. Secondly, each term in the gate must be set to its "false" condition (one term at a time) and the output tested for the false level. Tests insure that a gate not only functions in the "true" condition, but also that it will not work if one of the input terms is "false", or if one of the input diodes is in an open state.

Six of the circuit-board types used by FALT are identical with those used in the computer unit. These circuit boards are as follows:

Circuit Board	Quantity
Flip-flop	6
Crosspoint network	5
Power supply subassembly	1
Clock amplifier	1
Rectifier diode assembly	1
Transistor assembly	3

In addition, FALT uses the circuit boards described below.

(1) Power Supply Boards (Nos. 1 and 2). These two boards supply the d-c voltages used throughout FALT: power supply board No. 1 furnishes the regulated -12, -18, -50, -25, and +12 volts, while power supply board No. 2 furnishes the regulated +25, +6, -6, and +1.2 volts.

(2) Network Logic Board. The network logic board has three functions:

(a) It contains the matrix driver amplifiers used to drive the DU and DL lines.

(b) It contains the amplifiers that are used for the neon drivers and the visible-readout drivers. There are

two types of circuits: shunt-type drivers and series-type drivers.

(c) It furnishes the logic gating for FALT, (This is the network logic board's main function.)

(3) Network Amplifier Board. The network amplifier board contains the following circuits:

(a) Oscillator circuit (used for clock oscillator or flashing error-light).
(b) Threshold amplifier (samples F1, FO, and AP).
(c) Crosspoint circuit.
(d) Clock-trigger amplifier.
(e) Inhibit amplifier.
(f) Error-light amplifier.
(g) Mechanical tape reader clutch-brake driver.
(h) Crosspoint set-all amplifier.

The front panel of FALT mounts all switches and indicators used to control and monitor the testing of the computer, as shown in Fig. 9-3.

9-4 COMPUTER CHECK-OUT PROCEDURES AND EQUIPMENT

9-4.1 MAINTENANCE

As mentioned in par. 9-1, one of the three important requisites of reliability is dependable maintenance. Once a piece of fire control equipment has been manufactured and inspected and has passed its acceptance tests, it is ready to perform in a manner determined by its performance specifications and to a degree of reliability as determined by the design specifications and by the quality assurance program. However, as time goes on, deterioration due either to usage or to the passage of shelf-time occurs. The answer to this part of the problem of reliability is dependable maintenance.

A standard procedure for determining whether or not component replacement as a maintenance measure is necessary in fire control systems using analog computers is to test the computer by using the input from a problem simulator and verifying the outputs against a pre-computed digital check solution. Deviation from certain norms will indicate the necessity of maintenance in the form of replacement of components, or subsystems, or in re-working subsystem elements. The same general procedure may be applied to fire control systems using digital computers. However, in this particular case, the final solution will have been worked out at an earlier date by the computer itself, or by another digital computer.

Since fire control systems are relatively complex devices, and computers are versatile enough to solve three-dimensional problems with multiple variations in parameters, the trial-solution method described is not always as effective as might be desired owing to the complexity of the analysis necessary to determine trouble spots. The immediately following paragraphs describe alternative schemes that have proved to be highly successful.

9-4.2 MARGINAL CHECKING

One of the most powerful techniques for checking computers is called marginal checking. This has many advantages over the trial-solution method, particularly in regard to digital computers. In marginal checking, one or more parameters of the computer – usually power supply voltage – is varied above and below the normal tolerances while a test problem is running. Errors in a solution indicate that one or more components is marginally operative and should be replaced.

Marginal checking can be applied to analog computers but the entire computer cannot be checked because of the difficulty in tracking down the unit whose performance is marginal. Instead, operational units are designed to plug or patch into the system and be removed and marginally checked individually on a regular schedule. Equipment is available that automatically performs marginal checks on operational units of analog computers.

In the marginal checking of digital computers used in fire control systems, the entire computer is switched to a marginal check mode in which the existence of errors and the location of the offending component can be determined. Modular design of the computer facilitates replacement of the component units.

Since marginal checking is such an important part of maintenance of digital computers used in fire control systems, it is recommended that the reader consult the basic work in this area, which is well documented in the classical paper titled "Designing for Reliability". This paper was written in 1957 when transistor applications were not as common as at the present time. Consequently, not much mention is made of transistors, but the general philosophy is quite applicable to more modern transistor circuit design.

9-5 SPECIAL-PURPOSE CHECK-OUT EQUIPMENT

The following description of the REDSTONE Missile Firing Data Computer[6] is provided as a typical example of the use of special-purpose check-out equipment in a highly reliable computer system.

9-5.1 GENERAL DESCRIPTION

The REDSTONE Missile Firing Data Computer system (see Fig. 9-4) is designed to

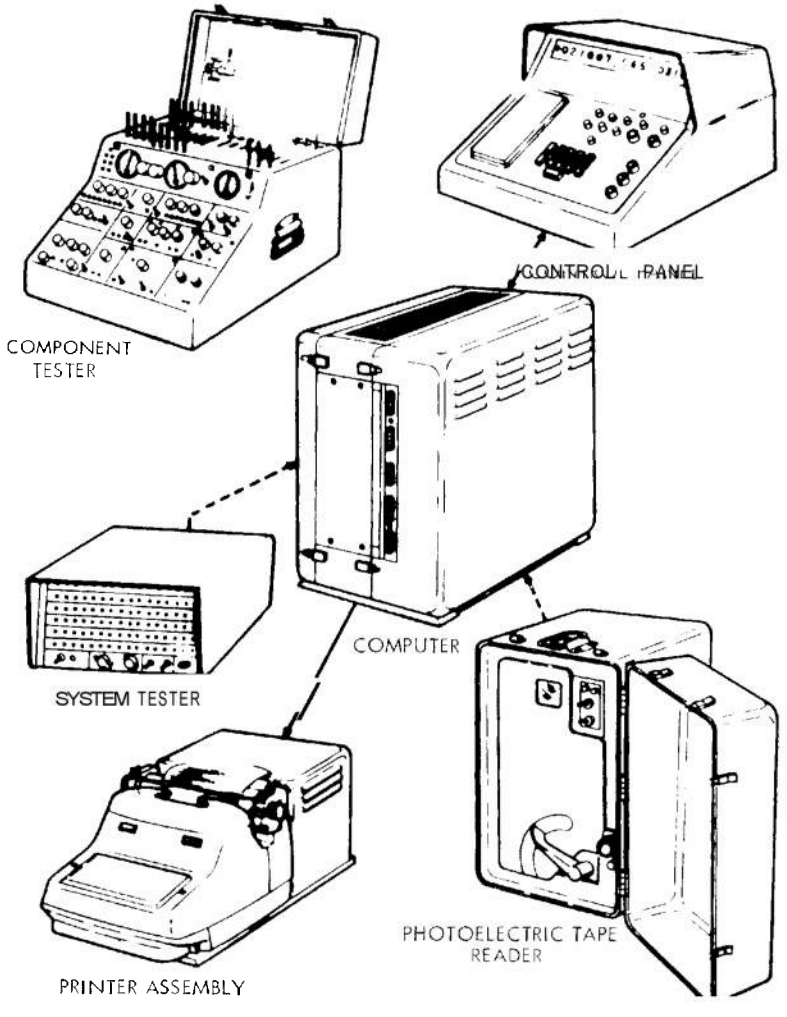


Figure 9-4. The REDSTONE Missile Firing Data Computer

compute the dial settings of the missile launch equipment. The system comprises the following equipment:

(1) General-Purpose Digital Computer – consists basically of (a) a rotating magnetic memory in which the problem information is stored and (b) circuits mechanized in accordance with specific logic equations to perform basic arithmetic operations and readout functions.

(2) Control Panel – provides controls for activating the computer, a decimal keyboard for entering numerical quantities specific to the problem, and miscellaneous controls and error indicators.

(3) Photoelectric Tape Reader – reads punched-tape information for entry into storage locations of magnetic memory.

(4) Printer Assembly – displays problem solutions in print.

(5) Facilities for supplying tape-punch or other external drive signals.

(6) System Tester – checks overall state of computer.

(7) Component Tester – tests computer etched-circuit panels.

9-5.2 DETAILED DATA

Computer and Control Panel

TYPE	General purpose, serial, single address	
PHYSICAL CHARACTERISTICS		
Size:	Computer	23 inches X 21 inches X 13 inches
	Control Panel	19.5 inches X 16 inches X 11 inches (max height)
** eight:	Computer	125 pounds
	Control Panel	30 pounds
Power:	3-phase, 400-cps, 120/208-volt, 4-wire system: 368 watts at 0.59 power factor (includes computer and control panel but not input-output equipment)	
Temperature:	Normal operating temperature: 9°F above ambient (Blower airflow: 100 cubic feet per min) Temperature warning thermostat setting: 115°F Power-supply thermostat setting: 130"F Memory run thermostat: memory deactivated at less than 55°F internal temperature	
MEMORY		
Type:	Rotary magnetic disk (2000 revolutions per minute)	
Capacity:	Permanent storage	3840 words
	(information can be modified only by tape reader)	
	Working storage	240 words
	High-speed storage	16 words
	Total	4096 words
	one-word arithmetic registers clock channel (3.5-microsecond synchronizing pulses) sector-origin channel	

	Bits, including sign, equivalent to approximately 12 decimal places
Command:	Bits per command stored 2 commands per word

COMM4NDS

Input:	"Start Tape Reader"
Output:	"Word Display", "Word Type", "Word Punch"
Arithmetic:	"Clear and Add", "Add", "Clear and Subtract", "Subtract", "Multiply", "Divide", "Shift Right", "Shift Left"
Information transfer:	"Store Word", "Interchange Registers", "Memory to High-speed Loop L", "Loop L to Memory", "Memory to High-speed Loop V"
Control:	"Transfer on Negative", "Transfer on Positive", "Transfer Unconditionally", "Store Address", "Halt and Transfer", "Extract"

TIMING

Access time (time to locate a memory cell and read its contents or write information thereon):

Information channels	15.77 milliseconds average 30.60 milliseconds maximum
High-speed channels	2.59 milliseconds average 4.23 milliseconds maximum

Operation time:

4dd-subtract	0.94 millisecond (excludes access time)
Multiply	18.8 milliseconds (excludes access time)
Divide	19.7 milliseconds (excludes access time)
Transfer control	1.41 milliseconds

INPIJT

Photoelectric paper-tape render (input to computer permanent storage and working storage)

Control panel keyboard (input to computer working storage)

(Decimal entry of numbers requires stored subroutine)

OUTPUT

Electric printer

Control-panel readout

(Decimal output of numbers requires stored subroutine)

Printer and Printer Drive Package

TYPE	Modified electric typewriter	
PHYSICAL CHARACTERISTICS		
Size:	Printer	17 inches X 14.5 inches X 12 inches (maximum height)
	Driver package	13 inches X 11 inches X 8 inches
Weight:	Printer	55 pounds
	Driver package	18 pounds
Power	115 volts, 60 cps, single-phase	
	Driver package	50 watts peak power 20 watts average power during printout 8-watt standby power
Temperature:	Printer	Ambient
	Driver package	Ambient

Photoelectric Tape Reader

PHYSICAL CHARACTERISTICS	
Size:	21.5 inches X 15 inches X 12 inches
Weight:	45 pounds
Input Power:	Single-phase, 60 cps, 115 volts, 75 watts average
Temperature:	Ambient
OUTPUT	200 characters per second read into computer
TAPE	5-channel, teletype-coded, punched paper Width 11/16 inch Length 8-1/2 inches between folds 10 characters per inch
MODES OF OPERATION	Fill, Verify
OUTPUT SPEED	200 characters per second

System Tester

PHYSICAL CHARACTERISTICS

Size:	19 inches X 19 inches X 9.5 inches
Weight:	33 pounds
Input Power:	3-phase, 400-cps, 120/208-volt, 4-wire system: 60 watts
Temperature:	Normal operation 9°F above ambient (Blower airflow: 20 cubic feet per minute)

Component Tester

PHYSICAL CHARACTERISTICS

Size:	20 inches X 19 inches X 14.75 inches
Weight:	52 pounds
Input Power:	3-phase, 400-cps, 120/208-volt, 4-wire system
Temperature:	Normal operating temperature 9°F above ambient (Mower airflow: 120 cubic feet per minute)

9-5.3 BASIC ELEMENTS

The REDSTONE Missile Firing Data Computer is a general-purpose solid-state digital computer. Such a digital computer can perform a large number of different operations and calculations by use of the basic arithmetic operations of addition and subtraction. The procedure for completing these operations is under the guidance of a control unit. The type of operation performed is dependent on the set of instructions or commands placed in the control unit. A change of instructions can be made without any physical change required in the computer proper.

Basically, operation of the cornputer is dependent on five sections: memory (storage) unit, arithmetic unit, control unit, input device, and output device (see Fig. 9-5). These sections are interconnected electrically and are under the control of the control unit.

The memory is of the magnetic-disk type and consists of a number of storage locations in which information can be stored and from which information can be extracted. The information is of two types: a number representing problem data or a number, termed "command", representing an operation to be performed. Therefore, the memory contains not only the data pertaining to a particular problem, but also the operations required to obtain the problem solution. To prevent erasing of permanently stored information under certain conditions of operation, the memory is arbitrarily divided into permanent and working storage areas.

The arithmetic unit consists of several temporary one-word registers (the accumulator, remainder register, and operand register), together with appropriate switching and control elements for carrying out basic arithmetic and logic operations. (A register is a device for retaining information.) The remainder register samples incoming and outgoing information, and is used as a temporary storage register in arithmetic operations.

The operand register has several functions: (1) application of incoming information (either commands or numbers) to the mem-

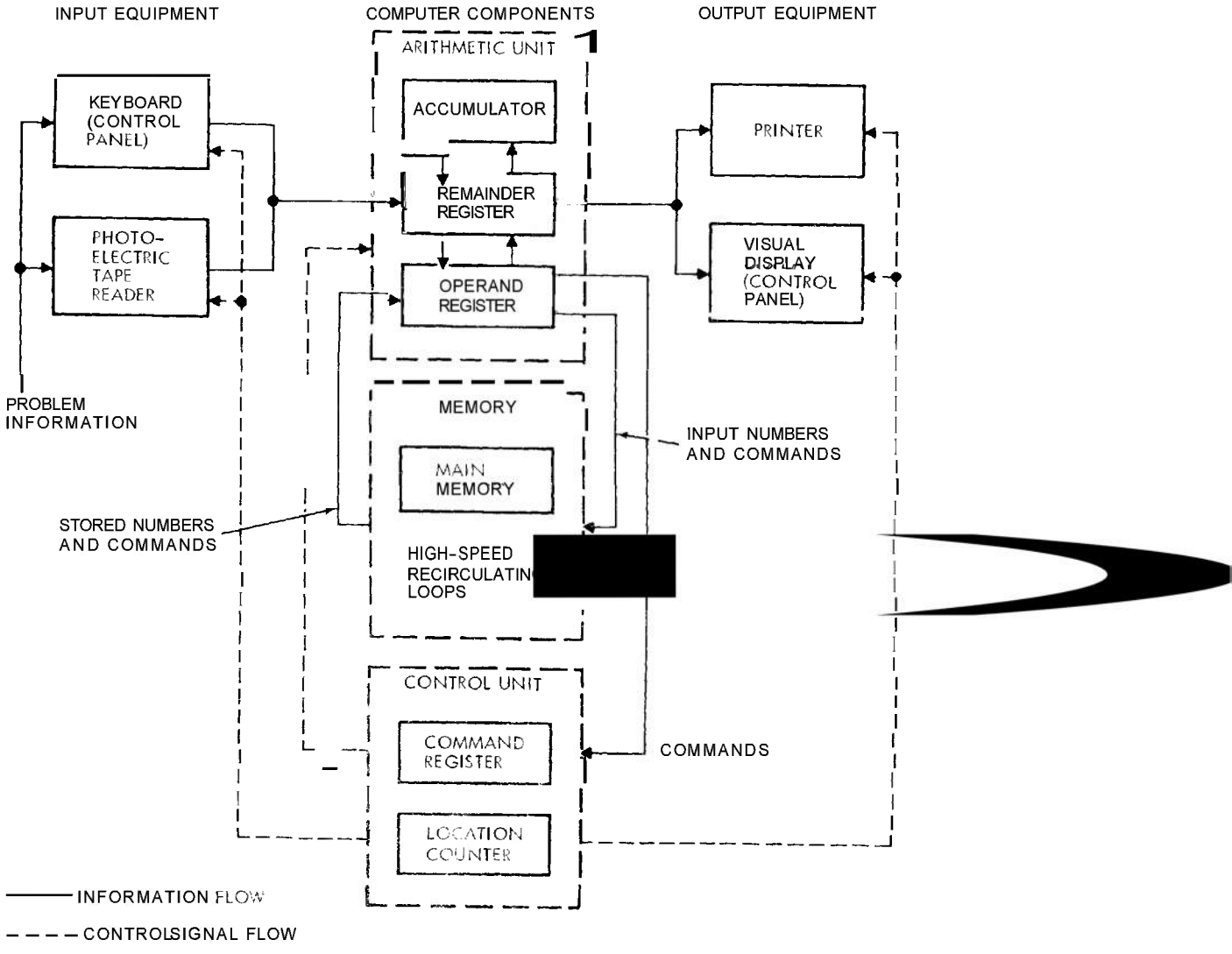


Figure 9-5. Block diagram of the basic computer system.

ory for storage, (2) application of commands selected from the memory to the control unit, and (3) in accordance with signals from the control unit, distribution of numbers that are to be operated on to the computational circuits of the arithmetic register.

The control unit consists of counters and registers that select each command in *sequence* from the memory and that apply control signals to the other computer elements for execution of the command. The main components of the control unit are the command register and the location counter. The command register is a one-word temporary storage register that holds the command to be executed. The location counter holds the memory location (address) of the command to be executed, increasing by one count each time a command is executed in sequence.

The input device is used to fill the computer memory with commands and numbers, to set the location counter to the address of the initial command, and to provide starting and stopping signals. A photoelectric reader and a control-panel keyboard with associated visual display are the input devices for this computer.

The output devices of the computer are a printer and the readout display on the control panel that print and display, respectively,

computational results. Provisions are also made for tape-punch output.

9-5.4 GENERAL METHOD OF OPERATION

In general, the computer system operates in the following manner (see Fig. 9-5). Various mathematical constants, and a plan (program) for the solution of the particular probleni are stored in the permanent storage location of the computer magnetic memory. This permanent information is fed into the computer one word at a time, by punched paper tape and the tape reader. The coded information is interpreted in the computer input register and forwarded in computer language to the location counter. The location counter in conjunction with a memory addressing unit causes the program to be stored on the memory in the order of location of the program.

The program consists of sequences of commands. These commands tell the coniputer which operation to perform, the memory location of mathematical constants and special routines in the permanent storage that are required for solution of the problem, the memory locationof numbers to be operated on, the memory locations fortemporary storage of these numbers and of intermediate results.

The problem parameters, i.e., data concerning the launch point and the target, are then entered into the computer- by means of the control-panel keyboard. When the PARAMETER or START buttons are pressed, computation automatically starts with the first command of tlie program. This command is stored in the memory location previously set into the location counter. The location counter, in conjunction with the memo~-addressing unit, locates this command, selects it from the memory and places it in the command register. The address part of the command specifies the location of the number io be operated on. The number is then selected froni the memory by means of the location counter and directed to the operand register. Here, it is ready to be operated upon by the computational processes.

The command also contains an operation code, i.e., a number that tells the coniputer- which operationto perform. From the command register, the code is transferred to the operation code register. The latter register, in conjunction with a decoding network, translates the operation code into the various separate control signals for carrying out the computation commanded. The number in the number register is then operated on and the result is stored in the accumulator. Remainders of division operations and the least significant digits of multiplication operations are stored in the remainder register.

Special commands cause the output register to receive the number, or part thereof, from the accumulator and activate the printer, tape punch, or visual display. The computation process is repeated for each command in turn until a "stop" command is encountered.

9-6 MEANS AND FACTORS TO BE CONSIDERED IN VERIFYING THE DESIGN OF REAL-TIME FIRE CONTROL COMPUTERS

Important to the achievement of high reliability in a real-time fire control computer is the evaluation process used to verify the computer design -- especially in regard to the settling time and accuracy characteristics required for various tactical situations. The fire control system designer should be aware of the means and factors to be considered in such an evaluation since the hardware-evaluation cost may easily exceed the original design cost before a fully acceptable item can be placed in the field.

The major solution factors used to evaluate real-time fire control computers are the solution-time and accuracy requirements for various tactical situations. The primary and worst-case tactical situations should be defined and analyzed for tlie tracking rates and accelerations that have to be followed by the coniputer. The output of the computer should then be measured and checked against the solution-time and accuracy requirements. These two factors are related directly to the resolution and the bandwidth of the computer and its solution of a fire control problem.

System testing or flight testing to ascertain the computer's capability of meeting these requirements is usually the final type of evaluation that is performed. In carrying out this evaluation, a real-time dynamic

tester that simulates typical input conditions should be employed. The computer output data should then be evaluated against the ideal *so*-lutions in terms of the response requirements.

The following are examples of tactical situations that could be used for evaluation purposes.

(1) For antiaircraft fire control
   (a) Crossover courses with various ranges out to 1500 meters at crossover.
   (b) Particular types of evasive courses.
   (c) Spot firing to test slewing rates.

(2) For helicopter fire control
   (a) Slalom courses.
   (b) Crossover courses with various ranges at crossover.
   (c) Spot firing to test slewing rates.

(3) For tank fire control
   (a) Situations that involve both stationary and moving ground targets at speeds of 5 to 40 mph, and both stationary and moving weapons at speeds of 5 to 25 mph.
   (b) Crossover courses with various ranges from 400 to 2500 meters.
   (c) Spot firing to test slewing rates.

9-7 CONCLUSION

In summary, it may be said that the reliability of fire control system computers depends on three elements – (1) design, (2) quality control, and *(3)* maintenance.

Design involves the use of such commonsense approaches as the use of derated components. In addition, it requires the application of more-sophisticated techniques involving the statistical analysis of components and subsystems and consideration of the interrelation of these factors. Design also takes into account environmental conditions to be encountered and uses corrective measures as, for example, vibration isolation to counteract hostile environment. Quality assurance, as the name implies, is a sustained, organized scheme of design rules, manufacturing controls and test procedures to assure that the quality of all components, subsystems, system interconnections, mounts, and housings are either equal to or better than the design specifications. Maintenance by means of systematic and sometimes automatic checking and a subsequent replacement of substandard components, and by routine servicing, keeps equipment at a condition of reliability equal to that guaranteed by the quality assurance team.

One other factor that affects reliability of almost any device is experience in its design, manufacture, and use. As time goes by, improvements are made in the design based on experience in the field. This is probably one of the main reasons why digital and analog computers for fire control use have reached such a high degree of reliability at the present time.

REFERENCES

1. M. F. Goldberg and J. Vacaro, Eds., Physics of Failure in Electronics, Vol. 1 (1964), Vol. 2 (1965), Vol. 3 (1966), Rome Air Development Center Series on Reliability. [Qualified requesters may obtain copies from the Defense Documentation Center (TISIR), Cameron Station, Alexandria, Virginia 22314. Other requesters may order copies from the Clearinghouse for Federal Scientific and Technical Information (CFSTI), Department of Commerce, Sills Building, 5285 Port Royal Road, Springfield, Virginia 22151.]

2. Lecture Notes; Integrated Electronics Lecture Series, Electronic Components Laboratory, U.S. Army Electronics Command, Fort Monmouth, N. J., 1966.

3. AMCP 706-327, Engineering Design Handbook, Fire Control Series, Section 1, Fire Control Systems – General.

4. Gun Direction Computer XM18 (FADAC), FCDD-361, Vol. I, Frankford Arsenal, June 1961.

5. Norman H. Taylor, "Designing for Reliability," Proc. IRE, Vol. 45, No. 6, June 1957.

6. Missile Firing Data Computer (REDSTONE), FCDD-340, Vol. I, Frankford Arsenal, September 1961.

PART III
THE REALIZATION OF A PROTOTYPE FIRE CONTROL SYSTEM BASED UPON A MATHEMATICAL MODEL

CHAPTER 10
PROBLEMS ASSOCIATED WITH THE MECHANIZATION OF MATHEMATICAL MODELS *

10-1 KINDS OF PROBLEMS ASSOCIATED WITH MECHANIZATION

The availability of a mathematical description for a system that presumably can be built without requiring any unrealistic advances in the current state of the art represents only a first step toward the realization of an operational system. This part of Section 3 (Part III) deals with the general types of problems encountered in transforming a mathematical description into a physical system.

Fortunately, many of these problems are of a relatively routine nature and in many instances subsystems having entirely adequate performance characteristics are available as standard commercial products. If an engineering group is to transform a mathematical model. into a physical system in a reasonable time and at a reasonable cost, it is essential that they be well aware of the variety of existing products capable of meeting the specifications of the design. They must, of course, also be familiar with the special requirements imposed upon the design by the conditions under which their system will operate, such as high temperature or humidity or the need to withstand high shock loads from being handled roughly.

Before commencing the physical realization of a mathematical model, the system designer must put the model into a form that is suited to the computational hardware. This operation is usually called programming, and is much more significant for digital computers than for analog machines. Programming an analog computer merely requires rearrangement of the equations so that the operations called for correspond to those available in the computer. For example, equations are usually rewritten so as to convert differentiation to integration.

In the case of a digital computer, programming is carried out in two stages. First, the equations are arranged for solution by the numerical methods described in par. 2-2 of Chapter 2. Then, each step of the procedure is written in machine language for entry into the computer memory. (See the example given in par. 4-5.3 of Chapter 4.) Because of the very different programming requirements between analog and digital mechanizations, even the initial mathematical model may be influenced by the choice between these two methods.

Realization of a complex system will undoubtedly be accomplished in several stages. Attempts to jump directly from a mathematical description to a final system are almost certainly doomed to failure if the system is even moderately complex and especially if it represents any significant departure from systems with which considerable past experience has been obtained. Consequently, the initial physical models usually take the form of relatively crude breadboards designed to demonstrate feasibility without giving particular attention to considerations of size, weight, cost, and other parameters that must be considered before production is initiated on a final version of the system. The components and subsystems employed in the breadboard are not necessarily those used in the final design. It is not necessary at this stage to consider environmental conditions and minimization of the numbers of components employed. To assist in the isolation of prob-

* By W. W. Seifert and Emery St. George, Jr.

lems, components will frequently be used in the breadboard that have higher performance than is actually required.

At this breadboard stage in particular, the designer who has the ability to improvise or innovate in order to assemble a simple and economical demonstration system is a great asset. Regardless of the capability with which initial conceptual and mathematical design of a system is carried out, unforeseen problems almost inevitably arise when an attempt is made to convert a design into hardware. Some of these problems arise because the limitations on mathematical analysis do not permit as complete a study as would be required to predict the exact performance of a system. Some of these limitations exist because knowledge in certain areas has not yet been developed to the point where completely adequate mathematical models can be formulated. Other limitations relate to the difficulty of obtaining numerical solutions even when a mathematical description is available. While modern computers are rapidly reducing this latter limitation, the effort and expense involved in studying complex systems are still very great and frequently mechanization of a system must proceed before as much analysis is done as might be desired. Consequently, it is highly desirable that a design move as rapidly as possible to the point where problems of mating components and subsystems become clear and an indication of overall feasibility can be obtained.

The next step toward transforming a design into a final operating system involves building a model that will meet the design specifications not only with respect to such features as accuracy and speed of operation but also in regard to size, weight, power consumption, and human-engineering features. This will still represent a largely custom-made model and the problems of producibility and operating reliability must still be faced. None the less, this prototype unit provides an important next step toward checking out the overall design and, if successful, may permit the designers to obtain some realistic field-operations data and thus appraise the effectiveness of their proposed design. Even at this stage in a design, some relatively major changes may be called for and it is still possible to incorporate improvements if it can be shown that they will lead to definite improvements in performance or production while not delaying the availability of the system unduly. In this regard, it should be noted that engineers are inclined to continue to change, improve, and modify systems almost without end unless someone really stops them. The project manager is thus apt to be faced with the problem of deciding when a system really needs further work before it will meet the desired specifications and when the engineers are merely acting as perfectionists. Sometimes, he has the even more difficult task of deciding that a design can never meet the specifications and that a completely new approach is therefore required.

After extensive testing of the prototype unit, the next step will be the production of a relatively small number of units for the purpose of gaining further information on the problems associated with producing the system in quantity rather than on an engineering-model shop basis. Here, new problems arise and the skills of engineers trained in production methods, materials properties, reliability, and quality control are required.

The final step represents volume production of the system. Once a system has reached this stage the only modifications necessary should be the relatively slight ones that are made to accommodate production procedures rather than to alter system performance. Unfortunately, many cases exist in which these supposedly trivial changes led to serious degradation in system performance. Consequently, they should be initiated only after very careful study and should be referred back to the system designer for concurrence.

10-2 COVERAGE OF REMAINDER OF PART III

Chapter 11 discusses some of the characteristics peculiar to computers incorporated in fire control systems while the remaining chapters of Part III are devoted to the presentation of several examples illustrating ways in which the problems were handled that arose in transforming system designs into operating systems.

CHAPTER 11
CHARACTERISTICS PECULIAR TO COMPUTERS USED FOR FIRE CONTROL APPLICATIONS

11-1 OVERALL DESIGN

A number of features of the logical design, component selection, and overall packaging of fire control computers should be distinguished from the case of general-purpose computers. For the most part, the design techniques, components, and systems described for analog and digital computers, digital differential analyzers, and analog-digital conversion (see Chaps, 4, 5, 6, and 7) are alike applicable to fire control computers, other special-purpose computers, and general-purpose computers. However, the particular requirements imposed on a fire control computer have dictated the more frequent choice of certain systems and components.

11-1.1 MECHANICAL ANALOG COMPUTERS

Historically, mechanical analog computers have played an important part in fire control systems. Mechanical analog devices have advantages of compactness and ruggedness. They are little affected by temperature variation, shock and vibration, and supply-voltage changes. Their relative inflexibility is not a disadvantage in fire control applications, where the form of the equations is fixed.

Input data are usually transmitted by synchros, and introduced by means of instrument servos. (Originally, input data were introduced by human operators using pointer-matching.) Such instrument servos form an important part of a mechanical or electromechanical analog computer. The block diagram of a typical instrument servo, employing two-speed synchro data and tachometer feedback, is shown in Fig. 11-1. The demodulator eliminates the carrier *so* that a lag-lead network can be introduced for the purpose of increasing the servo loop gain. The demodulator also provides for rejection of the quadrature component of the error voltage, which is often a serious problem in high-gain systems. The tachometer provides damping for the system; an alternative that is frequently employed is a mechanical damper.

AMCP 706-139[1] includes a description of a typical high-performance instrument servo, in this case employing damper stabilization. The performance is primarily specified by the system velocity constant (loop gain) K_v and the bandwidth BW. The velocity constant determines the error of the servo when the input shaft is turning at a constant velocity, and is also a measure of the error caused by a disturbing torque. The bandwidth is the frequency band in which the output amplitude is at least one-half of the input amplitude. For the servo described in Kef. 1, K_v is in excess of 10,000 sec^{-1} (i.e., (rad/sec)/rad) and BW is approximately 40 cps.

Nonlinear functions can be generated to great accuracy in a mechanical analog computer by the use of two- or three-dimensional cams, or by the use of linkages, The familiar three-dimensional ballistic cam is the heart of many fire control computers. Vector transformations may also be accomplished by assemblages of cams and linkages.

When the equations to be solved by the fire control computer are more complex, it may be necessary to provide additional ser-

* By E. St. George, Jr.

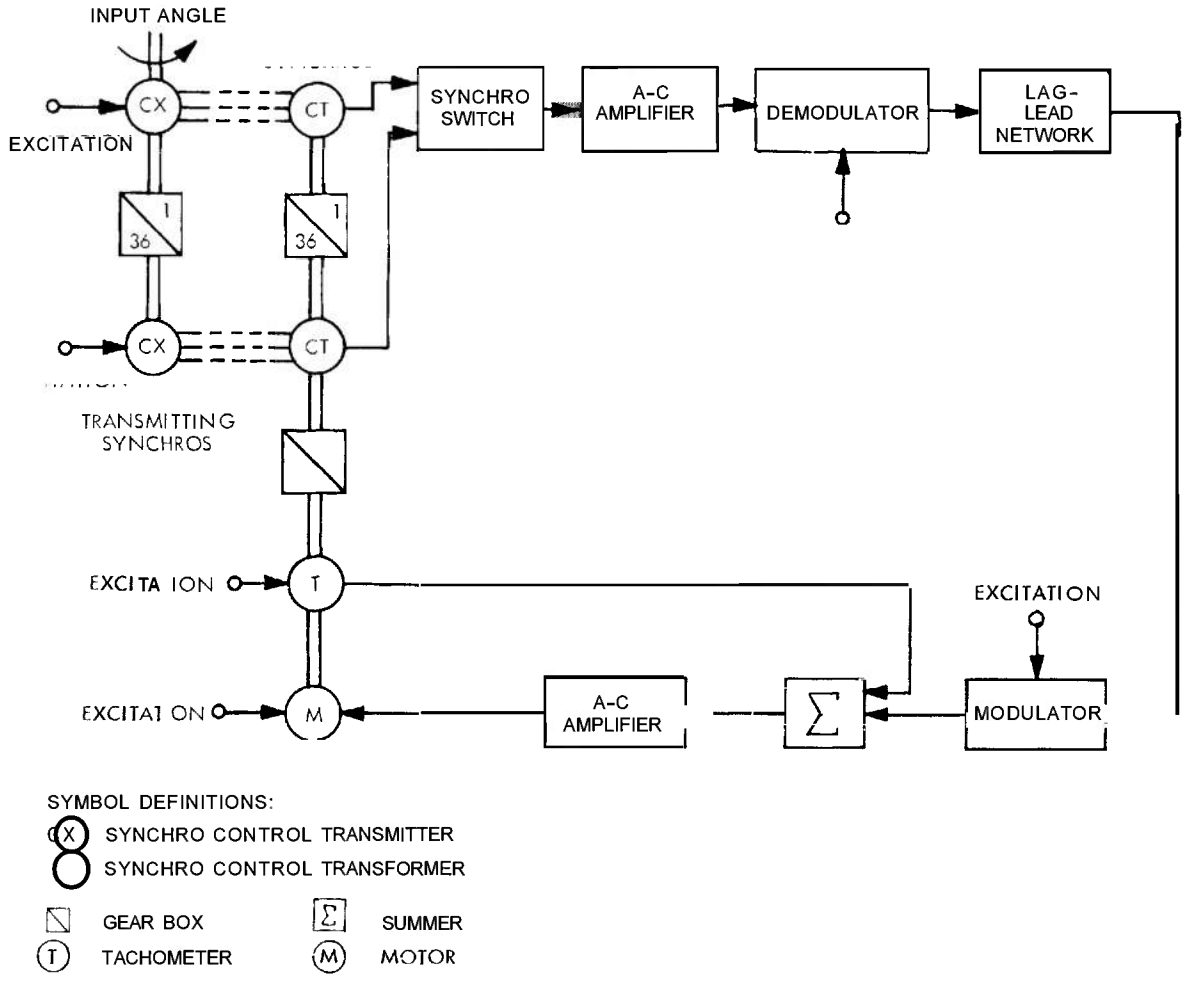


Figure 11-1. Functional diagram of a typical high-performance instrument servo.

vos within the mechanism as torque amplifiers. With this level of complexity, however, maintenance of the multiplicity of gears, bearings, cams, and other rubbing surfaces may become a problem. In these circumstances, the combination of electrical with mechanical computing devices becomes more attractive.

11-1.2 ELECTROMECHANICAL ANALOG COMPUTERS

Electromechanical analog computers are currently the most commonly employed type of computer in fire control applications. Both d-c and a-c (usually 400 cps) signals are employed, and mechanical computing elements are employed wherever their use is advantageous.

In fire control applications, two electromechanical components have proven particularly valuable. The first of these, the induction resolver, is a variable-coupling transformer in which the coefficient of coupling is accurately proportional *to* the sine of the shaft angle. A discussion of the errors in a computing network employing a resolver is included in AMCP 706-327, par. 4-4.18 [9].

The second important component is the a-c drag-cup tachometer. In this type of tachometer, a conductive cup rotates in the field of a primary winding which is excited from a constant a-c source. Eddy currents are induced in the rotating cup with an amplitude proportional to the speed of rotation, and are coupled to a secondary winding so located as to pick up the field of the eddy currents

but not that of the primary. Properly compensated drag-cup tachometers have high accuracy and exeellent resolution.

A precise integrating mechanism can be formed by employing a drag-cup tachometer as the feedback transducer in a rate servo (see Fig. 11-2). In such a rate-servo integrator, an input a-evoltage is compared with the tachometer output, and the resulting crror signal is amplified, demodulated, and passed through a compensating network. Thc modified signal is then modulated and amplified to a power level sufficient to control a two-phase induction motor which is coupled to the tachometer. The angle through which the motor-tachometer shaft has turned then represents thc time integral of the input voltage. The shaft may be coupled to mechanical computing devices or, if an electrical output is desired, transduecrs such as potentiometers, synchros, or resolvers can be geared to the motor, as indicated in Fig. 11-2. For high accuracy, the servo loop gain should be high; this requirement neccssitates that the residual noise in the tachometer be low, that the tachometer rotor bc rigidly coupled to the motor shaft -- preferably on a common shaft -- and that a quadrature-elimination circuit be employed (such as the demodulator-modulator combination in Fig. 11-2).

Detailed treatment of resolvers and drag-cup tachometcrs will bc found in Refs. 2, 3, and 4.

11-1.3 OTHER COMPUTER TYPES

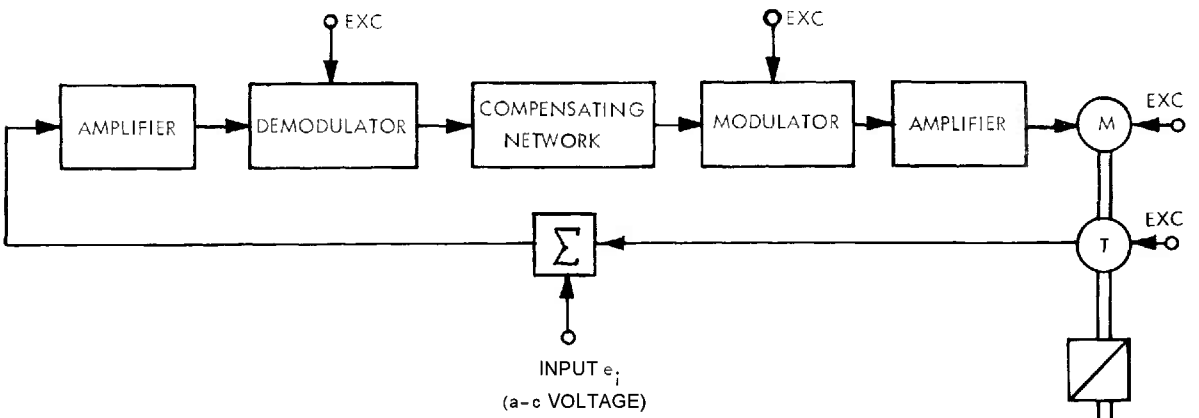
All-electronic analog computers and digital computers have until rccently seen little use in fire control systems. The reasons primarily have to do with time-response considerations, and are diseussed in pars. 11-3 through 11-3.3.

11-2 INPUT-OUTPUT CONSIDERATIONS

11-2.1 SOURCES OF DATA

The general elasses of the sources of data supplied to the fire eontrol computer are:

a. Target-tracking data.
b. Environmental and other semi-fixed data.
c. Command deeisions.
d. Weapon positional data.



POSITION TRANSDUCER	EQUATION SOLVED
POTENTIOMETER	$e_o(t) = K\int_o^t e_i(t)dt$
RESOLVER	$e_o(t) = K \sin\left[\int_o^t e_i(t)dt\right]$
	or $e_o(t) = K \cos\left[\int_o^t e_i(t)dt\right]$

NOTE: THE SYMBOLS USED IN THIS FUNCTIONAL DIAGRAM ARE DEFINED IN FIG, 11-1.

Figure 11-2. Functional diagram of a typical rate-servo integrator.

Target-tracking data are obtained from a radar, infrared, or optical tracking device. The basic operating principles of such devices are similar. The radiation-sensing device (the antenna or its equivalent) is movable in two axes, usually azimuth and elevation. An error signal is generated in each axis either by sequential scanning of a beam pattern displaced from the axis of the antenna, or by simultaneous comparison of four displaced beam patterns. Individual control systems controlled by the error signals position the elevation and azimuth axes so as to track the target. Thus, the position and velocity of the two tracking axes serve to establish the target motion.

Radar and pulsed-coherent-light systems can provide additional information as to the target range and range rate. Range information is available as the time interval between the transmitted and received pulses. If the transmitted pulse is used to gate a clock pulse source into a shift register, and the received pulse gates off the register, the contents of the register is a digital measure of the range. For analog computation, a conventional system employs a precise multivibrator to generate a time base. If the multivibrator is started by the transmitted pulse, the voltage at the time the return pulse is received is a measure of the range. A potentiometer servo can then be employed to track this voltage, giving range as a shaft angle.

Data on temperature, wind direction and velocity, barometric pressure, and other environmental data, including variations with altitude, are required as inputs to the computer. In addition, data as to the relative locations of weapon and tracker and, if the computer is to be used with more than one weapon, data on the muzzle velocity and exterior ballistics must be entered. In general, such data are gathered by human operators and are changed infrequently.

Once a target has been acquired, tracking and computation can continue automatically, and the computer can be designed to initiate firing at the optimum time. Only two command decisions are called for: to acquire and track a new target, and to hold fire. However, in order to facilitate these decisions, information from the tracker and computer must be transmitted to the command post. Such information would vary in different situations, but might include the present position and velocity of a target being tracked, and the predicted future time and position of a hit.

The weapon positional data constitute the computer outputs. The data ordinarily are in the form of the azimuth and elevation angles of the weapon axis.

11-2.2 TRANSMISSION OF DATA

It will be noted that most of the inputs and outputs are to or from a shaft angle. In analog equipment, the data transmission is almost universally by synchro means. The system is as shown in Fig. **11-1.** The one-speed synchro control transmitter (CX) is coupled to the axis of the tracker or other input device. In order to improve the accuracy of transmission, a high-speed control transmitter is coupled to the input axis through a high-precision anti-backlash gear mesh. A 36-to-1 ratio is indicated in the figure; 18:1 and 27:1 are other standard ratios. In the computer, two control transformers (CT) are driven by the instrument servo at corresponding gear ratios. A switchover network is provided in order to transfer the servo error signal from the high-speed to the one-speed synchro whenever the error magnitude exceeds the permissible range of the high-speed synchro. Thus, data are available to the computer as a shaft rotation.

In the case of data transmission to a weapon, a similar scheme is employed; but the CX's are driven by the computer output shafts, the CT's are coupled to the weapon axes, and the instrument servos are replaced by the weapon-pointing servos.

In digital systems, input and output shaft angles can be converted to digital data by one of the methods described in Chapter 8. A very high degree of accuracy, of the order of 0.005 deg, can be obtained with the optical type of shaft encoder whenever it is prepared to accept data, and, depending on the design of the encoder, data transmission may be either serial or parallel.

The angles of the weapon axes can also be converted to digital data by a shaft encoder. If this is done, the functions of the servo amplifier, except for power amplification, are taken by the computer. In some cases this extra load on the computer may not be desirable. If *so*, a special type of converter can

be employed which generates the required a-c voltage levels required to excite control transformers on the weapon. A conventional analog-type servo is then used for weapon pointing.

Many fire control systems employ direct mechanical transmission of data. For example, fire control systems have been designed in which the tracker is mounted on the weapon, but the tracker can be offset from the weapon axis by means of auxiliary servos. The computation is such that the tracking axis is offset by the lead angle from the weapon axis. Other systems incorporate the computer within the tracker, with direct mechanical coupling of tracking information, and synchro transmission of the output data.

When the major subunits of a fire control system are remote from one another, data transmission by telephone lines or radio link may be required. In such cases analog data may be multiplexed, employing commutation of signals, multiple carrier frequencies, or a combination of both. Digital data would be transmitted in serial form. If more than one channel is to be transmitted, multiplexing can be employed as with analog signals.

Command decisions may be transmitted by a voice communication link, or a simple pushbutton control. Environmental data are entered by hand, using dials or counters in analog equipment, and pushbutton banks or a contact-making typewriter in digital systems.

11-3 TIME-RESPONSE CONSIDERATIONS

11-3.1 REAL-TIME COMPUTATION

If the target and the weapon systems have the same, or approximately the same, vector velocity, computation in real time is of little importance. If their velocities differ appreciably, however, then real-time computation is necessary and the problem of minimizing the computation time is mandatory. The allowable computation time will be somewhat influenced by the relative range capabilities of the tracker and weapon. A simple example is shown in Example 11-1.

If the target is capable of performing evasive maneuvers, computation time may be of considerable significance. The following example for a lead-computing sight illustrates this point. For illustrative purposes assume:

1. A target that is capable of a maximum evasive maneuver of acceleration a_m.
2. A lead-angle computer whose target-velocity input $V_t(t)$ for an evasive maneuver beginning at $t = 0$ and ending at time t_1 is

$$V_t(t_1) = V_{to} + \int_0^{t_1} a_m \, dt$$

and whose lead-angle output $\delta_e(t)$ at any time less than t_1 is

$$\delta_e(t) = K \left[V_{to} + a_m t \left(1 - e^{-t/\tau_c}\right) \right]$$

where

V_{to} = the initial target velocity at $t = 0$
K = a scale factor
τ_c = the time constant associated with the time lag of the computer solution

The computer output can be expressed in normalized form as

$$\frac{\delta_e(t)}{K a_m \tau_c} = \frac{V_{to}}{a_m \tau_c} + T\left(1 - e^{-T}\right)$$

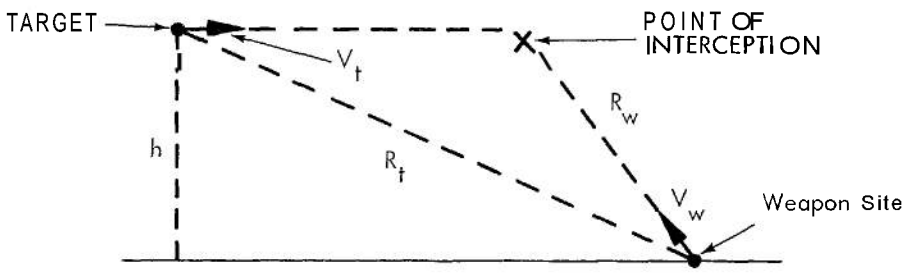
where

$$T = \frac{t}{\tau_c}$$

With V_{to} set equal to zero, this expression is plotted in Fig. 11-3. Note that the percentage error e^{-T} in the computer output reduces to about 0.7% after 5 time constants. As shown by Fig. 11-3, however, if the target were capable of reversing its acceleration within a few time constants, large errors would continue to exist.

Important contributions to the computation time of a fire control system are made by the time lags of the weapon-pointing servos and by the time lags of the tracker servos. In a fire control system which follows the standard block diagram (Fig. 11-4) of cascaded tracker, computer, and weapon-pointer, the time lags of each element are also cascaded, as shown in the figure. However, one or more of these elements may be enclosed in a computing loop. Loops of this sort are encountered in systems such as the weapon-mounted tracker described in par. 11-2.2. In such cases, the time lag is generally reduced be-

Example 11-1. Sample calculation of the maximum allowable computation time.



Given an aircraft target of altitude h and velocity V_t proceeding in a straight line such as to pass directly over a weapon site. If the maximum detectable range is R, and the target must, be hit by a projectile of average velocity V_w before the range closes to a minimum value R,, the maximum allowable computation time t_{cm} is given by

$$t_{cm} = \frac{1}{V_t}\left[\sqrt{R_t^2 - h^2} - \sqrt{R_w^2 - h^2}\right] - \frac{R_w}{V_w}$$

(For the basis of this relationship, see Derivation 11-1.)

For example, let

R_t = 100,000 ft

h = 20,000 ft

V_t = 900 ft/sec

R_w = h (Interception overhead)

V_w = 2000 ft/sec

Then t_{cm} = 99 sec

Thus, only when the maximum detectable range is very short will the maximum allowable computation time be a significant design factor.

cause of the characteristics of the closed loop. The rudimentary system discussed in Example 11-2 illustrates the principles involved, The standard techniques of servo systems analysis' can be employed to determine the time lag of such a feedback system.

It is convenient to express the computer or system time lag in terms of a single parameter. A number of parameters have been employed -- e.g., delay time, rise time, bandwidth, and settling time -- and for simple systems there are easy conversions between them. In the case of fire control computers, settling time is the most useful parameter, and is defined for the response to a step-function input. As shown in Fig. 11-5, the settling time is the time measured from the initiation of the step function to the second intersection of the response with the error tolerance band. The settling time is strongly influenced by the largest system time constant. Note that the system design would normally provide that the second overshoot fall within the error tolerance.

The necessity for rapid response in computing elements when high-speed targets are encountered is the same for either analog or digital computation. However, since the design problems are quite different, the two subsections which follow treat analog and digital computers independently.

Derivation 11-1. Derivation of the Relationship for Calculating the Maximum Allowable Computation Time.

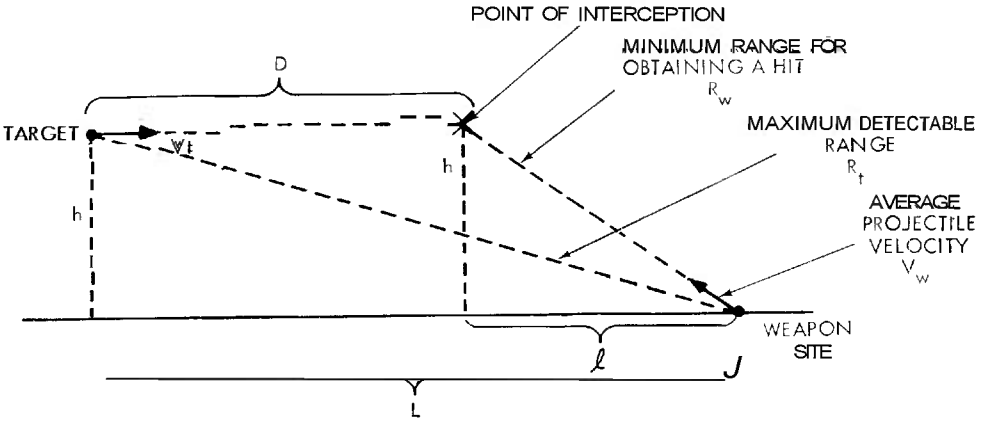


Fig. D11-1.1 Basic Geometry

From the geometry depicted in Fig. D11-1.1 in which an aircraft target of altitude h and velocity V_t is proceeding in a straight line over a weapon site, it is evident that

$$\ell + D = L \qquad \text{(D11-1.1)}$$

$$\ell^2 + h^2 = R_w^2 \qquad \text{(D11-1.2)}$$

and

$$L^2 + h^2 = R_t^2 \qquad \text{(D11-1.3)}$$

Therefore,

$$\ell = (R_w^2 - h^2)^{1/2} \qquad \text{(D11-1.4)}$$

$$L = (R_t^2 - h^2)^{1/2} \qquad \text{(D11-1.5)}$$

Substitution from Eqs. D11-1.4 and D11-1.5 into Eq. D11-1.1 shows that

$$D = (R_t^2 - h^2)^{1/2} - (R_w^2 - h^2)^{1/2} \qquad \text{(D11-1.6)}$$

Fig. D11-1.1 shows also that

$$D = V_t t_{cm} + V_t \left[\frac{R_w}{V_w}\right] \qquad \text{(D11-1.7)}$$

where t_{cm} is the maximum allowable computation time.

Therefore,

$$t_{cm} = \frac{D}{V_t} - \frac{R_w}{V_w} \qquad \text{(D11-1.8)}$$

$$\frac{1}{V_t}\left|(R_t^2 - h^2)^{1/2} - (R_w^2 - h^2)^{1/2} - R_w\right| \qquad \text{(D11-1.9)}$$

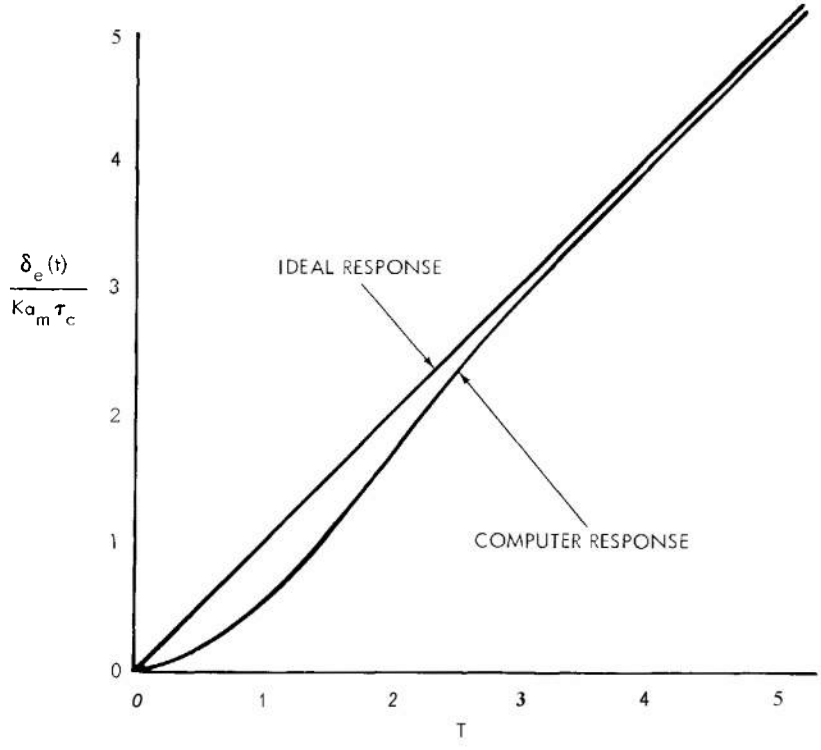
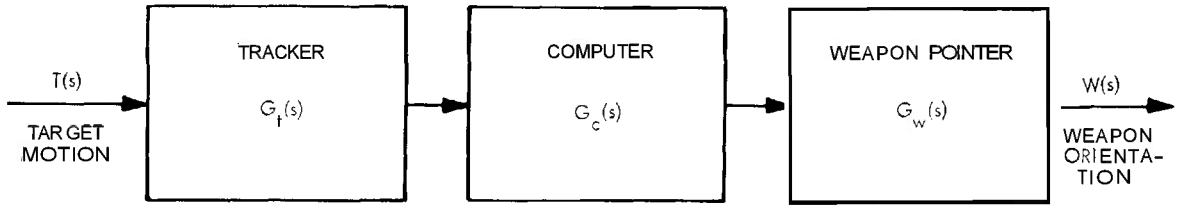


Figure 11-3. Effect of computer time lag with a maneuvering target.



THE SYSTEM RESPONSE CAN BE REPRESENTED IN LAPLACE-TRANSFORM NOTATION BY THE RELATIONSHIP

$$\frac{W(s)}{T(s)} = [G_t(s)]\,[G_c(s)]\,[(G_w(s)]$$

WHERE

s = THE LAPLACE-TRANSFORMER VARIABLE
$G_t(s)$ = THE TRANSFER FUNCTION OF THE TRACKER
$G_c(s)$ = THE TRANSFER FUNCTION OF THE COMPUTER
$G_w(s)$ = THE TRANSFER FUNCTION OF THE WEAPON POINTER

Figure 11-4. The standard configuration of a fire control system.

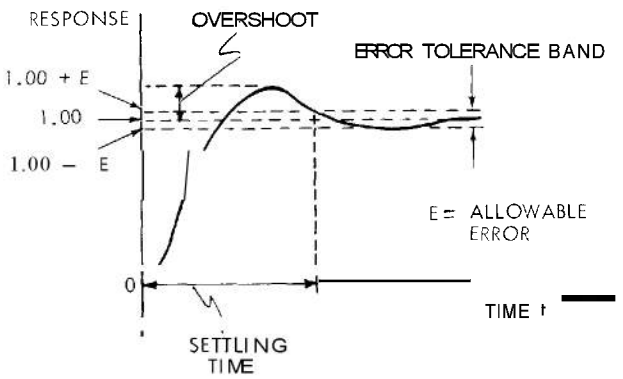


Figure 11-5. Settling time for a typical computer response to a step function.

11-3.2 CONSIDERATIONS ASSOCIATED WITH THE DESIGN OF ANALOG COMPUTERS FOR REAL-TIME OPERATION

Those portions of an analog computer for fire control applications which are wholly electronic are generally of such high-speed response that their time lags need not be considered. However, a different time-response characteristic becomes significant in all-electronic computing elements, that of drift. A fire control system, since it is principally open-loop and may have long computation periods, places extremely difficult requirements on electronic analogs. For this reason, most fire control systems employ electromechanical elements, the response time of which may be quite significant.

In general, the response of an electromechanical computer element is equivalent to the response of the servo that drives it. The cams, linkages, potentiometers, and the like are considered as an inertia and friction load on the driving servo. Par. 11-1.1 gives the block diagram and performance figures for a high-performance instrument servo. Not all instrument servos are of such high performance; however, it is relatively easy to secure a bandwidth of 10 cps and a velocity constant of 2000 sec^{-1} when this performance is required.

The speed of response of input and output elements such as radar-antenna drives and servos will probably be much lower than that of typical instrument servos. In general, the bandwidth of these power servos will be 1/3 (sometimes as much as 1/2) the lowest natural frequency of the structure that the servo drives.

If the input-output equipment is in existence, the servo bandwidth, or equivalent data, will ordinarily be available to the fire control system designer. In the absence of data, the response of the servo to an input sinusoid of varying frequency can be measured; the bandwidth is the frequency at which the output has been reduced to 3 db below the input.

In case a preliminary estimate is desired for developmental equipment, the lowest natural frequency can be calculated. The lowest natural frequency is normally determined by the moment of inertia of the moving part of the structure and by the compliance of the members that drive this load. The inertia is easily estimated by approximating the shape of the load structure with simple geometric forms. Compliance calculations can be quite complicated, but can be simplified by neglecting the less-compliant members. Usually the significant components in such a calculation are the teeth of the final gear mesh, the output shaft and bearings, the support structure for the inertia load, and (in the case of mobile equipment) the carriage and ground anchors. Having determined the inertia and compliance, the lowest structural natural frequency is obtained from the relation

$$\omega_n = \frac{1}{\sqrt{JC}} \qquad (11\text{-}1)$$

where ω_n = lowest structural natural frequency, rad/sec

J = moment of inertia of the moving components, slug-ft^2

and C = compliance of the members that drive the load, rad/ft-lb (compliance is the inverse of the spring constant)

Gyroscopic elements may also contribute time delays. In the case of a free gyro that is precessed by an electromagnetic torque motor, the lag between input signal and output velocity is solely the inductive lag of the torque motor and is usually relatively small. Floated single-degree-of-freedom integrating gyros have similar lags in the torque motor, but the major time lag is mechanical. The response of such a gyro to an input rate is given by

$$\frac{s\theta}{\omega_i} = \frac{H/B}{\frac{J}{B}s + 1} \qquad (11\text{-}2)$$

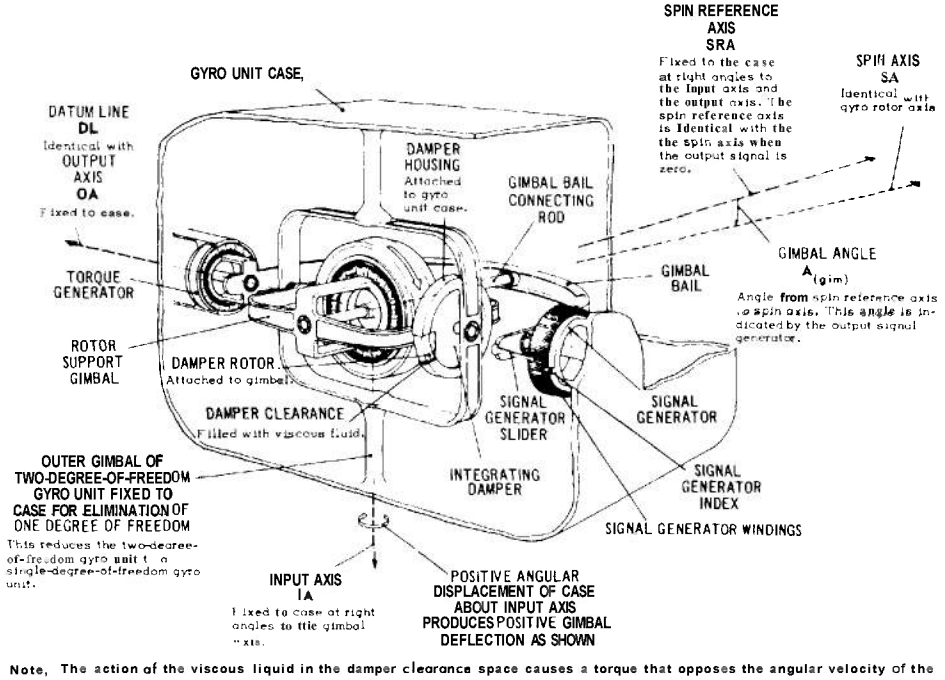
The parameters of Eq. 11-2 are defined below. The axes and components referred to are shown in Fig. 11-6.

θ = angular deflection of the output axis
ω_i = applied rate about the input axis
H = angular momentum of the gyroscopic element about the spin axis
B = rotational damping coefficient of the integrating damper
J = moment of inertia about the output axis
s = Laplace operator

From the form of Eq. 11-2, it is evident that the integrating gyro has a time constant of J/B. Such gyros are, however, customarily employed in a computing loop. A simple loop in which the output of the signal generator is fed back to the torque generator would reduce the time lag in accordance with Example 11-2. More complicated loops are discussed in Ref. 6.

An analog fire control computer is made up of assemblages of components like those previously discussed, arranged in cascaded chains and loops. The method for determining the time response of simple chains and loops has already been discussed. In analyzing more complex networks, it should be recognized that an analog computer is conceptually identical to a servomechanism. The methods developed for the analysis and design of servos can be employed directly in analog computer design. Since these techniques are covered in another handbook[5], only topics of particular interest will be mentioned here.

Servo analytical techniques are based on simple unity-feedback loops. Depending on the system and the results desired, the frequency response or locus-of-roots methods can be employed to determine the response of such simple loops. Many computer loops have non-unity elements in the feedback path. The techniques shown in Fig. 11-7 can be employed to convert a loop having a feedback transfer function H(s)* as in (A) to the unity-feedback form (B). The response $\theta_o(s)/\theta_i(s)$ is



Note, The action of the viscous liquid in the damper clearance space causes a torque that opposes the angular velocity of the gimbal. This torque is proportional to the magnitude of the gimbal angular velocity with respect to the unit case.

Figure 11-6. Essential elements of a single-axis integrating gyro unit[6].

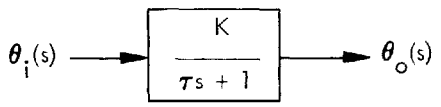
* It should be noted that H(s), the standard symbol for a feedback transfer function, is in no way related to H, the standard symbol for angular momentum.

Example 11-2. The response improvement that can be obtained by means of a closed loop.

A simple lag element of gain K and time constant τ has a transfer function G(s) given by

$$G(s) = \frac{K}{\tau s + 1}$$

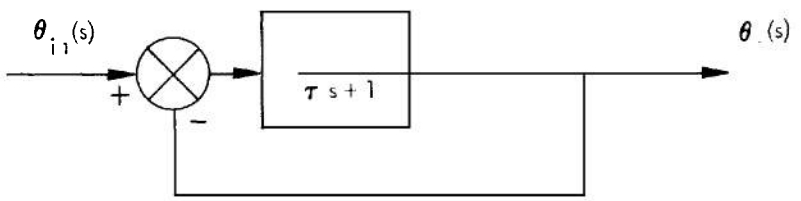
and a block diagram representation:



When enclosed by a unity-feedback loop, the overall transfer function $G_1(s)$ is given by

$$G_1(s) = \frac{G(s)}{1 + G(s)} = \frac{\frac{K}{\tau s + 1}}{1 + \frac{K}{\tau s + 1}} = \frac{K}{1 + \tau s + K} = \frac{K}{1 + K} \cdot \frac{1}{1 + \frac{\tau}{1 + K} s}$$

and the block diagram is



in which $K_1 = K/(1+K)$ and $\tau_1 = \tau/(1+K)$.

Thus, the addition of the closed loop permits the effective time constant to be divided by one plus the closed-loop gain. The amount of time-constant reduction that can be achieved is limited by instability of the loop. Compensation may be required to obtain a high speed of response.

identical for the two configurations. Therefore, the closed-loop response of G(s)H(s) can be determined by either of the methods described, and then modified by cascading the transfer function 1/H(s) (where $H(s) \neq 0$) to find the overall response $\theta_o(s)/\theta_i(s)$.

When the computer includes a minor loop enclosed by a major loop as in Fig. 11-8, the closed-loop response of the minor loop is determined first. In Fig. 11-8 the forward transfer function of the minor loop is $G_1(s)$ and the closed-loop responee of the minor loop is $G_{cl}(s) = G_1(s)/[1 + G_1(s)]$. The remaining forward-loop transfer function of the major loop is $G_2(s)$, and the major loop closed-loop response is given by

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{G_{cl}(s)\, G_2(s)}{1 + G_{cl}(s) G_2(s)}$$

If the frequency-response analysis is employed, the closed-loop response can be obtained readily by plotting $G_1(s)$ on a Nichols chart[5]. The angle and logarithmic magnitude of $G_{cl}(s)$ is read directly from the chart, and the corresponding angles and logarithmic magnitudes of $G_2(s)$ are added to obtain $G_{cl}(s)G_2(s)$. The Nichols chart is then employed again to obtain $\theta_o(s)/\theta_i(s)$. The pro-

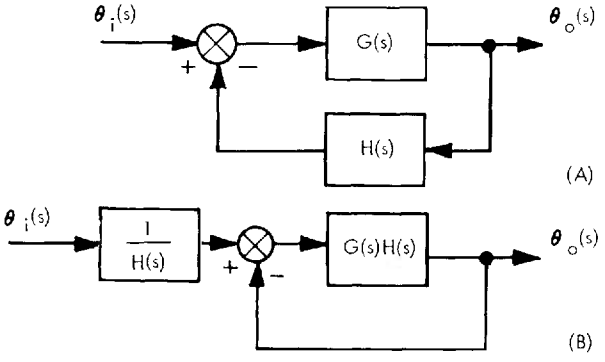


Figure 11-7. Formation of unity-feedback equivalent.

cess may be repeated if $\theta_o(s)/\theta_i(s)$ is enclosed by further loops.

Most analog-computer configurations can be solved by the application of the two techniques just described. However, in the case of very complicated configurations, it may be desirable to systematize the analysis by the use of signal-flow graphs[7], or even to employ simulation techniques[3]. If the problem is such that simulation techniques are justified, the response of the system to realistic, rather than mathematically tractable, inputs can be obtained. Frequently, simulation is employed as a means of determining the response of the system to a variety of signal and noise inputs, and also to examine the effect of parameter changes on the response. Usually, however, the design is first obtained by conventional techniques.

11-3.3 CONSIDERATIONS ASSOCIATED WITH THE DESIGN OF DIGITAL COMPUTERS FOR REAL-TIME OPERATION

Real-time operation imposes obvious requirements on the calculation speed of a digital computer. In addition, the requirement that the computer accept real-time data imposes additional design problems. The input problem will be considered first in this paragraph.

While both synchronous and asynchronous computers exist, the fact that major simplification in circuitry can be achieved in a synchronous computer makes this the preferred design. In a synchronous computer, all operations are controlled by a constant-frequency pulse source, or clock.

Depending on the logical design, a fixed number of clock pulses will be required to perform one computation, The time for one computation is known as the cycle time of the computer.

The computer may not generate output data after each cycle. The solution of differential equations is commonly carried out by an iterative process, *so* that several cycles may be required to generate a solution. The time between successive solutions is a fixed quantity for a given problem, but is set by the programming rather than the logic. This time will be defined as the solution time for the computer. It is a fixed delay, in contrast to

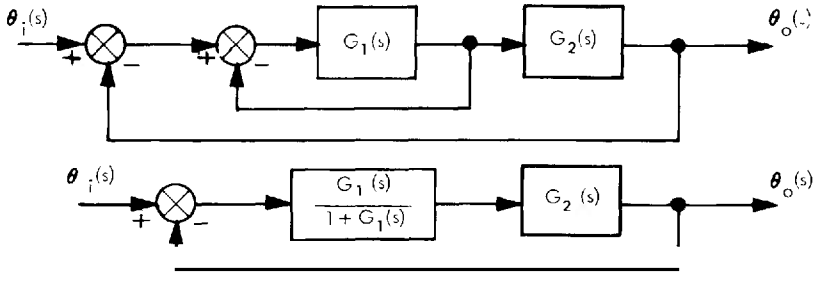


Figure 11-8. Treatment of minor loops.

the time lag -- or settling time -- of an analog computer as illustrated by Fig. 11-9.

The computer can accept input data at only one point in its cycle. The data accepted are normally read into storage and held until called for by the program. Thus, data are actually utilized at only one point in the solution cycle. In general, input data arrive at random intervals, and must be stored until such time as the computer can use the data.

Certain devices may be synchronized with the computer clock; for example, shaft encoders and analog-voltage-to-digital converters. In this case, the storage function is in effect performed by the analog signal.

Devices such as digital tachometers generate pulses at random intervals. Remotely originated data would ordinarily be read in at random times. In such cases, auxiliary storage can be provided to hold the data until they are read out by the computer. An example of a possible circuit for reading radar range information into a digital computer is given in Example 11-3.

11-4 ACCURACY CONSIDERATIONS

11-4.1 GENERAL CONCEPTS

The errors in fire control computers arise from a variety of sources. Some of these sources of error are peculiar to the type of computer chosen, whether analog or digital, while others may be identified with the input and output data, or with the mathematical model.

Errors associated with the mathematical model are by their nature predictable and their effect may be studied in advance of the equipment design by computation or simulation. For example, in Chapter 12, the T29E2 computer employs a sine-cosine approximation (i.e., the initial terms in a Fourier expansion) as a model of the ballistic trajectory. The effect of this approximation is completely predictable in advance, and rnay be studied for the knownranges of the variables. If required by the system accuracy specifications, the accuracy of the model could be imprcved; in this case, by adding terms in the Fourier series.

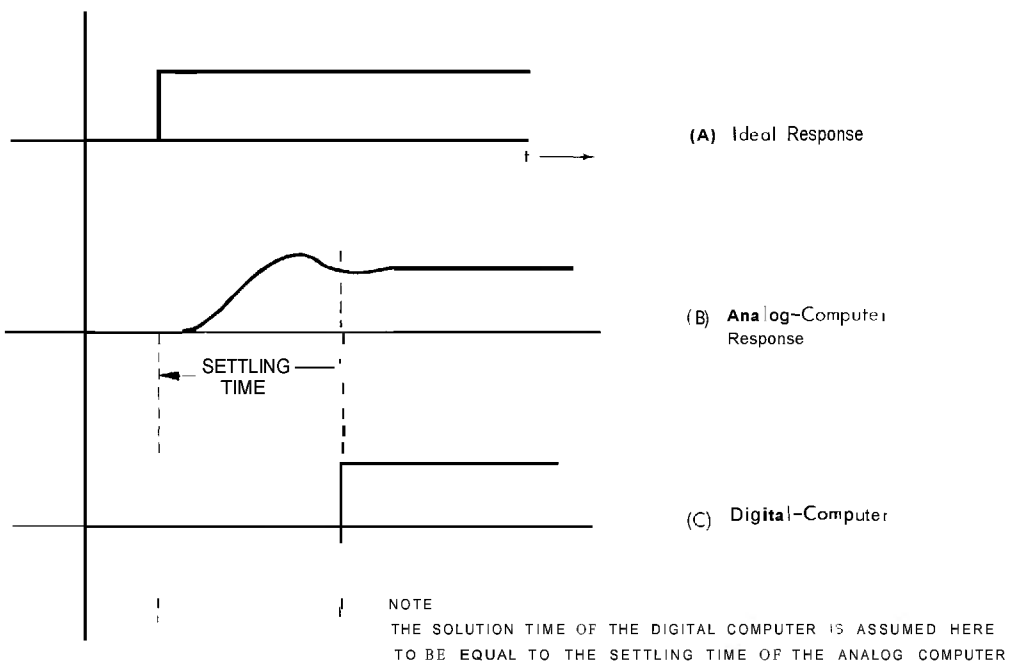


Figure 11-9. Comparison of the response characteristics of analog and digital computers.

Example 11-3. Radar range converter

A simple means of converting the range measurement of a radar system to digital form is indicated in Fig. E11-3.1. The radar pulse generator initiates bursts of microwave energy at regular intervals of a few microseconds. Each successive generated pulse and a small fraction of the transmitted pulse are fed to the range converter.

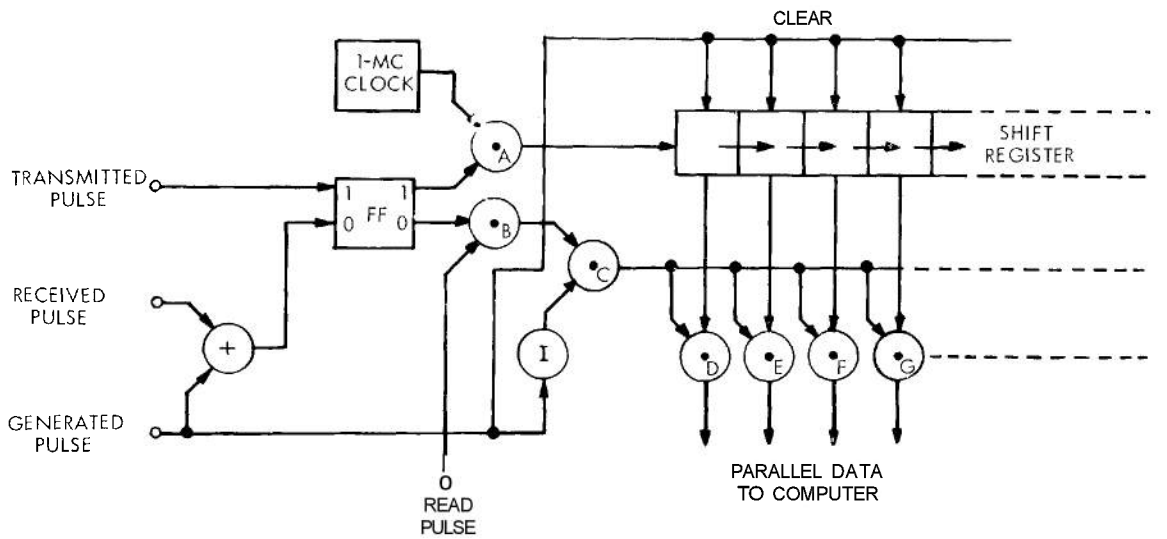


Figure E11-3.1. Logic diagram for a radar range converter.

After a delay determined by the path length (twice the range), the pulse returned from the target is detected by the receiver. This received pulse is also fed to the range converter. A period of 2500 microseconds is assumed between successive transmitted pulses, permitting ranges of over 200 miles.

The transmitted pulse sets the flip-flop, which is initially in the 0 state. The flip-flop 1 output gates pulses from a 1-mcps clock pulse generator through gate A into a shift register. The shift register has 12 bits in order to accommodate a maximum of 2500 clock pulses. Only four bits are shown in the figure.

The flip-flop is reset by the received pulse, thus shutting off the flow of clock pulses to the shift register. The shift register then holds a binary number which is a measure of the range. Since the received and transmitted pulses may occur at any time between successive clock pulses without changing the count, the inherent resolution of such a system is ±1 count. The resolution may be improved by raising the clock frequency.

Readout is accomplished at the instigation of a read pulse from the computer. The read pulse is generated once each cycle; in this case, 40 microseconds. To prevent an attempt to read out during the range measurement, the flip-flop must be in the 0 state in order to open gate B. Readout will occur not later than 40 microseconds after arrival of the received pulse. Note that readout is in parallel; if serial readout had been desired, equal delays could be inserted in the read-pulse line between gates D and E, E and F, etc., and the parallel outputs combined on a single line with OR gates.

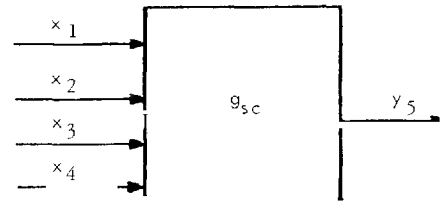
Example 11-3. (Continued)

The next shift register is cleared by the next generated pulse. This pulse also resets the flip-flop and, to prevent a false output, inhibits the readout through gate C. Since the generated pulse occurs just before the transmitted pulse, the circuit is ready to receive a new input. If no return pulse is received, the flip-flop will remain in the 1 state until the clearing pulse; thus, no readout is possible.

The generalized representation of a fire control system is a network of component units, each having multiple inputs and a single output. The output is functionally dependent on all the inputs. (This concept was introduced in Chapter 4 of Ref. 9.)

Of course, in the general case any function might be represented by the component unit. However, a typical computer would have one or more components which have a summing function. This configuration (see Fig. 11-10) provides a simple example of the techniques employed in adjusting a mathematical model to meet accuracy specifications with a minimum of equipment. If one starts with as nearly exact a model as can be devised, the first step is to determine the ranges of the variables x_1, x_2, etc. If it turns out that any of these variables are always less than the allowable error, these inputs can be immediately discarded. Other inputs may have a larger range, but still may be relatively small compared to the largest inputs. In such cases, linearization, replacement of the variable by a constant, or other approximations may be applicable. A word of caution is necessary here, however. If the maximum value of the smaller variable is small compared to the minimum value of the larger variable, under all conditions, then we may apply approximation techniques with complete safety. If, however, the two variables can approach each other under some conditions, the approximation may not be valid. If the joint probability of each combination of the two variables can be determined, the most probable error due to a given approximation can be obtained.

The choice of the input variables and the instrumentation with which to measure them may have an effect on the system errors. For example, if radar tracking is employed, range can be measured with very much greater accuracy than the accuracy of the pointing



$$y_5 = x_1 + x_2 + x_3 + x_4$$

OR, IN PERFORMANCE-OPERATOR NOTATION (SEE PAR. 4-4.4.2 OF REF. 9),

$$y_5 = g_{sc}(x_1, x_2, x_3, x_4)$$

WHERE g_{sc} = PERFORMANCE OPERATOR C THE SUMMING COMPONENT

AND

$$g_{sc}(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3 + x_4$$

Figure 11-10. Functional diagram of a summing Component.

angles. If optical or infrared tracking is employed, the opposite is true; in fact, range can only be roughly estimated. The computer design can be tailored to maximize the accuracy with the type of tracking employed by placing maximum dependence on the most accurate inputs.

The method by which the input variable is measured can also be of importance. If, for example, range is measured by a pulsed radar, then target scintillation, atmospheric effects, and multipath effects can introduce jitter, or noise, in the range measurement. Improved performance can often be attained by the use of a Doppler system in which the Doppler frequency is proportional to range rate. The continuous nature of the Doppler information averages out much of the noise. Sudden large fluctuations, such as might result from scintillation, can be more easily filtered from a rate than from a positional measurement. This technique of measuring

rates (either angular or range) is of general usefulness in fire control systems, Since the lead angle is proportional to the tracking rate, this rate must be computed if it is not measured directly. Computation of the rate from tracking position data requires differentiation of the position data. However, differentiation has the inherent property of increasing the noise which is present in the signal. Thus, techniques which directly generate a tracking rate have a considerable advantage. Such a system is the Vigilante tracking gyro described in Chapter 12. In this system, a feedback loop closed through a human operator generates the tracking rate by means of precessiontorques applied to a free gyro.

Noisy input data can be a major source of errors in fire control computers. Noise may be generated by nonidealities in tracking servos, by a human operator, or by propagation effects and target modulation in radar and infrared systems. The spectral distribution is of great significance. The high-frequency noise components are not usually significant since they are filtered by the computer and by the weapon drives. Careful design is usually necessary to smooth the input signals within the system pass-band. In an analog system, smoothing is accomplished by properly setting the bandwidth of the instrument servos or other electromechanical components *so* as to give the required filtering action. In a digital system, the desired filter transfer function is programmed into the computer.

A balance must be struck in the design of the filter networks between noise reduction and excessive time lag in the system. Fortunately, a fair degree of noise can be tolerated in the output. Since the systematic errors are most commonly predominant, a moderate amplitude of the noise error may enhance the engagement hit probability. The effect of increasing noise error is to enlarge the volume within which a given percentage of the bursts will probably fall. This volume is centered about apoint in space which is displaced from the target center by a distance determined by the systematic errors. As the noise increases, this volume can enclose the target, thus increasing the hit probability for an engagement with a sufficient number of shots *so* that the statistics are applicable. Further discussion of the relation of noise and systematic errors can be found in Chapter 4 of Ref, 9.

A final consideration in the design of computers which must respond to noisy inputs is the dynamic range. Filter elements must be so located in the computer that the noise amplitude cannot saturate a component, and prevent the transmission of the signal. Thus, filtering in the input sections of the computer reduces the dynamic range required of the follow-on components in a chain. Digital computers are subject to the same problem. In this case, the designer must make adjustments in his program rather than in the arrangements of components.

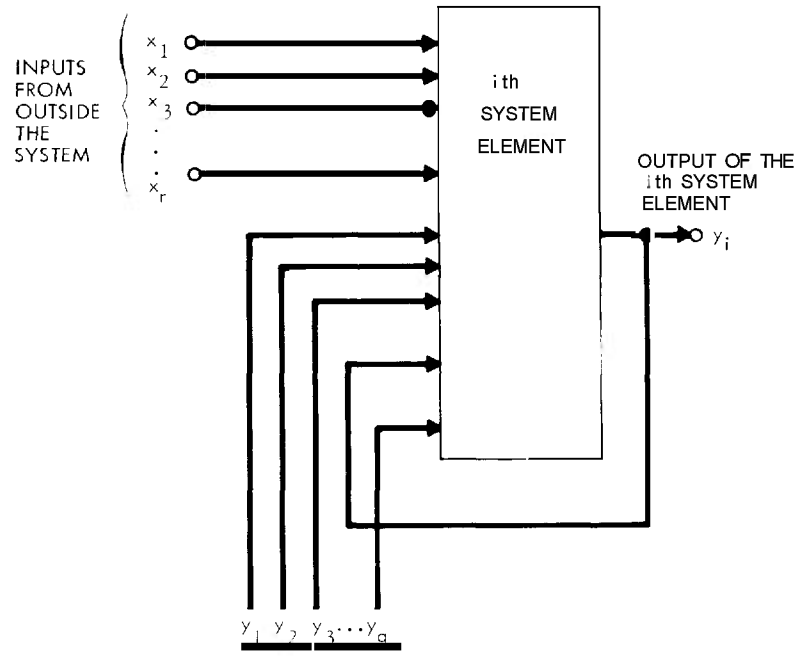
11-4.2 THE ACCURACY OF SOLUTIONS OBTAINED FROM ANALOG COMPUTERS

The discussion of the sources and propagation of errors in fire control systems, which was presented in Chapter 4 of Ref. 9, is directly applicable to analog computers. This discussion will be briefly reviewed in this paragraph. An analog computer can be considered to be made up of a combination of elements, each of which has the general form shown in Fig. 11-11. Each such element is defined as having one output (which, for the ith element, is y_i, as shown in the figure) and a number of inputs. Some of these inputs come from outside the computer, and are designated $x_1, x_2, \ldots, x_r$, while others come from other elements of the computer, and are designated $y_1, y_2, \ldots, y_r$. One of these latter inputs may be a signal fed back from the output of the element considered, i.e., y_i. The output is functionally dependent on all of the inputs, i.e.,

$$y_i = g_i (x_1, \ldots, x_r, y_1, \ldots, y_i, \ldots, y_q) \quad (11\text{-}3)$$

where

y_i = output of the ith element
g_i = performance operator of the ith element
$x_1, \ldots, x_r$ = r inputs from outside the computer



INPUTS FROM VARIOUS ELEMENTS OF THE SYSTEM, INCLUDING THE ith SYSTEM ELEMENT.

$$y_i = g_i(x_1, \ldots, x_r; y_1, \ldots, y_i, \ldots, y_q)$$

WHERE

g_i = PERFORMANCE OPERATOR OF THE ith SYSTEM ELEMENT

Figure 11-11. Functional diagram of a typical system element.

and

$y_1, \ldots, y_q$ = outputs of the q computer elements

In Chapter 4 of Ref. 9, an expression for the errors in such a system was derived. It consists of a set of q equations, one for each of the q elements. The error equation for the ith element is

$$\sum_{k=1} \frac{\partial f_i}{\partial y_k} \epsilon_{y_k} \qquad \sum_{n=1} \frac{\partial f_i}{\partial x} \epsilon_{x_n} + m_i \frac{\partial f_i}{\partial y_i} \qquad (11\text{-}4)$$

where f_i is a simplified notation for $f_i(x_1, \ldots, x_r, y_1, \ldots, y_i, \ldots, y_q)$, and is the implicit form of Eq. 11-3. That is (see Fig. 4-22 of Ref. 9),

$$f_i(x_1, \ldots, x_r, y_1, \ldots, y_i, \ldots, y_q) = y_i - g_i(x_1, \ldots, x_r, y_1, \ldots, y_i, \ldots, y_q) = 0 \qquad (11\text{-}5)$$

The other quantities in Eq. 11-4 are defined as follows:

ϵ_{y_k} = error in output y_k

ϵ_{x_n} = error in input x

m_i = error in the output of the ith element by virtue of its being nonideal

The partial derivatives are evaluated at some set of values of the inputs and outputs that satisfies the performance equation of the element concerned.

Eq. 11-4 is valid for static errors, i.e., for performance operators which do not include differential equations. Where differential equations are included and the dynamic errors are desired, it is convenient to employ an analysis in the frequency domain. In Chapter 4, Ref. 9, it was shown that the power spectral density of the output error for a

single-input, single-output computer element (as shown in Fig. 11-12) is given by

$$\Phi_{\epsilon_{yy}}(j\omega) = \left|R(j\omega)\right|^2 \Phi_{\epsilon_{xx}}(j\omega) \qquad (11\text{-}6)$$

where

$\Phi_{\epsilon_{yy}}(j\omega)$ = power spectral density of the error in the element output

$\Phi_{\epsilon_{xx}}(j\omega)$ = power spectral density of the error in the element input

$R(j\omega)$ = transfer function of the element, as defined by Fig. 11-12

If the input error is random with variance $\sigma^2_{\epsilon_x}$, Eq. 4-166 of Ref. 9 shows that its power spectral density is given by

$$\Phi_{\epsilon xx}(j\omega) = 2\pi \frac{d}{d\omega}\left\{\sigma^2_{\epsilon_x}(j\omega)\right\} \qquad (11\text{-}7)$$

Then the variance, $\sigma^2_{\epsilon_Y}$, of the output error is

$$\sigma^2_{\epsilon_Y} = \int_0^\infty \left|R(j\omega)\right|^2 \frac{d}{d\omega}\left\{\sigma^2_{\epsilon x}(j\omega)\right\} d\omega \qquad (11\text{-}8)$$

(Note that $\sigma^2_{\epsilon_Y}$ is independent of the variable of integration ω.) Eq. 11-8 can be expanded to the case of a computer having q elements and p inputs. For this computer system, a set of error equations can be written, one equation for each element. The equation for ith element is

$$\sigma^2_{\epsilon_{y_i}} = \sum_{n=1}^{p} \int_0^\infty \left|R_{i_n}(j\omega)\right|^2 \frac{d}{d\omega}\left\{\sigma^2_{\epsilon_{x_n}}(j\omega)\right\} d\omega$$

$$\sum_{k=1} \int_0 \left|R_{i_k}(j\omega)\right|^2 \frac{d}{d\omega}\left\{\sigma^2_{\epsilon_{y_k}}(j\omega)\right\} d\omega + \sigma^2_{m_i} \qquad (11\text{-}9)$$

where

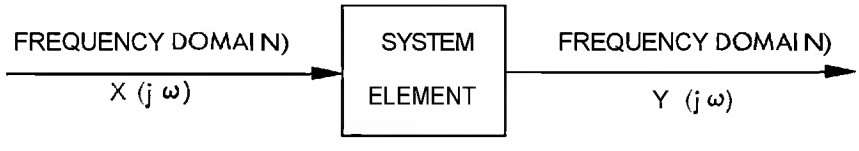
$\sigma^2_{\epsilon_{y_i}}$ = the variance of the error ϵ_{y_i} that is associated with the output y_i

$R_{i_n}(j\omega)$ = the transfer function that is associated with the ith element, and is measured between the input x_n and the output y_j

$\sigma^2_{E_{x_n}}(j\omega)$ = variance of the errore, that is associated with the" input x_n

$\sigma^2_{\epsilon_{y_k}}(j\omega)$ = variance of the error E_{y_k} that is associated with the output y_k that is also an input to the ith element

$\sigma^2_{m_i}$ = variance of the error of the ith element due to the element being nonideal



$$Y(j\omega) = R(j\omega)\, X(j\omega)$$

where

$X(j\omega)$ = Fourier transform of $x(t)$

$Y(jw)$ = Fourier transform of $y(t)$

$R(j\omega)$ = transfer function of the system element

Figure 11-12. Functional representation of a system element in the frequency domain.

If the superposition principle can be applied, the analysis of the computer errors is greatly simplified; fortunately, superposition does, in general, apply, However, care must be exercised in any error analysis to ensure that errors do not force the system into a nonlinear region. For example, a large noise error at the input to an amplifier may drive it into saturation, i.e., into a region of lower than normal gain. A signal introduced into the input will then be subjected to an error in amplification, and this error will be a function of the presence or absence of the noise error. Thus, for the example cited, superposition would not apply. Of course, the conditions of the example would be most disadvantageous for any computer, It is, in general, true that in a good, low-error computer design superposition will apply. In such a computer design, the errors can be separated into various classes and analyzed individually. The total error in any output is then found by superposing the total errors of each class. In fire control computers, it is convenient to separate the errors into systematic bias) and random (noise) components. The bias errors, in turn, are subdivided into static and dynamic errors. The static errors can be analyzed by the use of Eq. 11-4. This set of equations can be solved for the outputs of the computer (usually only two or three in number). The dynamic bias errors are best analyzed in the frequency domain. Eq. 13-9 can then be simplified for these errors. If the input error is $\epsilon_x(t) = \epsilon_x \sin(\omega_1 t)$, then it is possible -- from the definition of the response function $R(j\omega)$ -- to write the relationship

$$\epsilon_y(j\omega_1) = R(j\omega_1)\ \epsilon_x(j\omega_1) \qquad (11\text{-}10)$$

where $R(j\omega_1)$ is the value of $R(j\omega)$ at $\omega = \omega_1$. Ey. 11-9 then becomes

$$\sum^{p} R_i\ (j\omega_1)\ \epsilon_x\ (j\omega_1)$$

$$+ \sum^{a} R_{i_L}\ (j\omega_1)\ \epsilon_{y_k}\ (j\omega_1) + m_i(j\omega \qquad 11\text{-}11)$$

where $m_i(j\omega_1)$ is the error of the ith element due to the element being nonideal.

The noise errors are analyzed by direct application of Eq. 11-9.

The total error in an output variable can be determined by performing a root-square sum of the components. Usually, however, the bias and noise are retained as separate components.

Typical static errors in an analog computer are the linearity errors of potentiometers, null and transformation errors of resolvers, drift and nonlinearity of amplifiers, and zero offset errors in a variety of components. Dynamic bias errors in analog computers are principally associated with the bandwidth limitations of instrument servos and gyros, as discussed in par. 11-3.2. Noise errors are usually smaller than bias errors. The largest magnitude of noise is usually associated with the tracking device which provides one or more inputs to the computer. Other sources include potentiometer contact noise, gear backlash and tooth errors, and pickup of stray voltages (usually due to improper grounding procedures).

In computer design, a few points should be kept in mind to minimize the output error. Component ranges should be matched as closely as possible to the ranges of the input variables. The subtraction of two quantities which are of approximately equal magnitude should be avoided,

Noise sources within the computer should be avoided if at all possible. For minimum transmission of noise, servo and gyro bandwidths should be minimized. However, the same measures which reduce noise errors will increase dynamic bias errors. Therefore, a compromise bandwidth must be determined to give the best overall system performance. In some cases an adaptive system, in which the bandwidth is aatomatically (or manually) adjusted to conform to the noise amplitude, may be necessary.

In Chapter 4 of Ref. 9, a number of examples are given of error analysis in both simple and complex analog computer systems. Reference should be made to that chapter for further details.

11-4.3 THE ACCURACY OF SOLUTIONS OBTAINED FROM DIGITAL COMPUTERS

In theory, one can obtain from a digital computer accuracies as good as desired. In practice, of course, for a computer of given

computation speed and storage capacity, there must be a trade-off between speed and accuracy. In the case of a fire control computer, the requirements for solution time are often quite fixed if the computer must operate in real time. Therefore, the required solution time and the required accuracy specify the computer design within quite close limits.

The accuracy of a digital computer is usually specified in terms of the number of significant figures that are carried in the computation. In programming the computer, great care must be exercised to ensure that significant figures are not lost. In order to ensure that as many significant figures are carried in the case of a small number as are carried for large numbers, floating-point arithmetic is employed. In floating-point arithmetic, all numbers are less than one, and the placement of the decimal point is handled by associating with the number the proper power of ten. Thus 103950. would be written 0.10395×10^6, and 103.95 as 0.10395×10^3. This scheme preserves a constant percentage error in all quantities.

In digital addition, one more significant figure is sometimes produced in the sum than was present in the two numbers added. In order to prevent the registers from overflowing, the least significant digit must be dropped, or rounded off. The usual rules for round-off are as follows:

(a) Drop the least significant digit if it is less than 0.5.

(b) Add one to the next-most-significant digit if the least-significant digit is greater than 0.5.

(c) If the least-significant digit is exactly 0.5, add 1 to an odd next-most-significant digit, but add zero to an even next-most-significant digit.

This procedure ensures that, on the average, the round-off errors are balanced between high and low values.

It can be shown that the relative error (i.e., the actual error divided by the true value of the quantity) of an approximate number is never greater than a quantity determined by the number of significant figures. The expression is

$$\Delta \leq \frac{1}{2 \times 10^{n-1}} \qquad (11\text{-}12)$$

for a decimal number, where Δ is the relative error, and n is the number of significant figures. For a binary number,

$$\Delta \leq \frac{1}{2 \times 2^{n-1}} \leq \frac{1}{2^n} \qquad (11\text{-}13)$$

We have seen that in addition there is no loss of significant figures and there may be a gain of one; in subtraction, however, there is no general rule. If the numbers subtracted are nearly equal, there may be a complete loss of significance in the difference. For this reason, the ranges of variables to be subtracted must be very carefully observed. It is often desirable to rearrange the program *so* that subtractions are avoided. In the case of multiplication and division, it is possible to lose up to two significant figures. This is a general rule of thumb. In any particular instance, the loss might be one,. two, or no significant figures.

More complex operations are made up of combinations of addition, subtraction, multiplication, and division. The round-off errors in all operations can therefore be analyzed by breaking them down into combinations of simple operations. There are many pitfalls, however, and a successful computer program requires the judicious combination of analysis and experimentation with input numbers for which output numbers are known.

The solution of differential equations on a digital computer gives rise to another source of error. A number of solution methods based on the use of series expansions or on iterative procedures have been devised. The greater the number of terms retained in the expansion, or the greater the number of iterations, the greater will be the accuracy. The errors associated with these solution methods are therefore called truncation errors. Because of the wide variety of solution methods, it is difficult to make any further general statements. Par. 2-4 of Chapter 2 gives a description of several solution methods, and examples of the truncation errors.

A second source of truncation errors is in the representation of functions in a digital computer. In order to avoid excessive storage of functional data, interpolation by means of a power polynomial is commonly employed.

Interpolation formulas of the Lagrangian, Newtonian, or other forms may be employed, as described in par. 2-4.1. For these standard forms a remainder term has been derived, so that the truncation error for a given number of terms is readily calculable. In fact, it is sometimes advantageous to program the calculation of the remainder after each term; the computer then makes a decision whether or not to continue the series according to the magnitude of the error.

Very few errors in digital computation can be ascribed to equipment malfunction. A part of this error-free operation is inherent in the nature of digital computation; i.e., since all data is carried in two-state devices, deterioration of a device must be severe before it gives an improper indication. Also, computing components are commonly designed to fail in an indicative way. For example, a computer might employ +6 volts to represent 1 and -6 volts to represent 0, *so* that a power or device failure which would give 0 volts could be immediately discriminated against.

A further check against equipment failures is tlie provision of error-detecting codes. Coding theory demonstrates that an error in a single bit of a code can be detected by the provision of one extra (redundant) bit in the code. A two-bit error can be detected by two redundant bits, and *so* on. Additional redundant bits permit the detection of the particular bit in error, which allows the computer to correct its own errors. Normally, however, detection of an error stops the computation, and the incorrect word is read out as an aid in repair.

The simplest error-detecting code is the parity bit. If the code for each alphanumeric character is made up of four bits, we may add a fifth bit for error detection. The number of ones in the four-hit code is determined: if it is even, tile fifth bit is made zero; if odd, the fifth bit is made one. Thus, in the five-bit code the total number of ones is always even, and an odd number is an indication of an error in one bit. Most computers employ a fairly elaborate system of error-detecting codes and systems.

11-5 OPERATIONAL CONSIDERATIONS

The problems of computer design for field operation are not different in kind from those of normal service, except for the need for portability and for protection against the environment. However, all these problems are greatly magnified in degree. Field power sources, for example, vary much more widely in both frequency and voltage, necessitating high-quality regulating circuits.

The problem of maintaining complex electronic and electromechanical systems has led to the universal use of modular construction. In the field, the potential lack of trained personnel and the difficulty of supply of spare parts make a modular construction absolutely essential. A computer designed for field use must have means incorporated for detecting a defective module, which do not require appreciable technical knowledge on the part of the operator. It must be possible to replace a module easily with a minimum of disassembly. In order to minimize the supply problem, a minimum number of different types of modules should be employed.

Usually, the most difficult service problem is the location of the defective module. In digital computers, built-in error-detection schemes of the type described in par. 11-4.3 are usually provided. Also, marginal checking procedures are useful. In a marginal checking system, the computer is operated close io its tolerance limits under controlled conditions, in an attempt to induce detectable errors in the "weak link" coniponents, i.e., those that have deteriorated in performance so that they are more likely to cause errors.

In an analog computer, error tracing is more difficult. A check problem for which the signal amplitudes at the outputs and at important points within the computer are known can be helpful. However, some technical knowledge is required of the service personnel in order to locate such troubles as (for example) a noisy region in a potentiom eter which gives otherwise satisfactory performance,

REFERENCES

1. AMCP 706-139, Engineering Design Handbook, Servomechanisms, Section 4, Power Elements and System Design, par. 16-8.

2. J. E. Gibson, Control Systems Components, McGraw-Hill, New York, 1958.

3. S. A. Davis and B. K. Ledgerwood, Electromechanical Components for Servomechanisms, McGraw-Hill, New York, 1961.

4. AMCP 706-137, Engineering Design Handbook, Servomechanisms, Section 2, Measurement and Signal Converters, pars. 11-3.31 and 11-5.2.

5. AMCP 706-136, Engineering Design Handbook, Servomechanisms, Section 1, Theory.

6. See Ref. 4, par. 11-6.

7. J. G. Truxal, Control System Synthesis, McGraw-Hill, New York, 1955.

8. W.W. Seifert and C.W. Steeg, Jr., Control Systems Engineering, McGraw-Hill, New York, 1960.

9. AMCP 706-327, Engineering Design Handbook, Fire Control Series, Section 1, Fire Control Systems -- General.

CHAPTER 12
EXAMPLES OF MEANS USED TO MEET PARTICULAR TYPES OF DESIGN PROBLEMS

12-1 INTRODUCTION

This chapter employs discussions of actual fire control equipment to illustrate how particular types of design problems can be successfully met through the use of ingenuity. The five following items of fire control equipment are considered and illustrate solutions of design problems as indicated:

a. The Gun Data Computer T29E2, discussed in par. 12-2. The design principles and techniques employed for this computer provide an excellent example of the application of design ingenuity to the problem of performing a complex mathematical operation by means of a simple, compact, lightweight portable computer.

b. Lightweight Fire Control Equipment for Rocket Launchers, discussed in par. 12-3. The design approach employed for this fire control equipment exemplifies a situation where standard components can be advantageously used to a large degree, thereby simplifying supply and maintenance operations as well as shortening the time period required to arrive at a satisfactory design.

c. The Vigilante Computer Gyro/Platform System, discussed in par. 12-4. This system represents an excellent example of how design ingenuity can be employed to achieve one item of fire control equipment that can perform several functions--in this case, the generation of tracking rates, a complex coordinate transformation, and stabilization.

d. The Mark 20 Gyro Computing Sight, discussed in par. 12-5. This sight represents the realization of a relatively simple real-time computer for use against moving targets by a weapon station that is itself in motion. Its design principles, which have been extensively and successfully employed by the U.S. Air Force and the U.S. Navy, are currently being given consideration by the U.S. Army.

e. The Cant-Correction System of Ballistic Computer XM17, discussed in par. 12-6. This cant-correction system illustrates the modification and approximation of the mathematical model of a computer in such a way that the model will be simplified while still maintaining the accuracy required of the computer.

12-2 GUN DATA COMPUTER T29E2

The Gun Data Computer T29E2 computes firing parameters for the 105 mm and 155 mm howitzers from input data obtained from observer sightings of the target. The computer is a compact, portable, electromechanical analog device. Its design required the exercise of ingenuity in keeping the computer simple, and therefore light and compact.

The computer performs two functions. First, it determines the location of the target with respect to the gun battery; this is essentially a coordinate transformation operation. Secondly, it computes the ballistic trajectory in order to determine the coordinates of the gun line, which are required to hit the target.

The basic inputs to the computer are the target position in rectangular coordinates. Since these coordinates may not be known directly, three input conditions are provided:

(1) Condition A - The target coordinates are known (this is the basic condition).

(2) Condition B - An observer located at a point the coordinates of which are known measures the range, azimuth, and vertical angles of the target.

(3) Condition C - An observer with his compass at an unknown position, but with a known reference point in view, determines the distances from the target to the reference along his sight line and perpendicular to this sight line, and also the difference in height between the target and the reference point. The coordinates involved are depicted in Fig. 13-1 of Chapter 13.

From the input data--in either Condition A, B, or C--the target-location section of the Gun Data Computer T29E2 computes the north and east rectangular coordinates of the target with respect to the gun, and enters these two coordinates--termed, respectively, "Target Northing" (ANI) and "Target Easting" (ΔEI)--in the gun-order section, as indicated in Fig. 12-1. After addition of the battery-parallax corrections,* the resultant distances, AN and ΔE, are resolved in order to determine the gun azimuth and the target range.

In the gun-order computer, a wind-correction term and the base-point azimuth are summed with the target azimuth to generate

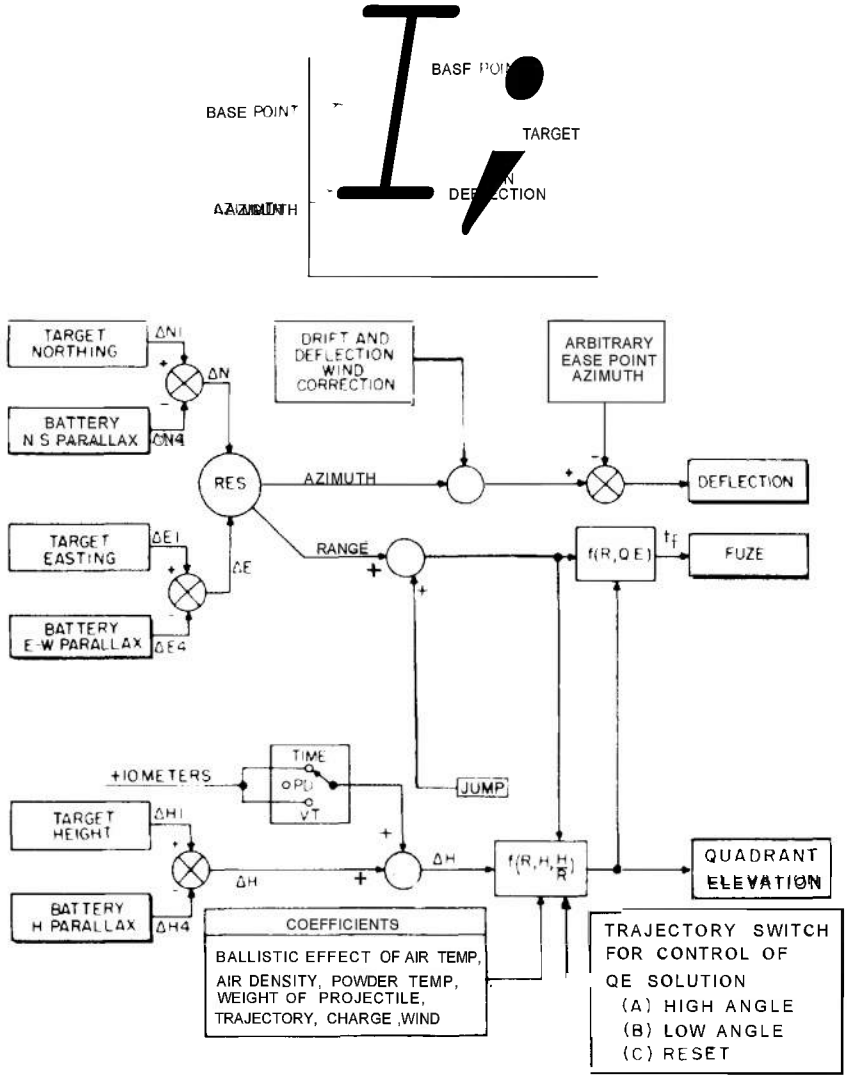


Figure 12-1. Simplified functional diagram depicting the generation of gun-order computations by the T29E2 computer.

* See Ref. 4 for a discussion of parallax and means employed for its compensation.

the gun deflection which is a computer output.

The range R and target height H are used to compute the gun-elevation angle, denoted quadrant elevation QE. Specifically, the quadrant elevation depends upon R, H, and $\frac{H}{R}$ as shown by the subsequent discussion.

The final computation shown in Fig. 12-1 is the determination of the time of flight t_f which can be expressed as a function of range and quadrant elevation (see Fig. 12-2). Time of flight is employed to set the fuzes of time-fuzed projectiles.

Since the computation of quadrant elevation requires the determination of the ballistic trajectory, considerable ingenuity is required to devise a mechanization that is simple and compact. As a starting point, the equations of the projectile trajectory in a vacuum have been derived in Fig. 12-2. The final expression in Fig. 12-2 gives the range of the vacuum trajectory as a function of the variables $\frac{H}{R}$, QE, and muzzle velocity MV. Since It and H are measured and QE is to be determined, a feedback type of computation is employed. In general, two solutions for QE will be obtained, one low-angle (between 200 and 500 mils) and one high-angle. (between 900 and 1150 mils). A trajectory switch *(see* Fig. 12-1) is provided to select either high or low angle. In cases for which two high-angle solutions exist, the trajectory switch can be turned to "Reset" which injects a signal transferring the solution to the higher of the two trajectories.

The real (nonvacuum) trajectory is affected by air density, air temperature arid range' wind (i.e., wind in the direction of the range vector), as well as by changes in muzzle velocity and other effects. Variations in the weight of the projectile and in propellant temperature appear as changes in muzzle velocity. Since weight variations in the projectile are accompanied by changes in the ballistic coefficient, weight changes are also introduced as equivalent changes in air density.

Since the ballistic corrections are empirical for the most part, they would normally be introduced by a series expansion of the independent variables, the ballistic factors being introduced as the coefficients of the expansion. The form of the equation of the vacuum trajectory (see Fig. 12-2) suggests a Fourier series expansion, and this was in fact employed.

The basic form of the Fourier series is

$$f(z) = \sum_{p=1}^{\infty} a_p \sin(pz) + \sum_{p=0}^{\infty} b_p \cos(pz) \quad (12\text{-}1)$$

The coefficients, in the case of empirical data, may be evaluated by dividing the empirical function into r equal segments. If the value of the independent variable is z_q at the qth point, and the corresponding value of the function is f_4, the coefficients of Eq. 12-1 are

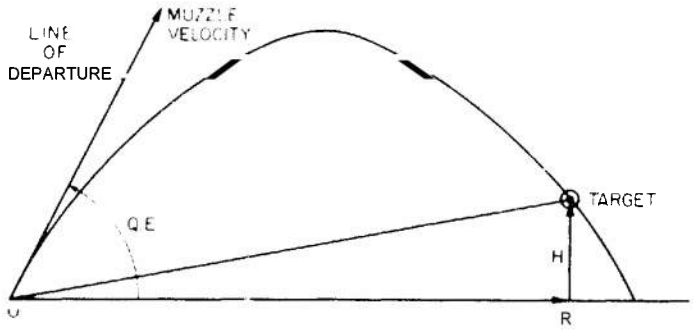
$$\left.\begin{aligned} a_p &= \frac{2}{r} \sum_{q=1}^{r} f_q \sin(pz_q) \\ b_p &= \frac{2}{r} \sum_{q=1}^{r} f_q \cos(pz_q) \\ b_o &= \frac{1}{r} \sum_{q=1}^{r} f_q \end{aligned}\right\} \quad (12\text{-}2)$$

where r is the number of equal segments chosen.

In the approximation employed in the T29E2 Computer, only the zero- arid first-order terms were employed, and z was set equal to $k\theta$, where k is approximately 2 and is the quadrant elevation QE. Then Eqs. 12-1 and 12-2 are

$$\left.\begin{aligned} f(k\theta) &= a_1 \sin k\theta + b_o + b_1 \cos k\theta \\ a_1 &= \frac{2}{r} \sum_{q=1}^{r} f_q \sin z_q \\ b_1 &= \frac{2}{r} \sum_{q=1}^{r} f_q \cos z_q \\ b_o &= \frac{1}{r} \sum_{q=1}^{r} f_q \end{aligned}\right\}$$

The trajectory equation expressed as a group of truncated Fourier series then becomes



IN A VACUUM:

$R = (MV \cos QE)t_f$ ---------- (1)

$H = (MV \sin QE)t_f - \frac{1}{2}gt_f^2$ -------- (2)

FROM (1)

$t_f = \frac{R}{MV \cos QE}$

SUBSTITUTE IN (2)

$H = MV \sin QE \frac{R}{MV \cos QE} - \frac{1}{2}g\frac{R^2}{(MV)^2 \cos^2 QE}$

$\frac{H}{R} = \frac{(MV)^2 \sin QE \cos QE - \frac{g}{2}R}{(MV)^2 \cos^2 QE}$

OR

$\frac{g}{2}R = -\frac{H}{R}\left[(MV)^2 \cos^2 QE\right] + (MV)^2 \sin QE \cos QE$

BUT

$\sin QE \cos QE = \frac{1}{2}\sin 2QE$

AND

$\cos^2 QE = \frac{1}{2} + \frac{1}{2}\cos 2QE$

THEREFORE,

$\frac{g}{2}R = -\frac{H}{R}\left[(MV)^2\left(\frac{1}{2} + \frac{1}{2}\cos 2QE\right)\right] + \frac{(MV)^2}{2}\sin 2QE$

$$\boxed{R = -\frac{H}{R}\left[\frac{(MV)^2}{g} + \frac{(MV)^2}{g}\cos 2QE\right] + \frac{(MV)^2}{g}\sin 2QE}$$

Figure 12-3. Equations of projectile trajectory in a vacuum.

$$R(1 + c\sin k\theta + d\cos k\theta) + ac\sin k\theta + ad\cos k\theta + a\sin k\theta + b\cos k\theta - \frac{H}{R}(f\sin k\theta + g\cos k\theta) - \frac{H}{R}j(1 + c\sin k\theta + d\cos k\theta) + H(m\sin k\theta + n\cos k\theta) + Hp(1 + c\sin k\theta + d\cos k\theta \qquad (12\text{-}3)$$

where θ is the quadrant elevation QE; k is approximately 2; a, b, and j are functions of air density, air temperature, range wind, and muzzle velocity; c, d, α, f, g, m, n, and p are constants; R is the ground range; and H is the height of the target above the origin. All of the coefficients in Eq. 12-4 are adjustable for powder charge, and for high or low trajectory.

Eq. 12-4 can be mechanized by the extremely simple electromechanical computer shown in Fig. 12-3. The resolver in the figure is driven through a gear ratio k from the quadrant elevation shaft. The output voltage e_o from the resolver is given by the expression

$$e_o = (\alpha - \frac{Hf}{R} + Hm - e_o c)\sin k\theta + (b - \frac{Hg}{R} + Hn - e_o d)\cos k\theta \qquad (12\text{-}5)$$

Solving for e_o gives

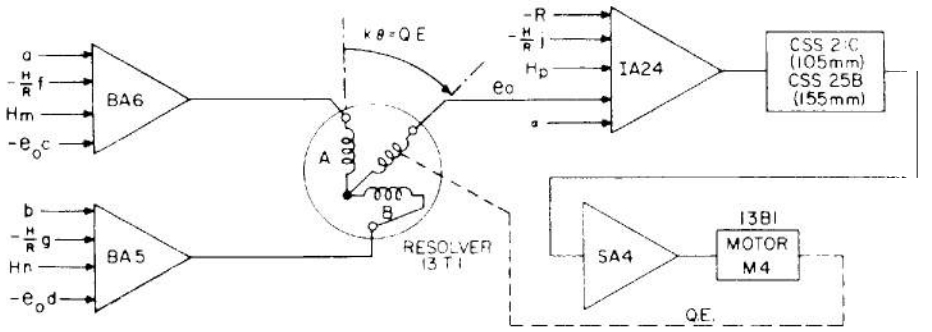


Figure 12-3. Schematic diagram of the quadrant-elevation loop of the T29E2 Computer.

$$e_o = \frac{a \sin k\theta + b \cos k\theta}{D} - \frac{H}{R}\left[\frac{f \sin k\theta + g \cos k\theta}{D}\right] + H\left[\frac{m \sin k\theta + n \cos k\theta}{D}\right] \qquad (12\text{-}6)$$

where $D = 1 + c \sin k\theta + d \cos k\theta$. When the servo reduces its error signal to zero,

$$e_o + Hp + a - \frac{H}{R}j - R = 0 \qquad (12\text{-}7)$$

Combining Eqs. 12-6 and 12-7 shows that

$$R = a + \frac{a \sin k\theta + b \cos k\theta}{D} - \frac{H}{R}\left[\frac{f \sin k\theta + g \cos k\theta}{D} + j\right] + H\left[\frac{m \sin k\theta + n \cos k\theta}{D} + p\right]$$

which can be converted to Eq. 12-4 by multiplying both sides by D.

Thus this expression, which appears to be complex mathematically, was mechanized very simply. The T29E2 Computer design as a whole is an excellent example of design ingenuity applied to the problem of performing a complex mathematical operation by means of a simple, portable computer.

12-3 LIGHTWEIGHT FIRE CONTROL EQUIPMENT FOR ROCKET LAUNCHERS'

The lightweight fire control equipment for rocket launchers described here was developed to provide a simple means of pointing rocket launchers to a desired azimuth and elevation angle. The device is modified from a standard telescope and mount.

Rocket launchers are rotatable about an azimuth axis perpendicular to the base, and about an elevation axis perpendicular to the azimuth axis. Since the base may not be accurately leveled, the firing azimuth and elevation data, which are determined for a horizontal reference plane, must be corrected for mislevel, or cant, of the mount.

The computation is acconiplished by a telescope mounted on a mechanical gimbal system, with the gimbal angles measuring crosslevel (cant), elevation, and azimuth—reading from the launcher in toward the telescope. The cant-correction gimbal is equipped with a level bubble in the crosslevel vial (see Fig. 12-4).

The elevation gimbal is supplied with a similar level vial. Leveling of these two axes gives a level reference plane, from which the given elevation data can be measured by means of the elevation scale and micrometer (or vernier) (see Fig. 12-4). An azimuth base point is established by emplacing a pole of known azimuth hearing. The launcher azimuth is then measured by malting readings with the azimuth scale and micrometer at the pole and at the launcher boresight.

The system of axes of the launcher and of the fire control equipment is shown in Fig. 12-5. In addition to the elements already described, the telescope incorporates a reticle with vertical and horizontal cross-hairs, arid a reticle lighting device. The azimuth and elevation scales can be slipped for zero-setting. A scale) is also provided on the

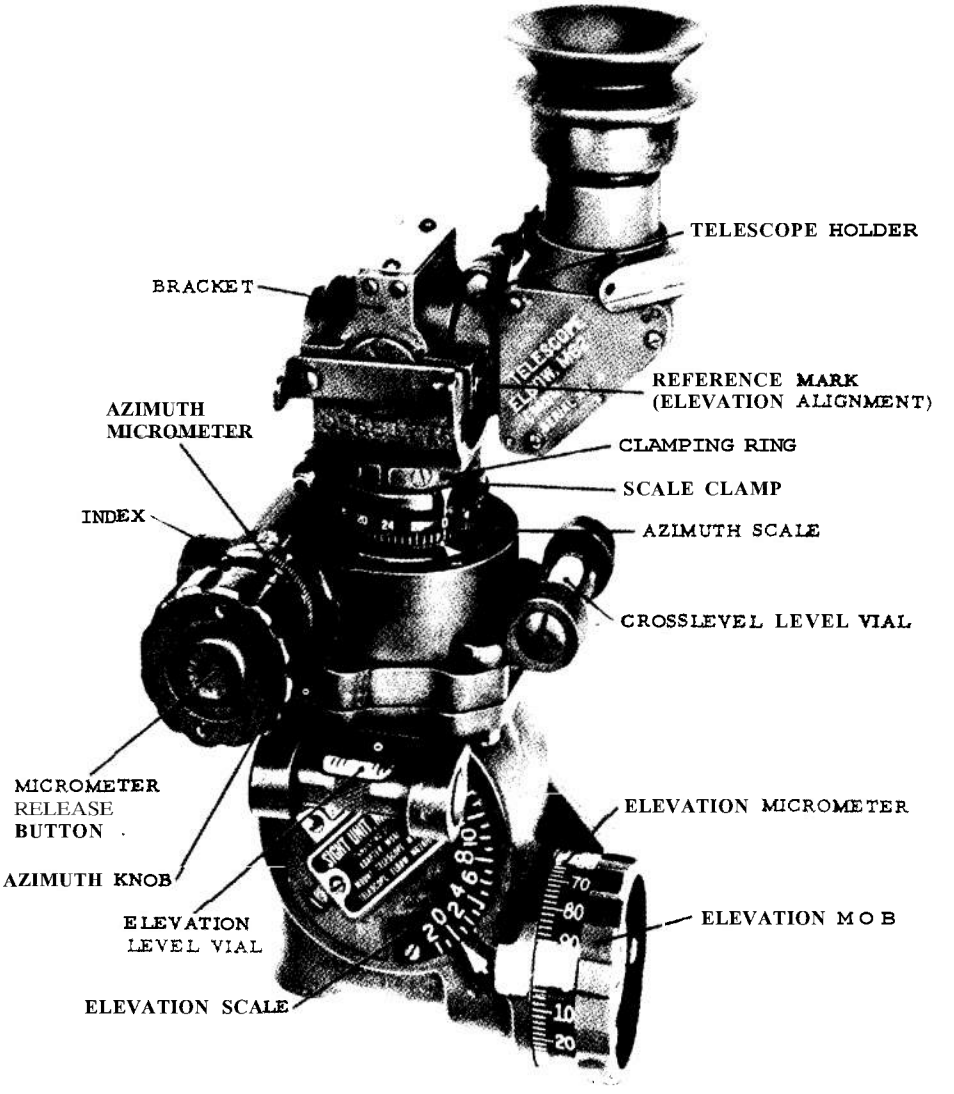


Figure 12-4. Sight Unit M34A2; left rear view.

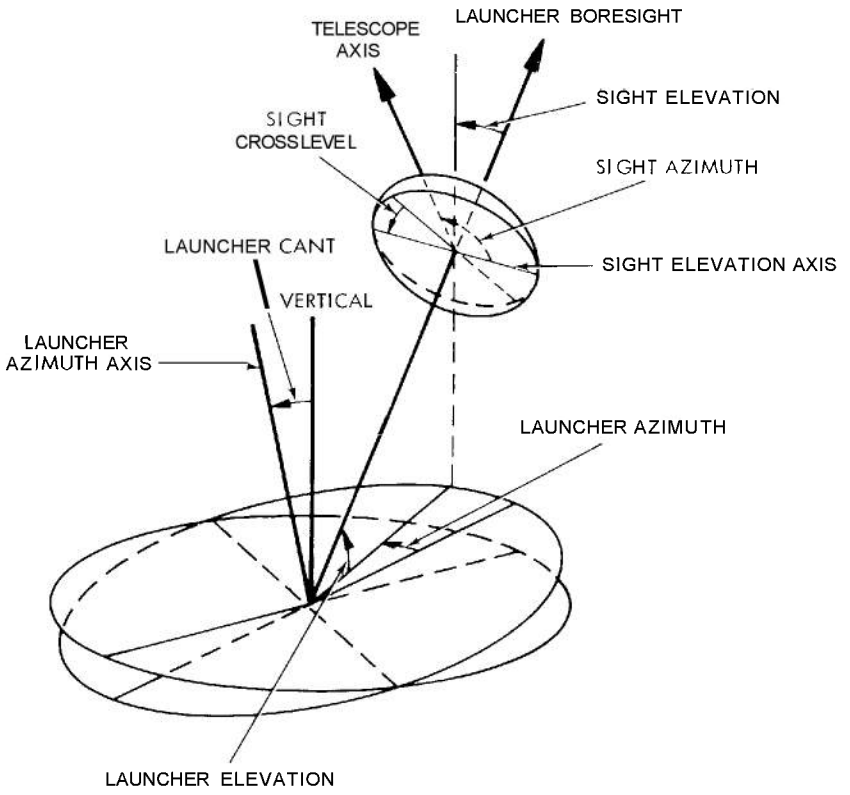


Figure 12-5. Launcher and sight axes systems.

crosslevel axis.

In operation, the launcher is first boresighted on a distant object. The telescope is then sighted on the same object, and the azimuth arid elevation scales are zeroed. The crosslevel axis is zeroed, arid the desired azimuth deflection from the base point is set in on the azimuth scale. The launcher is then rotated in azimuth to bring the base point in line with the telescope; the crosslevel should then be readjusted if necessary, and the launcher azimuth corrected. The desired elevation angle is then set in on the elevation scale, and the launcher is then elevated until the elevation level bubble is centered. Again, a crosslevel correction should be made, if necessary.

The rocket launcher fire control equipment described herein exemplifies the rugged, simple equipment which is of particular usefulness in forward-area combat. The design of this equipment is particularly notable in that it makes great use of standard components, thus simplifying supply and maintenance operations.

12-4 VIGILANTE COMPUTER GYRO/ PLATFORM SYSTEM

In any fire control system which is employed against moving targets, the tracker and the section of the computer which is concerned with tracking are required to define the vector velocity of the target in some convenient coordinate system. The information available is generally the azimuth arid elevation angles of a gimballed tracker, and the range' to the target. The computation is greatly simplified by a separation of the velocity vector into two components: one which lies along the line of sight and is therefore the range rate, and one which is normal to the line of sight. The computation then divides into:

(1) A prediction of the time of flight, based on range and range-rate data and also on ballistic parameters.
(2) A prediction of future position, obtained by extrapolation from the present position by the velocity normal to the line of sight.

The velocity normal to the line of sight can be determined by the measurement of three angular velocities. Two of these are immediately available as the azimuth and elevation velocities of the tracking mount. The third angular velocity is the rotational velocity of the tracking mount about the line of sight. It can be measured directly by a gyroscope, or it can be obtained indirectly by geometrical relationships from the azimuth and elevation velocities.

In the fire control system used withthe Vigilante Antiaircraft Weapon System,* tracking is accomplished optically, with a separate radar range measurement. The normal method of computation would be to measure the azimuth and elevation angular velocities by means of tachometers on the tracking mount,** smooth this data, and perform the necessary vector resolutions by means of resolvers and instrument servos. The Vigilante system substitutes a gyroscopic tracking system, which in a single device provides for most of the complex vector resolutions, and gives a number of secondary benefits as well.

Previous employment of gyroscopes in fire control systems has been principally in ship-, air- and tank-borne applications in which the gyros, in addition to tracking functions, provide a stable platform reference for the system. The mobility of the Vigilante weapon has required some sacrifice in stability of the mount, as compared with that of a fixed-emplacement gun. Referencing of the tracking lint. to a gyroscope greatly reduces the disturbances introduced by gun-reaction forces. The primary advantage of the gyroscopic system is, however, the simplification of computation.

The Vigilante fire control system employs three gyros. One is a vertical reference with a pendulous element. This gyro eliminates the need for accurate leveling of the mount, and is of no further concern to the present discussion. The second, or tracking, gyro is a two-axis free gyro, equipped with pickoffs and torquers on both axes, and is mounted on a two-axis platform. The platform servos receive their error signals from the gyro pickoffs, and cause the platform to remain closely perpendicular to the gyro spin axis. The third gyro is a single-degree-of-freedom rate gyro, and is mounted on the same platform, oriented so as to measure the angular rate of the platform normal vector. This measured angular rate of the platform normal vector can be transformed into the angular rate of the tracking-gyro spin axis by means of a coordinate transformation that makes use of the error signals fed to the platform servos. Since these error signals are small, the vector transformations can be approximated by simple. linear operations.

In Chapter 4, par. 4-6 of AMCP 706-327[7], the tracking equations were derived in the orthogonal coordinate system $\vec{1a}$, $\vec{2a}$, $\vec{3a}$, where 2a is a unit vector- directed along the tracking line, i.e., the spin axis of the tracking gyro. If D_s is the range to the tracking point, the velocity of this point is

$$\frac{d}{dt}\vec{D}_s = \frac{d}{dt}(D_s\vec{2a}) = \dot{D}_s\vec{2a} + D_s\dot{\vec{2a}} \qquad (12\text{-}8)$$

Eq. 12-8 is identical with Eq. 4-269 of Ref. 7. The scalar velocity $\dot{D}_s$ is the range rate along the tracking line $\vec{2a}$, while $D_s\dot{\vec{2a}}$ is the vector velocity normal to 2a. From Eq. 4-271 of Ref. 7, this velocity is

$$D_s\dot{\vec{2a}} = (S_1E_{1a} + D_s\nu_{1a})\vec{1a} + (S_1E_{3a} + D_s\nu_{3a})\vec{3c} \qquad (12\text{-}9)$$

where, as defined in Ref. 7,

S_1 = a constant whose value is chosen so that the tracking system will have a

* See Chapter 4, par. 4-6 of AMCP 706-327 for a full description of this system

** An alternative approach would be to match the operator's rate command signals against the tachometers, and use either the command or response, modified by tlic tracking-servo error, as the velocity data.

response time compatible with the requirements of a human operator and an optimum compromise between fast settling and noise attenuation

E_{1a} and E_{3a} = the tracking-error components along the 1a and 33 axes, respectively

$v_{1a} = V_{1a}/D_s$, $v_{3a} = V_{3a}/D_s$ where V_{1a} and V_{3a} are the components of the smooth target velocity $\vec{V}_s$ along the $\vec{1a}$ and $\vec{3a}$ axes, respectively

But,

$$\vec{2a} = \omega_{3a}\vec{1a} - \omega_{1a}\vec{3a} \qquad (12\text{-}10)$$

(Eq. 4-275 of Ref. 7)

where ω_{1a} and ω_{3a} are the $\vec{1a}$ and $\vec{3a}$ components, respectively, of the angular velocity $\vec{\omega}_a$ with which the gyro coordinate system rotates with respect to an inertial coordinate system. Division of Eq. 12-9 by D_s and comparison with Eq. 12-10 shows that

$$\left.\begin{aligned} \omega_{1a} &= \frac{S_1 E_{3a}}{D_s} \\ \text{and} \\ \omega_{3a} &= \frac{S_1 E_{1a}}{D_s} + v_{1a} \end{aligned}\right\} \qquad (12\text{-}11)$$

In accordance with basic gyroscope principles (see Fig. 2-12 of Kef. 7 for example), a torque applied about an axis perpendicular to the spin axis induces a resultant angular velocity, or precession, about an axis that is mutually perpendicular to the torque and spin axes. The magnitude of tlic velocity is equal to HT, where T is the applied torque and H is the spin momentum of the gyro.

As shown in Fig. 12-6, Eqs. 12-11 are solved by two position servos which are compliance-coupled to tlie gyro gimbals. The vortical servo exerts a torque on the lateral gimbal; the vertical gimbal then precesses at a rate ω_{1a} proportional to this torque. The lateral servo produces a similar rate at the lateral gimbal.

As shown in Fig. 12-7, each precession servo is mounted on a gimballed platform and exerts a torque on the gyro gimbal through a rack, cam, force spring, and linkage. The angles between the gyro gimbals and the platform are measured by electrical pickoffs, one of which is shown in Fig. 12-7. The pickoff outputs, designated δ_t and δ_e, provide the error signals that drive the platform servos so as to follow the tracking line. Since the platform closely follows the gyro, the need for bails is eliminated, but the torque is applied to the gimbal about a platform axis rather than a gyro axis. The transformation relationships given by Eqs. 4-296 of Ref. 7 show that

$$\left.\begin{aligned} T_{1a} &= T_{1p} - T_{2p}\,\delta_t \\ T_{2a} &= T_{1p}\,\delta_t + T_{2p} + T_{3p}\,\delta_e \\ T_{3a} &= -T_{2p}\,\delta_e + T_{3p} \end{aligned}\right\} \qquad (12\text{-}12)$$

where the T's are torques about axes designated by the subscripts.

There are no torques applied about the $\vec{2p}$ axis; hence, $T_{1a} \cong T_{1p}$ and $T_{3a} \cong T_{3p}$. The torque error about the $\vec{2a}$ axis produces no precession effect.

The right-hand sides of Eqs. 12-11 may now be rewritten in platform coordinates, yielding

$$\left.\begin{aligned} \omega_{1a} &= -\frac{S_1 E_{3p}}{D_s} - v_{3p} \\ \omega_{3a} &= \frac{S_1 E_{1p}}{D_s} + v_{1p} \end{aligned}\right\} \qquad (12\text{-}13)$$

These quantities expressed in platform coordinates are the ones available in the computer.

A further complication in the platform azimuth servo is the provision of a means of introducing the lead angle. This is necessary because the platform is mounted in the turret, which provides the azimuth angle for the gun. Therefore, the platform azimuth servo must remove the turret motion as well as the gyro motion.

As mentioned previously, the angular rate ω_{2a} about the tracking line must also be determined in order to complete the definition of the vector velocity of the tracking point.

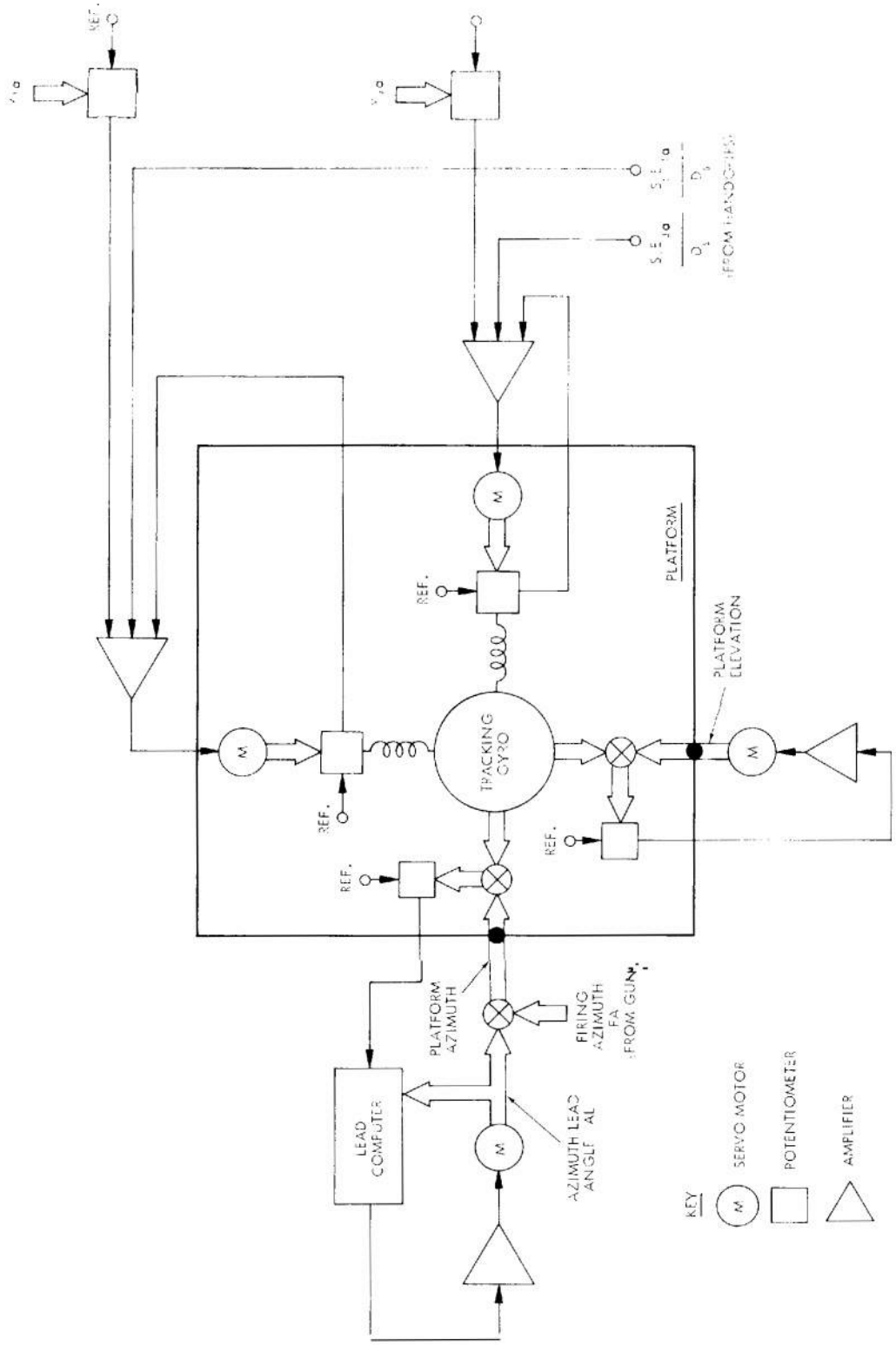


Figure 12-6. Functional diagram of the tracking-gyro/platform system.

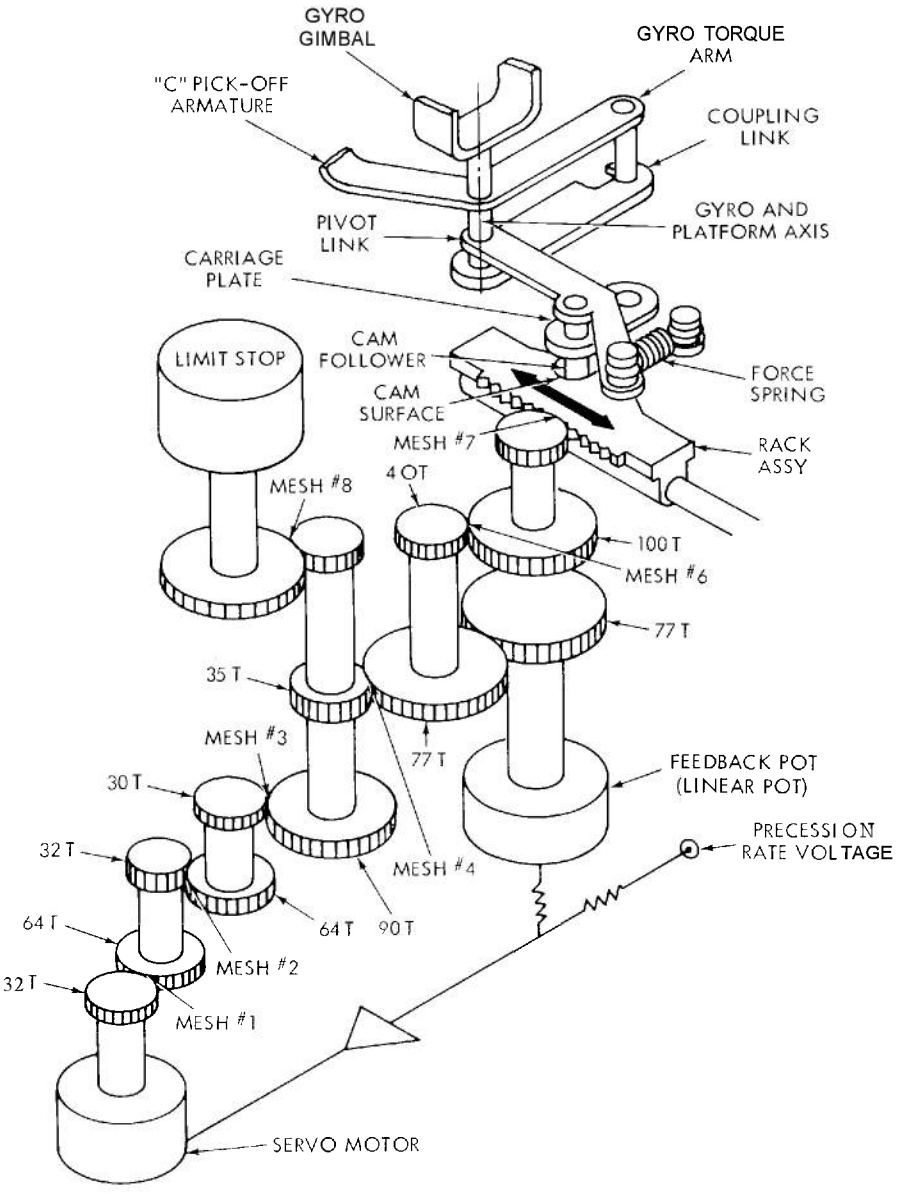


Figure 1%-7. Schematic diagram of the tracking-gyro/platform system.

The angular velocity ω_{2a} is measured approximately by a rate gyro mounted on the platform. The gyro reading must be corrected for several error terms, however.

A rate gyro is a single>-degree-of-freedom gyro in which the single gimbal is restrained by a spring force. A rate applied about an axis that is mutually perpendicular to the spin axis and the gimbal axis produces a torque which is exerted against the spring restraint. A pickoff is employed to measure the displacement δ_r of the gimbal, which is in turn proportional to the applied rate, i.e.,

$$\delta_r = \frac{H}{k}\omega_i \qquad (12\text{-}14)$$

where H is the spin momentum, k is the spring constant, and ω_i is the applied rate.

The rate gyro is mounted on the platform with its input axis perpendicular to the two platform gimbal axes. Ref. 7, par. 4-6 derives the geometrical relationships that relate the rate ω_{p2p} about the platform normal axis to the rate about the tracking line, or gyro spin axis. For small platform servo errors, the expression is, from Eq. 4-311 of Ref. 7,

$$\omega_{2a} \cong \omega_{p2p} + \omega_{1a}\,\delta_t + \omega_{3a}\,\delta_e \qquad (12\text{-}15)$$

where ω_{2a} = the angular rate about the tracking line
ω_{p2p} = the angular rate about the platform normal axis
ω_{1a} = the angular rate about the gyro elevation axis
ω_{3a} = the angular rate about the gyro azimuth axis
δ_t = the platform azimuth servo error
δ_e = the platform elevation servo error

Eq. 12-15 can be transformed into variables which are available in the computer by introducing Eq. 12-13 and neglecting products of error terms. This yields

$$\omega_{2a} = \omega_{P2P} - v_{3p}\,\delta_t + v_{1p}\,\delta_e \qquad (12\text{-}16)$$

in which ω_{p2p} is measured by the rate gyro; v_{1p}, and v_{3p} are available from the tracking computer; and δ_t and δ_e are provided by the platform servo error signals.

A further correction is required for the rate gyro. An input rate causes the gyro to precess against the restraint through the angle δ_r. The input axis is now displaced from the 2p axis, and measures a component of ω_{p1p} as well. For a small displacement, the input rate is $\omega_{p2p} - \omega_{p1p}\,\delta_r$.

The restraining torque on the gyro gimbal is provided by a torque motor. The torque-motor excitation is provided through an amplifier from the gyro signal δ_r, thus generating the equivalent of a spring restraint. The loop is stabilized by an additional feedback of the integral of δ_r. The torque equation on the rate-gyro gimbal is then

$$\omega_{2p2} - \omega_{p1p}\,\delta_r = k_1\,\delta_r + k_2 \int \delta_r\,dt \qquad (12\text{-}17)$$

where k_1 and k_2 are constants that include the gyro spin momentum.

Transformation of the first of Eqs. 12-11 into platform coordinates yields

$$\omega_{PIP} = \frac{S_1 E_{3p}}{D_c} - v_{3p} \qquad (12\text{-}18)$$

Multiplication by δ_r then yields, when the error products are dropped,

$$\omega_{p1p}\,\delta_r = -v_{3p}\,\delta_r \qquad (12\text{-}19)$$

Thus, the gyro torque equation can be corrected by the addition of a term $v_{3p}\,\delta_r$ to the right-hand side, yielding

$$\omega_{p2p} - \omega_{p1p}\,\delta_r = k_1\delta_r + k_2 \int \delta_r dt + v_{3p}\,\delta_r \qquad (12\text{-}20)$$

Since the first two terms of the right-hand side of Eq. 12-20 appear as outputs in the rate-gyro loop, a mechanization of ω_{p2p} has been obtained. Then Eq. 12-16 can be employed to compute ω_{2a}, the angular velocity of the tracking line.

All variables of Eqs. 12-20 and 12-16 are available either from the gyro or from the tracking computer. The mechanization is shown in Fig. 12-8.

Thus, the gyro/platform assembly in the Vigilante fire control system provides (a) an integrator for the generation of tracking rates, (b) a complex coordinate transformation, and (c) a stabilizing element. This is an excellent example of the need for ingenuity in combining many functions into one element.

12-5 MARK 20 GYRO COMPUTING SIGHT'

Gun Sights Mk 20 Mods 6 and 7 are gyroscope lead-computing sights using a single electrically driven gyroscope as the principal element. The Mod 6 is used with 20 mm antiaircraft machine guns, the Mod 7 with 40 mm guns *(see* Figs. 12-9 and 12-10). These sights can, however, be used with other ballistics when appropriately modified and suitably mounted. The principal function of the gun sight is to compute lead angle and superelevation, and to offset the line of sight from the

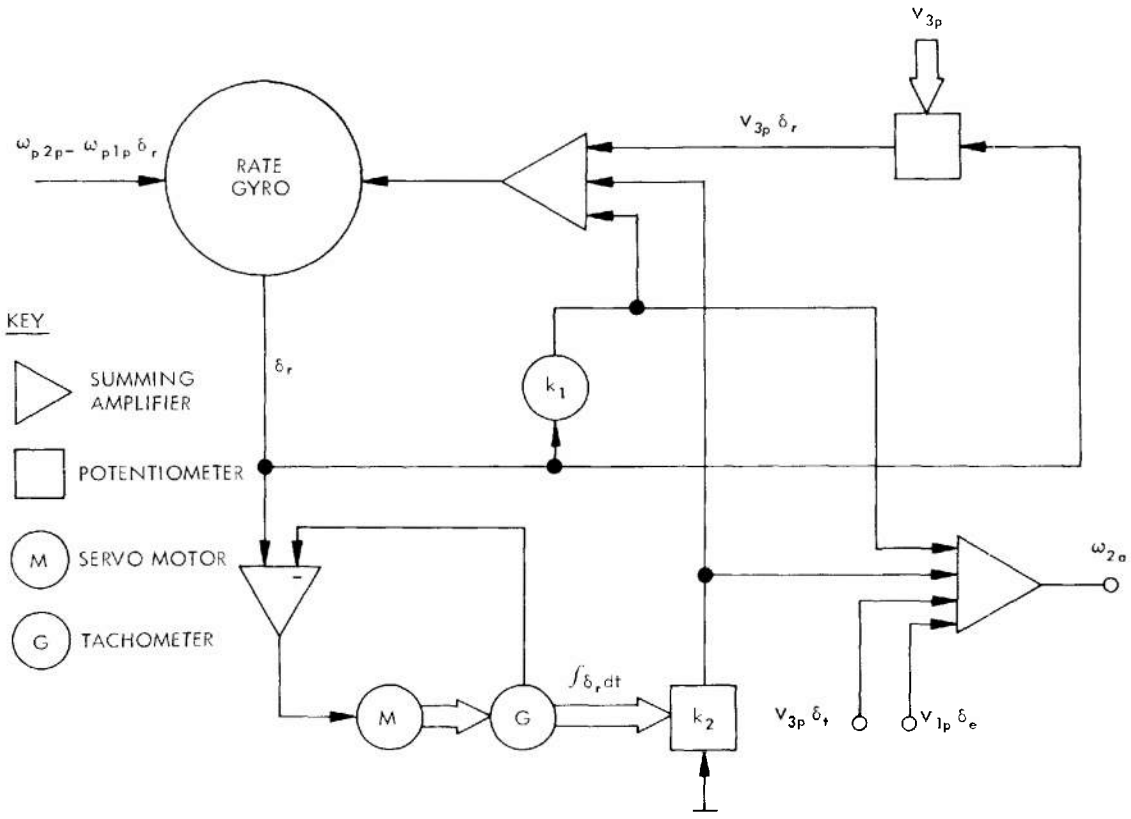


Figure 12-8. Computation of rate about the tracking line.

axis of the gun bore, accordingly. Then, with the line of sight following a moving target, the guri leads the target and is elevated the proper amount to secure hits. The operation of the sight thus relieves the gunner of the duty of estimating lead and superelevation, and eliminates many of the approximations associated with a fixed line of sight. The sights are adapted primarily for use against aircraft, motor torpedo boats, and other high-speed targets, but can be used as well against slow-moving or fixed targets.

Fig. 12-11 shows the elements of the problem solved by the Mk 20 sight. The function of the gun sight is to offset the line of sight from the axis of the gun bore in such a manner that tlic gun is constantly aimed to hit the target when the line of sight is kept on the target. The advance position is offset from the present position by the lead angle. This angle lies on the tilted gun-target plane which contains the present target position and the advance position. The gun bore must be pointed above the advance position to allow for the curved path of the projectile due to gravity. The additional elevation necessary for this purpose is the superelevation. Thus, the offset of the line of sight must include two elements: lead and superelevation. The size and direction of the lead and the size of the superelevation angle are computed by means of a gyroscope, the operation of which is described in the paragraphs which follow.

The gyroscope used in the gun sight employs a small electric motor, the moving parts of which are relatively heavy and correspond to the flywheel of the gyroscope top. The gyro is mounted on pivots in a gimbal, as shown in Fig. 12-12, and the gimbal is pivoted in the sight case. Thus, the spin axis of the gyro can be tilted up or down about the gyropivots and can be moved to the right or left about the gimbal pivots. With the motor running, the direction of the spin axis tends to remain

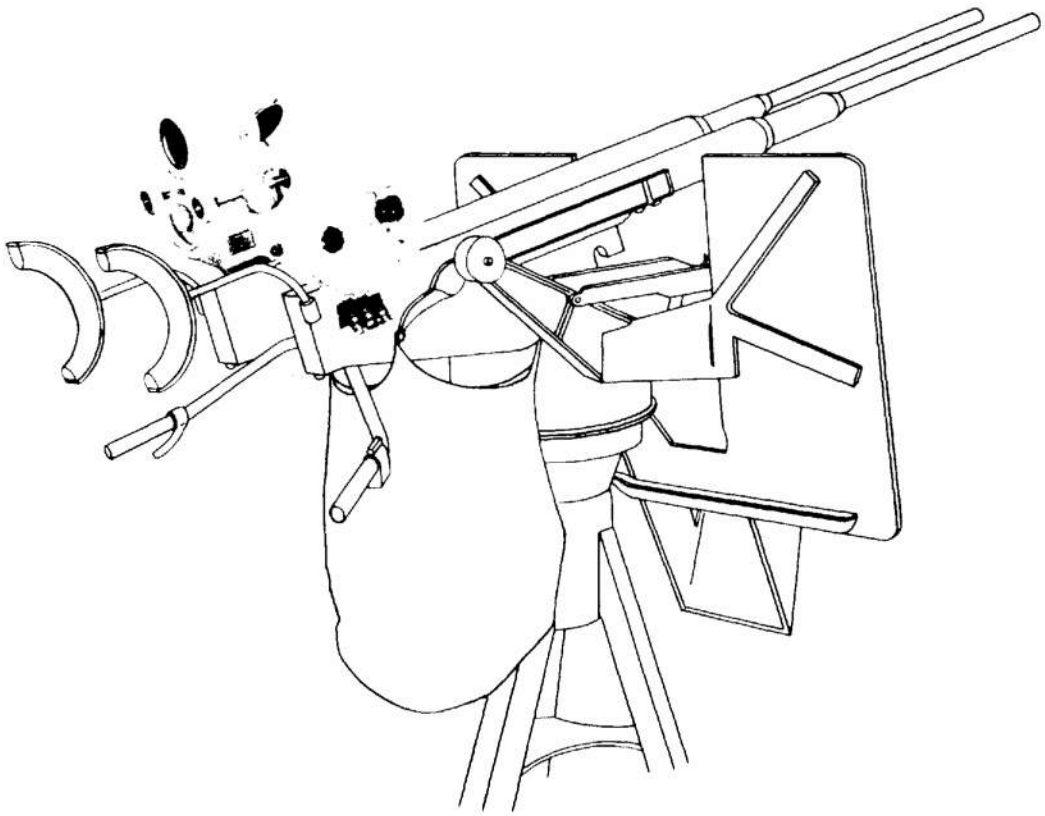


Figure 12-9. Gun Sight Mk 20 Mod 6 mounted on a twin 20 mm gun.

fixed in space, no matter how the sight case is moved. However, the spin axis can be moved in any direction if a properly directed precessing force is applied to the gyro or gimbal.

The principal precessing force is exerted by the range' magnet which is mounted on the sight case directly in front of the gyro. A sectional view of the range magnet is shown in Fig. 12-13. Its pole pieces are located close to the surface of a copper-covered eddy-current disk which is rigidly mounted on the forward end of the gyro shaft. The disk thus rotates in a magnetic field set up by electric current in the winding of the range' magnet. When an electric conductor is moved in the field of a magnet, electric currents are induced in the conductor and these currents react with the field to produce a magnetic force tending to stop the movement. Thus, the range magnet exerts a force on the eddy-current disk, tending to stop rotation of the gyro.

Suppose now that the sight case is moved to swing the magnet to the left, as seen from the front in Fig. 12-14. Since the part of the disk opposite the magnet pole is moving up, the retarding force. exerted by the magnet is directed down. But a downwardly directed force moves the gyro spin axis to the left, in accordance with gyroscopic principles. Thus, the gyro axis follows the magnet. The same result is obtained when the sight case is moved in any direction; the gyro axis always follows the magnet axis.

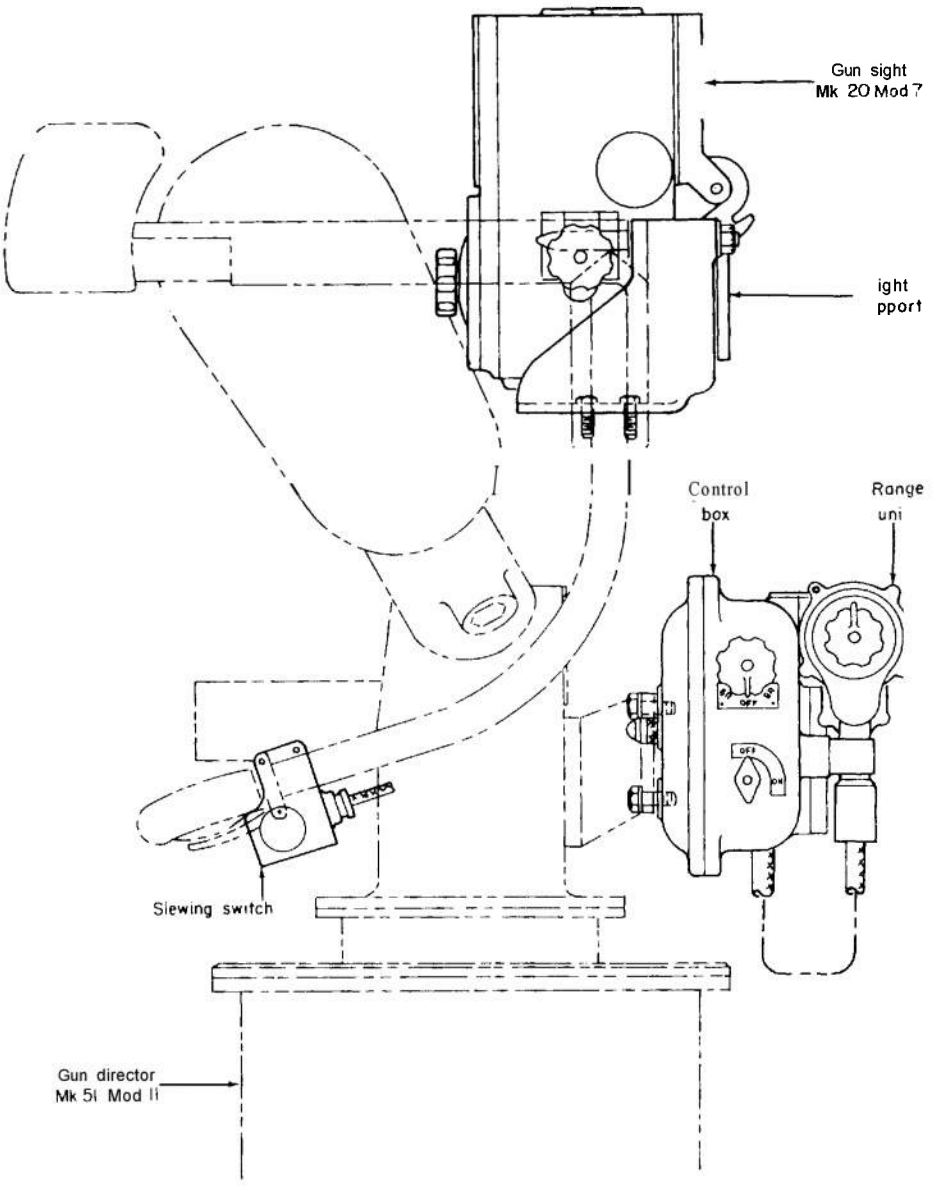


Figure 12-10. Gun Sight Mk 20 Mod 7, with adapter equipment, mounted on Gun Director Mk 51 Mod 11, for use with 40 mm guns.

The speed at which the gyro moves while following the magnet is called the precession rate, and depends on the strength of the precessing force exerted by the magnet. If the precessing force increases, the precession rate increases; if it decreases, the precession rate decreases. If the precessing force is uniform, the gyro axis follows the magnet axis at a uniform rate.

The precessing force increases as the strength of the magnet current increases, and also depends on the distance the magnet has moved from the center of the eddy-current disk. With the magnet centered on the disk, the force is zero; but as it moves away from the center, the force steadily increases. If the magnet is carrying a uniform current and traveling at a uniform speed, it must move

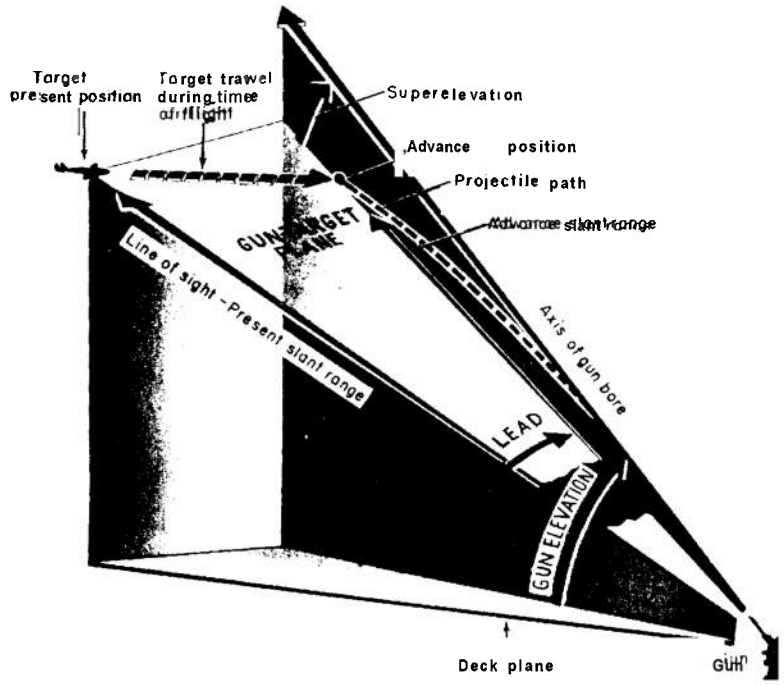


Figure 12-11. The fire control problem solved by Gun Sight Mk 20.

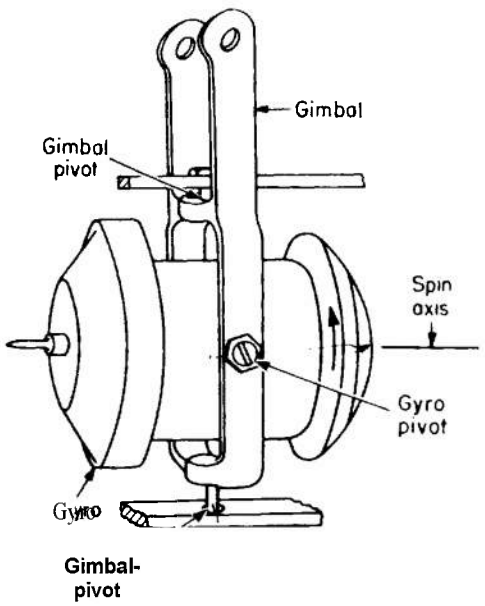


Figure 12-12. The gyro and gimbal used in Gun Sight Mk 20.

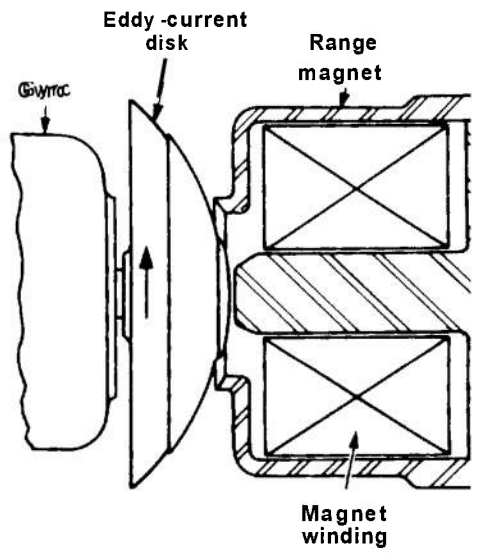


Figure 12-13. The range magnet and eddy-current disk.

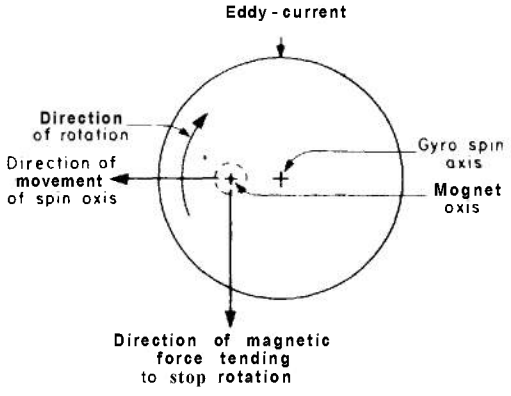


Figure 12-14. Themannerinwhich the gyro follows the magnet.

through a certain definite angle fromthe center of the disk before the precessing forceis strong enough to move the gyro at the same speed, The gyro then follows the magnet with a definite angle of lag which does not change unless the magnet current or magnet speedis altered.

The angle of lag of the gyro can be increased by increasing the speed of movement of the magnet and can be reduced by moving the magnet at a slower speed. Since the magnet is mounted on the sight case which is mounted on the gun, the speed of movement of the gun controls the size of the gyro lag if there is no change in the magnet current. The direction of the gyro lag indicates the direction of movement of the gun.

The gyro lag can be controlled by changing the magnet current. An increase in current increases the precessing force and speeds up the movement of the gyro. It thus reduces the gyro lag necessary to move the gyro at the same speed as the gun.

In summary, as the gun is trained and elevated, the gyro axis follows the magnet axis but lags behind it. The angle of lag is controlled by the speed of gun movement and the strength of the magnet current. The direction of lag indicates the direction of gun movement.

12-5.1 COMPUTATION OF LEAD

The required lead angle can be computed approximately by multiplying the time of flight of the projectile to the advance position by the rate of angular speed of the target about the gun position. Thus, if time of flight is two seconds andthe rate is 50 mils per second, the lead is close to 100 mils. However, this simple relation is a very rough approximation. In these gun sights, a rnore accurate computation is used, but it depends on the same two factors: the angular speed of travel of the gun and the time of flight.*

The preceding discussion has shown that the gyro lag depends on the movement of the gun and on the magnet current. If, then, the magnet current is made to depend on the time of flight, the gyro lag is controlled by the same two factors which deterrnine lead. By properly shaping the eddy-current disk and the parts of the magnetic structure, and by controlling the magnet current in relation to time of flight, the gyro lag is made directly proportional to lead.

Referring again to Fig. 12-11, the target is shown traveling at a substantially constant altitude but its distance from the gun is decreasing. Therefore, the line of fire is tilted up at a steeper angle than the line of site, and the' lead must include an elevation component as well as a traverse component. This relation is shown in Fig. 12-15, in which the line of site, the line of fire, and the axis of the gun are projected on a plane at right angles to the line of site, The example is a typical case since lead is usually a combination of elevation and traverse components.

In following the target, the line of site moves from the present position toward the advance position as shown in Fig. 12-15, and its movement therefore has the same elevation and traverse components as the lead. The gun follows a different path since its bore is displaced from the line of site by the lead and superelevation. However, lead and superelevation change very slowly during a firing run,

* *See* Appendix 12-1 for the true equations.

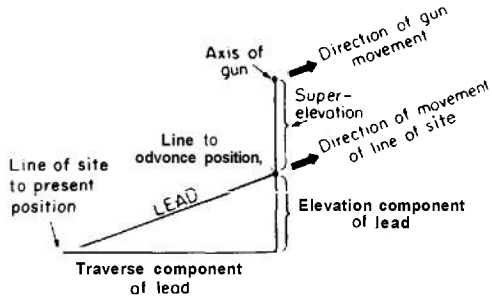


Figure 12-15. Geometrical relationships between lead and superelevation.

and the path of the gun is practically parallel to the path of the line of site. Therefore, the gun movement has the same elevation and traverse components as the line-of-site movement and the lead. As already noted, the direction of gyro lag indicates the direction of movement of the gun. Gyro lag thus has the same elevation and traverse components as the gun movement and the lead, and also indicates the direction of the lead.

12-5.2 TIME OF FLIGHT AND MAGNET CURRENT

It is shown in par. 12-5.1 that, in order to compute the lead angle, time of flight must be known and the range-magnet current must be controlled accordingly. Time of flight to the advance position can be determined from the firing tables of 20 mm and 40 mm guns if the range to the advance position is known. The advance position can be located if present slant range and the course and speed of the target are known.

Gun Sights Mk 20 Mods 6 and 7 are calibrated to give best accuracy on incoming targets having a passing range of 500 yards and a speed corresponding to that set on the target-speed knob of the sight. For Gun Sight Mk 20 Mod 6 an additional calibration is provided (without superelevation change) for outgoing targets of low speed at short and medium ranges. Gun Sight Mk 20 Mod 7 with 40 mm ballistics does not require a different calibration for outgoing targets of low speed at short and medium ranges.

Time of flight to the advance position does not fit into any reasonable mathematical expression which can be used in the gun sights. Instead, a figure known as nominal time of flight (symbol T_n) must be used. Nominal time of flight depends on actual time of flight, present slant range, advance slant range, target speed, and other factors. The nominal time of flight for 20 mm guns with Gun Sight Mk 20 Mod 6 and for 40 mm guns with Gun Sight Mk20 Mod 7 are given in Table 12-1, as functions of the present slant range R and the target velocity.

In the Mod 6 tabulation, T_n is given for maximum and minimum target speeds on incoming targets and for minimum target speed only on outgoing targets. The Mod 7 tabulation is for both incoming and outgoing targets.

Magnet current is supplied by the range circuit which includes the range rheostat. The rheostat dials are marked with the values of present slant range given in Table 12-1. For each range and target speed setting, the rheostat introduces the proper resistance to make the precessing force for any given gyro displacement proportional to the corresponding value of $1/T_n$. By this means, when present slant range and target speed have been estimated and the range rheostat has been set accordingly, the time-of-flight factor is entered in the computation.

It should be noted that the assumed target course and speed are used only to compute nominal time of flight, and thus they affect the size but not the direction of the lead. The direction of the lead is determined by the direction of movement of the gun. The gun's movement is determined by the actual course of the target.

12-5.3 COMPUTATION OF SUPERELEVATION

Superelevation (see Fig. 12-11) depends on two elements: the gun elevation angle and nominal time of flight. For a given gun elevation, superelevation is approximately proportional to nominal time of flight. For a given time of flight, it is greatest when gun elevation

TABLE 12-1. NOMINAL TIME OF FLIGHT VS RANGE DATA.

(a) 20 mm Guns and Gun Sight Mk 20 Mod 6

	INCOMING TARGET		OUTGOING TARGET
	Target Speed		
	200 Knots	600 Knots	200 Knots
R	T_n	T_n	T_n
200	0.23	0.23	0.23
400	0.51	0.51	0.53
800	1.21	1.19	1.38
1200	2.12	2.04	2.60
1600	3.24	3.06	4.10
2000	4.56	4.20	

(b) 40 mm Guns and Gun Sight Mk 20 Mod 7

(For an incoming or outgoing speed of 200 knots)

R	T_n
200	0.24
400	0.475
800	0.99
1200	1.55
1600	2.18
2000	2.85
2400	3.57
2800	4.38
3200	5.75
3600	6.10

NOTES:

1. The present slant range R is given in yards
2. The nominal time of flight T_n is given in seconds.

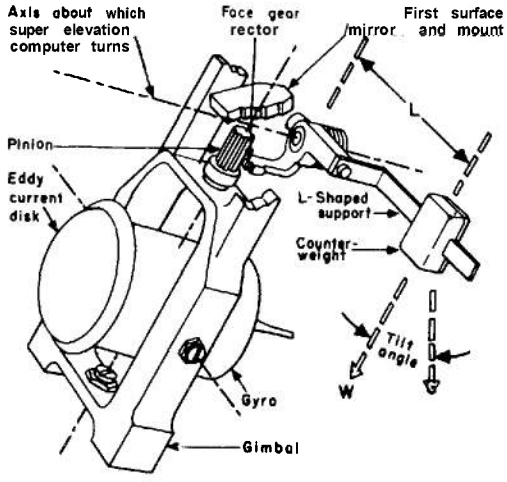
is zero and grows smaller as the gun is elevated.*

In Gun Sights Mk 20 Mods 6 and 7, superelevation is computed by the face gear sector, L-shaped support, and counterweight, as shown in Fig. 12-16. Gear teeth on the face gear sector mesh with a pinion mounted on the gimbal. Thus the counterweight which is carried on the L-shaptd support exerts a force tending to rotate the gimbal and gyro about the gimbal pivots. The turning force can be increased by moving the weight farther out on the L-shaped support and can be decreased by moving the weight toward the pivot.

Target speed has an appreciable effect on the superelevation correction. In view of this fact, the superelevation computer is so constructed that the counterweight can be positioned on the L-shaped support by turning the target speed knob.

The tilt of the sight case as the gun is elevated has the same effect as moving the weight toward the pivot. It reduces the effective lever arm and so reduces the turning

* The equation used for superelevation in the gun sights is $K T_n \cos E_g$, where K is a constant, dependent upon the target speed knob setting, T_n is nominal time of flight, and E_g is gun elevation. This equation does not give the exact value of superelevation (*See* Figures 12-2.4 and 12-2.5 of Appendix 12-2).



L = Effective lever arm of the superelevation computer
LW = Torque acting in gyro assembly due to force of gravity G
W = G (cos. of tilt angle)

Figure 12-16. Operation of the superelevation computer.

force. This reduction in turning force is in direct proportion to the reduction in superelevation required by the increased elevation of the gun. Thus, the tilt of the sight case automatically proportions the turning force of one of the elements which determine superelevation.

The second factor, nominal time of flight, is introduced by the range magnet. The turning force exerted by the superelevation weight is directed to the gimbal and gyro by means of gears as already noted and as seen from the front in Fig. 12-17. It therefore precesses the gyro downward until the range magnet exerts an equal force in the oppositedirection. With the two forces balanced, the gyro comes *to* rest with a definite displacement from the magnet axis which depends on the turning force and on the magnet current. Since magnet current represents nominal time of flight, it supplies the time of flight factor necessary for the computation. The relation between magnet current and time of flight is such that the resulting displacement is directly proportional to superelevation.

Small errors in the computed superelevation are introduced by the displacement of the gimbal to the right or left in the computation of lead. This displacement raises or lowers the superelevation weight above or below the nominal horizontal. The tilt of the sight case due to cross-level roll or pitch of the base on which the sight is mountedalso has a slight effect on the size and direction of the computed superelevation. However, these errors are usually very small.

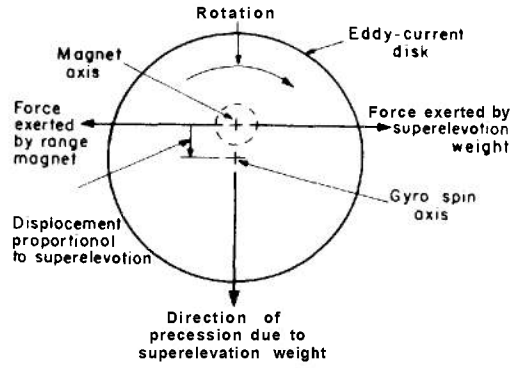


Figure 12-17. Effect of the superelevation computer on the position of the gyro spin axis.

In summary, the displacement of the magnet axis from the gyro axis is the result of two displacements. The lead displacement is proportional to the lead angle and is caused by the movement of the gun in following the target. Its direction is determined by the direction in which the gun is moving. The superelevation displacement is all in elevation and is proportional to the superelevation angle. It is caused by the precessing force exerted by the superelevation computer. The total displacement is proportional to the desired displacement between the gun bore and the line of site, and its direction is the same as that of the desired displacement.

12-5.4 DISPLACEMENT OF THE LINE OF SIGHT

The displacement of the line of sight with respect to the axis of the gun bore is accomplished by the movement of the reflector glass shown mounted on the gyro gimbal in Fig. 12-18. The reflector glass rotates in the same direction that the gimbal turns on its pivots in the sight case. In addition, the reflector glass is connected to the gyro by a link which tilts it

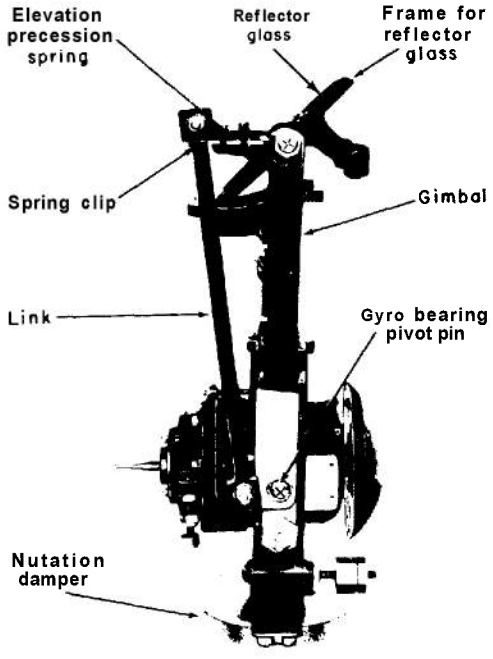


Figure 12-18. The linkage between the reflector glass and the gyro in traverse and elevation.

up or down as the gyro tilts in the gimbal.

The relation of the reflector glass tothe rest of the optical system is shown in Fig. 12-19. The beam of light from the lamp is reflected by the first surface mirror and passes through the cylinder lens and reflector glass to the Mangin mirror which reflects and collimates the light rays, causing the reticle image to appear on the reflector glass.

Because of the collimating action of the Mangin mirror there is no dispersion of the reflected beam; it is made up of parallel rays all of which are reflected at the same angle. If the gunner holds his eyes at the center of the beam, he will see the reticle image at the center of the glass. If he moves his head toward the edge of the beam, theimagewill appear to move toward the edge of the glass; but, since the direction of all parts of the beam is the same, the direction of the line of sight does not change. Thus, considerable movement of the gunner's head is possible without his losing sight of the image or changing the direction of the line of sight.

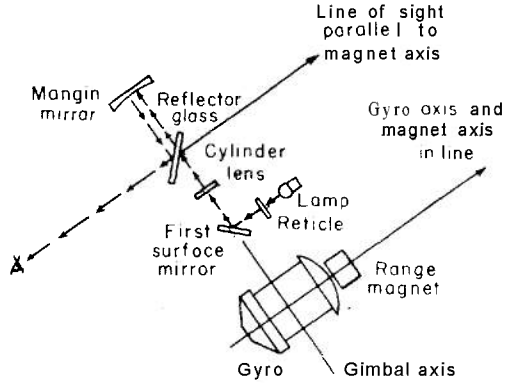
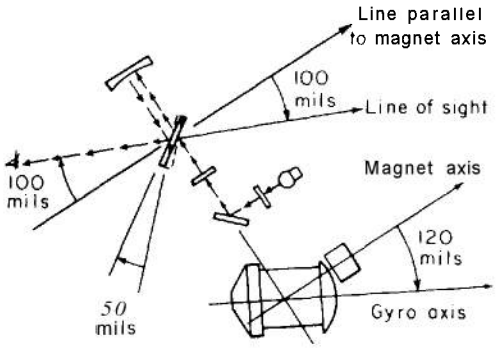


Figure 12-19. The line of sight with zero gyro-axis displacement.

The direction in which the beam of light is reflected depends on the angle) at which the reflector glass is tilted. For zero gyro-axis displacement, as shown in Fig. 12-19, the angle is such that the beam is reflected along a line parallel to the axis of the range magnet. Therefore, the line of sight is parallel to the magnet axis. If the reflector glass is tilted 50 mils from this position in the direction shown in Fig. 12-20, the reflected beam tilts up 100 mils from its former position. As a result, the line of sight is depressed 100 mils below the direction of the magnet axis. The angle of depression or elevation of the line of sight is always twice the angle through which the reflector glass is moved from its zero position.

As noted inpar. 12-5.3, the gyro-axis displacement is proportional to the desired displacement of the line of sight... It is not made equal to the line-of-sight displacement but is always greater by an arbitrarily added percentage>which is determined by the design of the magnet and eddy-current disk and by calibration of the range-magnet current. The added percentage is the sigma factor, the purpose of which is described in par. 12-5.5. In elevation, the sigma factor is 20 percent; in traverse, it is approximately 29 percent. Because of the inherent characteristics of the gyro gimbal system, it is not possible to make the sigma factor the same in bothdirections, but this does not affect the accuracy of the lead computation.

In elevation, the sigma factor is taken care of by the design of the linkage which



NOTE:
FOR ILLUSTRATIVE PURPOSES, THE ANGLES SHOWN HERE HAVE BEEN EXAGGERATED.

Figure 12-20. The line of sight with 120 mils gyro-axis displacement.

connects the reflector glass to the gyro in such a manner that it tilts the glass one mil for each 2.4 mils of gyro displacement. Thus, in the example shown in Fig. 12-20, a gyro displacement of 120 mils tilts the glass 50 mils and depresses the line of sight 100 mils.

In traverse, there is no linkage which can be proportioned to take care of the sigma factor. In traverse, the sigma factor is introduced by the choice of proper constants in the optical system. Sigma is the ratio of the distance from the reticle to the first surface mirror over the distance from the first surface mirror to the Mangin mirror.

Since the sigma factor is introduced by the magnet system and is then taken out by the elevation linkage and the collimating system, and since the displacement of the line of sight from the axis is the same in size and direction as the desired displacement of the line of sight from the gun bore, it only remains to mount the sight on the gun in such a position that the magnet axis is parallel to the gun bore. Then, with the line of sight following the present position of the target, the gun is constantly aimed at the proper point in space.

At large elevation lead angles, an elevation precession spring applies a centering torque about the elevation axis. This action results in a substantial reduction in quadrature error without adversely affecting the other performance characteristics. The elevation precession spring is shown in Fig. 12-18. It is secured to the right side of the gyro gimbal support bracket by the locknut which supports the right side of the reflector frame. The other end of the spring passes through a small hole in a clip on the end of the reflector frame arm.

A traverse precession spring operates in a similar manner for large traverse lead angles.

12-5.5 FUNCTIONS OF THE SIGMA FACTOR

The sigma factor is arbitrarily introduced in the design of the gun sight to improve its tracking characteristics. The gyro lag depends on the speed of movement of the gun as explained earlier. If the gun is moved smoothly, as it should be moved in tracking a target, the gyro lag changes gradually and smoothly. But if the gun is moved in a jerky or erratic manner, the gyro lag increases with every forward jerk and decreases with every decrease in gun speed. A sudden change in gyro lag changes the precession rate of the gyro, but it produces no sudden change in the position of the gyro axis. The effect is much the same as if the gyro were pulled along by a very flexible rubber band connected to the gun barrel. The gyro follows the jerky gun movement smoothly at a speed corresponding roughly to the average speed of the gun.

If the lag of the line of sight were made exactly equal to the gyro lag, the line of sight would be parallel to the gyro axis and it, too, would move smoothly even though the gun movement were erratic. There would be no indication to warn the gunner that he was not tracking properly. He would continue his erratic operation and the direction of his fire would be equally erratic.

With a sigma factor of 20 percent, the gyro lag is 1.20 times the lag of the line of sight or, stated, the lag of the line of sight is five-sixths of the gyro lag. Thus, the angle between the line of sight and the gyro axis is equal to one-sixth of the gyro lag. Now, suppose the gun is given a sudden jerk that increases the gyro lag by 24 mils. The line of sight responds instantly by moving 4 mils farther away from the gyro axis. The reticle image moves 4 mils off the target and the gunner

knows immediately that he is not tracking properly. Thus, the introduction of the sigma factor enables the gunner to recognize poor tracking and to correct it promptly.

12-6 CANT-CORRECTION SYSTEM OF BALLISTIC COMPUTER XM17 [6]

The design of a fire control computer—regardless of whether the computer is destined for field artillery, antiaircraft, tank, or air-to-ground (e.g., helicopter) applications—requires an analysis of the various functional aspects of the application concerned. For the categories noted, these functional aspects would include the following:

For Field Artillery and AA Applications

Towed vs. self-propelled weapon
Type of weapon
Type of target(s)
Range(s)
Interfaces

For Tank Applications

Model of tank
Type of weapon
Type of target(s)
Range(s)
Interfaces

For Air-to-ground Applications

Type of aircraft
Aircraft speed and altitude
Type of weapon(s)
Type of target(s)
Target speeds

After this analysis of functional aspects has been completed, the development of the computer mathematical model can proceed. This model should, of course, provide for (a) such fire control corrections as kinematic lead (if moving targets are involved) and superelevation, and (b) modifications in the fire control solution for deviations from standard conditions of air density, air temperature, wind, etc. For field artillery, antiaircraft, and tank applications, the effect of cant is an additional factor that should be taken into consideration in the fire control solution, for cant is a subtle condition that does introduce target-miss components. In the process of designing and fabricating a computer to correct for cant, it will therefore be necessary and desirable to modify and approximate the mathematical mode in such a way that the computer will remain as simple as practicable while providing the required accuracy.

As an illustrative example of the results of such a process, the paragraphs which follow summarize the background of Ballistic Computer XM17 and describe the method employed in its design to correct for the cant of a tank. In addition, the accuracy of the cant-correction system is analyzed. Finally, the factors that should be considered before undertaking an improved design are discussed.

12-6.1 BACKGROUND OF BALLISTIC COMPUTER XM17

Because of gravity, hitting a target with a projectile requires that the gun from which a projectile is to be fired be elevated above the line of sight (see Fig. 12-21). The fire control system now standard for the M60 series tanks includes a ballistic computer (the M13A1D) which accepts range from the range finder and computes the superelevation Φ_s required for hitting the target. Superelevation is then introduced into the weapon by virtue of the fact that the computer output depresses a laying cross in the gunner's sight line with respect to the axis of tube. When the gunner restores the cross to the target, the weapon is then at the proper elevation (see Fig. 12-22).

Ballistic Computer M13A1D generates superelevation as a function of range only. As a result, errors occur because effects other than range--such as gun jump, muzzle-velocity variations, parallax due to offset of the sighting device from the gun, tube bend, etc.--have been ignored. In addition, lateral effects such as drift, lateral parallax, etc., are not included. The major shortcoming of the M13A1D Computer is, however, that effects on the firing data of the out-of-level condition of the weapon (cant) are not corrected.

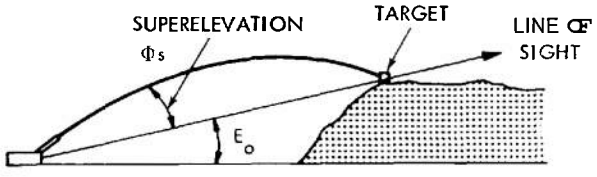


Figure 12-21. The geometry associated with the superelevation correction.

Because of the shortcomings of the Ballistic Computer M13A1D, Frankford Arsenal, under the sponsorship of Army Tank-Automotive Center (ATAC), directed the development of the improved Ballistic Computers XM16 and XM17. These devices, which account for all the effects discussed above, are identical except that the XM16 is designed for use with a periscope while the XM17 provides corrections to Direct Fire Telescope XM108. The ensuing discussion pertains solely to the XM17 Computer.

12-6.2 THE DESIGN USED FOR THE CANT-CORRECTION SYSTEM

The purpose of the Ballistic Computer XM17 is to determine the angles in both azimuth and elevation that are necessary for properly offsetting the gun tube from the line of sight to the target. Corrections that require the gun to be displaced in elevation from the line of site include superelevation, tube bend, vertical jump, and vertical parallax. Corrections requiring azimuth shifts between the gun and the line of site include lateral jump, lateral parallax, and drift.

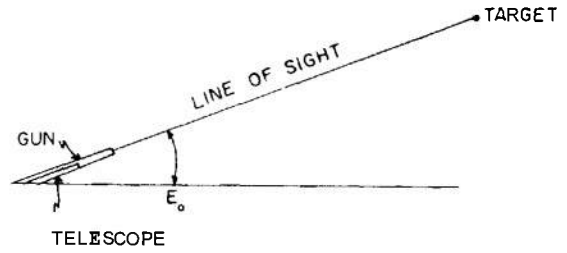
It should be noted that these corrections fall into two categories: those that always occur in the same direction with respect to the deck of the tank, and those that always occur in the same direction with respect to level. Jump is an example of an effect that is tank oriented. No matter whether the tank is positioned on a hill or on level ground, it is assumed that the gun tube will always jump by the same magnitude and in the same direction with respect to the deck of the tank. As a result, corrections which are made in gun elevation and azimuth to compensate for the effect do not depend on cant.

On the other hand, as contrasted with the tank-oriented corrections, superelevation, compensation for tube bend and compensation for drift are all gravity-oriented and must, therefore, be applied to the weapon in a level coordinate system. This point is illustrated by the discussion which follows and by Fig. 12-23.

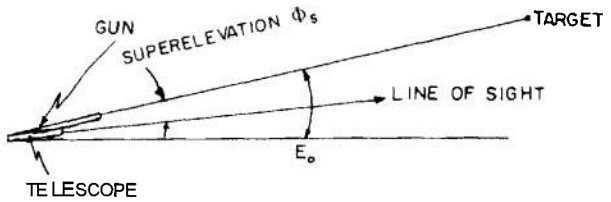
Assume that the target lies at the position E_{o_1} and A_{o_1} with respect to the tank. (E_o is measured in a vertical plane with respect to level, while A_o is measured in a horizontal plane with respect to an arbitrary azimuth reference.) Geometric conditions (range and height of target) and environmental effects (meteorological conditions, muzzle velocity, etc.) necessitate that if a hit is to be obtained on the target, the axis of the weapon must be displaced from the line of site by the angles AE, and ΔA_o. Omitting from consideration the tank-oriented corrections, these angles are determined entirely by ballistic data as found in the firing table (superelevation and drift) and the effect of gravity on the gun tube. The angles are completely independent of the manner in which the gun trunnions may happen to be canted. As a result, the elevation displacement AE, must always be made in a vertical plane and the azimuth displacement ΔA_o must always be made in a horizontal plane.

If the tank were always level, the proper introduction of **AE,** and ΔA_o into the sighting system would present no problems. A level tank is, however, a rare case and it is for this reason cant correction is required if optimum accuracy is to be obtained. The results of not correcting the gravity-oriented effects for cant can be seen from the discussion which follows.

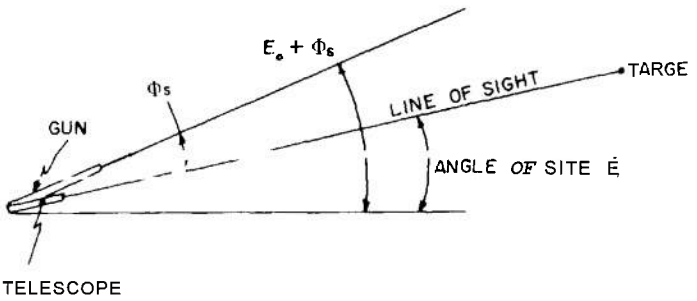
Assume that ΔE_o has been computed on the basis of range and tube bend. This value **AE,** is applied directly to a weapon whose elevation axis (trunnion) is level. As shown on Fig. 12-24(A), this results in the weapon being elevated vertically so that its muzzle travels from P_1 to P_2. Now, assume that the weapon is placed on a slope, as shown on Fig. 12-25, so that the gun trunnion is no longer level. A computer such as the M13A1D which does not correct for cant will generate a



(A) TELESCOPE BORESIGHTED WITH CROSS IN TELESCOPE ON TARGET



(B) TELESCOPE DEPRESSED BY COMPUTER OUTPUT, CROSS IN TELESCOPE NO LONGER LAYED ON TARGET



(C) GUN ELEVATED UNTIL TELESCOPE CROSS IS LAYED ON TARGET

Figure 12-22. The mechanics of introducing the superelevation correction.

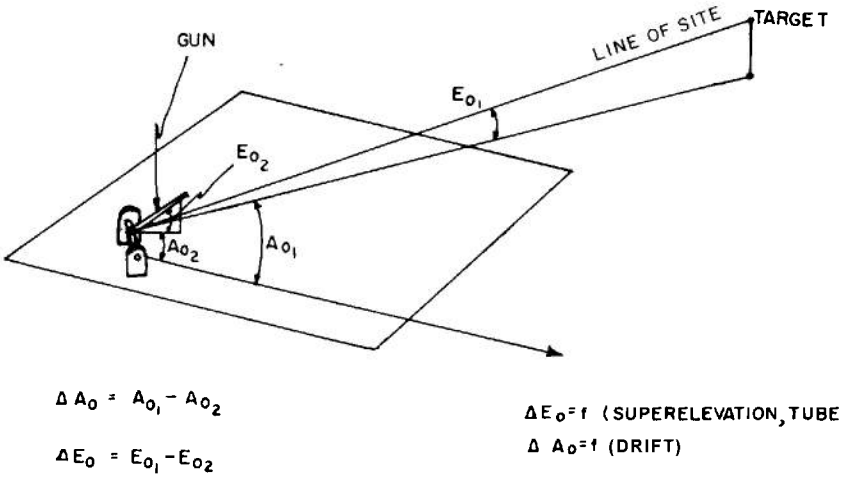


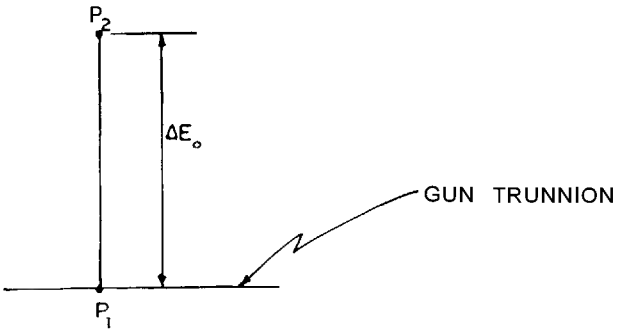
Figure 12-23. The geometry associated with the azimuth and elevation corrections.

value AE, without regard to whether or not the trunnion is level. As can be seen on Fig. 12-24(B), elevating the muzzle of the gun from P_1 to P_2 under this condition results in a wrong elevation and a large azimuth error as well. Conversely, an azimuth correction if not properly cant-compensated will result in an elevation error. The paragraphs which follow discuss the principle of operation of the cant-correction system.

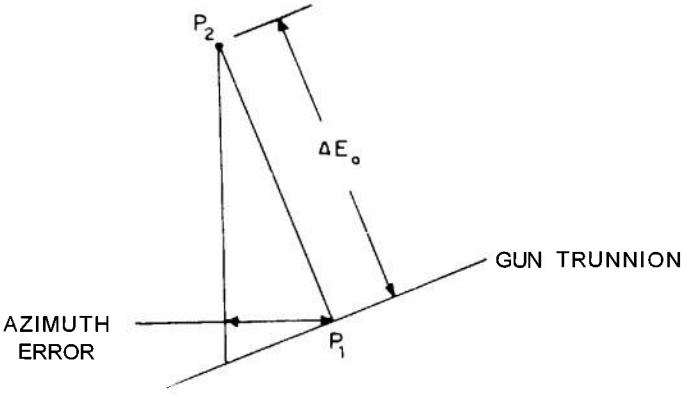
Figure 12-25 depicts a tank gun positioned on a hill of slope α. Initially, it is assumed that the output of the computer is zero and, hence, the reticle cross of the XM108 Telescope is at its boresight position. The gunner has traversed and elevated the weapon until the telescope cross lies on the target and, as a result, the gun is pointed directly at the target. The computer is energized and generates on the basis of its input (range, tube bend, etc.) values of ΔE_o and ΔA_o which, as previously noted, *(see* Fig. 12-23) are the required offset of the gun axis from the lint. of site as measured in a level coordinate system. This requires that the muzzle of the gun be moved from point P_1 to P_3 (see Fig. 12-26). As the tank is constructed so that the gun can only elevate perpendicular to the deck, it is impossible to elevate the weapon vertically through AE, to point P_2 and then traverse horizontally through AA, to P_3. Rather, it is required that first the weapon be traversed in a plane parallel to the deck of the tank to P_2' and then elevated perpendicular to the deck to point P_3. Briefly, it can be seen from Fig. 12-26 that while it is determined that the weapon is to be elevated ΔE_o and traversed ΔA_o, the output of the computer must be in terms of angles measured in planes that are parallel and perpendicular to the deck of the tank, i.e., the output must be cant corrected. These angles are, as shown on Fig. 12-26, AE, and AA,. In the discussion that follows, the subscript o; i.e., E_o, A_o, etc., will denote angles measured in planes perpendicular and parallel to the level, while the subscript g; i.e., AA_g, AE_g, etc., will denote angles measured in planes that are perpendicular and parallel to the deck of the tank.

The XM17 Computer first computes the required vertical and lateral deflections in level coordinates (ΔE_o and AA,) and then transforms these corrections into gun coordinates through the utilization of the following relationships:

$$\Delta A_c = \Delta A_o + \Delta E_o \sin C \qquad (12\text{-}21)$$



(A) APPLYING THE ELEVATION CORRECTION TO A LEVEL GUN



(B) APPLYING THE ELEVATION CORRECTION TO A NONLEVEL GUN

Figure 12-24. The application of the elevation correction.

$$\Delta E_c = \Delta E_o \cos C - \Delta A_o \sin C \qquad (12\text{-}22)$$

where AA, and AE, are azimuth and elevation corrections, which are approximately equal to AA, and ΔE_g. This will be explained in a subsequent paragraph. The symbol C in Eqs. 12-21 and 12-22 represents cant, the inclincation of the gun trunnion. While the meaning of the term cant is generally understood, it should be noted that there are a number of ways in which it may be defined and measured (see Fig. 12-27). While the usual definition of cant is the inclination of the trunnion with respect to level as measured in a vertical plane, it is not uncommon to define cant as the angle between the trunnion and level in a plane that is perpendicular to the deck of the vehicle. Thus, it is important to know precisely how cant is measured prior to using Eqs. 12-21 and 12-22.

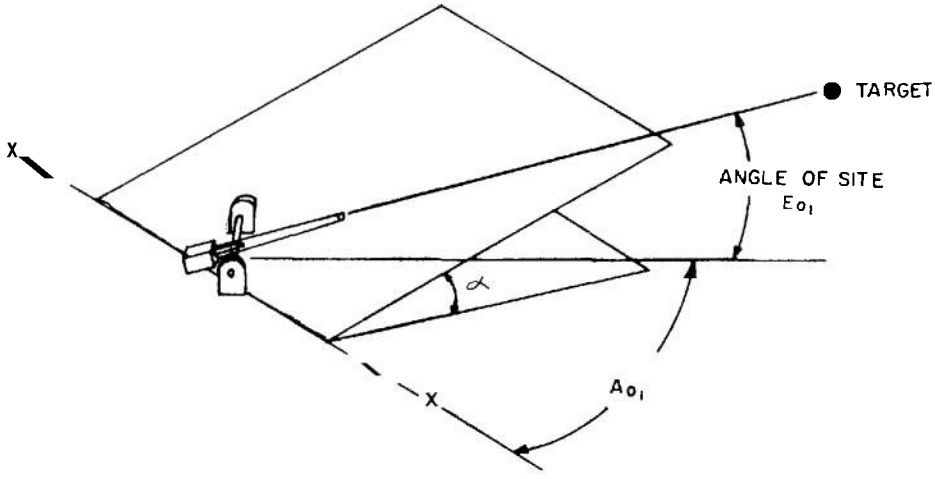


Figure 12-25. The weapon positioned on a slope.

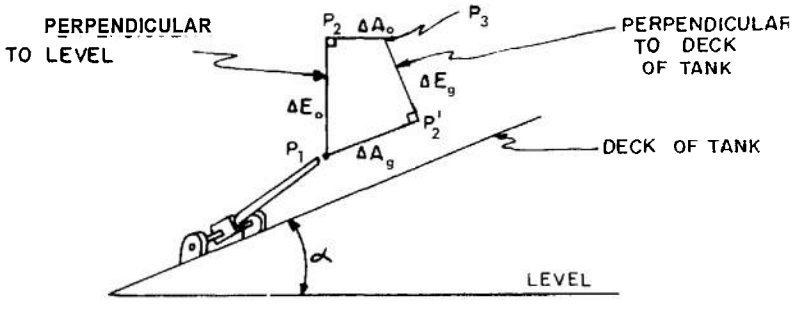


Figure 12-26. Movements of a weapon positioned on a slope.

In the XM17 Computer System, the value to be utilized in Eqs. 12-21 and 12-22 is determined by a pendulum unit which is mounted to the roof of the tank and, as shown on Fig. 12-28, rotates about an axis which is perpendicular to the trunnion and parallel to the turret deck. As a result, cant as used in the approximate equation is not measured in a vertical plane.

Fig. 12-29 shows a schematic arrangement of the Direct Fire Telescope XM108. This instrument is an articulated telescope which has a stationary eyepiece and an objective end that elevates with and always remains parallel with the axis of the gun. The objective end includes a mechanism that contains a reticle mounted on slides so that the reticle may be moved laterally and vertically. The reticle is laid on the target by the gunner. When the telescope is boresighted, the gunner's line of sight through the cross is parallel to the bore of the gun. A ΔE_c signa

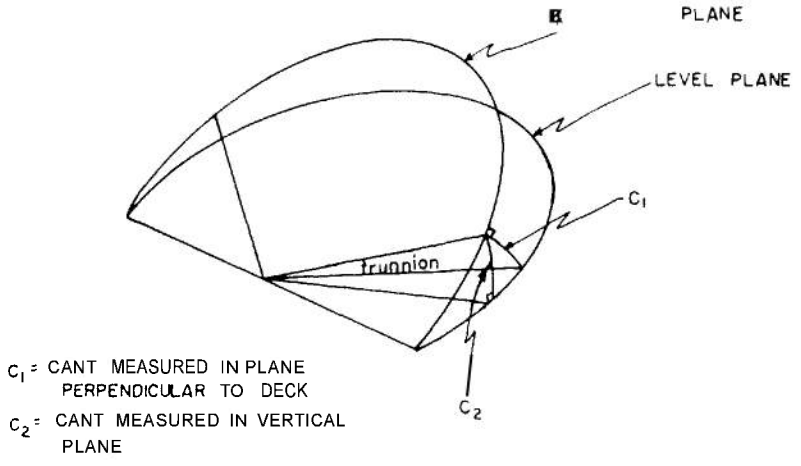


Figure 12-27. Two methods of defining cant.

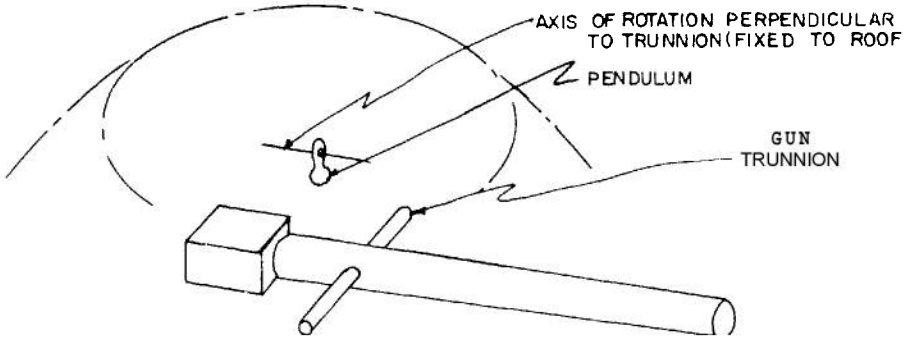


Figure 12-28. Mounting of a cant-measuring unit.

emanating from the computer drives the cross in the vertical direction, while a AA, signal drives it laterally. It is important to note that at all times these movements take place in a plane that is perpendicular to the axis of the gun.

12-6.3 ACCURACY ANALYSIS OF THE CANT-CORHECTION SYSTEM

The analysis of the theoretical accuracy of the cant-correction capability of the XM17 Computer starts with the assumption that the following initial conditions exist:

1. The tank is positioned on a hill with a slope.
2. Initially the computer output is zero so that the reticle is boresighted with the gun.
3. The gunner lays the aiming cross in the telescope reticle and, hence, directly points the gun at a target that is at an angle E_{01} above the horizontal as measured in a vertical plane.
4. The target's azimuth with respect to the X - X axis *(see* Fig. 12-25) as measured in a horizontal plane is A_{01}.

The geometric conditions, described above, are shown in Fig. 12-30(A). A project-

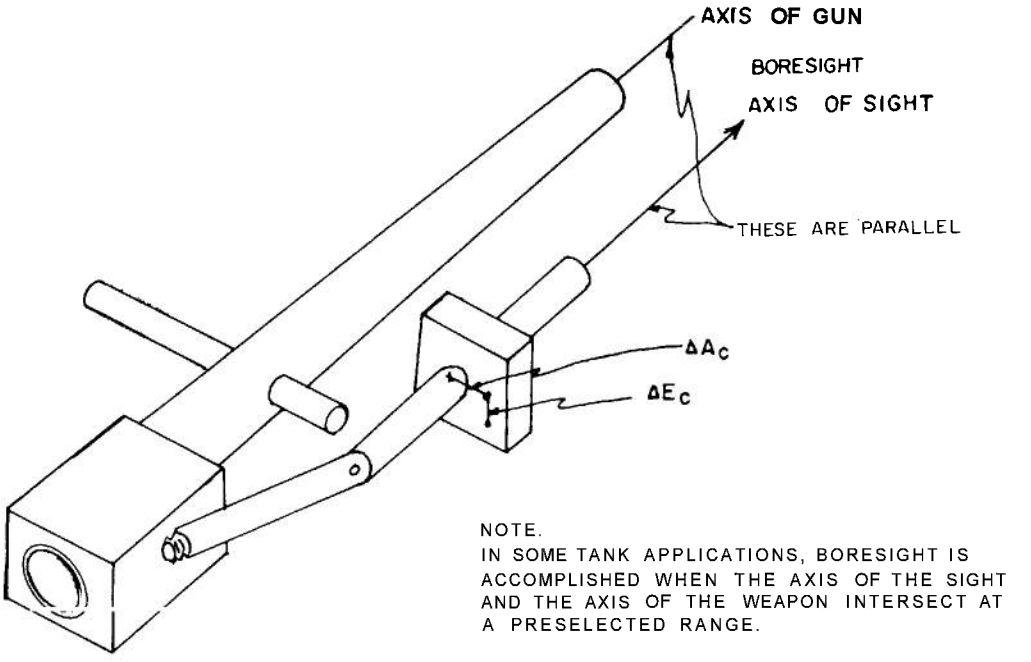


Figure 12-29. Relation of the gun and the direct-fire telescope.

ed view of the solid diagram shown in Fig. 12-30(A) is given in Fig. 12-30(B). The first step of the computation is to convert the coordinates of the gun tube as measured in the level plane (A_{o_1} and E_{o_1}) to its coordinates as measured in the plant. of the tank deck (A_{g_1} and E_{g_1}). This transformation is made by utilization of the following equations:

$$\sin E_g = \cos \alpha \sin E_o - \cos E_o \sin A_o \sin \alpha \quad (12\text{-}23)$$

$$\tan A_g = \cos \alpha \tan A_o + \tan E_o \sec A_o \sin \alpha \quad (12\text{-}24)$$

Next, the rotation of the pendulum about its axis, as shown in Fig. 12-31, and hence the value of C utilized by the computer can be computed by the equation

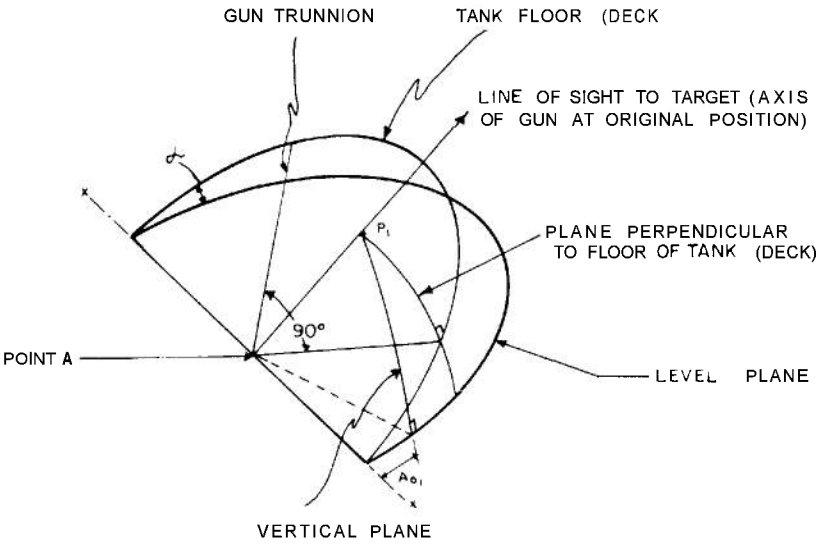
$$\tan C = \tan \alpha \cos A, \quad (12\text{-}25)$$

As previously described (see Fig. 12-23), the proper laying of the gun in order to obtain a hit on the target requires that the axis of the tube be displaced from the line of site by the angles AA, and ΔE_o.

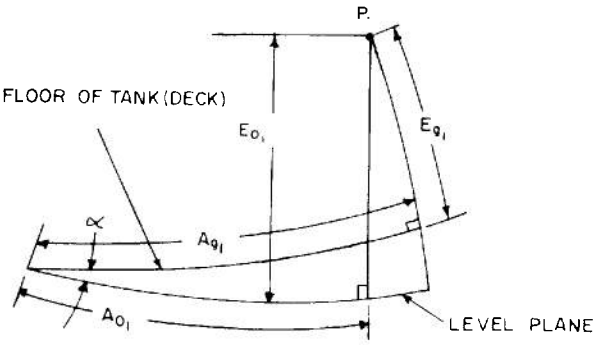
In actual practice, values of ΔA_o and ΔE_o are, as previously explained, determined by the computer on the basis of range tube bend. The AA, and AE, values are converted into ΔA_{c_1} and ΔE_{c_1} by means of Eqs. 12-21 and 12-22. Signals thus derived by the computer are then used to drive the telescope reticle, as shown in Fig. 12-29. The paragraphs which follow describe an analysis of the accuracy achieved by this mechanization.

As previously explained and as shown in Fig. 12-32, the reticle and its drives are always positioned in a plane that is perpendicular to the axis of the gun tube. The ΔE_c signals emanating from the computer will, therefore, drive the reticle cross from point P_1 to P_2 while the A A, signal will then drive it to P_3.

Following this movement, the gunner will traverse and elevate the gun until his line of sight through the cross again lays on the target. When these movements have been completed, the line of sight through the cross has returned to the position A_{g_1}, E_{g_1}, but the gun is now at a new position A_{g_2}, E_{g_2}. The



(A) GEOMETRY OF A WEAPON ON A SLOPE AND AWEAPON AT ORIGINAL POSITION



(B) PROJECTION ON A SPHERE CENTERED AT POINT A

Figure 12-30. The geometry associated with cant correction.

mathematical relationship between ΔA_{c_1}, E_{g_1}, E_{g_2}, and ΔE_{c_1} is expressed by the equation

$$\tan E_{g_1} = \tan (E_{g_2} - \Delta E_{c_1}) \cos \Delta A_c \qquad (12\text{-}26)$$

Inasmuch as ΔA_{c_1} is invariably small, this equation reduces to

$$E_{g_1} \cong E_{g_2} - \Delta E_{c_1} \qquad (12\text{-}27)$$

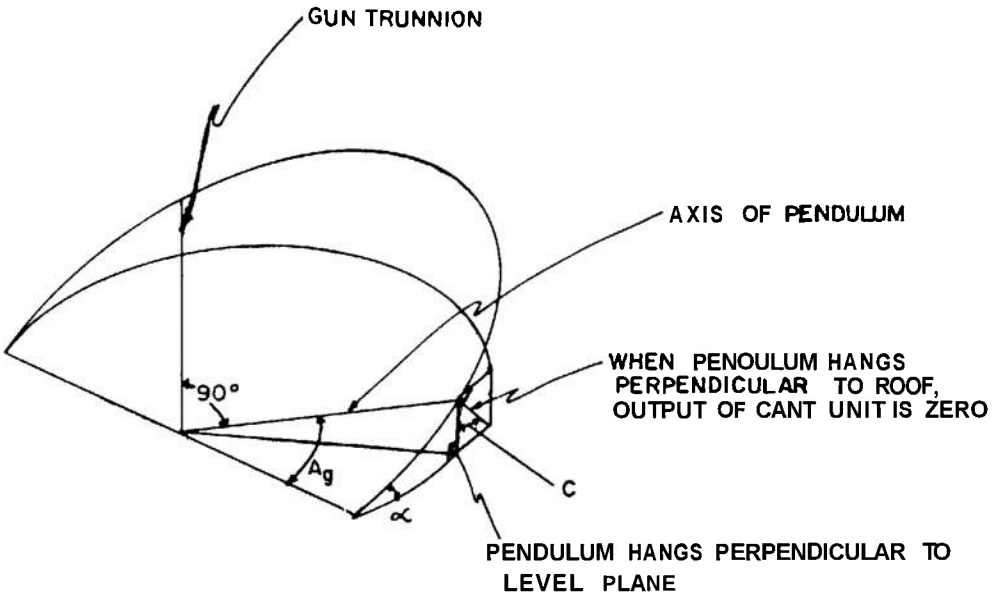
Therefore,

$$\Delta E_{c_1} \cong \Delta E_{g_1} \qquad (12\text{-}28)$$

where ΔE_{g_1} is $E_{g_2} - E_{g_1}$.

The conversion from ΔA_c to ΔA_{g_1} may be accomplished by use of the relationship

$$\tan \Delta A_{g_1} = \frac{\tan \Delta A_{c_1} \cos \Delta E_{c_1}}{\cos (E_{g_2} - \Delta E_{c_1})} \qquad (12\text{-}29)$$



THE ANGLE THROUGH WHICH THE PENDULUM ROTATES WHEN MOVING FROM A POSITION PERPENDICULAR TO THE ROOF TO A POSITION PERPENDICULAR TO THE HORIZONTAL IS THE "C" INPUT TO COMPUTER.

$$\text{TAN } C = \text{TAN } \alpha \text{ Cos } A_g$$

Figure 12-31. The geometry associated with a pendulum.

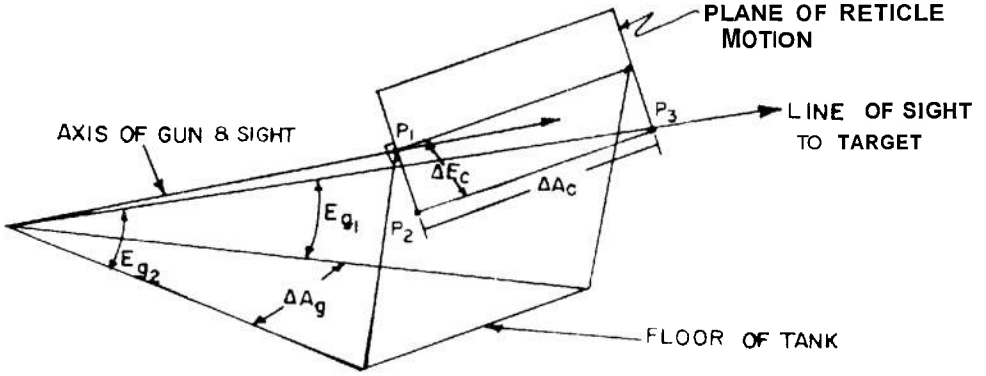


Figure 12-32. The arrangement of the reticle drive.

Once the amount that the gun tube has been moved from its original position has been computed, its new position in the tank coordinate system can then be computed from the following relationships:

$$A_{g_2} = A_{g_1} + \Delta A_{g_1} \qquad (12\text{-}30)$$

$$E_{g_2} = E_{g_1} + \Delta E_{g_1} \qquad (12\text{-}31)$$

The net effect of the gun tube's motion is shown in Fig. 12-33.

Obviously, as the gun traverses, its orientation with respect to the slope will change. When this occurs, the cant measured by the pendulum will also change. (As can be seen from Eq. 12-25, tan C is a function of A_g.) When the value of C utilized by the computer in Eqs. 12-21 and 12-22 changes, the values of ΔA_c and AE, outputted by the computer will also change. In actual practice, the pendulum output is changing while the gun is being traversed and, hence, the gunner can, with one motion, lay on the target. In a mathematical analysis, however, a computation necessitating several iterative steps is required. These steps are as follows:

Step 1. On the basis of the original wtapon position A_{g_1} the output of the pendulum C_1 is computed by Eq. 12-25.

Step 2. On the basis of ΔA_o, ΔE_o and C_1, the quantities AA,, and ΔE_{c_1} are computed by Eqs. 12-21 and 12-22.

Step 3. ΔA_{c_1} and ΔE_{c_1} are converted to ΔA_{g_1} and ΔE_{g_1} by Eqs. 12-28 and 12-29.

Step 4. With ΔA_{g_1} and ΔE_{g_1} determined, values of A,, and E_{g_2} are computed from Eqs. 12-30 and 12-31.

Step 5. As the vehicle has been traversed from A_{g_1} to A_{g_2}, the value of C has changed from C_1 to C_2.

Step 6. Steps 2, 3, 4 and 5 are repeated until the value of C derived from step 5 is the same as the value of C used in step 2. When this occurs, the iteration has been completed and the values of A_{g_n} and E_{g_n} computed in step 3 are considered correct values.

Once the final position of the weapon in the tank coordinate system (A_{g_n}, E_{g_n}) has been computed, the next step is to convert this position into the level coordinates (A_{o_n}, E_{o_n}). This is done by the equations

$$\sin E_o = \cos \alpha \sin E_g + \cos E_g \sin A_g \sin \alpha \qquad (12\text{-}32)$$

and

$$\tan A_o = \cos \alpha \tan A - \tan E_g \sec A_g \sin \alpha \qquad (12\text{-}33)$$

The final step in the accuracy analysis is to compute the actual movement of the weapon, i.e.,

$$\Delta A'_o = A_{o_n} - A_{o_1} \qquad (12\text{-}34)$$

$$\Delta E'_o = E_{o_n} - E_{o_1} \qquad (12\text{-}35)$$

and to compare these values with the values of AA, and ΔE_o originally chosen. The differences between these quantities are the errors in the cant-correcting mechanism.

Based on the procedure described, the theoretical performance of the cant-correcting capabilities of the XM17 system were computed. Two sets of conditions were assumed. These conditions are as follows:

Condition No. 1

Slope of hill $\alpha = 5^o$
Angular height of target above weapon $E_{o_1} = 5^o$
Elevation correction ΔE_o = 25 mils
Azimuth correction ΔA_o = 3 mils

Condition No. 2

Slope of hill $\alpha = 15^o$
Angular height of target above weapon $E_{o_1} = 10^o$
Elevation correction ΔE_o = 50 mils
Azimuth correction ΔA_o = 5 mils

The parameters of Condition No. 1 were chosen on the basis of estimates relative to "average" operating conditions. Condition No. 2 represents what is believed to be a severe, but nevertheless possible combination of operating conditions. In connection with Condition No. 2, it should he noted that 50

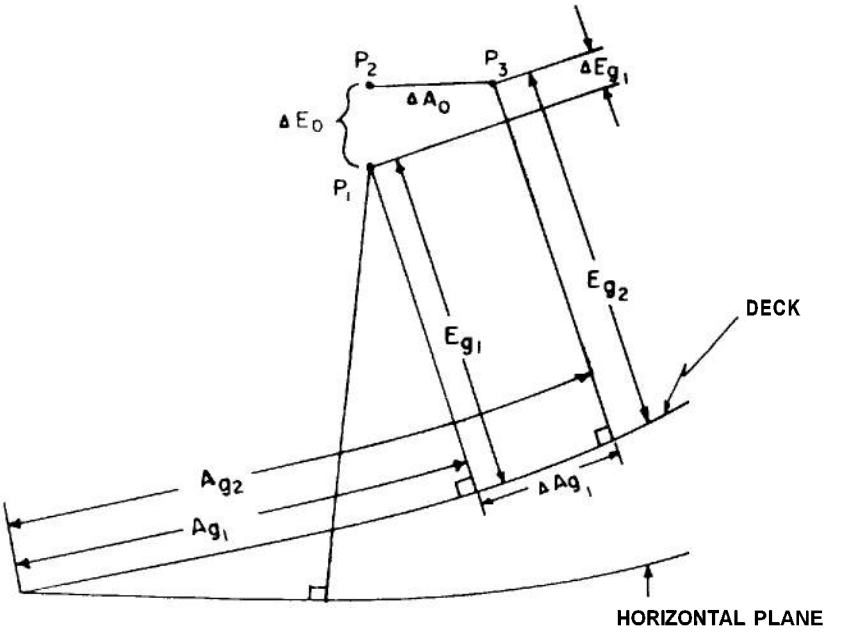


Figure 12-33. The movement of the gun from P_1 to P_3.

mils is the maximum elevation correction that can be accommodated by the existing XM17 system. Moreover 50 mils is more than adequate for engaging targets at realistic maximum range for even the slowest ammunition now under consideration. As previously described, drift is currently +,he sole contributor to AA,; 3 mils is a realistic value for this quantity.

As the error produced by the fire control system is dependent on how the tank is oriented with respect to axis X-X, computations were performed assuming the target was at various values of A,.

Fig. 12-34 shows the azimuth error produced by the XM17 Computer when it is operating under Condition No. 1. This error is plotted as a function of A,. As indicated, the maximum theoretical azimuth error is about 0.07 mil. The corresponding maximum theoretical elevation error under Condition No. 1 is about 0.02 mil. Figure 12-35 gives azimuth and elevation errors based on Condition No. 2. As indicated, the maximum theoretical azimuth error under Condition No. 2 is about 0.5 mil and the maximum theoretical elevation error is about 0.07 mil.

The matter of reducing azimuth errors (if this is ultimately deemed desirable) has been considered. This was done as described below.

The relationship expressed by Eq. 12-24 is

$$\tan A_g = \cos \alpha \tan A_t + \tan E_o \sec A_o \sin \alpha$$

Assume that E_O and A_O are to be changed by small angles AE, and AA,. This will, of course, result in a change of A, by the amount AA,. This is expressed by the equation

$$\tan (A_g + \Delta A_g) = \cos \alpha \tan (A_o + \Delta A_o) + \tan (E, + \Delta E_o) \sec (A, + \Delta A_o) \sin \alpha \quad (12\text{-}36)$$

If the changes (ΔA_g, AE,, ΔA_o) are small, it may be assumed that the higher powers of these quantities ($AA_{,}^2$ ΔA_g^3, etc.) are too small to be considered. If this is true, the trigonometric functions in Eq. 12-36 can be expressed by the first two terms of Taylor's Series, i.e.,

$$f(x + h) - f(x) + h \frac{df}{dx}$$

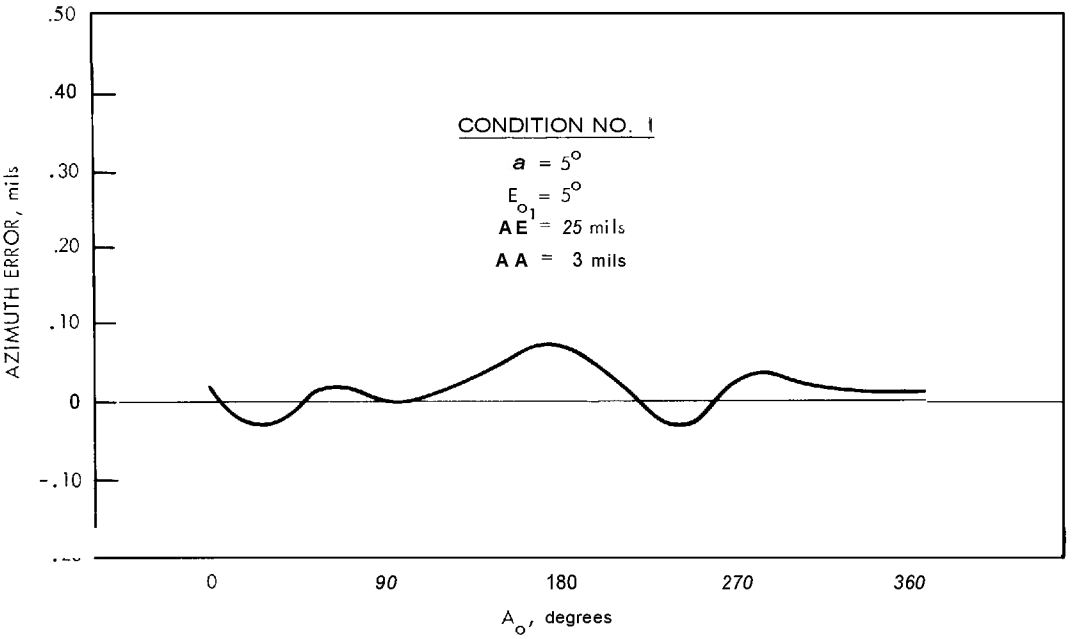


Figure 12-34. The theoretical azimuth error produced by the XM17 Computer when it is operating under Condition No. 1.

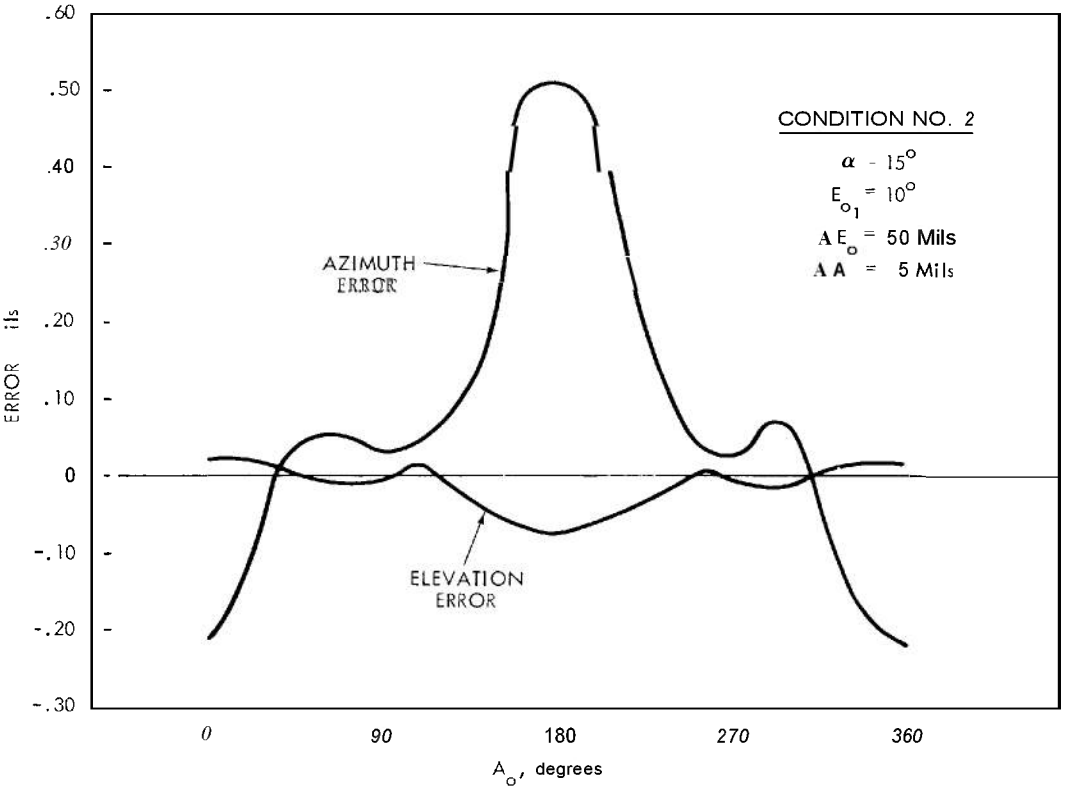


Figure 12-35. The theoretical azimuth and elevation errors produced by the XM17 Computer when it is operating under Condition No. 2.

Equation 12-36 then can be written in the form

$$\tan A_g + \Delta A_g \sec^2 A_g = \cos a (\tan A_o + \Delta A_o \sec^2 A_o) + [\tan E_o + \Delta E_o \sec^2 E_o] \left[\sec A_o + \Delta A_o \tan A_o \sec A_o\right] \sin a$$

Following through with this approach and dropping other small terms, it will be found that

$$\Delta A_g \cong \Delta A_o \cos a + \Delta E_o \sin C \sec^2 E_o \quad (12\text{-}37)$$

which, except for the cos α and the sec^2 E, term, is identical with Ey. 12-21, the azimuth cant-correction equation utilized in the XM17 Computer.

Fig. 12-36 shows the azimuth errors resulting under condition No. 2 from including the sec^2 E_o term alone and also both the cos a and sec^2 E, terms simultaneously. As can be seen from Fig. 12-36, including the sec^2 E, term alone will reduce the maximum azimuth error in Condition No. 2 from about 0.5 to about 0.3 mil; adding the *cos* a term as well reduces this error still further to 0.2 mil. While not shown on the curve, maximum elevation errors rise slightly from 0.07 to 0.12 mil.

The problem of whether to refine the computer by the substitution of Eq. 12-37 for Eq. 12-21 will depend on both the need for the refinement as well as the complexity introduced by the added mechanism. As it is mechanically undesirable to measure a and since α varies between 0 and 15°, one solution may be to utilize some average value of cos a such as 0.98. This will result in small errors (2% of ΔA) when a = 0 but will also reduce the error at a = 15° by the same amount.

A precise measurement of E, would add some Complexity to the equipment. Providing

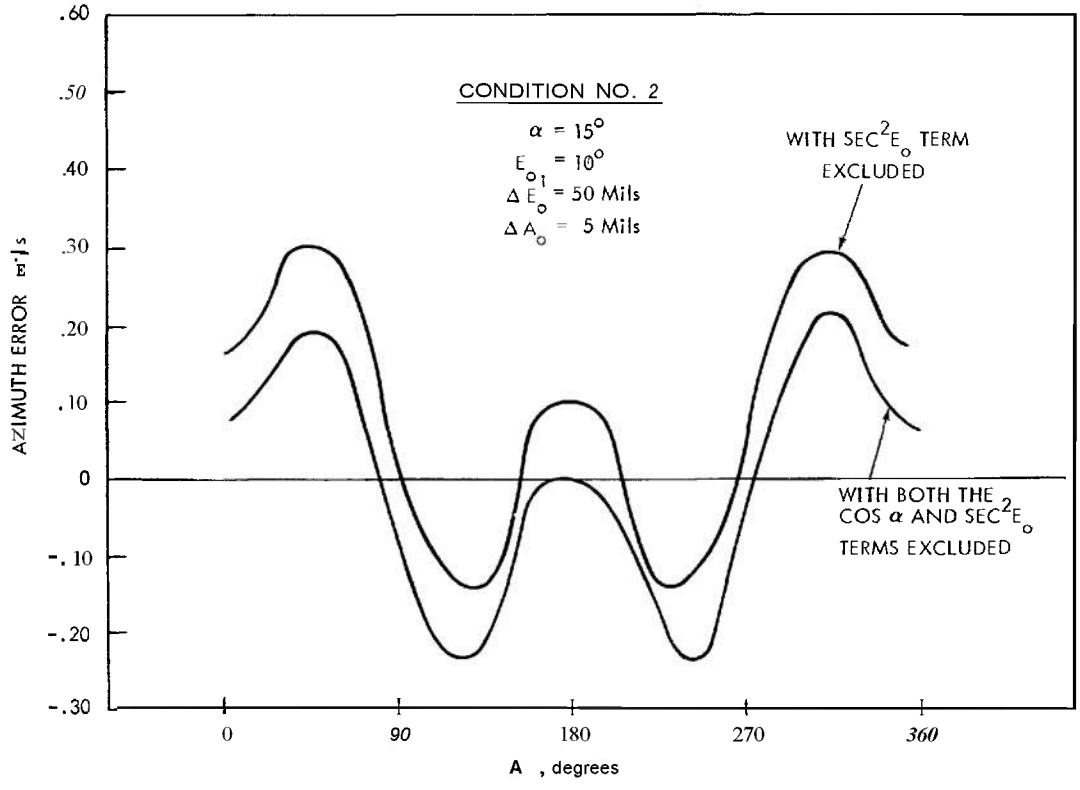


Figure 12-36. The azimuth errors that result under Condition No. 2 from excluding the sec^2 E, term alone and also the *cos* α and sec^2 E, terms simultaneously.

a mechanism for measuring E, would appear to be unwarranted except for the fact that ATAC has sponsored a study by Frankford Arsenal, relating to reduction of ballistic errors caused by assuming that trajectories do not change their shape, despite the fact that the target is above or below the weapon. As the mechanism required for eliminating this error requires measurement of angle of site E_s, it may be feasible to obtain the $\sec^2 E_s$ correction with very little extra equipment.

12-6.4 FACTORS TO BE CONSIDERED BEFORE UNDERTAKING AN IMPROVED DESIGN

The errors cited in the preceding discussion can be reduced by building a somewhat more sophisticated computer. Before embarking on such a program, one must ask the question, "Is better accuracy necessary when consideration is given to frequency and magnitude of the errors and variables that are introduced into the fire control problem under practical conditions?". At this point, a full system study of the tank gunfire problem would be in order. Such a study should consider specific existing or developmental guns, ammunition, and fire control systems. In addition, such a study should take into account the practical statistical variables and practical conditions encountered in a real system operating in the field. Some examples of these variables and conditions are: gunfire dispersion including those contributions due to meteorological and wind variations; errors in measuring target angle and target range; bias and random errors in the computer and data transmission links; the probable occurrence pattern of tank cant angle, target elevation angle, target azimuth angle, target range; performance reliability under field conditions of all active components in the system; and the importance of nonrandom variations to field operations.

Once such a study has been carried out in a careful manner, the relations of the many variables in the inevitable compromise that exists in any system will be better known. The improved understanding of those relations will provide a better basis for the application of any required engineering judgment, and an optimum system can be selected for development. Only those systems for which a definite field need has been established should be studied in this manner.

If such studies are carried out for future systems, three things are achieved. First there is a good estimate as to the hit probability of the system if it were developed. Second, the requirements or performance specifications including reliability of the components in the system are defined. Third, the development is initiated with confidence that the optimum approach has been selected.

APPENDIX 12-1
THE MATHEMATICS OF LEAD COMPUTATION

Figure A12-1.1 illustrates the lead-angle problem solved by the Gun Sight Mk 20. In this figure, superelevation is assumed to be zero and the axis of the gun and the gyro axis are therefore represented in the gun-target plane which includes the line of sight, the gun, and both the present and future positions of the target. The quantities represented by symbols in this figure are as follows:

B_a = bearing of the gun bore from any convenient reference line in the gun-target plane

B_p = bearing of the line of site from the same reference line

G_y = bearing of the gyro axis from the same reference line

$\frac{d(B_a)}{dt}$ = rate of angular movement of the gun

$\frac{d(B_p)}{dt}$ = rate of angular movement of the line of sight

$\frac{d(G_y)}{dt}$ = gyro precession rate

L = lead angle

L_g = gyro lag

S = sigma factor

R_p = present slant range

R_a = advance slant range

$\frac{dR_a}{dt}$ = rate of change of advance slant range

V = target speed

V_a = rate of movement of advance position

T_a = time of flight to advance position

a = angle between target course and line of fire

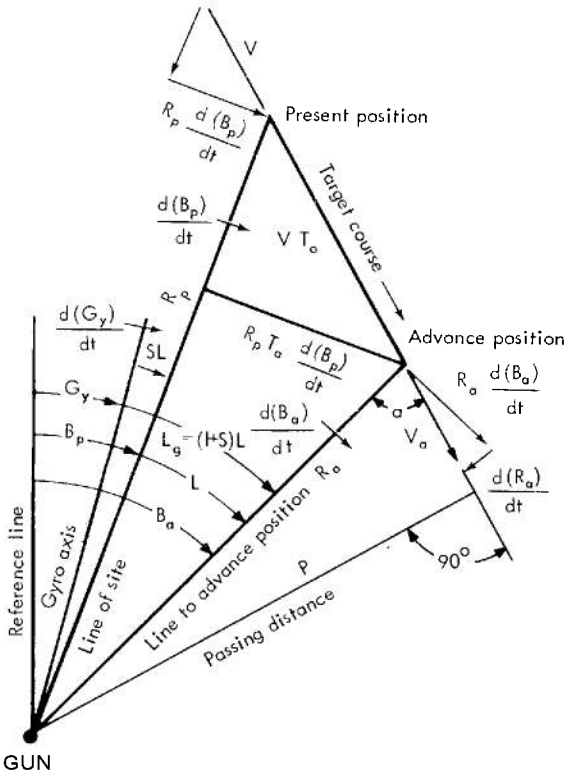


Figure A12-1.1 Diagram of the lead-angle problem solved by Gun Sight Mk 20.

In the gun sight, the gyro and reflector glass are mechanically connected in such manner that

$$L_g = (1 + S) L_p$$

where L_p is displacement of the line of site from the axis of the gun. Then the sight will solve the fire control problem if the gyro precession rate is controlled to make L_p equal to lead angle L for all points on the course of the target; i.e., the following equation must be satisfied:

$$L_g = (1 + S) L \tag{1}$$

Since the problem is solved by controlling the gyro precession rate, it is necessary to derive from Eq. 1 an expression for precession rate. From Fig. A12-1.1 and Eq. 1:

$$G_y \quad B_a - L_g$$

$$B_a = (1 + S) L$$

$$= B_a - (B_a - B_p) - SL$$

$$B_p = SL$$

$$\frac{d(G_y)}{dt} = \frac{d(B_p)}{dt} - \frac{S\,dL}{dt} \tag{2}$$

which is the desired expression for precession rate.

In the gun sight, precession rate depends upon the gyro lag L_g and magnet current I in accordance with the following equation:

$$\frac{d(G_y)}{dt} = KI^k \sin \frac{L_g}{K_s}$$

in which K, k and K_s are constants. K_s can be controlled by design of the magnetic structure and in practice is made equal to (1+S). Thus $L_g K_s = L$. To simplify the equation, also substitute $1/T_n$ for KI^k. The equation then becomes

$$\frac{d(G_y)}{dt} = \frac{\sin L}{T_n} \tag{3}$$

The quantity T_n is called nominal time of flight and it can be given any desired value by varying the magnet current. The problem, then, is to find values of T_n which make the precession rate in Eq. 3 the same as that in Eq. 2 for any point on the target course. Combining Eqs. 2 and 3 and solving for T_n shows that

$$\frac{d(B_p)}{dt} \quad \frac{dL}{dt} \quad \frac{\sin L}{T_n}$$

$$T_n = \frac{\sin L}{\dfrac{d(B_p)}{dt} - S\dfrac{dL}{dt}} \tag{4}$$

which is the value sought. However, Eq. 4 is not in convenient form for numerical computation. It is desirable to express it in terms of quantities which can be easily determined from range tables and an assumed target course and speed. From Fig. A12-1.1,

$$L = B_a - B_p$$

$$\frac{dL}{dt} = \frac{d(B_a)}{dt} - \frac{d(B_p)}{dt}$$

Eq. 4 then becomes

$$T_n = \frac{\sin L}{\dfrac{d(B_p)}{dt} + S\left[\dfrac{d(B_p)}{dt} - \dfrac{d(B_a)}{dt}\right]} \tag{5}$$

To obtain a value of $d(B_p)/dt$, consider the small triangle at the present position in Fig. A12-1.1 in which the velocity of the target V has been resolved into two components: one parallel to the line of site and the other at right angles to it. The right-angle component is

$$(R_p) \quad \frac{d(B_p)}{dt}$$

Then, in the similar right triangle having VT_a as its hypotenuse, the base must equal

$$R_p \; T_a \quad \frac{d(B_p)}{dt}$$

and

$$\sin L \quad \frac{R_p \; T_a}{R_a} \quad \frac{d(B_p)}{dt}$$

Therefore,

$$\frac{d(B_p)}{} = \frac{R_a}{} \; \frac{\sin L}{} \tag{6}$$

To obtain a value for $d(B_a)/dt$ it is necessary to consider the movement of the advance position. The advance position moves along the target course ahead of the target, but since T_a decreases as R_a decreases, the distance VT_a between the present and advance positions is not constant. The advance position therefore moves at a rate V_a which is equal to target speed plus the rate of change of VT_a

$$V_a = V + \frac{d(T_a)}{dt} = V + V \; \frac{d(T_a)}{d(R_a)} \; \frac{d(R_a)}{dt} \tag{7}$$

In the small triangle at the advance position in Fig. A12-1.1, the velocity V_a has been resolved into two components: one parallel to the line between the gun and the advanced position, and the other at right angles to it. The parallel component is

$$\frac{d(R_a)}{dt} = V_a \cos a \tag{8}$$

Substituting Eq. 8 in Eq. 7 and solving for V_a gives

$$V_a = V + V \frac{d(T_a)}{d(R_a)} V_a \cos a$$

$$V_a = \frac{V}{1 - V \dfrac{d(T_a)}{d(R_a)} \cos a} \tag{9}$$

The right-angle component of V_a is

$$R_a \frac{d(B_a)}{dt} = V_a \sin a$$

$$= \frac{R_p}{VT_a} V_a \sin L \tag{10}$$

Substituting Eq. 9 in Eq. 10 and solving for $d(B_a)/dt$ yields

$$\frac{d(B_a)}{dt} \frac{R_p}{R} \frac{\sin L}{\left[1 - V \dfrac{d(T_a)}{d(R_a)} \cos a\right]}$$

Substituting Eqs. 6 and 11 in Eq. 5 shows that

$$T_n = \frac{T_a}{\dfrac{R_a}{R_p} + S\left[\dfrac{R_a}{R_p} - \dfrac{R_p}{R_a}\left(\dfrac{1}{1 - V \dfrac{d(T_a)}{d(R_a)} \cos a}\right)\right]} \tag{12}$$

Eq. 12 can be used to compute T_n for any given value of R_a and for an assumed target course and speed. For each value of R_a, T_a can be taken from the range tables. The derivative $\dfrac{d(T_a)}{d(R_a)}$ is the rate of increase of time of flight with respect to range, expressed in seconds per yard. It can be obtained with sufficient accuracy by comparing the times of flight at points in the range table at each side of R_a. V is the assumed target speed in yards per second. Angle a can be determined from the expression

$$\sin a = \frac{P}{R_a}$$

where P is the assumed passing distance of the target course. R_p can be computed from either of the following expressions:

$$R_p = \sqrt{P^2 + (VT_a + R_a \cos a)^2}$$

$$R_p = \sqrt{(R_a)^2 + 2 R_a VT_a \cos a + V^2 (T_a)^2}$$

Since the sigma factor S is not the same for elevation and traverse, an intermediate value must be used in Eq. 12. In the Mk 20 Sights this value is 0.25.

By computing R_a and T_a for a number of values of R_a, a curve can be plotted giving T_n for corresponding values of R_a. It is then possible to adjust the position of the range magnet and to calibrate the magnet circuit in such manner that $KI^k = 1/T_n$ for any value of R_p set into the range box. When this has been done the sight computes the lead angle for all values of present ranges.

APPENDIX 12-2
CALIBRATION CHARACTERISTICS OF GUN SIGHT MARK 20 MOD 6 AND DATA ON LEAD ANGLE AND TIME OF FLIGHT FOR 20 MM BALLISTICS

The curves shown on Figs. A12-2.2 through A12-2.5 were calculated from 20 mm ballistics to determine the desirability of adapting the gun sight to the different types of target approach shown in Fig. A12.2.1.

From this investigation, it has been concluded that making the sensitivity and superelevation moment a function of the type of approach is unnecessary. For the assumed tactical use of the gun sight at ranges up to 2000 yards, such features are of little value. Furthermore, this inclusion would complicate the sight mechanically and would probably increase the class A errors. The complete problem is analyzed in the paragraphs which follow.

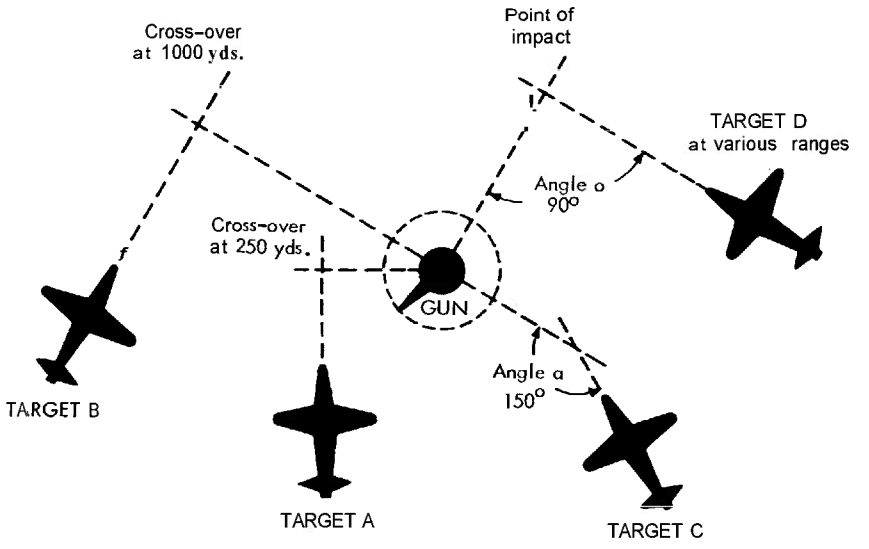
Curves are included for sensitivity and superelevation for target speeds of 300 and 600 miles per hour, computed for the four types of target approach shown in Pig. A12-2.1 and defined below:

1. Curve A—Incoming solution for a target flying a straight course and passing the gun at 250-yards minimum range (Target A).
2. Curve B—Incoming solution for a target flying a straight course and passing the gun at 1000-yards minimum range (Target B).
3. Curve C—Solution for a series of targets at various ranges, each flying so that the angle between target path and line of sight to the advance position (angle a) is 150^{o}. This curve shows the correct sensitivity and superelevation during the incoming part of each path (Target C).
4. Curve D-solution for a series of targets at various ranges, each flying so that the angle between target path and line of site to the advance position (angle a) is 90^{o}. This curve shows the correct sensitivity and superelevation during the cross-over part of each path (Target D).

It should be noted that, at long distances, the first three curves of each set are the same. That is, even a target heading for cross-over at 1000 yards is, prediction-wise, an incoming target while at long range. As range decreases to 1000 yards, curve B approaches cross-over curve D. As range decreases to 250 yards, curve A also approaches curve D.

Thus, it appears that if the range setter had a control box producing two calibrations corresponding to curves B and D, and if during the course of firing at a target he used the curve B calibration when the target path was directed principally along the line of sight and the curve D calibration when it was principally across the line of sight, then he would obtain the best overall results. It will be noted that these two calibrations are sufficiently far apart to justify different calibrations of the sight.

However, it seems more likely that in actual use the operation would not be as described above. Rather, the range setter would probably establish the type of problem, when the target was acquired, as incoming, if the attack were directed against his own area, or cross-over, if it were not, and would have little opportunity to make further changes during the problem. For most cases, the curve D calibration would be unsuitable for this kind of use because, for any target with a cross-over distance of 1000 yards or less, most of the problem would still be essentially incoming and would fall in the closely grouped



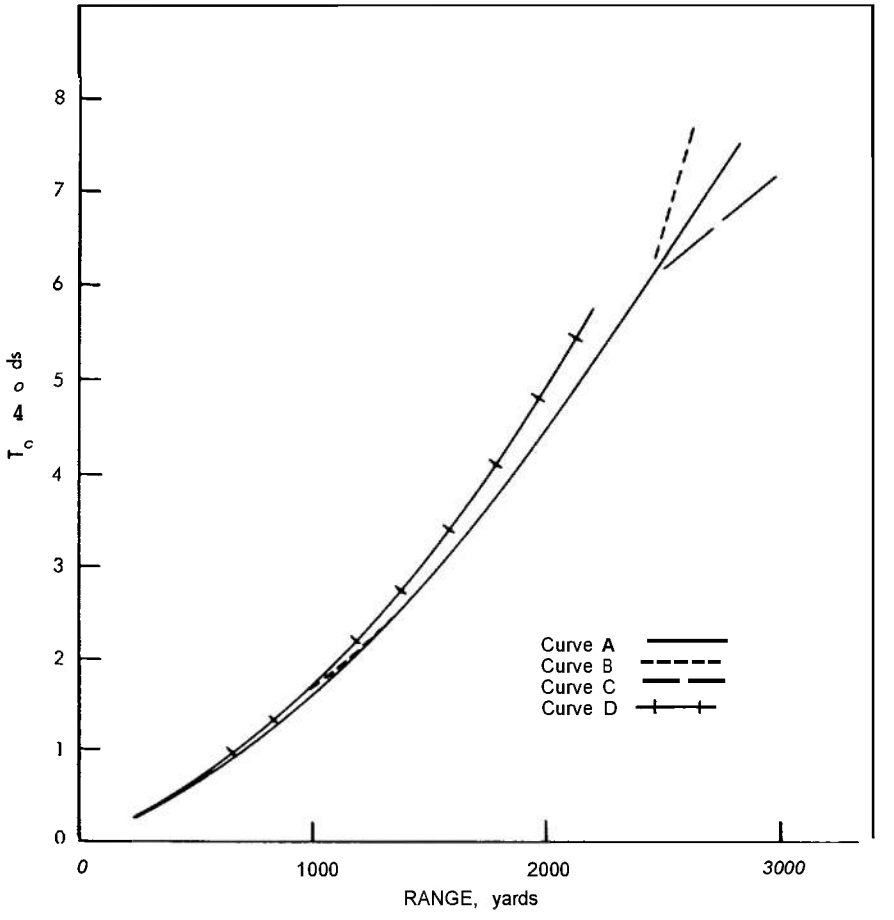


Figure A12-2.2. T_n vs range for a target speed of 300 mph.

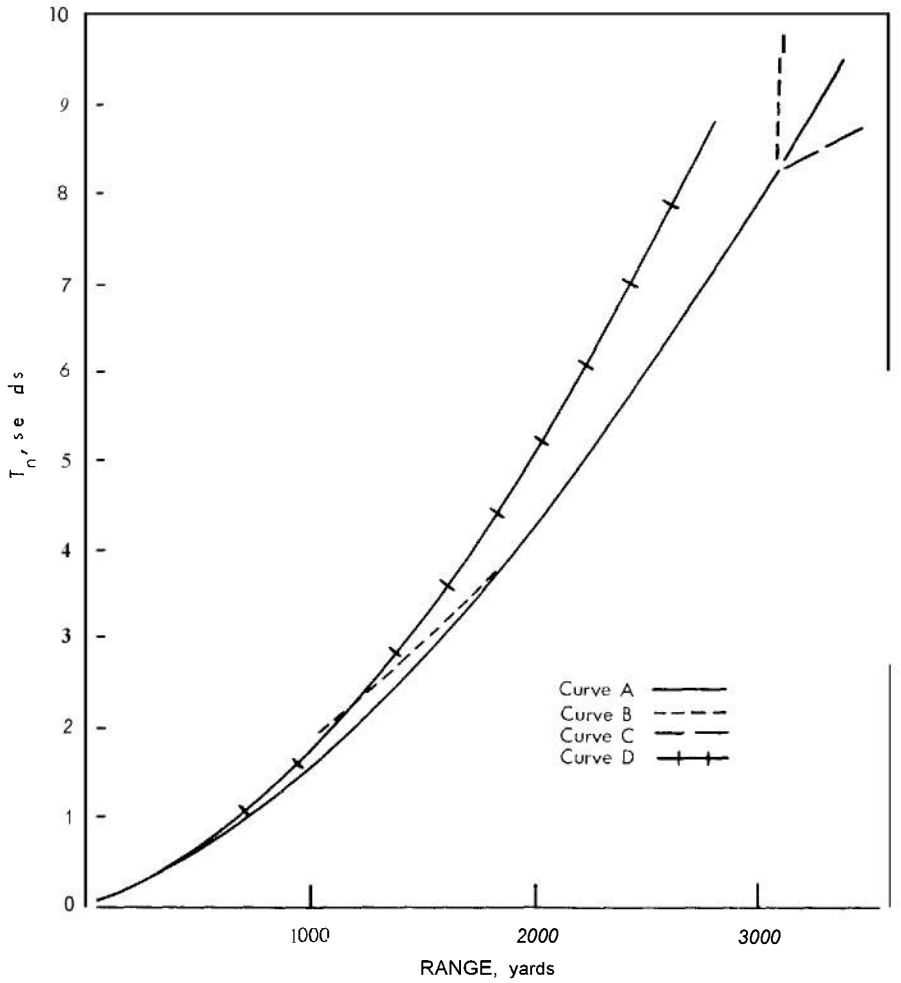


Figure A12-2.3. T_n vs range for a target speed of 600 mph.

set of curves A, B, and C. The only cases in which the curve D calibration could properly be used during the entire problem would be on targets with cross-over ranges of over 1500 yards. These are probably of little importance for 20 mm guns.

This manner of setting agrees with training practice, where an incoming course is considered as one which comes directly toward tlie gun, while a cross-over course is one where the target passes at perhaps 500 yards.

Thus, it appears that if a single setting is to be made for each complete problem, the best results will be obtained from curves A and B for cross-over distances of 250 and 1000 yards, or from curve C which is a good approximation of both. These curves are all so near to each other that there seems no justification for providing more than one calibration.

Comparison of the curves for 300 mph and 600 mph shows that both sensitivity and superelevation, particularly the latter, depend largely on target speed. It appears desirable to include some means of malting operating adjustments to the superelevation moment in order to make the sight adaptable to combat conditions.

To accomplish this, a target-speed knob has been included on the sight. This knob adjusts both superelevation and sensitivity for target speeds from 200 to 600 knots. The change of sensitivity with target-speed setting

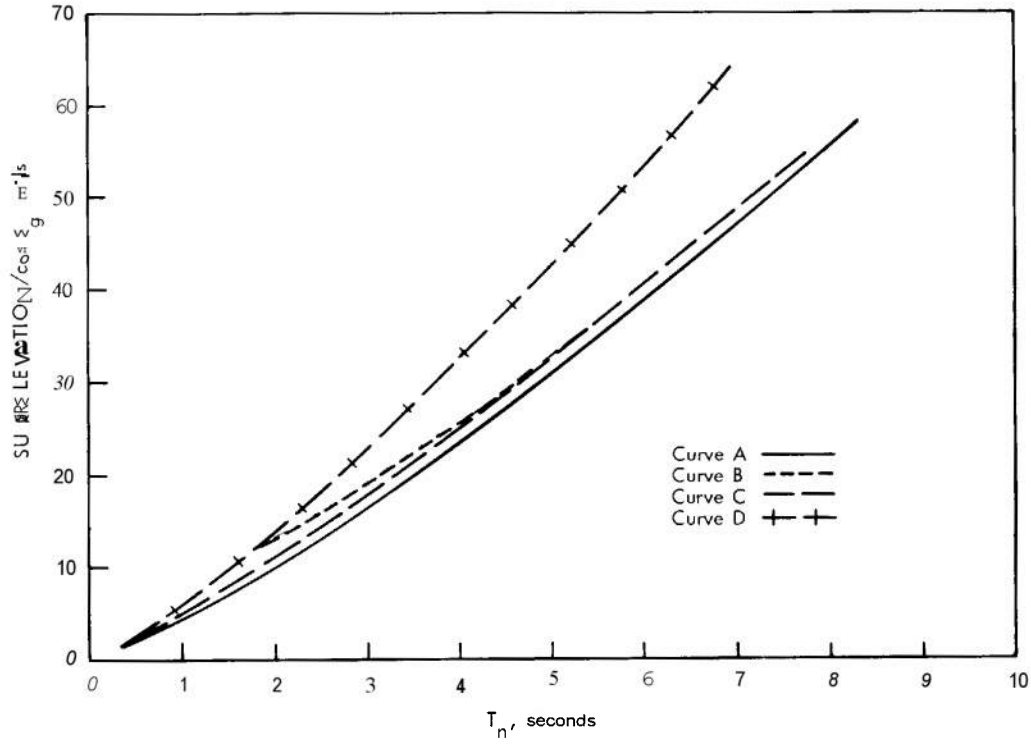


Figure A12-2.4. Superelevation vs T_n for a target speed of 300 mph.

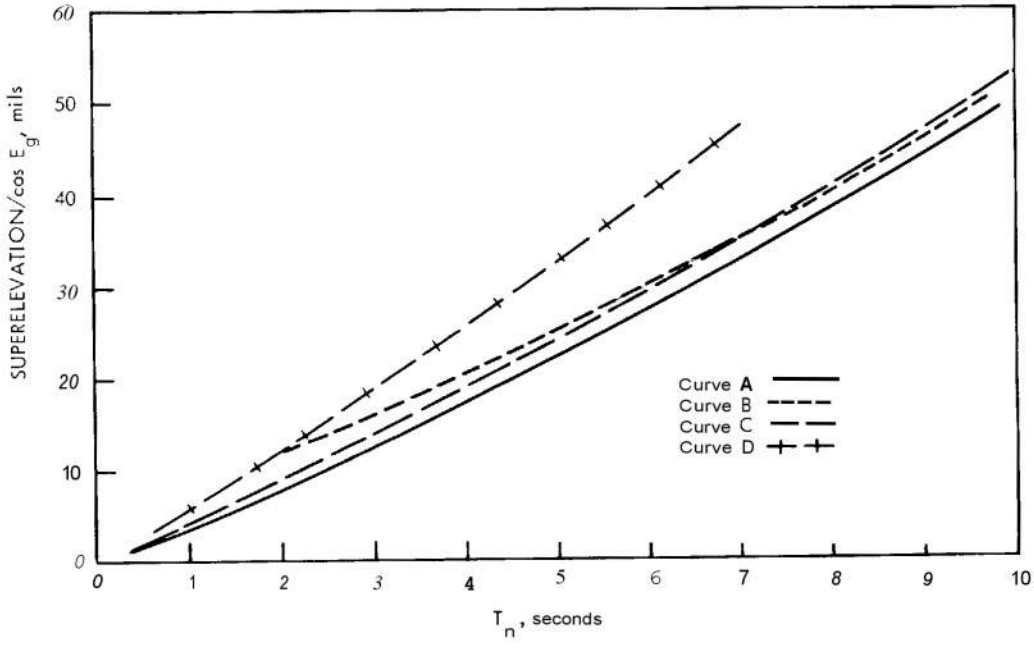


Figure A12-2.5. Superelevation vs T_n for a target speed of 600 mph.

is shown on Fig. A12-2.6 and the change of superelevation on Fig. A12-2.7.

In addition, a self-restoring switch was incorporated in the range box. When pushed, this switch changes the sight sensitivity for targets which have passed cross-over to that shown by the solid line curve on Fig. A12-2.8. The theoretical curves for each target speed are shown dashed. In the past, sights had been calibrated only for the incoming portion of the target's path, and, since the required outgoing sensitivity curve is quite different, no hits could be expected once the target has passed cross-over. Use of this switch will extend the useful coverage of the sight. The superelevation moment is not changed for outgoing targets because of the increased complication of the mechanism.

The curves shown on Fig. A12-2.9 indicate the relationships which exist between elapsed time and lead angle for target speeds of 200, 400, and 600 knots. The sight solution for outgoing targets is shown by the dashed curves. The target is assumed to be on a straight course with 500 yard minimum cross-over range. A second presentation of this data is made on Fig. A12-2.10, in which lead angle is plotted against present range.

Curves of maximum lead angle against cross-over distance are given on Fig. A12-2.11 for target speeds of 200, 400, and 600 knots. Bear in mind that the lead angle limits are 25^{o} in Gun Sight Mk 20 Mod 6; therefore, the target conditions for which the gun sight provides a solution are easily visualized from Figs. A12-2.9 through A12-2.11.

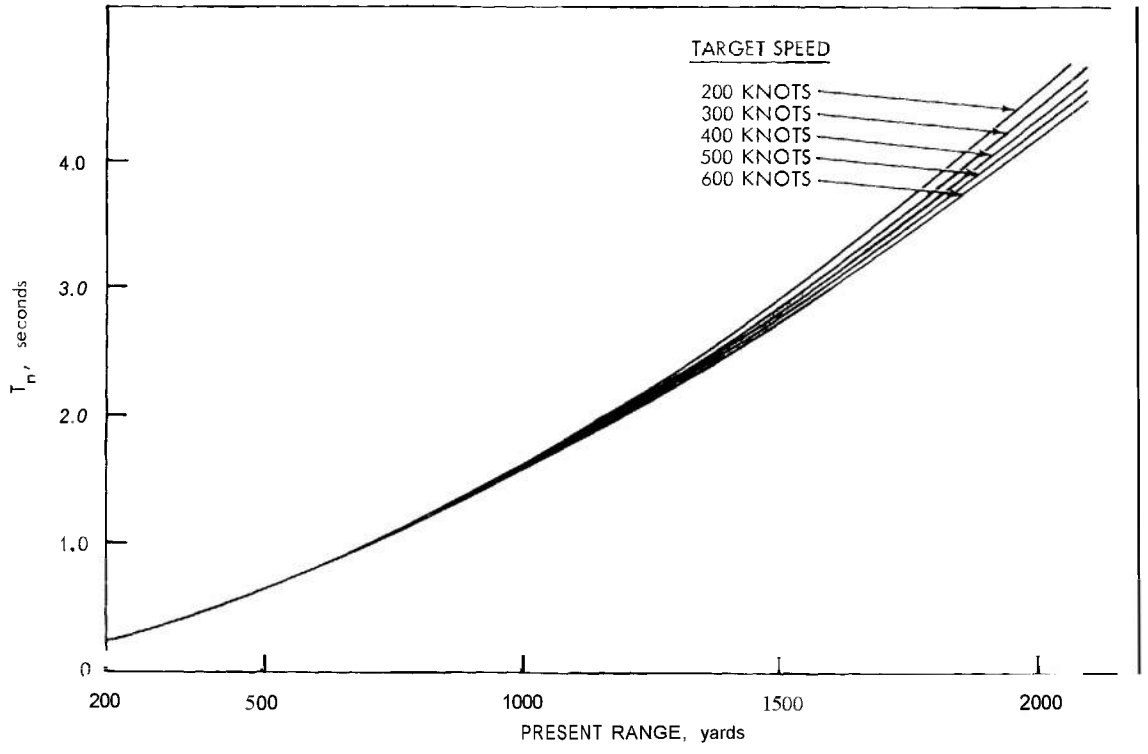


Figure A12-2.6. T_n vs present range for an incoming target.

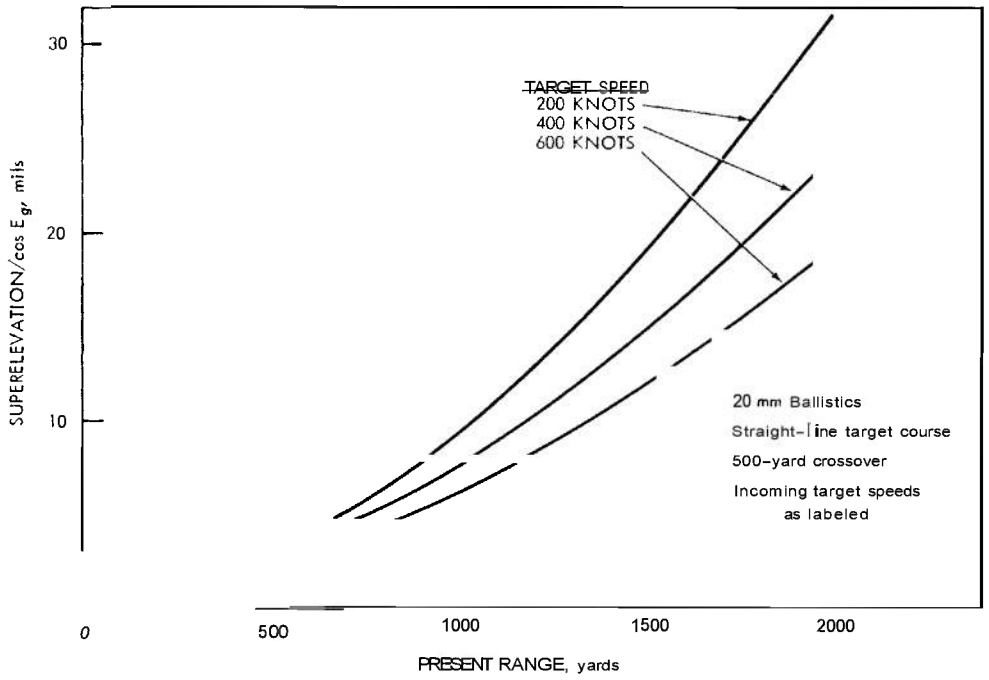


Figure A12-2.7. Superelevation vs range.

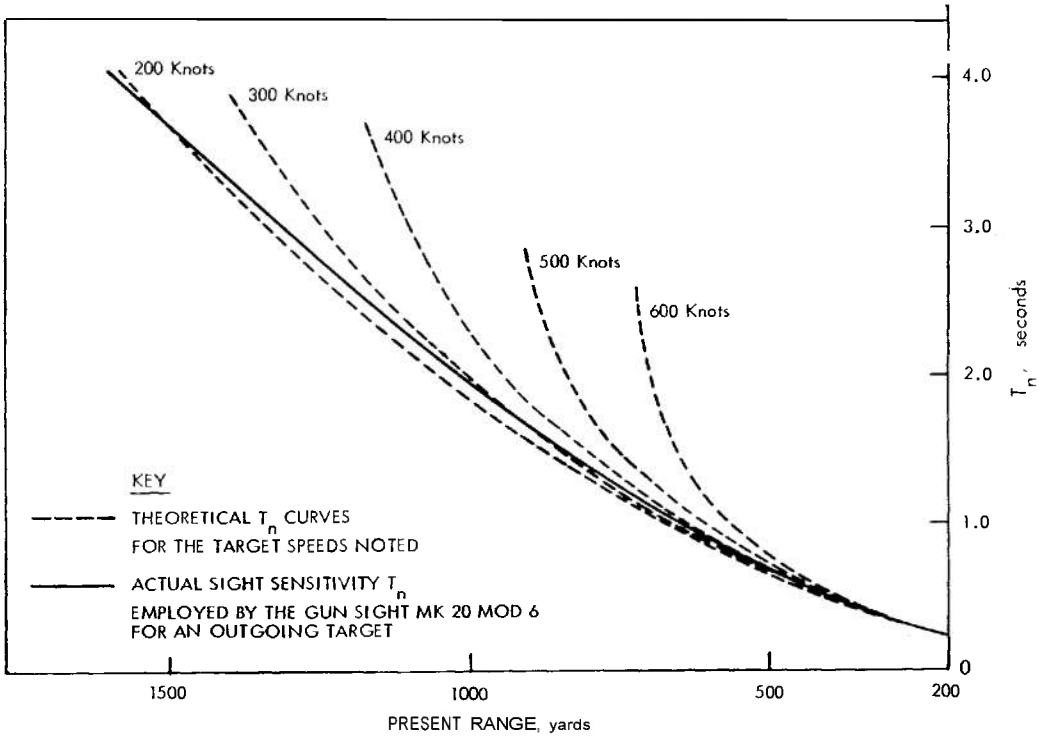


Figure A12-2.8. T_n vs present range for an outgoing target.

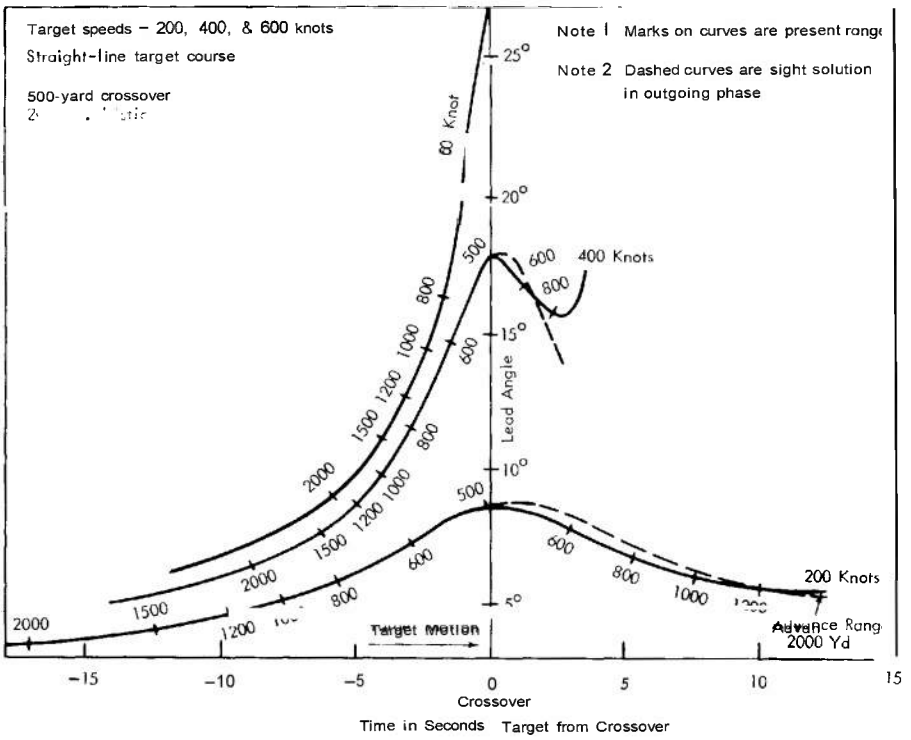


Figure A12-2.9. Lead angle vs time.

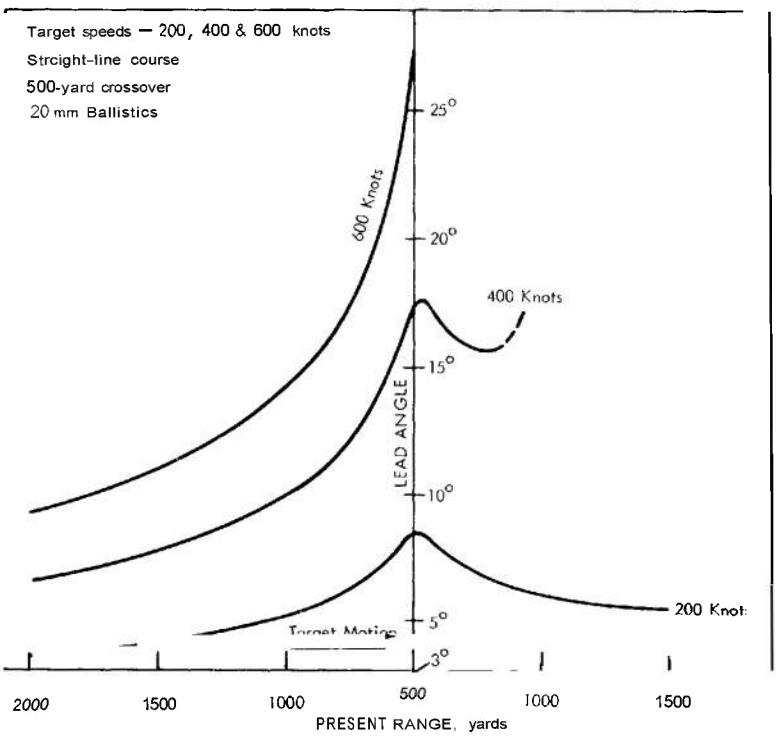


Figure A12-2.10. Lead angle vs range.

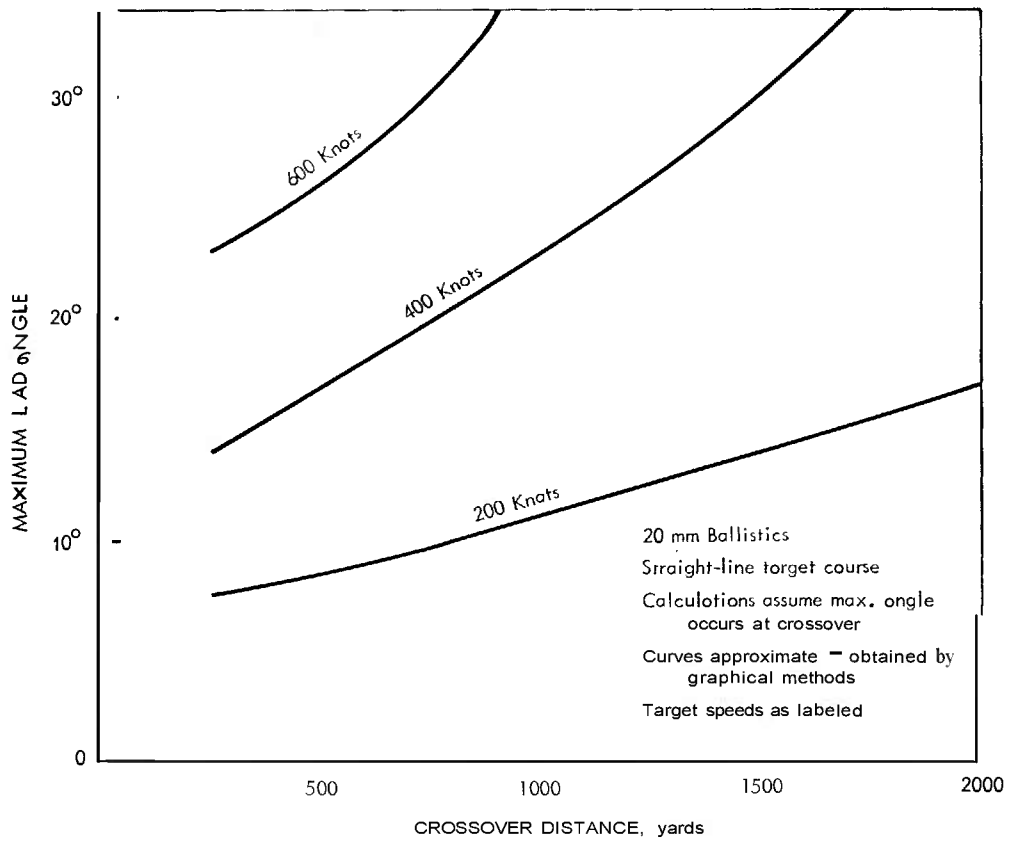


Figure A12-2.11. Maximum lead angle vs crossover.

REFERENCES

1. Notes on Development Type Materiel – Gun Data Computer T29E2 and Related Equipment, Frankford Arsenal Report No. FCDD-324, Sept. 1957 (Unclassified).

2. T. v. Karman and M. A. Biot, Mathematical Methods in Engineering, McGraw-Hill Book Co., Inc., N.Y., 1940.

3. Lightweight Fire-Control Equipment for Rocket Launchers, Frankford Arsenal Report No. FCDD-351, Dec. 1958 (Unclassified).

4. AMCP 706-331, Engineering Design Handbook, Fire Control Series, Compensating Elements.

5. Gun Sight Mark 20 Mods 6 and 7, Maintenance Manual, OP1870A, Bureau of Ordnance, Department of the Navy, Washington, D.C., December 1952.

6. W. Sperling and J. C. Glynn, Ballistic Computer XM17 – Analysis of Cant Correction Accuracy, Report M64-1-1, Frankford Arsenal, U.S. Army, July 1963.

7. AMCP 706-327, Engineering Design Handbook, Fire Control Series, Section 1, Fire Control Systems – General.

CHAPTER 13
EXAMPLE OF A PROBLEM INVOLVING THE INTERCONNECTION OF A COMPLEX SYSTEM

13-1 INTRODUCTION

One of the problems considered in Chapter 12 was the fire control problem for field artillery. In Chapter 12, this problem was solved by the use of a simple analog computer in which the ballistic equations of motion were approximated by a truncated Fourier series. The same problem can be solved with greater accuracy and flexibility by means of a digital computer with, however, considerable increase in complexity. A digital computer which has been designed to perform this function is the FADAC (Field Artillery Digital Automatic Computer).

The basic problem is the same as that considered in Chapter 12. Data on the weapon characteristics and on the firing site are set into the memory of the digital computer in advance of the computation. Just prior to the computation, data for the individual mission--i.e., specific target, projectile, and charge data--are entered into the memory. Provision is made for correction of data from observations of firing.

The target data are accommodated in any one of three modes:

(1) Mode A - Target location is specified in the standard grid system; the components being designated Easting, Northing, and Height.

(2) Mode B - Target location is specified by the range, azimuth, and vertical angle from an observation post of known location.

(3) Mode C - Target location is specified with respect to a reference point of known position, the coordinates being the distance from the target to reference along the observer's sight line, the lateral displacement of the reference from the sight line, and the difference in height between the target, and the reference point.

The various coordinates used are shown in Fig. 13-1. The geometrical conversions are readily carried out by the computer. Data in the form of Modes B or C are converted to the grid system as in Mode A. Next, the known location of the battery is used to compute the target range, azimuth, and height, as referenced to the battery. This information is then used in the trajectory calculation.

In a digital computer, trajectory computations can be carried out by step-by-step integration. It is not necessary to approximate the trajectory as in the analog computer. In the analog computer, the approximate trajectory was solved by a closed-loop computation (see Fig. 12-3), in which quadrant elevation is continuously adjusted to match the range and other input parameters. In the digital computer, the computation starts with an initial assumed quadrant elevation; determines the corresponding range; and compares this computed range with the actual range. The difference in range is used to determine a new quadrant elevation, and the trajectory computation is then repeated until the range error has been reduced to the desired accuracy. The digital computation is thus more time- consuming than the analog, but provides greatly increased accuracy, both from the elimination of most of the approximations as well as from the increase in the accuracy of individual computing operations. Moreover, the increase in time is on the order of a few seconds and is negligible in the time frame of field artillery operations.

A choice is possible between a general-purpose digital computer and a digital differential analyzer (DDA) in this application. The DDA would be less complex and would have greater speed in the solution of the trajectory

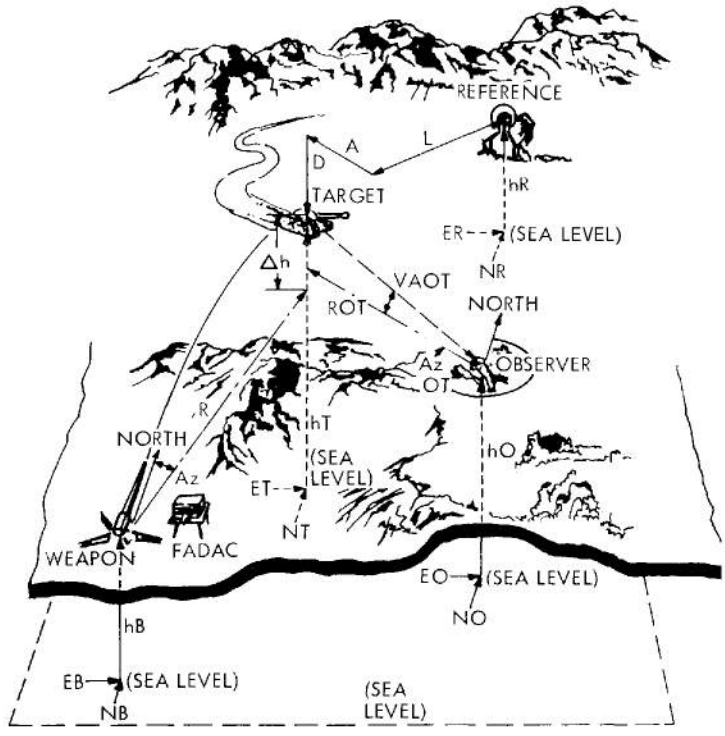


Figure 13-1 Coordinates used in solving artillery problems.

problem. The DDA would, however, be far less flexible than the general-purpose computer in performing auxiliary computations and in the ability to adapt to new weapons or new data formats. The reason for this difference lies in the fact that the DDA is preprogrammed with a particular problem setup and can be reprogrammed only by wiring or component changes. The general-purpose computer, on the other hand, has a stored program. Reprogramming is a simple matter: the new data are read into the computer memory by means of a punched paper tape. Thus, the general-purpose computer can perform surveying computations or even compute a payroll when not required for its primary function.

In summary, the FADAC performs many more functions than an analog computer in a unit which is comparable in size, but considerably more complex. The FADAC has greater accuracy and flexibility than is possible with an analog computer, and is more flexible than a DDA.

The paragraphs which follow provide a more detailed description of the computations performed by FADAC and a brief description of the general layout and circuit design of the computer.

13-2 TRAJECTORY COMPUTATIONS

The FADAC is a general-purpose transistorized digital computer. Programs are entered through a high-speed photoelectric punched-paper-tape reader contained in a separate unit, the Memory-Loading Unit. Specific inputs are entered through a mechanical tape reader and a manual keyboard, both of which are integral with the computer. Output is via a Nixie-tube numerical indicator readout, or by teletype.

The field artillery fire control problem is solved in three steps: data entry, geometrical and trajectory computations, and correction of input data.

Input data are entered by the program tapes for permanent storage of weapon char-

acteristics, and by either the keyboard or the mechanical tape reader::: for semipermanent storage of firing-site data and individual mission data. Program tapes have been prepared for the following weapon types:

105 mm howitzer 175 mm gun
155 mm howitzer HONEST JOHN rocket
8 inch howitzer LITTLE JOHN rocket
280 mm gun

Addition of other weapon types to this inventory is done by programming the weapon characteristics and preparing a new tape.

Data are required for all possible combinations of gun, projectile, and charge. The data stored internally are:

Standard muzzle velocity
Maximum range
Projectile drag function
Ballistic coefficient function
Drift function
Time-fuze-setting function
Relationship of muzzle velocity to powder temperature
Relationship of muzzle velocity to projectile weight

The semipermanently-stored firing-site data are obtained by surveying techniques at the actual site, and are expressed in the standard grid system. The data required are:

Location of weapons
Location of observation points
Location of fixed reference points
Meteorological data
Measured or calibrated muzzle velocity
Projectile ballistic coefficient factor

Meteorological data must be updated from time to time. Normally, gross changes are entered every two hours, or as new data become available.

The data required for each mission are:

Target location**
Projectile weight**
Powder temperature::::: (when available)
Projectile and fuze type

The starred items are subject to correction in the third step of the computation.

The general plan of the trajectory computation is straightforward. An initial quadrant elevation is assumed. The ballistic equations are then integrated step-by-step until the trajectory intersects the horizontal plane passing through the target. Any of the numerical integration techniques described in Chap. 2 can be employed. The computed range is then compared with the input range, and the difference is employed to adjust the quadrant elevation to a new trial value. The computation then iterates until the computed range approaches the input range to within the desired accuracy. The required settings are then displayed on the readout. In general, either a high-angle or low-angle trajectory may be employed.

The geometrical data which locate the target and reference points are converted to the grid system, if they were determined by Modes B or C. The grid coordinates are then converted to target range, azimuth and height, with the weapon location as reference.

The ballistic equations are first solved in a range-height coordinate system (i.e., in a vertical plane intersecting the weapon and target). A correction is made for lateral motion after completion of the trajectory computation. This order is followed since it results in a simpler set of equations.

The forces acting on the projectile which are considered in the integration of the ballistic equations are gravity, aerodynamic drag, and the Coriolis force due to the earth's rotation. The gravity force is assumed to be invariant with height. The aerodynamic drag is assumed to be opposite to the velocity vector and to be a function of the relative velocity of the projectile with respect to the air, the air density, the ballistic coefficient of the projectile corrected for its weight, and the drag function of the projectile. The drag function in turn depends upon air temperature and relative velocity. Of these independent variables, the projectile velocity is taken as the last value computed, starting from the input muzzle velocity. The wind, air density, and temperature are interpolated from the meteorological input data. The remaining quantities are available as input data. The Coriolis force is a function of the projectile velocity, the latitude, and the azimuth of the plane of the trajectory. From the vector sum of these forces and the known weight of the projectile, the acceleration is determined. The acceleration is then integrated over a

* The mechanical tape reader is employed only for the introduction of meteorological data.

time increment to obtain the velocity, and the velocity integrated to obtain the position.

At each step of the integration, the vertical position is compared with the height of the target. If the projectile is above the target, the computed velocity and position are employed in the next step of the integration. If the projectile falls below the target, the range error is determined at this point. The quadrant elevation is then corrected by a function which relates it to range and height of the target. Generally, only two or three iterations are required to converge to the final trajectory.

The lateral corrections are now performed. These corrections are functions of projectile spin; earth's rotation; ballistic value of the crosswind; and the computed time of flight, quadrant elevation, and target azimuth.

The final step in the computation is the application of correction terms to the input data. One of these correction schemes, known as replot, is employed when a target has been successfully hit. The computed trajectory is stored in the memory, and the procedure determines just where, along this arc, the target is located. The replot function is performed by FADAC in the following manner:

(1) The rectangular coordinates of the target just hit, stored in memory, are displayed.
(2) The operator plots the Easting and Northing on a contour map and compares the height with the map contours.
(3) If the height does not agree with the contour map, the operator enters a new height into the computer.
(4) The program extends the previously-computed trajectory to the new height and stores the resulting target coordinates.
(5) Steps 1 through 4 are repeated until the operator is satisfied that the target coordinates match the contour map.

The other correction scheme is known as registration. In the registration correction, small adjustments in the weapon settings are made from observations of firing. In precision registration, a target of known location is bracketed. The corrections for deflection and fuze time thus determined are used unchanged for other targets, and are simply added or subtracted at the end of the computational procedure. The range correction is adjusted proportionately for use with targets at other ranges.

An alternative method of registration is known as high-burst registration. In this method, two observers report the azimuth and vertical angles of the center of a burst pattern. The corrections thus determined are applied in the manner just described. A correction for the assumption of a constant ballistic coefficient is applied to the time-of-flight result.

After completion of the trajectory calculations and application of corrections, the results are displayed in the visual readouts. These outputs are deflection, quadrant elevation, powder charge, and fuze time if a time or delay fuze is employed.

13-3 COMPUTER DESIGN

FADAC is a small-scale, general-purpose computer with special features, particularly in the inputs and outputs, which make it more applicable to field-artillery fire control. The significant specifications are given in Table 13-1.

As in any general-purpose computer, the major divisions of the computer are the input, output., arithmetic unit, memory, and control. These divisions are illustrated in Fig. 13-3. Both input and output accommodate punched paper tape in both teletype and ASCii* codes, as well as magnetic tape and other FADAC's. In addition, the control panel provides switches for input, and numerical displays for output. Output is also available on external lines.

The control unit stores instructions read from the memory, and controls the routing of data and the operation of the other units. The control unit includes an instruction (I) register in which instructions read from the memory are stored. Instructions are interpreted by the control unit which then reads the required data from memory into the ar-

* American Standard *Code* for information interchange.

TABLE 13-1. FADAC SPECIFICATIONS.

a. Type. General-purpose, transistorized digital computer; serial by bit, parallel by function, allowing 12,800 one-word execute (add, subtract, etc.) operations per second.

b. ** eight. Approximately 210 pounds.

c. Size. Approximately 5 cubic feet. (For dimensions, see Fig. 13-2).

d. Power. Three-phase, 4-wire, 400-cps system; 120/208 volts, approximately 750 watts.

e. Temperature. -25" to 125°F(external ambient at sea level); with rear cover installed, to -40°F. Automatic temperature protection is provided.

f. Commands. One command per word; each command contains both address of operand and address of next command (1 + 1 system).

g. Numbers. Straight binary for internal operations; automatic conversion to other codes for input-output; two's complement notation for negative numbers.

h. Word Length. Thirty-three (33) binary digits, including parity bit, sign bit, and 31 binary digits for absolute numerical value.

i. Memory Type. Rotating magnetic disk; 6000 rpm nominal, nonvolatile. Automatic frequency and voltage protection is provided.

j. Storage Capacity. Sixty-four (64) channels of 128 words each (8192 words) consisting of 48, 52, or 60 channels designated as permanent storage (Read only) and 16, 12, or 4 channels as working storage. Also provided are two 16-word high-speed loops for rapid access, live 1-word registers for arithmetic operations and control, one 2-word register for output display-information storage.

k. Pulse Repetition Rate. Nominal pulse repetition rate 460 kilo-pulses per second.

l. Input. Input rate, mechanical tape reader, approximately 10 characters per second; other sources, approximately 4250 characters per second, maximum.

m. Output. Output rate is approximately 4250 characters per second (maximum).

thmetic unit and instructs the arithmetic unit o perform the desired operation.

The arithmetic unit is comprised of three registers: the accumulator A, the lower accumulator L, and the number register N. All of these registers are addressable.

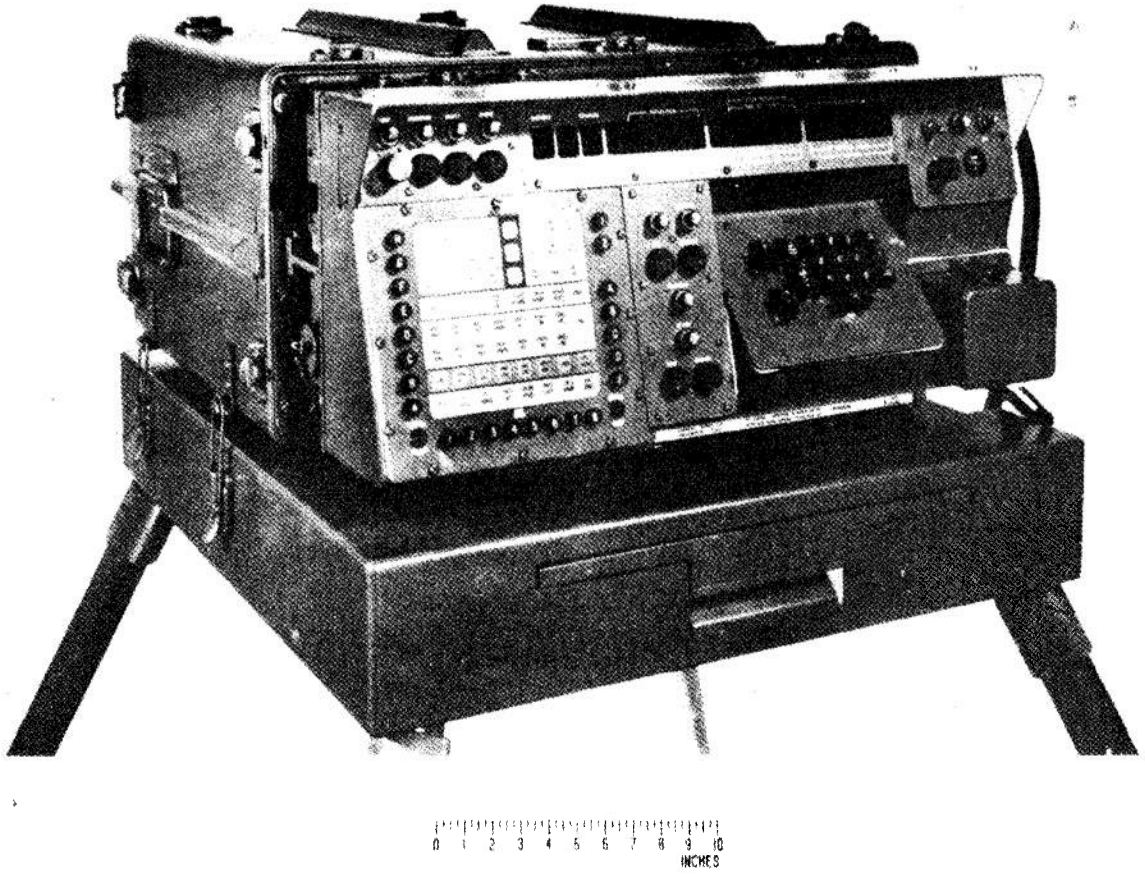
(1) The accumulator A is the primary register. It is employed in all arithmetic, logical, and decision operations and serves also as one of the input-output buffers.

(2) The lower accumulator L is an extension of A in some operations; in other operations it functions independently of A. It also serves as part of the control for input-output.

(3) The number register N holds the second operand for some operations. It serves in program-control transfer operations and also as one of the input-output buffers. Because of the comparatively infrequent use of the latter two registers in machine operations, they can be employed as additional rapid-access storage.

The registers, A, L, N, and I, are physically realized as recirculating loops in the memory unit. The detailed operation is explained in a later paragraph.

The memory unit (see Fig. 13-4) consists of a double-sided oxide-covered disk driven by a motor between lower and upper head-



SIZE	24 INCHES BY 14 INCHES BY 34 INCHES
WEIGHT	APPROXIMATELY 210 POUNDS
POWER	APPROXIMATELY 750 WATTS
	3-PHASE, 400 CPS, 120/208-VOLT, 4-WIRE SYSTEM

Figure 13-2. Computer physical characteristics.

plates housing a series of write and read heads. The disk rides on a self-generated air bearing developed as it rotates at 6000 rpm between the two headplates in an air gap only five ten-thousandths of an inch wider than the disk. The headplates and disk are machined and lapped flat to within two light bands or approximately twenty-two micro-inches. A number of other critical dimensions have tolerances of half a ten-thousandth of an inch.

Data are written on the magnetic disk by current pulses in the write head which magnetize the oxide coating of the disk directly under the head. Binary ones and zeroes are distinguished by the polarity of the magnetic flux. This type of recording is designated return-to-zero RZ. Return-to-zero recording requires only simple driving circuits, but

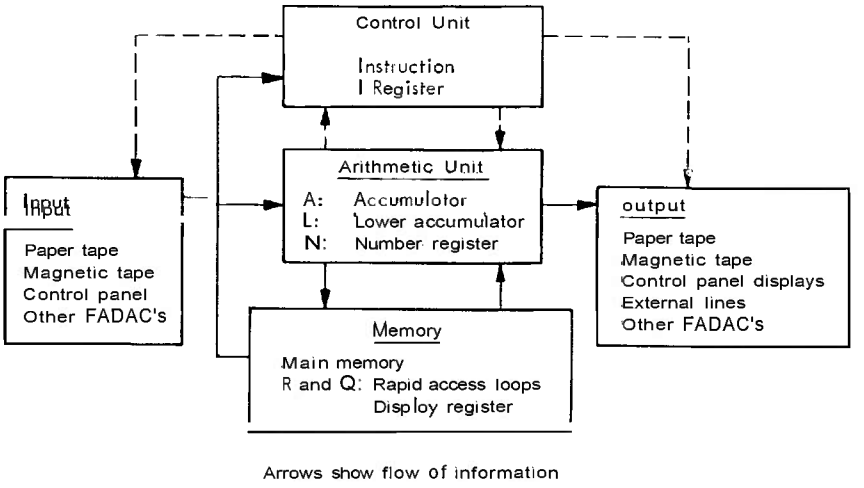


Figure 13-3. Functional diagram of FADAC system.

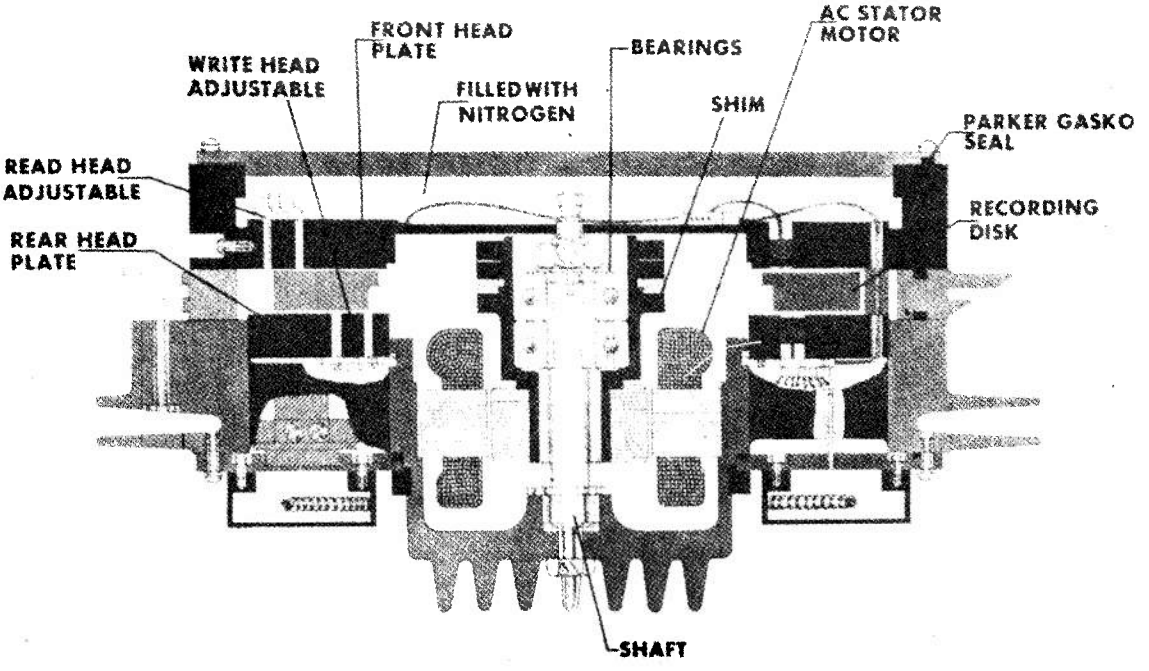


Figure 13-4. Magnetic memory detail.

has the disadvantage that a continued seyucnce of ones or zeroes may saturate the head *so* that data are lost. This is a serious problem in magnetic tape readers, but is less important in the short tracks of the magnetic disk.

The memory disk has 64 tracks for data storage. Each track has a storage capacity of 128 36-bit words (4608 bits). 16, 12, or 4 tracks are assigned to temporary storage and 48, 52, or 60 tracks to permanent storage. In addition, the recirculating-loop registers listed in Table 13-2 are provided for the control and arithmetic units. Timing and sector marking pulses are permanently recorded on the disk. The timing pulses synchronize all computer operations, while the sector pulses mark the beginning of each of the 1-word sectors into which each track is divided.

The recirculating loops are used to form registers that have characteristics similar to delay-line storage units (see Chapter 4). A typical recirculating loop is shown in Fig. 13-5. This loop, which forms the A register, holds one word plus an extra bit. Each word, whether numeric or instructional, holds a binary number of 31 bits, plus a sign bit, plus spacer and parity bits. There is thus a 36-bit capacity in the loop. As shown in Fig. 13-5, 33 bits are stored on the disk and 2 bits in flip-flops. The spacing between the read and write heads is tied in with the prerecorded timing track which effectively divides the space between the heads into 33 bit spaces. A bit written by the write head arrives at the read head 33 clock pulses later, and is then read into flip-flop A_X. The original bit is transferred through flip-flops A_X and A_D by two clock pulses, and is then rewritten on the disk. The data can be read out while stored in the flip-flops; thus, any bit is available every 33 clock pulses or, in this case, every 22.3 microseconds.

The assignment of channels on the magnetic disk to act as registers is the principal device employed in the FADAC to reduce the size and complexity of the computer. The space occupied by a bit on the magnetic surface of the disk is obviously much less than that required for a transistor flip-flop or magnetic core plus driver. The access time, as noted above, is a disadvantage, but is far outweighed, in a portable computer, by the advantages of reduced size and complexity.

The detailed circuitry of the FADAC is of conventional conceptions. The major items are transistor flip-flops, diode logic boards, read and write amplifiers, conversion equipment for teletype output, Nixie drivers, and power supplies. The logical design is based on the use of serial logic which is compatible with the serial readout of the memory. Serial logic is most economical of equipment, since data are processed through a single logical element bit-by-bit, rather than through multiple logical elements simultaneously, as is the case in parallel logic. The penalty is, of course, the greater length of time required for serial computations.

TABLE 13-2. MEMORY CONTENTS.

Register	Capacity	
R	16 words	may be stored into by program
Q	16 words	
A	1 word	
L	1 word	
N	1 word	
D_0, D_1	2 words	
Main Memory	8192 words	
I	1 word	may not be stored into by program
X	1 word	

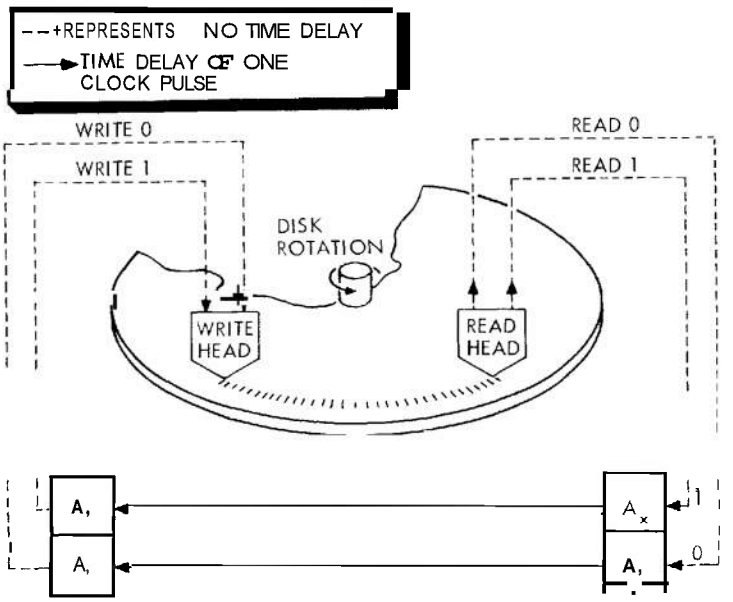


Figure 13-5. Typical recirculating loop register.

13-4 CONCLUSIONS

The FADAC is an example of the adaptation of a general-purpose digital computer to the problem of a field-portable computer designed to solve the field artillery fire control equations. The approach taken in FADAC is to minimize the equipment requirements through the use of serial logic and through the use of recirculating loops for register requirements in the arithmetic and control units. An extremely compact memory is secured through the employment of a magnetic disk which provides permanent storage, temporary storage, eight recirculating loops, and the clock pulse generator. Moderately high-speed computing elements are employed *so* that the total computation time, despite the serial logic and serial access to the memory, is not excessive.

Further reduction in size and complexity is achieved through simple input/output equipment. Input is by manual keyboard or by a simple low-speed mechanical tape reader. Primary output is to a Nixie visual readout. Some added complexity is introduced by the requirement for teletype output as well. Since the semi-permanent memory need be loaded only when the program is changed, a separate memory-loading unit is provided, consisting of a high-speed photo-electric reader for punched paper tape.

The FADAC, in summary, represents a practicable compromise between flexibility and equipment simplicity. The use of a magnetic disk to provide most of the storage registers resembles the configuration of a typical DDA. However, the general-purpose configuration gives a great increase in flexibility at a small increase in complexity as compared with a DDA. In comparison with analog computers, the FADAC is comparable in size and weight but provides a major increase in the accuracy of computation.

INDEX

D

M

N

(AMCRD-TV)

FOR THE COMMANDER:

OFFICIAL:

P. R. HORNE
Colonel, GS
Chief, HQ Admin Mgt Ofc

CHARLES T. HORNER, JR.
Major General, USA
Chief of Staff

DISTRIBUTION:
Special

U. S. GOVERNMENT PRINTING OFFICE: 1976 O - 409-611

ENGINEERING DESIGN HANDBOOKS

Listed below are the Handbooks which have been published or are currently under preparation. Handbooks with publication dates prior to 1 August 1962 were published as 20-series Ordnance Corps pamphlets. AMC Circular 310-38, 19 July 1963, redesignated those publications as 706-series AMC pamphlets (e.g., ORDP 20-138 was redesignated AMCP 706-138). All new, reprinted, or revised Handbooks are being published as 706-series AMC pamphlets.

No.	Title
100	*Design Guidance for Producibility
104	*Value Engineering
106	Elements of Armament Engineering, Part One, Sources of Energy
107	Elements of Armament Engineering, Part Two, Ballistics
108	Elements of Armament Engineering, Part Three, Weapon Systems and Components
110	Experimental Statistics, Section 1, Basic Concepts and Analysis of Measurement Data
111	Experimental Statistics, Section 2, Analysis of Enumerative and Classificatory Data
112	Experimental Statistics, Section 3, Planning and Analysis of Comparative Experiments
113	Experimental Statistics, Section 4, Special Topics
114	Experimental Statistics, Section 5, Tables
115	Basic Environmental Concepts
116	*Basic Environmental Factors
120	*Design Criteria for Environmental Control of Mobile Systems
121	Packaging and Pack Engineering
123	*Hydraulic Fluids
125	Electrical Wire and Cable
127	*Infrared Military Systems, Part One
128(S)	*Infrared Military Systems, Part Two (U)
130	Design for Air Transport and Airdrop of Materiel
134	Maintainability Guide for Design
135	Inventions, Patents, and Related Matters
136	Servomechanisms, Section 1, Theory
137	Servomechanisms, Section 2, Measurement and Signal Converters
138	Servomechanisms, Section 3, Amplification
139	Servomechanisms, Section 4, Power Elements and Systeni Design
140	Trajectories, Differential Effects, and Data for Projectiles
145	*Dynamics of a Tracking Gimbal System
150	Interior Ballistics of Guns
160(S)	Elements of Terminal Ballistics, Part One, Kill Mechanisms and Vulnerability (U)
161(S)	Elements of Terminal Ballistics, Part Two, Collection and Analysis of Data Concerning Targets (U)
162(S-RD)	Elements of Terminal Ballistics, Part Three, Application to Missile and Space Targets (U)
165	Liquid-Filled Projectile Design
170(C)	Armor and Its Application to Vehicles (U)
175	Solid Propellants, Part One
176(C)	Solid Propellants, Part Two (U)
177	Properties of Explosives of Military Interest
178(C)	†Properties of Explosives of Militarv Interest. Section 2 (U)
179	Explosive Trains
180	"Principles of Explosive Behavior
185	Military Pyrotechnics, Part One, Theory and Application
186	Military Pyrotechnics, Part Two, Safety, Procedures and Glossary
187	Military Pyrotechnics, Part Three, Properties of Materials Used in Pyrotechnic Compositions
188	*Military Pyrotechnics, Part Four, Design of Ammunition for Pyrotechnic Effects
189	Military Pyrotechnics, Part Five, Bibliography
190	*Army Weapon Systeni Analysis
195	*Development Guide for Reliability, Part One
196	*Development Guide for Reliability, Part Two
197	*Development Guide for Reliability, Part Three
198	*Development Guide for Reliability, Part Four
199	*Development Guide for Reliability, Part Five
200	*Development Guide for Reliability, Part Six
201	"Rotorcraft Engineering, Part One, Preliminary Design
202	*Rotorcraft Engineering, Part Two, Detail Design
203	*Rotorcraft Engineering, Part Three, Qualification Assurance
205	*Timing Systems and Components
210	Fuzes
211(C)	Fuzes, Proximity, Electrical, Part One (U)
212(S)	Fuzes, Proximity, Electrical, Part Two (U)
213(S)	Fuzes, Proximity, Electrical, Part Three (U)
214(S)	Fuzes, Proximity, Electrical, Part Four (U)
215(C)	Fuzes, Proximity, Electrical, Part Five (U)
235	*Hardening Weapon Systems Against RF Energy
239(S)	*Small Arms Ammunition (U)
240(C)	Grenades (U)
241(S)	*Land Mines (U)
242	Design for Control of Projectile Flight Characteristics
244	Ammunition, Section 1, Artillery Ammunition--General, with Table of Contents, Glossary and Index for Series
245(C)	Ammunition, Section 2, Design for Terminal Effects (U)
246	†Ammunition, Section 3, Design for Control of Flight Characteristics
247	Ammunition, Section 4, Design for Projection
248	†Ammunition, Section 5, Inspection Aspects of Artillery Ammunition Design
249	Ammunition, Section 6, Manufacture of Metallic Components of Artillery Ammunition
250	Guns--General
251	Muzzle Devices
252	Gun Tubes
255	Spectral Characteristics of Muzzle Flash
260	Automatic Weapons
270	Propellant Actuated Devices
280	Design of Aerodynamically Stabilized Free Rockets
281 S-RD)	Weapon System Effectiveness (U)
282	†Propulsion and Propellants
283	Aerodynamics
284 C)	Trajectories (U)
285	Elements of Aircraft and Missile Propulsion
286	Structures
290 C)	Warheads--General (U)
291	Surface-to-Air Missiles, Part One, System Integration
292	Surface-to-Air Missiles, Part Two, Weapon Control
293	Surface-to-Air Missiles, Part Three, Computers
294 S)	Surface-to-Air Missiles, Part Four, Missile Armament (U)
295 S)	Surface-to-Air Missiles, Part Five, Countermeasures (U)
296	Surface-to-Air Missiles, Part Six, Structures and Power Sources
297(S)	Surface-to-Air Missiles, Part Seven, Sample Probleni (U)
327	Fire Control Systems--General
329	Fire Control Computing Systems
331	Compensating Elements
335(S-RD)	*Nuclear Effects on Weapon Systems (U)
340	Carriages and Mounts--General
341	Cradles
342	Recoil Systems
343	Top Carriages
344	Bottom Carriages
345	Equilibrators
346	Elevating Mechanisms
347	Traversing Mechanisms
350	Wheeled Amphibians
355	The Automotive Assembly
356	Automotive Suspension
357	Automotive Bodies and Hulls

* *UNDER PREPARATION--not available*

† *OBSOLETE--out of stock*