

AD-A064 131

CONTROL DATA CORP HAMPTON VA
PREDESIGN OF THE SECOND-GENERATION COMPREHENSIVE HELICOPTER ANA--ETC(U)
OCT 78

F/6 1/3

DAAJ02-77-C-0058

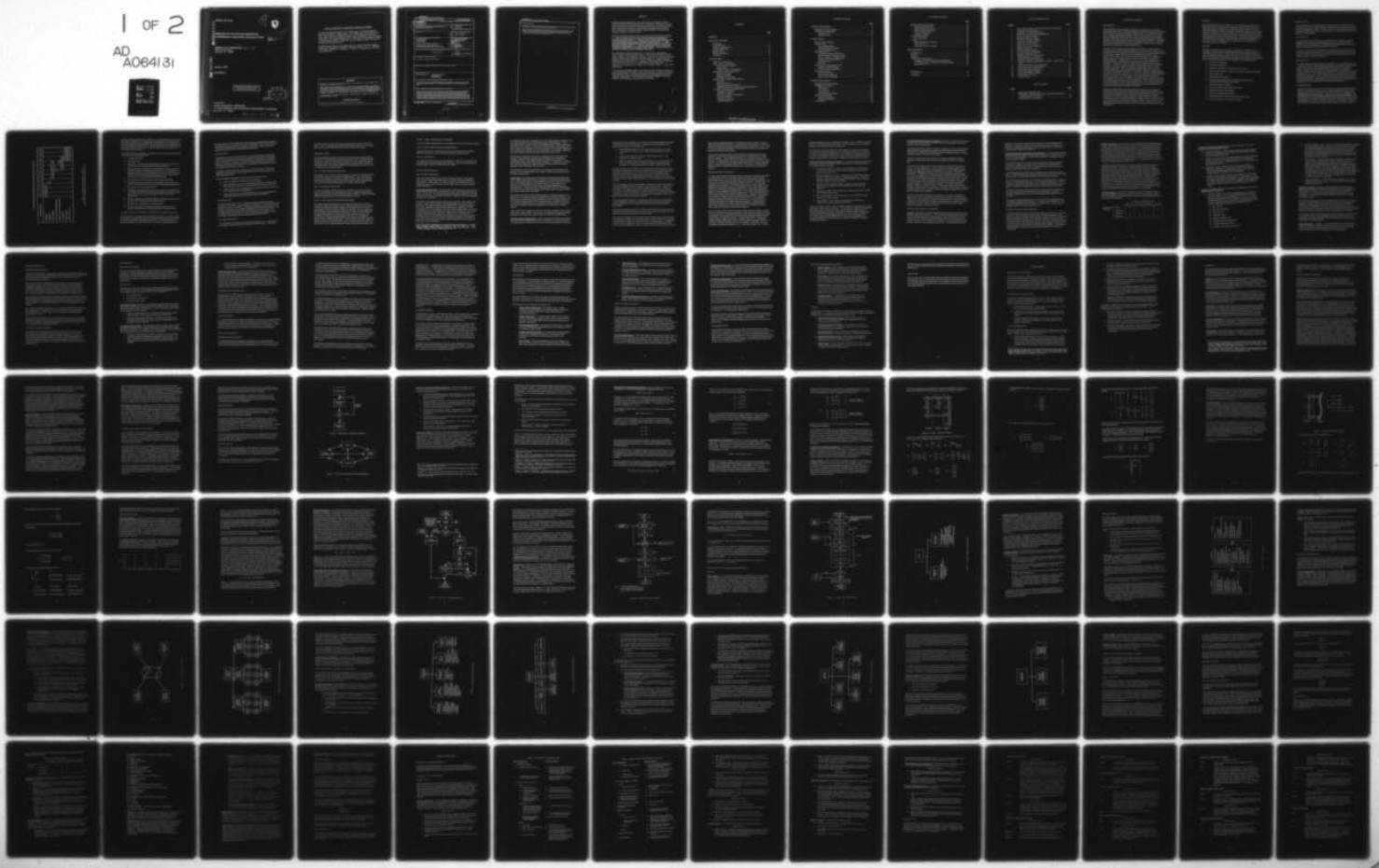
UNCLASSIFIED

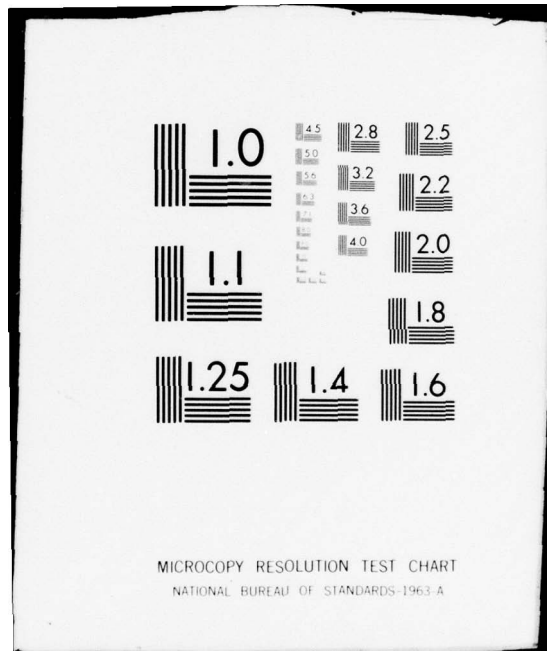
USARTL-TR-78-43

NL

1 OF 2

AD
A064131





USARTL-TR-78-43

AD A 064131

12
SC



PREDESIGN OF THE SECOND-GENERATION
COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM

LEVEL 21

CONTROL DATA CORPORATION
2101 Executive Drive
Hampton, Va. 23666

New 411 055

DDC FILE COPY

October 1978

Final Report

Approved for public release;
distribution unlimited.

DDC
RECEIVED
FEB 2 1979
A

Prepared for
APPLIED TECHNOLOGY LABORATORY
U. S. ARMY RESEARCH AND TECHNOLOGY LABORATORIES (AVRADCOM)
Fort Eustis, Va. 23604

79 01 29 019

APPLIED TECHNOLOGY LABORATORY POSITION STATEMENT

This report summarizes the results of the predesign phase of the Second-Generation Comprehensive Helicopter Analysis System. The predesign phase was conducted to provide: improvements to the Government-written functional specification, conceptual system design, definition of necessary computer program configuration items, development specifications, and a baseline development plan. This phase will be subsequently followed by the development validation, maintenance, and user application phases. The report is considered to be technically sound.

Technical program direction was provided by Mr. E. E. Austin, Contracting Officer's Representative (Technical) of the Applied Technology Laboratory, Mr. H. I. MacDonald, Team Leader, and Messrs. D. J. Merkley, P. H. Mirick, A. E. Ragosta, and W. D. Vann of the project team.

DISCLAIMERS

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

Trade names cited in this report do not constitute an official endorsement or approval of the use of such commercial hardware or software.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return it to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
18 1. REPORT NUMBER USARTL TR-78-43	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
5. TITLE (and Subtitle) PREDESIGN OF THE SECOND-GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM		9 5. TYPE OF REPORT & PERIOD COVERED Final Report	
7. AUTHOR(s)		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Control Data Corporation 2101 Executive Drive Hampton, VA 23666		8. CONTRACT OR GRANT NUMBER(s) 15 DAAJ02-77-C-0058 new	
11. CONTROLLING OFFICE NAME AND ADDRESS Applied Technology Laboratory U.S. Army Research and Technology Laboratories (AVRADCOM) Fort Eustis, VA 23604		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 63211A 1L263211D157 17 18001 EK	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 164p.		12. REPORT DATE 11 October 1978	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		13. NUMBER OF PAGES 158	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15. SECURITY CLASS. (of this report) Unclassified	
18. SUPPLEMENTARY NOTES Prepared in cooperation with: Kaman Aerospace Corporation Old Windsor Road Bloomfield, CT 08002		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report summarizes the efforts and results of the predesign of the Second-Generation Comprehensive Helicopter Analysis System (also referred to as the System). Attention is focused on the solution to problems that are inherent in the design, development, maintenance, validation, and use of a system that satisfies the objectives and requirements set forth in the Statement of Work. The objectives and design considerations are presented, along with a comprehensive design that will enable the realization of all objectives through the use of dynamic component coupling techniques, a comprehensive but simple user-oriented control language, and an extensive library of technical modules. System usage has been described in three levels: Basic, Intermediate, and Advanced. Suggested responsibilities and relationships of the various agencies during the Development Phase			

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411 055
79 01 29 019 LB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued)

are presented; also development schedules, documentation, and testing requirements are included. The concept of a Baseline Review Board and its activities are offered to enhance quality assurance control.

Based on the results of the Predesign Phase, the Second-Generation Comprehensive Helicopter Analysis System has been determined by Control Data Corporation and Kaman Aerospace Corporation to be a feasible system that will provide users with a viable vehicle for current and future endeavors in the prediction of performance, stability and control, acoustics, loads, and aeroelastic stability characteristics of rotorcraft.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This document was prepared by a team effort of the Professional Service Division of Control Data Corporation (CDC) and Kaman Aerospace Corporation (KAC) personnel to present the conclusions obtained from the predesign activities on the Second-Generation Comprehensive Helicopter Analysis System. These activities were performed under Contract DAAJ02-77-C-0058 to the Applied Technology Laboratory (ATL).

Additional documents delivered under this contract but not published included (1) Interim Technical Report for the Second-Generation Comprehensive Helicopter Analysis System, (2) Baseline Type A System Specification for the Second-Generation Comprehensive Helicopter Analysis System, (3) Draft Type B5 Development Specification for the Second-Generation Comprehensive Helicopter Analysis System, (4) Baseline Development Plan for the Second-Generation Comprehensive Helicopter Analysis System. These documents are available through the Applied Technology Laboratory.

The CDC team members are Messrs. L. Warren Haley, CDC Manager; John R. Mitchell, Project Leader; Mark A. Anderson and Jimmie C. Deaver, Senior Analysts. KAC members are Messrs. Alex Berman, Principal Research Engineer; and Nicholas Giansante, Research Engineer. Technical program direction was provided by Mr. E. E. Austin, Contracting Officer's Representative (Technical) of ATL; Mr. H. I. MacDonald, Team Leader; and Messrs. D. J. Merkley, P. H. Mirick, A. E. Ragosta, and W. D. Vann of the project team.

Two concurrent predesign efforts were performed under Contracts DAAJ02-77-C-0057 and DAAJ02-77-C-0059 by teams from Computer Sciences Corporation/Bell Helicopter Textron and Science Applications Incorporated/Boeing Vertol Company, respectively. Those efforts are reported in USARTL-78-41 and USARTL-78-42 bearing the same titles as this report.

CONTENTS

	<u>Page</u>
PREFACE	1
EXECUTIVE SUMMARY	7
Background	7
Predesign Tasks	13
System Design Summary	14
System Capabilities	24
System Usage	25
System Development	26
Conclusion	33
SYSTEM DESIGN	34
Objectives of the System	34
Major Design Considerations	34
Processing Requirements	34
Design Goals	36
Technical Design Considerations	37
Mathematical Basis of the System	39
Sequential Coupling	40
Dynamic Coupling	40
Modeling of Components	53
System Design Concept and Architecture	59
Overview	59
System Operation	63
Engineering Data Management	65
Application Executive Components and Relationships	69
Technical Components and Relationships	77
Technical Capabilities	77
Couplings of Analyses	78
Run Time Coupling Considerations	78
Technical Modules	79
Technical Subroutines	83
Utility Subroutines	83

CONTENTS (Continued)

	<u>Page</u>
SYSTEM CAPABILITIES	84
Satisfaction of Requirements	84
Availability of Capabilities	84
Executive Area	84
Technical Area	89
SYSTEM USAGE	97
System Control Language	97
Levels of Use	97
Basic System Usage	98
Intermediate System Usage	99
Advanced System Usage	100
Illustrations of Use	101
Preliminary Design Performance	101
Linear Stability Preliminary Design	103
Loads, Acoustics, Detailed Design	104
Nonlinear Aeroelastic Stability	105
Damage Effects	106
Summary	106
Development and Maintenance Aids	106
Structured Modular Design	107
Functional Data Independence	107
Resource Utilization	108
Executive Overhead	108
Small Problem Efficiency	109
Large Problem Capability	109
SYSTEM DEVELOPMENT	112
Organizational Responsibilities	112
Development Contractor	112
Subcontractors	118
Government	122
Level of Effort	122
Development of Facilities	127
Development Schedule	127
First-Level Release	127
Second-Level Release	127
Documentation	130

CONTENTS (Continued)

	<u>Page</u>
Quality Assurance and Control	134
Functional Design Review	136
System Design Concurrence	136
Preliminary Design Review	137
Critical Design Review	137
Program Production	137
Testing Requirements	141
Error Resolution	145
Training	146
Government Internal Training	146
User Training	147
Interactive Aids for Tutoring	147
RISK ASSESSMENTS	148
Developmental	148
Hardware	149
Software	149
Technical Risk Assessment	150
Adequacy of Advanced Technology Development	150
Cost Effectiveness of Advanced Technology Methods	151
Operational Costs	151
REFERENCES	153
GLOSSARY	155

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Initial Overall Schedule, Second-Generation Comprehensive Helicopter Analysis System	10
2	Illustration of a Sequential Analysis	41
3	Illustration of Dynamically Coupled System	41
4	Sample of Explicit Coupling	47
5	Example of Implicit Coupling	51
6	Logic Flow of Execute Statement	56
7	Logic Flow of Set-Up Phase	58
8	Logic Flow of Active Phase	60
9	System Concept and Architecture	61
10	System Processing	64
11	Data Base Record Formats	67
12	Data Base Record Relationships	68
13	Application Executive CPCIs	70
14	Executive Supervisor Components	71
15	Batch Subsystem Components	74
16	Interactive Subsystem Components	76
17	Analysis and Design Organization - 6 months - Control Data/ Kaman	114
18	First-Level Release Peak Organization	115
19	Second-Level Average Organization	116
20	Bi-Monthly Levels of Manpower	126
21	First-Level Release Schedule	129
22	Second-Level Release Schedule	131
23	Quality Assurance Loop	135
24	Levels of Tests	142

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Satisfaction of Requirements	85
2	Relationships of System Developer, Technical Subcontractor, Government and TAG/GIWG	123
3	Technical CPCI Level Schedule	128

EXECUTIVE SUMMARY

BACKGROUND

The Government and the helicopter industry need a capability to accurately predict helicopter loads, aeroelastic stability, stability and control, performance, and acoustics for a variety of aircraft configurations. This capability is necessary to reduce engineering development risk for new helicopters, to minimize delays in deployment of new aircraft, to reduce reliability and maintainability problems of operational aircraft, and to prevent undue restrictions of operational capabilities of Army helicopters due to unsolved technical problems. Although the primary requirement is for accurate prediction, economy and reliability in prediction are important secondary requirements.

Presently, the Government and the helicopter industry use a wide variety of engineering application computer programs to meet most of their analysis needs. Computerized analysis methods have been developed by Government organizations and by industry under Government and/or industry sponsorship. Due to limitations of current theory and implementation thereof, each manufacturer applies empirical correction factors to achieve correlation with existing experimental data. Consequently, these methods are often applicable only to the types and sizes of helicopters for which these empirical factors were developed. Furthermore, these methods have not emphasized documentation, diagnostics, advanced software techniques, configuration control, or user convenience features. This has led to unnecessary duplication of analysis methods development since the capability developed within one organization is not readily transferable to another. As a result, a welter of computer programs exists to do each major analysis task. Each of these programs can be operated by only a small user community and few have been adequately verified by correlation with test data. In addition, comparison of such methods is difficult due to differences in notation as well as approach. Finally, most methods have a limited capability to account for interactions such as the coupling of the rotor with advanced flight control systems, fuselage motions, and inadvertent high-frequency pilot inputs (pilot-coupled oscillations).

Despite these deficiencies, the growth over the past ten years of computer technology and the development of sophisticated computerized analysis methods have played an important role in the analysis of advanced rotary wing aircraft. Such analysis methods as the Bell Helicopter Textron Rotorcraft Flight Simulation (C-81), the Lockheed-California Company REXOR, and the Sikorsky Aircraft Company Normal Modes (Y-200) are examples of the state of the art in comprehensive analysis mathematical models. These, referred to as first-generation comprehensive analyses, do not, however, satisfy the needs of Army and industry.

Objectives

The primary objectives of this effort are: (1) to develop and demonstrate a Second-Generation Comprehensive Helicopter Analysis System that will be a major step toward satisfaction of the need for accurate prediction of loads, aeroelastic stability, stability and control, performance, and acoustics of helicopters of various sizes and rotor types; and (2) to provide each of the major helicopter manufacturers and Government users with an operational capability using the System at his own computer facility. Successful accomplishment of these objectives will provide an analysis system which can subsequently be evolved by further development into a system that is more reliable and economical, as well as accurate.

Approach

In order to satisfy the above objectives, a project was established for the development of a computer-implemented Second-Generation Comprehensive Helicopter Analysis System. This System will provide a unified treatment of loads, aeroelastic stability, stability and control, performance, and acoustics, and will be applicable to all stages in the research, development, improvement, and employment of helicopters. Key concepts for this project include:

- systematic development
- thorough documentation
- exhaustive validation by comparison with test data
- use of modern computer hardware and advanced software techniques
- data management
- configuration management
- varying levels of complexity in the analysis techniques and representation of helicopter components
- computer program modularity
- user aids including diagnostics and graphics
- standardized engineering notation
- engineer readable program coding
- development keyed to Government and industry users
- coupled aerodynamic and dynamic analysis.

Program Phases

The Second-Generation Comprehensive Helicopter Analysis System effort consists of six phases: Planning, Predesign, Development, Validation, Maintenance, and User Applications. Figure 1 presents the initial overall schedule and major milestones for the life-cycle phases of the System. A description of each of the phases is set forth below.

Planning Phase

The specific needs for the System were defined and an approach to be taken throughout the development was tentatively established. The Government/Industry Working Group (GIWG) was established and participated in an advisory capacity to formulate the overall approach to be taken.

An Initial Type A System Specification was written with the advice of the GIWG detailing the functional capabilities which the System should possess, and each of the six industry companies represented on the GIWG provided the Army with some comments on technical approach.

Predesign Phase

A multiple-contract predesign effort was conducted. Contractual efforts were: to improve the Initial Type A System Specification, to define the feasible First-Level, Second-Level, and Long-Range System capabilities, to design the System, to define the CPCIs which comprise the System, to produce an associated set of Type B5 Development Specifications, and to produce a Baseline Development Plan. The Government project team was advised by the GIWG to enhance user orientation, and by a Technical Advisory Group (TAG) to enhance the technical approach. The Government reviewed results of this phase, prepared a revised Type A Specification, and formulated tentative requirements for experimental data to determine CPCI and System level accuracy.

Development Phase

During this phase the First-Level and Second-Level System capabilities will be developed in accordance with the Type A System Specification defined in the Predesign Phase and in general accordance with AMCP 70-4 - Research and Development Software Acquisition - A Guide for the Material Developer. This release near the end of the Development Phase has been designated the Second-Level Release. An interim release of the System at about the mid-point in the Development Phase has been designated the First-Level Release. The First-Level Release is expected to have the computer program structure (executive

SECOND-GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM

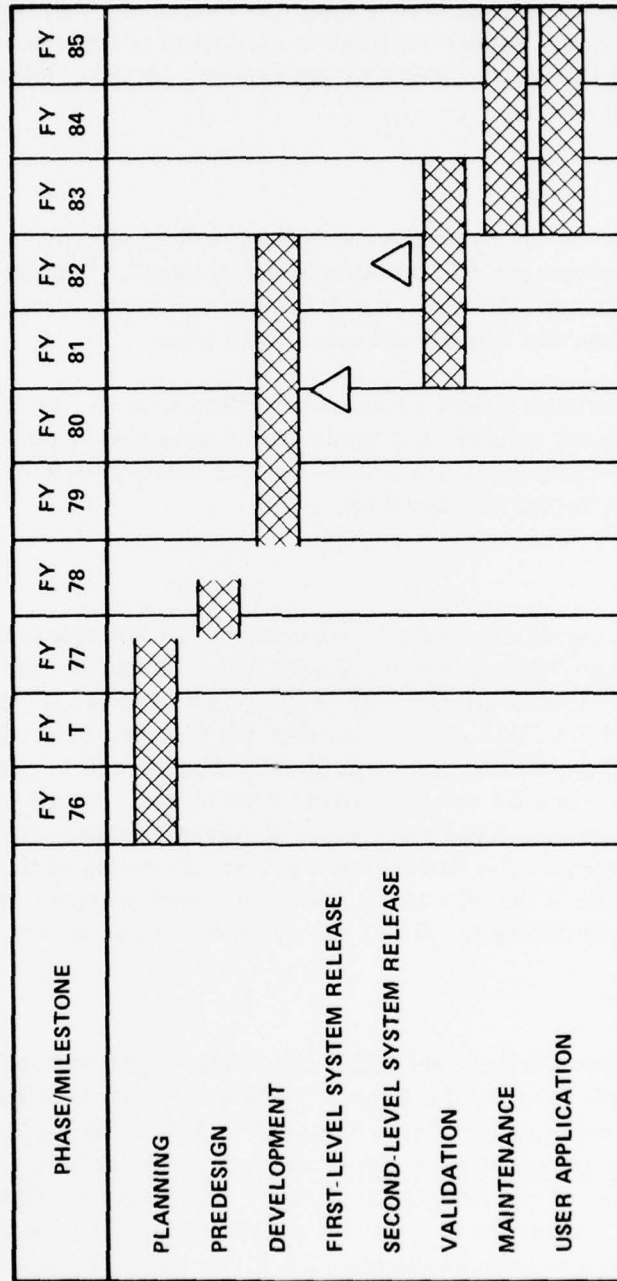


Figure 1. Initial Overall Schedule, Second-Generation Comprehensive Helicopter Analysis System.

program) largely completed with modules for aircraft components and analysis components available up to a certain level of complexity. The technology of many of these modules would be expected to be state of the art, comparable to that used throughout industry today. The second half of the Development Phase could then be devoted, primarily, to improving the technology of the First-Level Release of the System.

The Development Phase contractor, expected to be one of the Predesign Phase contractors, will be responsible for:

- Designing the System
- Identifying CPCI's
- Preparing a Type B5 Development Specification for each CPCI, for both First-Level and Second-Level System capabilities
- Recommending those CPCI's to be developed by the Development Phase contractor, those by subcontractors, and those to be Government-furnished based on the premises that few, if any, First-Level System CPCI's will be Government-furnished and few, if any, Second-Level System CPCI's will be developed by subcontractors
- Developing those CPCI's approved by the Contracting Officer
- Determining that each CPCI meets the functional requirements and quality assurance provisions of its Type B5 Development Specification
- Integrating all CPCI's into the System
- Conducting functional demonstration of the System to demonstrate to Government and industry that the System meets the functional requirements and quality assurance provisions of the Type A System Specification
- Defining a unified documentation approach and editing documentation for each CPCI to promote uniform high standards
- Implementing a configuration management plan
- Providing training and maintenance support to Government and industry users during the initial portion of the Validation Phase.

The GIWG and the TAG will continue to advise the Government project team.

The Government will monitor the development of the System in detail, and the Government will approve the Type B5 Development Specifications produced by the Development Phase contractor for each CPCI. The Government will, in addition, participate in the evaluation of and exercise selection approval of

subcontractors for CPCI development. The Government will prepare to assume full responsibility for the System during the Maintenance Phase. The Government will finalize requirements for, and sponsor acquisition of, experimental data necessary to determine CPCI and System level accuracy.

Validation Phase

The objectives of the Validation Phase are to establish within the Government/industry user community an operational capability with the System, contribute to the validation of the accuracy and operating cost of the System, and provide inputs from the user community to the Development Phase contractor and the Government project team to maximize user orientation of the System during the Development and Maintenance Phases.

Helicopter manufacturers under contract to the Government will validate the applicability of the System to their helicopter types by conducting correlation with experimental data.

These helicopter manufacturers along with Government users will:

- Achieve an operational capability with the System
- Apply the System to current rotary wing R&D efforts, in parallel with other methods of analysis, to evaluate the effectiveness of the System
- Identify minor errors and deficiencies, determine corrective measures, and recommend their implementation
- Identify major errors and deficiencies and make recommendations to the Government project team that the System be modified.

Maintenance Phase

The Maintenance Phase will be a continuous activity consisting of System correction, modification, and development in response to errors and deficiencies identified by the user community. Further advancements in the state of the art in rotary wing analysis and computer technology will also be incorporated. The responsibility for maintenance will be assumed by the Government. The Government will serve as the focal point for dissemination of documentation and advice on operational problems encountered using the System.

User Applications Phase

At the beginning of this phase, the Government/industry user community will have attained a mature operational capability with the System. Under their

own funds, users will utilize the System capabilities for their own analysis needs. They will continue to provide the Government with input to the maintenance activity so that the System will continue to meet their needs.

PREDESIGN TASKS

The first contractual phase of the overall program was the predesign effort. The objectives of the predesign contract were to provide a candidate technical approach, to define the First-Level, Second-Level, and Long-Range System capabilities, to produce required specifications, and to produce a detailed development plan. To meet the objectives of the predesign contract, the work was performed in six defined tasks.

Task I - System Requirements Definition

The purpose of Task I was to conduct synthesis, analysis, trade-offs, and risk assessments of Initial Type A System Specifications. Task I uses defined activities and premises to produce an interim technical report that contains recommended revisions and deviations to the Initial A System Specifications, as well as an initial development plan and a design analysis report.

Task II - Functional Design Review

A two-day Functional Design Review was conducted to review the results of Task I. The contractor also presented the Task I results to a joint one-day meeting of all contractors in the predesign effort, members of the Government/Industry Working Group and the Technical Advisory Group.

Task III - CPCI Type B5 Development Specifications

The Initial System Specifications were revised with approved changes and deviations to establish a Baseline Type A System Specification and to prepare a Baseline Development Plan compatible with the Baseline Type A System Specification. System synthesis, analysis, trade-offs, and risk assessment were conducted to allocate System capabilities to CPCIs, and to develop the functional requirements and quality assurance provisions for each CPCI. This effort culminated in the production of draft Type B5 Development Specifications to establish the feasibility of the design in accordance with the Baseline Type A System Specification. Also, an interim technical report was prepared to contain a design analysis report and recommended revisions to the Baseline Type A System Specification and the Baseline Development Plan.

Task IV - System Design Review (concurrency)

A five-day System Design Review was conducted to review the results of Task III.

Task V - Review Type B5 Development Specifications

Approved changes were incorporated into each Draft Type B5 Development Specification to obtain a set of Type B5 Development Specifications.

Task VI - Final Review

The results of Tasks III and V were presented to a joint two-day meeting of all the Contractors in the predesign effort, members of the Government/Industry Working Group, and the Technical Advisory Group.

SYSTEM DESIGN SUMMARY

Major Design Considerations

The major design considerations of the System are detailed in the Baseline Type A System Specification (Reference 1). However, those performance requirements, design goals, and technical design considerations which had a significant impact on the feasibility of the System and on the design approach are highlighted here.

Performance Requirements - The performance requirements are those requirements from the Type A System Specification which are demonstrable following system development. The requirements that most greatly influenced the system design were the General Functional Capability, Particular Functional Capability, and Interactive Capability.

The General Functional Capability (GFC) can be described as the ability of the System to analyze arbitrary rotorcraft configurations in a variety of flight conditions. The approach taken in designing the GFC eventually impacts the the overall extendability and flexibility of the System. Specifically, if the development of a new or modified rotorcraft analysis configuration is a multiple job-step operation involving the user in extensive use of features of the host computer operating system, then the average user will avoid using the capability because of the additional host system features that must be learned. On the other hand, if use of the General Functional Capability is localized in the System and involves use of normal system capabilities, the user would be more likely to use that feature.

¹ Control Data Corporation: Baseline Type A System Specification for the Second Generation Comprehensive Helicopter Analysis System (in response to Task IIIa, CDRL A008, contract DAAJ02-77-C-0058), Control Data Corporation, Hampton, Virginia 23666, and Kaman Aerospace Corporation, Bloomfield, Connecticut 08002; January 27, 1978.

The Particular Functional Capability (PFC) also requires special attention since it provides the user with a set of standard analysis configurations. This encourages initial use of the System by providing simple, readily-used analysis tools to the user. However, even in the case of the PFC careful consideration must be given to the approach. If the PFC analyses are too rigidly defined, the user may find that none quite fits the problem that is to be solved. Therefore, provision should be made for run-time modification of the PFCs.

The Interactive Capability also played a significant part in the design of the System. The specification requires that the System have an Interactive Capability; however, that capability shall not preclude the execution of any engineering analysis in the batch mode of operation. Consequently, the design of the System incorporates a special CPCI that will provide conversational operation without impacting batch operations.

Although these particular capabilities received special consideration in the design, all of the requirements were carefully considered and integrated into the design of the System.

Design Goals - The design goals of the System are those requirements of the Type A System Specification which, though not demonstrable through testing, can be evaluated qualitatively through extensive use over the life of the system. Some of the requirements that were determined to be design goals were: longevity, cost effectiveness, ease of use, hardware independence, and maintainability. Though not strictly "programmable" features of the System, these goals must be taken into consideration during design, and specific steps can be taken toward their realization.

Specifically, the design must be resource efficient - fitting the System to the problem that is to be solved instead of fitting the problem to the System. The design must be modular and easily extended so that new technical capabilities can be added without extensive costly reprogramming. Computer hardware and software dependencies must be minimized and centralized in a few easily replaceable modules to enhance transportability between unlike computer systems.

These and all other design goals have been carefully considered in the Control Data/Kaman system design. In particular, the Application Executive concept, discussed in the body of the report, meets all of the above requirements.

Technical Design Considerations - There are numerous decisions that must be made during the development of problem formulations, execution of analyses, and interpretation of results that can only be made by knowledgeable engineers.

The System must be developed so as to allow for convenient engineering control of the System at all levels of usage. A language must be developed which specifies the analysis to be performed by:

- establishing the order of major steps in the analysis (e.g., blade modal analysis, iterate to trim, and harmonic analysis of specified loads)
- specifying the dynamic component representations (e.g., rotor, fuselage, and control system)
- specifying major decisions during execution (e.g., test for trim and modify controls, interpret solution and insert structural damage parameters, and perturb solution for nonlinear stability analysis).

In addition to this basic capability, the user must have the capability to invoke the basic and often-used analyses with a simple input. Any particular problem formulations developed at a user site or delivered with the System must be retrievable and executable with temporary changes in any element of the procedure such as: use a different component representation, use a different mathematical algorithm, modify input data, or any combination of the preceding changes.

Thus, the engineer will have the capability to perform any analysis required by his technical judgment for the solution of a particular problem. It will not be necessary to perform an overly complex analysis that would be wasteful of resources or an inadequate analysis in which there would be little confidence. If an analysis results in predictions which the engineer has technological reasons to question, he may easily rerun it with changes to appropriate levels of complexity of components or numerical methods.

Prior to production running of a large number of similar cases, the engineer may conveniently make trial runs using several problem formulations and, by comparing results, he may select the most efficient and effective problem formulation.

The above-mentioned capabilities will ensure the maximization of the technical effectiveness of the System and will result in costs which are significantly lower than those of present generation analytical methods.

Each major component of the helicopter should be represented at several levels of complexity in order to give the engineer the capability to establish a problem solution which is adapted to a particular analysis' requirements. In addition to varying levels of complexity, some of the component representations should allow the user a selection of analytical methods, such as modal or finite element.

This choice of alternate methods is important for several reasons. There is often no universal acceptance of any one method and an engineer may select one method for efficiency and another method for increased accuracy. There is also the matter of personal preference, and the choice of methods is necessary for universal acceptance of the System.

The technical capabilities of the System have been separated into "technical modules", each of which represents the analysis of a component or a mathematical procedure. These technical modules will be directly addressable by the user, and automatically and exactly coupled to perform the complete system analysis. In addition to the technical modules which will be delivered with the System, the user will have the option to add to the System any other technical modules desired, whether they represent new components or new methods of analysis.

Mathematical Basis of the System

The basic mathematical formulation used as a basis for the design of the System for performance, stability and control, loads, acoustics, and aeroelastic stability problems is a set of second-order ordinary differential equations. In many cases a time domain numerical solution is required. In other cases, the equations are converted to the frequency domain or into a set of algebraic equations prior to solution. In the most general cases, where solutions to the differential equations are required, the equations for each of the major components may be developed and coupled into a set of simultaneous differential equations representing the complete system. Each of the component representations may include nonlinear and periodic effects. A numerical algorithm is then required to obtain time history solutions to the equations. The solutions may be carried to steady state and then iterated upon to a trim condition, transient or perturbed solutions may be required, or it may be required to interrupt the solution and modify structural parameters. When the basic solution of the equations of motion has been completed, postprocessing of data is often required to obtain such information as the harmonic content of loads, stability parameters, and acoustic response.

When an analysis that is performed within a single technical module is coupled to other analyses simply by passing data from one analysis to another, this type of coupling is called "sequential" coupling and the technical modules are called "stand-alone" or sequential modules. The technical modules that fall in this category include simple analyses of a complete system (e.g., simple preliminary design performance analysis, Coleman ground resonance analysis, or a simple or complex analysis of a blade or fuselage). In addition, any post-processing mathematical algorithm (e.g., harmonic analysis, FFT, far field

acoustic prediction) is also a "stand-alone" analysis, since it interacts with other analyses only through the passage of data. Iteration through a set of sequential analyses is also included in this classification.

Many analyses will be performed by linking together an arbitrary combination of analyses of individual components. It is incorrect in these cases to attempt to analyze each component separately, but the complete system must be solved as a unit. The capability to perform this type of dynamic coupling between the physical components of the helicopter is crucial to the success of the System.

Objective of Dynamic Coupling Capabilities - In order to establish a mathematical basis for the design of the System, the following goals have been identified for the method of dynamic coupling:

- Independent Component Analysis - The modeling of each component must be completely independent of other component representations which may be coupled to it
- Multiple Methods of Analysis - There must be no limit on the methods of analysis which may be used in modeling components. Each component may make use of finite element, Myklestad, modal, Galerkin and other methods, without restriction
- Nonlinearities and Periodic Effects - There must be no practical restriction of the nonlinear or periodic effects modeled in each component
- Automatic Coupling - The coupling of the components must not require special inputs from the user
- Time and Frequency Domain Applications - The method must apply equally well to both regimes
- Exact Method - There must be no approximations built into the system which are not subject to user control.

There is a method available which is simple and exact and satisfies all the goals established in the previous paragraph. This method is a modification of methods which have been in use for many years in finite element synthesis and has recently been applied to general components in frequency domain applications. It is shown that this method applies equally well to time domain problems. The method starts with the differential equations of each component and, through a simple set of transformations, develops the equations of motion of the complete system.

System Operations for Solution of Equations - In order to achieve the above goals, it is necessary for the System to perform initiation and control, set up, and solution processing operations.

The control operations include the initiation of the Set Up Phase, determination of the problem type (differential equation or eigenproblem), execution of the eigenproblem or differential solution processing, outputting of data for restart, and determination of the end of the solution.

In the Set Up Phase the system will form the coupled coefficient matrices, define the coupled degrees of freedom, and identify and input the component data.

The solution processing is determined by the type of problem that is being solved. For an eigenproblem, a user-selected mathematical module will be executed to output eigenvectors and eigenvalues. For a differential equation solution, the differential equation will be solved repeatedly. During this processing, a mathematical module and active component modules specified by the user are executed with transformations to and from the coupled matrices and degrees of freedom being performed automatically by the System. At each time step a test is made to determine if a program checkpoint is to be output. The determination will be made based on selected user input or a standard default parameter. If the condition is satisfied, all the data necessary to perform a restart operation is stored. Following the checkpoint processing, an end-of-solution test will be performed. This test will be a function of the type of problem and may test a number of rotor revolutions, elapsed time, or may test for a steady-state condition.

External to the problem the user will often specify a criteria judgment test which may check for a specified trim condition and compute new controls and return to the "active phase" to repeat the problem solution. Other noncriteria modules will be available which perform such functions as: perturb the initial conditions to produce a Floquet matrix; obtain derivatives for an External Model; and introduce damage parameters.

System Design Concept and Architecture

Overview - The software design concept chosen for the System is that of a data directed "Application Executive". This approach permits the engineering user to direct the operation of the system through a set of simple inputs. These inputs permit the arbitrary configuration and execution of both executive and technical functions of the System and, in addition, provide comprehensive data storage, retrieval, and manipulation capabilities. In developing this design

approach, the technical requirements of the System were identified and categorized as "technical modules" and the functions necessary to support the arbitrary configuration and coupling of the technical modules were identified and placed in the Application Executive.

Application Executive Components and Relationships - The Application Executive was then functionally aligned into five major components: the Executive Supervisor, the Batch Subsystem, the Interactive Subsystem, the Restart Facility, and the Graphics Package.

The Executive Supervisor provides all utility functions required for System operation. In addition, it controls system initialization and termination operations, determines the mode of system operation and selects either the Batch or Interactive Subsystem for execution.

The Batch Subsystem provides batch and remote batch processing of system inputs. When interfacing with the Batch Subsystem, the user can expect control inputs and data to be thoroughly diagnosed before processing begins.

The Interactive Subsystem provides conversational diagnosis and correction of errors, and otherwise aids the user in data and problem definition. Tutorial interaction, information describing analysis components selected for use, and graphic presentation of engineering data will enhance interactive usability.

The Restart Facility provides comprehensive restart features protecting the user from unplanned interruptions of analysis processing. In addition, automatic checkpointing will occur and user-specified interruptions will be provided.

The Graphics Package prescribed for the System provides support for both interactive and offline graphic devices. The approach recommended is a set of graphic subroutines that will generate a "neutral" file, which can then be postprocessed to a variety of graphic devices.

Technical Components and Relationships

All of the technical functions of the System reside in a collection of "technical modules". There are several general types of technical modules. Each technical module is partitioned into functional modules for ease of programming and for economical operation. When technical modules are specified for a rotorcraft analysis, the Application Executive will cause the required modules to be executed and will perform the input and output, sequential and dynamic couplings, and all similar functions required by the modules. A description of these components and certain operational considerations are provided in this section.

Technical Capabilities - The System has been designed with the engineering user in mind. The System will provide the user with the greatest possible flexibility in modeling and analyzing helicopter problems. For standard and production analysis, the user will have the convenience of extremely simple control inputs. At the disposal of the engineering user will be a library of technical modules (or CPCIs), each of which will be an analytical representation of one or more aircraft components, a method of analysis, or a numerical algorithm. Within the scope of the available CPCIs, the user will be able to specify any combination of (compatible) component representations, any method of analysis, and any numerical processing of the resulting data.

The System will be delivered with a set of validated Particular Functional Capabilities; however, the System is highly oriented toward the General Functional Capability and the PFC's are simply special cases consisting of a prescribed set of control inputs which may be simply addressed. Additional PFC's may be established at each user's installation and adapted to the user's particular needs simply by storing the appropriate set of control statements and giving this set a unique name for future reference. When the user prescribes a problem (including solution method and the required data) with System Control Language statements, all or part of the problem may be designated as a PFC which may be accessed at any time in the future. When it is desired to re-execute this PFC, only the name need be referenced along with any desired changes in the model, physical data or condition, analysis method or numerical processing, and the problem is re-executed.

Technical Modules - There are four distinct categories of technical modules. Each of these technical modules consists of at least two functional modules. The four categories of technical modules and the functional modules associated with each are listed below:

TECHNICAL MODULES TYPES

		Diff. Eq.	Eigenvalue	Sequential	Criteria
FUNCTIONAL MODULES	Active	X			
	Processing			X	X
	Coefficient	X	X		
	Definition	X	X	X	X

Purposes of the Technical Module Types - The four categories of technical modules are briefly described as follows:

- Differential Equation - These modules represent individual physical or analysis components where a coupled set of differential equations is to be formulated and solved.
- Eigenvalue - These modules represent individual physical components where a coupled, linear, constant coefficient set of differential equations are desired. An eigenvalue analysis is to be performed on the coefficient matrices.
- Sequential - These modules perform stand-alone functions as described previously. The algorithms which perform numerical solutions of differential equations (and use the "active modules" of components) are included in this classification.
- Criteria - These modules are used in controlling overall problem logic and are included in a generalized "IF" type statement. They will include such functions as iterative trim algorithms, formulation of Floquet matrices by varying initial conditions, and computation of quasi-linear stability derivatives by perturbations of a trimmed system.

Purpose of Functional Modules - The four types of functional modules are briefly described as follows:

- Definition Module - The definition module must be a part of all technical modules. It is not an executable program but supplies necessary information to the Executive. It is, in effect, a kind of documentation of the CPCI, and appears in the data base of the System. The information contained in this module is listed below (detailed definitions are given in the body of this report).
 - (1) Name of CPCI
 - (2) Narrative Description
 - (3) Input Data List
 - (4) Output Data List
 - (5) Degree of Freedom List
 - (6) Implicit Coupling Relationships
 - (7) Expected Coupled Variables
 - (8) Variability of Coefficient Matrices

- Coefficient Module - These modules are used in the differential equation and eigensolution problems. After the Executive has used all the data in the definition modules, established tables of variables, allocated core, formed transformation matrices, and the other necessary functions, the coefficient modules are called upon to actually compute the constant matrix coefficients as well as any other coefficient data required by the active modules.
- Active Module - These modules play the same role in the differential solution process as do the user-supplied subroutines commonly required in present differential equation solution algorithms. These active modules perform whatever analyses are required to compute the highest derivative vector in the equations, given all of the lower derivatives. They use the constant coefficients already generated, and may include any time-varying or periodic functions, table loop-ups, and nonlinearities of any kind. Active modules for rotor, fuselage, or engine/drive system will contain call statements to aerodynamic or engine performance subroutines.
- Processing Modules - These modules used in sequential or criteria technical modules are, in effect, ordinary routines which perform specified computations.

Technical Subroutines - Most of the technical functions of the System are performed by the Technical Modules described above. There are certain of these functions, however, which are performed by ordinary FORTRAN subroutines. Both modeling and utility functions are performed by these subroutines. The modeling subroutines also will have a definition module associated with them.

The airmass computations which are performed during the active phase of the differential equation solution are performed by one of a set of subroutines. In addition, in the Engine/Drive System technical modules, the engine performance computations are carried out in a similar manner by a user-selected subroutine. These subroutines are developed and validated in a manner identical to the technical modules, and are also considered to be CPCI's.

This capability allows for the flexibility of the user to choose a rotor analysis and a fuselage analysis and to independently select airmass analyses as appropriate. The same flexibility exists in selecting drive system dynamics and engine performance analyses.

Utility Subroutines - In addition, a set of subroutines have been identified as CPCIs which perform a number of utility functions, and may be called by technical modules, technical subroutines, or the executive CPCIs. They include such functions as matrix operations and data checking.

SYSTEM CAPABILITIES

Satisfaction of Requirements

Careful system synthesis and analysis has ensured the satisfaction of all requirements of the Type A System Specification. The following summary identifies the major concerns of the specification and the way in which they have been resolved in the system design.

The General Functional Capability has been provided in the System through a comprehensive set of control inputs termed the System Control Language. Through this language, the user can define any arbitrary rotorcraft analysis configuration and direct its execution. Data base management capabilities have also been provided to permit storage of the analysis in the data base for later use and to provide for the storage and maintenance of engineering data.

The Particular Functional Capability has been provided in the System through the storage of specialized procedures in the data base for later recall and execution. Using this feature, the system developer will define all standard analyses and their procedures in a "Master Data Base" file which will be delivered with the System.

The Detailed Functional Capabilities (DFC) are provided through various stored PFC procedures. By definition, a set of related DFCs are grouped to form the specification for a PFC. Thus, a DFC is formed by selecting specific options within a PFC.

The External Model Functional Capabilities (EMFC) have not as yet been fully defined in the specifications provided. However, the general requirements for EMFCs are satisfied by specialized technical modules which will output data in a form usable by other computer programs.

Availability of System Capabilities

The System will be produced in two releases. The First-Level Release will provide most of the Application Executive Capabilities for use in a batch processing environment and a significant number of technical capabilities. It would not include both finite element and module analysis approaches. The Second-Level Release will provide interactive operation and will extend the technical capabilities to include more complex analysis methods and component representations.

SYSTEM USAGE

System Control Language

The System Control Language (SCL) provides the user with a comprehensive interface to the System helicopter modeling and data base management capabilities. These include the control of execution sequence in the analysis of a variety of aircraft configurations and the ability to define data base entry formats and their content, and retrieval and update of values in these data base entries.

Levels of Use

The System provides three levels of user interface permitting system usage with a minimum of training while simultaneously providing extended features to the experienced user. The three levels of use are as follows:

- a. Basic System Usage
- b. Intermediate System Usage
- c. Advanced System Usage

Basic System Usage - The basic level of system usage provides the engineer with the ability to introduce rotorcraft physical component data to the system and invoke standard analysis procedures. The SCL statements which provide these capabilities are of two types:

- a. Sequence Control Statements - providing the engineer with control over the order in which analysis procedures are executed
- b. Data Base Maintenance Statements - providing the engineer with the ability to add, change, or delete physical characteristics data residing in his data base file.

Intermediate System Usage - The intermediate level of system usage provides the engineer with the ability to define specialized rotorcraft analysis configurations and procedures. An expanded set of SCL statements and two specialized data base records are used to implement these capabilities.

- a. Helicopter Model Definitions (HMD) - The HMD is used to describe an arbitrary rotorcraft analysis configuration to the System. An HMD will identify the mathematic technique, analysis method, and component analysis technical modules and subroutines that are to be used.

- b. **Stored Procedure Definition (SPD)** - The SPD is used to describe a set of Sequence Control statements for storage in the data base for subsequent recall via the CALL statement.

Advanced System Usage - The advanced level of system usage provides the research engineer with expanded Sequence Control and Data Base Maintenance capabilities and with the ability to install new technical capabilities. An additional data base record format is provided for this purpose. This record provides for the installation of new technical modules and subroutines. Termed the Technical Module Definition (TMD) or Definition Module, the record will identify input, output, degrees-of-freedom, and coupling relationships for a specific technical module or subroutine and make the functional portion of that module logically available for use.

Development and Maintenance Aids

Normal system development and maintenance activities often result in the introduction of errors in existing, tested processes. These activities encompass the installation of new capabilities and modification and deletion of existing capabilities. Often, new capabilities require new or modified record formats and thus, changes impact other processes and proliferate throughout the system. These problems are answered in part by the capabilities provided to the intermediate and advanced user of the System. But, in addition to these capabilities, the system development and maintenance teams will have a set of "Data Base Definition" statements available which will permit the definition of new record formats and modification of existing record formats.

Resource Utilization

The modular design of the System and extensive use of dynamic loading throughout the Application Executive to control the residency and nonresidency of system components results in the minimization of executive memory overhead. Although memory utilization will vary during system execution, it is estimated that a typical analysis problem can be solved in less than 95K bytes of memory.

SYSTEM DEVELOPMENT

Organizational Responsibilities

The organization for the development phase should be a project-oriented organization designed to maximize the utilization of resources but still provide all the necessary functions for successful development. A formal organization

is established for clarity of job assignments, minimizing unnecessary interactions, controlling changes and establishing responsibilities and direction. To ensure that the above areas are defined, specific statements and assignments must be made for participants in the Second Generation Comprehensive Helicopter Analysis System.

The Development Contractor will provide management and control of the Development Phase under the auspices of the Government and within the Statement of Work (SOW). Management and control features (other than company policies and procedures) must include subcontract management, analysis, design, programming, testing and documentation to ensure that the resultant products for the System are acceptable, reliable, and standardized to be transportable and maintainable.

The Development Contractor, with defined responsibilities of management and control for all activities, standards and deliverable products will be responsible for the development of the executive area.

A Technical Subcontractor to the Development Contractor should be utilized for the technical area to provide the expertise that is required for rotorcraft technology. The Technical Subcontractor should be an integral part of the Development Phase team. Other technical subcontractors can provide particular rotorcraft expertise for consultation and development. Utilizing the concept of Development Phase Contractor and Technical Subcontractor, definitive allocations of effort can be made.

The Technical Subcontractor will be issued a Statement of Work that will be a subset of the Government's Statement of Work and contract provisions, and will establish the overall objectives, assignments and expectations of the work to be performed by the Technical Subcontractor. The Statement of Work will be oriented to work in helicopter technology and technical CPCIs, as the Development Phase Contractor will be working in the Executive area.

All types of formal communication to the Government that are stated in the Development Plan will be the responsibility of the Development Contractor. However, the Technical Subcontractor should have the responsibility to adhere to activities (communications, progress and cost reports, formats, standards, etc.) of the Development Plan with the Development Contractor in the same manner as the Development Contractor will adhere to the plan with the Government.

The Technical Subcontractor will have the primary responsibility to develop technical CPCIs for the First-Level Release with the assistance, as necessary, of technical consultants or contracting in specific areas of expertise.

During the Development Phase for the Second-Level Release, the Technical Subcontractor will continue with the responsibilities as defined for the First-Level Release for those CPCIs that are not contracted by the Government. In addition, the Technical Subcontractor will develop the preliminary Subsystem Specifications for those technical CPCIs that are contracted by the Government for the Second-Level Release and provide assistance for their completion.

Government-sponsored technical CPI contractors will be responsible for providing specialized rotorcraft expertise for the Second-Level Release. As is generally known, the various participants in the rotorcraft industry have specialized talents and expertise that may not be industry-wide. These specialized talents and expertise will be required to develop technical CPCIs, particularly the more advanced technical CPCIs. A premise of the Statement of Work for the Predesign effort was that few, if any, Second-Level System CPCIs will be developed by subcontractors. It is suggested that the Government-sponsored technical CPI contractors be required to adhere to defined standards to ensure that the final delivered products are standardized. (Note: Items that are not required for delivery but are produced by the contractor can be the contractor's format.) The responsibilities of the Government-sponsored CPI contractors will begin with their receipt of an approved preliminary CPI Subsystem Specification and continue through development and integration of the CPI.

Development Schedule

The activities and events, based upon the Type A System Specification, Statement of Work in the Predesign effort, the system design concept, the program hierarchical structure, and military standard documents, have been established to form the schedule for the First- and Second-Level Releases.

The First-Level System Release will provide a system which makes extensive use of state of the art rotary-wing technology and software techniques. The First-Level Release is expected to contain an executive program and technical modules that will provide the level of sophistication and capabilities comparable to those currently in use by the helicopter industry. The First-Level (1A) System Release for IBM equipment is scheduled for release 32 months after beginning the Development Phase. The First-Level (1B) System Release for CDC equipment is scheduled for delivery 36 months after the beginning of the Development Phase.

The Second-Level System release will provide more advanced rotary-wing technology and software techniques than the First-Level System. It will complete the executive system and incorporate additional functional capabilities using advance state of the art engineering analysis.

The Second-Level System release will occur near the end of the Development Phase, which is approximately 54 months after its beginning. The Second-Level System will be operable on IBM and CDC equipment. Interim releases have not been scheduled. It is possible and suggested that interim releases be made to provide on-site evaluation.

Documentation

Documentation for large and complex systems can take many different forms for the same purpose. Documentation is primarily used to (1) provide developers with documents that can be reviewed at significant developmental milestones to determine that requirements are met and (2) record technical information to allow coordination of later development and use/modification of the system. Documentation should provide uniformity of format and content particularly within a project as large as the Second-Generation Comprehensive Helicopter Analysis System.

The specifications are the vehicles that dictate the capabilities that will be produced for the System. As such, all input from the various interested agencies and users is necessary and required before the specifications are baselined.

The documents recommended for the System are:

1. Type A System Specification - The baseline Type A System Specification as provided from the Predesign Phase will be used as the document that defines the system requirements and operational capability.
2. System Specification - The System Specification will be produced during system design to identify CPCIs, allocate requirements of the Type A System Specification and specify the complete overall design for the System.
3. Subsystem Specifications - The Subsystem Specifications (comparable to Type B5 Development Specifications) will be provided for CPCIs or a group of similar CPCIs.
4. Program Maintenance Manual - The Program Maintenance Manual will describe the computer programs in a detailed, technical presentation to assist the maintenance programmer in his functions.
5. User's Manual - The primary purpose of the User's Manual is to serve the needs of the user group with documentation sufficient to utilize both the executive system capabilities and technical modules.

6. Theoretical Manual - The purpose of this manual is to provide a concise description of the methods that can be employed in the solution of problems.
7. Test and Implementation Plans - Two types of test plans should be formally documented: (1) Acceptance Test Plan for the System and (2) CPCI Test and Integration Plans for the Computer Program Configuration Items.
8. Test Analysis Reports - Test Analysis Reports describe the status of the computer program system after the Acceptance and Integration tests and provide a presentation of capabilities and deficiencies for review by staff and management personnel.
9. Development Plan - The Development Plan is a planning document that organizes and describes the development effort and provides standards and techniques that can be distributed to participating agencies.
10. Computer Program Documentation - Information and data should be written into the program source listing. The information applies to design, data and flow charts for each program module.

Quality Assurance and Control

Quality assurance and control should begin with the initiation of the project and continue until its completion in the areas of objectives, requirements, design, programming, testing and documentation. Quality assurance should begin with the first specification (Type A System Specification).

It is recommended that a Baseline Review Board be formed to review and critique design products in the Development Phase for approval by the contracting agency of the Government. The Baseline Review Board can be composed of Government personnel, development contractor, and (as appropriate) sub-contractors and CPCI contractors. In addition, the Technical Advisory Group and members of the Government/Industry Working Groups can be contributing members. The Baseline Review Board would review and approve products at formal reviews.

Functional Design Review - The initial effort for the Development Phase involving system synthesis, analysis, risk assessment and trade-offs will result in the publication of a draft System Specification, draft Acceptance Test Plan, revisions to Type A System Specification and revisions to the Development Plan which will be reviewed at a Functional Design Review by the Baseline Review Board.

System Design Concurrence - The drafts of the System Specification (design) and Acceptance Test Plan that were produced for the Functional Design Review (FDR) will be revised, if necessary, to conform to the results of the FDR. The System Design Concurrence by the Baseline Review Board is required to ensure that the final system concept and direction of the system design and test plan meet with the approval of the Government.

Preliminary Design Review - Each subsystem (CPCI) identified and defined within the System Specification will undergo a preliminary subsystem design. A Preliminary Design Review will be held for each subsystem (CPCI) to ensure that the direction of the CPCI meets the requirements as defined in the Type A System Specification and the System Specification.

Critical Design Review - The detailed subsystem design occurs after preliminary design approval and before programming for a CPCI. The results of the development of detailed Subsystem Specifications and Test and Integration Plans will be reviewed by the Baseline Review Board at a Critical Design Review. The purpose of this review is to assure the accuracy and adequacy of CPCIs prior to their actual development and to ensure that the developing System continues to meet all requirements that are placed upon it.

It has been estimated that for the First-Level Release and Second-Level Release the Baseline Review Board would convene 13 times over a 15-month period and 14 times over a 14-month period, respectively.

Quality assurance for program development will be based on the techniques of hierarchical structured concepts, analysis and design walk-throughs, Chapin logic flow charts, pseudo code prologues for programmable modules, standardized FORTRAN coding techniques, four levels of tests and scheduled units of work.

Testing Requirements

The testing for the Second-Generation Comprehensive Helicopter Analysis System should be detailed, comprehensive, and structured to verify the accuracy of the code and adequacy of the design. Tests for the programmed System should begin at the lowest programmable level and continue through successively higher levels in a meaningful test hierarchy. An Acceptance Test Plan should be written for the System during the early stages of the Development Phase to be reviewed at the Functional Design Review.

Four levels of testing should be utilized:

- a. Module Testing - Module testing exercises the module through its full range of inputs and outputs and evaluates its performance for any necessary correction. Each and every path, decision, and code of a module is exercised during module testing.
- b. CPCI Testing - CPCI testing exercises the CPCI to validate that it is correctly interpreting input data, successfully performing its processing tasks, and providing arithmetic and logical accuracy as well as statistics for storage utilization and CPU timing.
- c. Integration Tests - The objective of the integration test is to add a tested module or CPCI into the System, exercise it as thoroughly as possible, determine the adequacy of analysis for technical CPCI modules upon which the System is based and prove that the CPCI performs all of its processing tasks.
- d. Acceptance Tests - The objective of the acceptance test is to demonstrate and verify that the programmed System operates according to the specifications and is correctly installed. Acceptance testing is the final quality assurance provision for a particular level of the System.

Training

Training must be complete, structured, and formalized and encompass concepts through usage. This training will prepare the Government to assume maintenance of the System and to provide subsequent training to users.

- a. Understanding the System Concept - This training provides an overview of the System concepts and the functions of the CPCIs (both executive and technical).
- b. Module and Structured Concepts - This training will provide an understanding of how a system and a program are developed using the modern structured techniques.
- c. System Installation - System Installation Training will provide the knowledge required to install the system onto different host computers.
- d. Modifying the Software System - This training will provide the information to modify the system for the purpose of adding or changing technical CPCIs and the Executive.
- e. System Usage - This training provides the potential system user with the knowledge required to enter the System, process data, checkpoint if required, and evaluate output results.

The interactive version of the system will have the capability to tutorially guide the user in the use of the System with descriptive information about the operation of the System.

CONCLUSION

Based on the results of the Predesign Phase, the Second-Generation Comprehensive Helicopter Analysis System has been determined by Control Data Corporation and Kaman Aerospace Corporation to be a feasible system that will provide the rotorcraft industry and users with a viable vehicle for future endeavors in rotorcraft technology.

SYSTEM DESIGN

OBJECTIVES OF THE SYSTEM

The primary objective of the Second-Generation Comprehensive Helicopter Analysis System (SGCHAS or System) is to provide to the rotorcraft research and development community a system that will be a major step toward satisfaction of the need for accurate prediction of loads, aeroelastic stability, stability and control, performance, and acoustics of rotorcraft of all sizes and rotor types for all rotorcraft life-cycle phases. In addition, the System is to provide the ability to analyze arbitrary rotorcraft component test configurations.

MAJOR DESIGN CONSIDERATIONS

The specific processing requirements, cost and accuracy goals, computer hardware and software considerations, and other design considerations are defined in considerable detail in Reference 1. These considerations were allocated to three major categories:

- a. Processing Requirements - the quantitative goals and requirements that can be demonstrated through testing.
- b. Design Goals - the qualitative goals defined for the System which, though not demonstrable, impact the design and development of the System.
- c. Technical Design Considerations - the specific quantitative and qualitative technological requirements and goals which require special consideration.

Processing Requirements

The basic processing requirements of the System are described in the section of the Baseline Type A System Specification entitled "FUNCTIONAL CAPABILITIES". Briefly, they are defined as follows:

- a. General Functional Capability - the ability of the System to accept inputs describing an arbitrary configuration of helicopter and other analysis components, their physical characteristics and coupling relationships, and the logical sequence of problem solution, and to accurately solve the particular analysis problem that is described.

¹ Control Data Corporation; Baseline Type A System Specification for the Second Generation Comprehensive Helicopter Analysis System (in response to Task IIIa, CDRL A008, contract DAAJ02-77-C-0058), Control Data Corporation, Hampton, Virginia 23666, and Kaman Aerospace Corporation, Bloomfield, Connecticut 08002; January 27, 1978.

- b. Particular Functional Capability - the ability of the System to analyze predefined rotorcraft analysis problems.
- c. Detailed Functional Capability - the ability of the System to analyze any one of many specifically detailed rotorcraft analysis configurations (e.g., a single-rotor helicopter having a four-bladed articulated main rotor, and a two-bladed teetering tail rotor, and analyzed for preliminary design performance characteristics).
- d. External Model Functional Capability - the ability of the System to generate data in a form that is readily usable by programs and processes outside the SGCHAS.
- e. Cost and Accuracy Assessment Functional Capability - the ability of the System to provide a priori and after-the-fact estimation of processing costs, and to provide an assessment of the accuracy of a particular analysis through the correlation of analysis results with experimental data.
- f. Diagnostic Capability - the ability of the System to diagnose three levels of errors (fatal, warning, and informative) and permit the user to define the level of error that is to terminate processing.

In addition to these functional capabilities, the specification identifies other processing requirements in the section entitled "SOFTWARE". These requirements are as follows:

- a. Restart Capability - the system design must provide the ability to restart processing following any interruption (provided checkpoint data has been retained) and to request the output of additional data or modification of existing data prior to reinitiating processing.
- b. Graphic Capability - the System design must provide both online and offline graphics capability for effective use of the System.
- c. Interactive Capability - the System design must provide for interactive use of the System through teletype, graphics CRT, and nongraphic CRT terminals without compromising the batch processing capabilities.

Design Goals

The system design goals for the SGCHAS are defined and implied throughout the Baseline Type A System Specification (Reference 1) and in the Statement of Work (Reference 2) for the predesign of the SGCHAS. A summary of these goals is in the paragraphs that follow:

Longevity - One of the principal goals affecting the Control Data and Kaman approach to the system design has been that of longevity. It is expected that the SGCHAS will have a 15-year or longer life span. Thus, if the System is to obtain and maintain widespread acceptability throughout its existence, it must be sufficiently generalized and flexible to encourage the development and integration of new rotorcraft analysis capabilities.

Cost Effectiveness - The goal of cost effectiveness is closely related to the goal of system longevity since the system will be used only as long as similar capabilities are not provided at a lower cost. In order to ensure the long-term cost effectiveness of the System, steps must be taken during the design of the system to minimize manpower and computer resource utilization and maximize computational efficiency and accuracy.

Ease of Use - Any system that is going to interface with people on a regular basis requires the design team to pay careful attention to the human/machine interface. The SGCHAS requires particular attention to the various levels of use and system capabilities with which the user must interact. Any user interface language developed for the System must provide simple, readily identified statements through which the user can direct system operation.

Hardware Independence - The goal of hardware independence must be recognized and addressed early in the design effort. Although 100 percent hardware independence is not practical, steps can be taken during design and development to minimize and localize hardware dependencies and, thus, maximize system transportability.

Maintainability - The maintainability of the System is another design goal that is interwoven with the goal of longevity. Many systems die early due to their complexity and lack of growth potential. To be maintainable, a system must

¹ Control Data Corporation; Baseline Type A System Specification for the Second Generation Comprehensive Helicopter Analysis System (in response to Task IIIa, CDRL A008, contract DAAJ02-77-C-0058), Control Data Corporation, Hampton, Virginia 23666, and Kaman Aerospace Corporation, Bloomfield, Connecticut 08002; January 27, 1978.

² Anon.; Statement of Work, Predesign of the Second Generation Comprehensive Helicopter Analysis System, DAAJ02-77-C-0058, Section F, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia 23604; September 8, 1977.

be characterized by functional modularity; that is, each programmable module performs a single function (or limited number of related functions) and is limited in size and complexity. Such modularity must be designed into the system.

Technical Design Considerations

In addition to the formal requirements for the System, as specified in the Type A System Specification, there are other considerations related to technical issues which have had a major impact on the System design. These are discussed below. It is shown that these design considerations increase technical capability and decrease user cost.

Engineer Orientation - There are numerous decisions that must be made during the development of problem formulations, execution of analyses, and interpretation of results that can only be made by knowledgeable engineers. The System has therefore been developed so as to allow for convenient engineering control of the System at all levels of usage.

The System Control Language (SCL) has been designed as a small set of meaningful and convenient statements. This language specifies the analysis to be performed by: establishing the order of major steps in the analysis (such as blade model analysis, iterate to trim, and harmonic analysis of specified loads); specifying the dynamic component representations (such as rotor, fuselage, and control system); and specifying major decisions during execution (such as test for trim and modify controls, interrupt solution and insert structural damage parameters, and perturb solution for nonlinear stability analysis).

The SCL for basic and often-used analyses may be entered into the data base as Stored Procedure Definitions (SPD) and may be invoked and executed by a single statement. The subset of the SPDs which are designated as Particular Functional Capabilities (PFCs) will be formulated and delivered as part of the System. In addition, any other particular problem formulations developed at a user site may be named and stored for future use. The user may retrieve from the data base and execute any SPD and may, if desired, make temporary changes in any element of the procedure such as: selection of a different component representation, modification of mathematical algorithm, or modification of input data. Alternatively the engineering user may easily set up a complete problem formulation at run time which is appropriate to the problem at hand. Thus, the engineer will have the capability to perform any analysis required by his technical judgement for the solution of a particular problem. The user will be able to select the order of major analyses, the level of complexity, and the

method of analysis of each component and numerical algorithm. It will not be necessary to perform an overly complex analysis that is wasteful of resources or an inadequate analysis in which there will be little confidence.

If an analysis produces predictions which the engineer has technological reasons to question, it may be easily rerun, changing appropriate levels of complexity of components or numerical methods. Prior to production running of a large number of conditions, the engineer may conveniently make trial runs using several problem formulations and, by comparing results, select the most efficient and effective problem formulation. The above-mentioned capabilities will insure that the System will maximize the technical effectiveness of the System and will result in costs which are significantly lower than those of present-generation analytical methods.

Technical Modules - Each major component of the helicopter should be represented at several levels of complexity in order to give the engineer the capability to establish a problem solution which is customized to specific needs. In addition to levels of complexity, some of the component representations should allow the user a selection of analytical methods. In the case of the rotor or airframe, for example, the user should have the choice of either a modal or finite element representation.

This choice of alternate methods is important for several reasons. First, there is no universal acceptance of any one method and, in fact, it may not have been shown technically that one method is superior to others. Second, under certain situations, an engineer may select one method for efficiency and for other situations may select another method for increased accuracy. This choice is obviously in the realm of the engineer and this option must be available. Third, there is the matter of personal preference, and the choice is necessary for universal acceptance of the System.

The technical capabilities of the System are separated into units (technical modules) related to individual helicopter components or analysis methods. The linkages and coupling of these technical modules are handled by the Application Executive.

The System Concept presented places no limitations on the components which may be represented in a technical module. A simple vibration absorber model or a complete helicopter may each be a single technical module. In fact, some of the modules established in the Development Plan represent simple complete helicopter analyses. There are reasons, however, why it is not recommended that each physical component be represented in individual modules. The two possible extremes for levels of complexity of technical modules are: many modules, each representing the smallest possible hardware component; or a

single complex module representing the entire helicopter with many input options. The first extreme is not practical from the aspect of program maintenance and user convenience. The second is impractical because of poor core efficiency and increased risk of erroneous logical input. The technical approach recommended in this report is a compromise between these two extremes which optimizes the combination of user convenience and resource efficiency. In addition to the technical modules which will be delivered with the System, the user will have the option to add to the System any other technical modules he desires, whether they represent new components or new methods of analysis.

Duplication of Capabilities - Whenever major "stand-alone" analyses exist, it was chosen not to attempt to duplicate these capabilities in this System since the effort required could be better spent elsewhere. Finite element analysis (e. g. , NASTRAN) and computational fluid dynamics are examples of existing technology which fall in this category. The System is designed so that the output of a detailed finite element analysis (reduced mass and stiffness matrices or a modal analysis) or tables of aerodynamic coefficients may be directly input. Provisions are included, however, for user convenience, for simpler analyses of this type or for modifications such as fuel usage or cargo jettison without rerunning a major external program.

MATHEMATICAL BASIS OF THE SYSTEM

The basic mathematical formulation for performance, stability and control, loads, acoustics, and aeroelastic stability problems used as a basis for the design of the System is a set of second-order differential equations. In many cases, a time domain numerical solution is required. In other cases, the equations are converted to the frequency domain or into a set of algebraic equations prior to solution.

In the most general cases, where solutions to the differential equations are required, the equations for each of the major components may be developed and coupled into a set of simultaneous differential equations representing the complete system. Each of the component representations may include nonlinear and periodic effects. A numerical algorithm is then required to obtain time history solutions of the equations. Depending on the application, the solutions may be carried to steady state and then iterated upon to trim. For other applications, transient or perturbed solutions may be required, or it may be required to interrupt the solution and modify structural parameters.

When the set of differential equations is linear, it may be converted to a frequency domain problem and then subjected to an eigenanalysis. Other special procedures, such as ground resonance analysis, modal analyses, or use of the method of undetermined coefficients, may require special mathematical algorithms.

When the basic solution of the equations of motion has been completed, post-processing of data is often required to obtain such information as the harmonic content of loads, stability parameters, or acoustic response.

All of the above-mentioned analyses are treated in the System by a set of technical modules that are coupled by the Application Executive under the control of the user. In the following sections, the crucial issue of coupling of analyses and components is discussed.

Sequential Coupling

Analyses that are performed within a single technical module are coupled to other analyses simply by passing data from one analysis to another. This type of coupling is called "sequential" coupling and the technical modules are called "stand-alone" or sequential modules.

The technical modules that fall in this category include simple analyses of a complete system (such as a simple preliminary design performance analysis, a Coleman ground resonance analysis, or a simple or complex analysis of a blade or fuselage). In addition, any post-processing mathematical algorithm (such as harmonic analysis, FFT, far field acoustic prediction) is also a "stand-alone" analysis since it interacts with other analyses only through the passage of data.

Iteration through a set of sequential analyses is also included in this classification. A simple illustration of blade modal analysis, performance analysis, a trim algorithm which changes controls, and a harmonic analysis of loads, is schematically shown in Figure 2.

Dynamic Coupling

Many analyses will be performed by linking together an arbitrary combination of analyses of individual components. It is not correct in these cases to attempt to analyze each component separately; the complete system must be solved as a unit. This type of coupled system may be represented schematically as in Figure 3.

The capability to perform this type of coupling between the physical components of the helicopter is crucial to the success of the System.

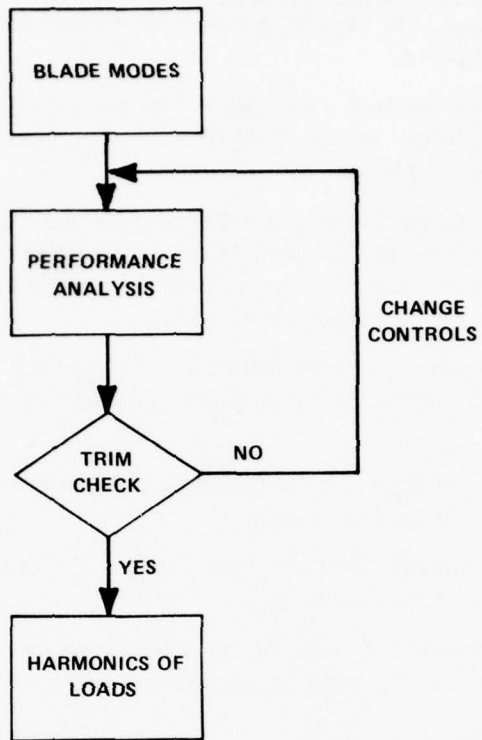


Figure 2. Illustration of a Sequential Analysis.

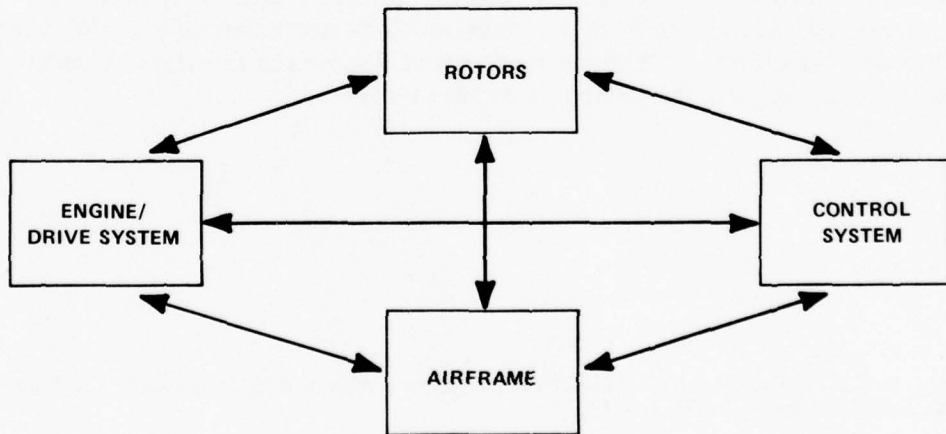


Figure 3. Illustration of Dynamically Coupled System.

Objective of Dynamic Coupling Capabilities - In order to establish a basis for the design of the System, the following goals have been established for the method of dynamic coupling.

- a. Independent Component Analysis - The modeling of each component must be completely independent of other component representations which may be coupled to it.
- b. Multiple Methods of Analysis - There must be no limit on the methods of analysis which may be used in modeling components. Each component may make use of finite element, Myklestad, modal, Galerkin or other methods, without restriction.
- c. Nonlinearities and Periodic Effects - There must be no practical restriction of the nonlinear or periodic effects modeled in each component.
- d. Automatic Coupling - The coupling of the components must not require special inputs from the user.
- e. Time and Frequency Domain Applications - The method must apply equally well to both regimes.
- f. Exact Method - There must be no approximations built into the system which are not subject to user control.

Basis of the Method - Methods of analyzing systems of components by separately analyzing the individual components have been available for many years. The most commonly used family of methods was initiated by Hurty in 1964 (Reference 3). These methods all use "modes" of the components. Various methods use free or constrained modes with or without mass loading and "constraint" modes as introduced by Hurty. (See References 4, 5, 6 for typical methods and summaries.) These methods are extremely useful when dealing with structures with large numbers of degrees of freedom or when separate analyses are performed at remote sites.

³ Hurty, W. C., "Dynamic Analysis of Structural Systems by Component Mode Synthesis," Jet Propulsion Lab., Technical Report 32-530, Jan. 15, 1964.

⁴ Benfield, W. A. and Hruda, R. F., "Vibration Analysis of Structures by Component Mode Substitution," AIAA Journal, Vol. 9, No. 7, July 1971.

⁵ Goldman, R. L., "Vibration Analysis by Dynamic Partitioning," AIAA J. 7 (6), 1152, 1154, 1969.

⁶ Hou, Shou-njen, "Review of Modal Synthesis Techniques and a New Approach," Shock and Vibration Bulletin No. 40, Dec. 1969.

References 7 and 8 have made comparative studies of the accuracies of the various methods available. The accuracy of each method depends on the types and the number of modes used and the particular characteristics of each of the substructures. Under differing conditions different methods will provide greater accuracy or greater efficiency or both greater accuracy and greater efficiency.

The following is a list of considerations related to the application of these methods to the SGCHAS:

- a. Since a modal analysis is required, the concept of alternate modeling methods is compromised.
- b. The capability to treat component damping is limited.
- c. The capability to treat component nonlinearities is limited.
- d. Normal modes of nonlinear or periodic structures are not rigorously defined.
- e. Rotor (not blade) modes are required but these are not generally defined.
- f. Modal analyses of certain structures are not appropriate or convenient (e.g., control systems).

It has been concluded that modal synthesis is not appropriate for this application.

There is a method available, however, which is simple and exact, and satisfies all the goals established in the previous paragraph. This method is a modification of methods which have been in use for many years in finite element synthesis (for example, see Reference 9) and has recently been applied to general components in frequency domain applications (Reference 10, 11). It will be shown that this method applies equally well to time domain problems.

⁶ Hou, Shou-nien, "Review of Modal Synthesis Techniques and a New Approach," Shock and Vibration Bulletin No. 40, Dec. 1969.

⁷ Benfield, W. A., Bodley, C. S., and Morosow, G., "Modal Synthesis Methods," Presented at the Space Shuttle Dynamics and Aeroelasticity Working Group Symposium on Substructuring, Marshall Space Flight Center, Alabama, Aug. 30-31, 1972.

⁸ Hurty, W. C., Collins, J. D., Hart, G. C., "Dynamic Analysis of Large Structures by Modal Synthesis Techniques," Computers and Structures 1 (4), 535-563, Dec. 1971.

⁹ Przemieniecki, J. S., Berke, L., "Digital Computer Program for the Analysis of Aerospace Structures by the Matrix Displacement Method," AFDL report No. FDLTDR64-18, April 1965.

¹⁰ Berman, A., "Vibration Analysis of Structural Systems Using Virtual Substructures," The Shock and Vibration Bulletin 43, NRL, Washington, D.C., June 1973.

¹¹ Berman, A., Giasante, N., "CHIANTI - Computer Programs for Parametric Variations in Dynamic Substructure Analysis," The Shock and Vibration Bulletin 47, NRL, Washington, D.C., Sept. 1977.

Derivation of Coupled Equations of Motion - Each component of the System is represented by a set of matrix differential equations of the form

$$M_i \ddot{V}_i + C_i \dot{V}_i + K_i V_i = F_i \quad (1)$$

where M_i , C_i , K_i are coefficient matrices which may be functions of the state variables, \dot{V}_i , V_i and time, t . F_i is the forcing function and includes the external loads and any nonlinear terms which are not included on the left-hand side of the equations and forces at the boundary. The elements F_i will generally be functions of t , \dot{V}_i , V_i .

The complete coupled system is represented by a corresponding set of differential equations

$$M_c \ddot{V}_c + C_c \dot{V}_c + K_c V_c = F_c \quad (2)$$

The coupled system equations, (2), may be obtained from the component equations, (1), by a straightforward transformation. Consider a matrix T_i which is constant and which transforms the coupled variables to the variables of the component i , as follows:

$$\begin{aligned} \ddot{V}_i &= T_i \ddot{V}_c \\ \dot{V}_i &= T_i \dot{V}_c \\ V_i &= T_i V_c \end{aligned} \quad (3)$$

This transformation is made possible through a naming convention through which the coupled Degrees of freedom have the same variable name in the representation of each component. T_i is then just an implementation of this naming convention.

The requirement that T_i be constant implies that the transformation may not be used to transform from a rotating coordinate system to a fixed system. This is no hardship since such transformations will be performed within the individual component representation. (This effect is discussed in more detail in a later section.)

In many applications (see examples in following sections), V_i is simply a subset of V_c and thus T_i is a rectangular matrix of unit and null elements. Substituting the relationships (3) in Eq. (1) and premultiplying by T_i^T results in:

$$T_i^T M_i T_i \ddot{V}_c + T_i^T C_i T_i \dot{V}_c + T_i^T K_i T_i V_c = T_i^T F_i \quad (4)$$

When Eq. (4) is summed over all components, the result is Eq. (2), the complete coupled system with the following relationship:

$$\begin{aligned}
 M_c &= \sum_i T_i^T M_i T_i \\
 C_c &= \sum_i T_i^T C_i T_i \\
 K_c &= \sum_i T_i^T K_i T_i \\
 F_c &= \sum_i T_i^T F_i
 \end{aligned}
 \tag{5}$$

Thus, it is seen that the individual component equations may be simply transformed into the exact equations of the complete system. It should be noted that the above equations may also be derived using Lagrange's method where the system kinetic energy, potential energy, and dissipation function may be written as:

$$\begin{aligned}
 \frac{1}{2} \dot{V}_c^T (\sum_i T_i^T M_i T) \dot{V}_c \\
 \frac{1}{2} V_c^T (\sum_i T_i^T K_i T) V_c \\
 \frac{1}{2} \dot{V}_c^T (\sum_i T_i^T C_i T) \dot{V}_c
 \end{aligned}$$

Implementation Considerations - When M_i , C_i , K_i are constants, the transformations to M_c , C_c , K_c may be carried out prior to the differential equation integration. This will be the case for many of the simpler problems. Also, when a linear eigenproblem is being formed, these matrices will, by definition, be constants. In this case the system is transformed into the frequency domain by converting Eq. (2) into

$$(-\omega^2 M_c + i\omega C_c + K_c) V_{c_0} = F_{c_0}(\omega)$$

Even for more complex systems, one can expect most of these matrices to be constant. An advanced rotor representation, an adaptive control system, and a fuselage on its landing gear are examples of components which will have variable matrix coefficients. When this situation exists, Eq. (5) must be evaluated at each time increment during the equation solution. It is, of course, inefficient to completely reevaluate all the matrices of Eq. (5) when only one

or two of them actually vary. In practice, the Application Executive will have information as to which components have constant coefficients and will initially form the summations over these components, as shown below.

$$\left. \begin{aligned} M_{c_0} &= \sum_{i=i_c} T_i^T M_i T_i \\ C_{c_0} &= \sum_{i=i_c} T_i^T C_i T_i \\ K_{c_0} &= \sum_{i=i_c} T_i^T K_i T_i \end{aligned} \right\} \begin{array}{l} i_c \text{ refers to components} \\ \text{with constant coefficients} \end{array}$$

$$\text{then} \quad \left. \begin{aligned} M_c &= M_{c_0} + \sum_{i=i_v} T_i^T M_i T_i \\ C_c &= C_{c_0} + \sum_{i=i_v} T_i^T C_i T_i \\ K_c &= K_{c_0} + \sum_{i=i_v} T_i^T K_i T_i \end{aligned} \right\} \begin{array}{l} i_v \text{ refers to components} \\ \text{with varying coefficients} \end{array}$$

This process is discussed in detail in the section entitled "System Operations for Solution of Equations".

The transformation matrices, T_i (see more detailed discussion below), are of the order $N_i \times N_c$ (number of degrees of freedom in the component multiplied by number of degrees of freedom of the coupled system). Most of the elements will be zeroes and most of the nonzero elements will be unity. It is extremely inefficient to store these matrices and to perform all the multiplications by 0 and 1. The T matrices are readily interpreted in terms of a simple and concise decision table, and the effective operations are efficiently performed by a straightforward algorithm.

Formation of Transformation Matrices - Because of the use of a uniform standard notation throughout all the technical modules, it is possible for the Application Executive to automatically form the T matrices. (Note that the T matrix is used here in a symbolic sense; actual implementation will be a more efficient algorithm as discussed in the previous paragraph.) The coupled degrees of freedom may be explicit or implicit as discussed and illustrated below.

Explicit Coupling - The same degrees of freedom may appear in more than one component. When this occurs, the Executive will recognize this fact through the transformation matrices and will couple the systems at these points. As examples, the hub degrees of freedom will appear in the rotor component and the airframe component, and the pitch horn displacement will appear in the rotor and the control system (end of control rod). When such duplication of degrees of freedom appears the transformations constrain these degrees of freedom to be, in fact, equal.

Consider a simple example of this type of coupling, exemplified by three spring mass systems connected at coordinates x_1 and x_3 as illustrated in Figure 4.

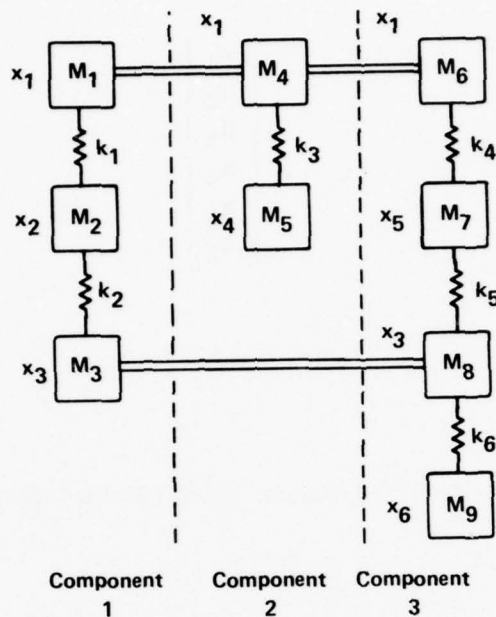


Figure 4. Sample of Explicit Coupling.

The M_i , K_i and V_i matrices for the three components may be written (the C matrices are all 0 in these simple examples):

$$\begin{aligned}
 M_1 &= \begin{bmatrix} M_1 & & \\ & M_2 & \\ & & M_3 \end{bmatrix} & M_2 &= \begin{bmatrix} M_4 & \\ & M_5 \end{bmatrix} & M_3 &= \begin{bmatrix} M_6 & & & \\ & M_7 & & \\ & & M_8 & \\ & & & M_9 \end{bmatrix} \\
 K_1 &= \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1+k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} & K_2 &= \begin{bmatrix} k_3 & -k_3 \\ -k_3 & k_3 \end{bmatrix} & K_3 &= \begin{bmatrix} k_4 & -k_4 & 0 & 0 \\ -k_4 & k_4+k_5 & -k_5 & 0 \\ 0 & -k_5 & k_5+k_6 & -k_6 \\ 0 & 0 & -k_6 & k_6 \end{bmatrix} \\
 V_1 &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} & V_2 &= \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} & V_3 &= \begin{bmatrix} x_1 \\ x_5 \\ x_3 \\ x_6 \end{bmatrix}
 \end{aligned}$$

A coupled degree of freedom vector is formed by listing each variable only once, as follows:

$$V_c = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

The transformation matrices then become (in symbolic form)

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Carrying out the transformation of Eq. (5) the coupled coefficient matrices become

$$M_c = \begin{bmatrix} M_1 + M_4 + M_6 & 0 & 0 & 0 & 0 & 0 \\ 0 & M_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_3 + M_8 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & M_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & M_9 \end{bmatrix}$$

$$K_c = \begin{bmatrix} k_1 + k_2 + k_4 & -k_1 & 0 & -k_3 & -k_4 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 & 0 & 0 \\ 0 & -k_2 & k_2 + k_5 + k_6 & 0 & -k_5 & -k_6 \\ -k_3 & 0 & 0 & k_3 & 0 & 0 \\ -k_4 & 0 & -k_5 & 0 & k_4 + k_5 & 0 \\ 0 & 0 & -k_6 & 0 & 0 & k_6 \end{bmatrix}$$

These may, of course, be verified as correct by comparing them with the complete coupled system.

It should be noted that while the illustration shows only unconstrained components, no such limitation is implied. Any or all of the masses may be connected to ground with springs and dampers. This will add terms to the diagonal elements of K_i and C_i . The processing as shown is not changed.

If external forces are applied to the component, the F_i vectors may be written (identifying the forces with the same subscript as the masses):

$$F_1 = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad F_2 = \begin{bmatrix} f_4 \\ f_5 \end{bmatrix} \quad F_3 = \begin{bmatrix} f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix}$$

The coupled forcing functions (from equation 5) then becomes:

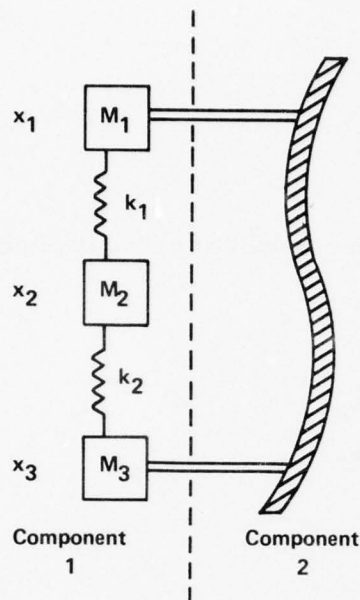
$$F_c = \begin{bmatrix} f_1 + f_4 + f_6 \\ f_2 \\ f_3 + f_8 \\ f_5 \\ f_7 \\ f_9 \end{bmatrix}$$

The transformation matrices may be readily formed completely automatically by comparison of the list of variable names associated with each component and the list associated with the coupled variables. The coupled variable may also be formed automatically by a comparison of all the component variables. All the necessary information (i.e., the name of the degrees of freedom of each component) is available to the Executive.

Implicit Coupling - In some models of components, the coordinates which interface other components may not be specific degrees of freedom. For example, when a modal representation is used to model a fuselage, the hub degrees of freedom do not appear as explicit degrees of freedom in the fuselage equations of motion. However, these hub coordinates may be expressed as a linear combination of the generalized modal displacements which are the degrees of freedom of the fuselage equations. The coefficients will be the modal amplitudes at the hub locations. These implicit relationships are used to form the transformation matrices.

The types of modes used in any particular application will be those deemed by the analyst to be the most representative of the structure and which will yield the best analytical results. They may be free-free modes or a combination of constrained plus rigid body modes and static deformations due to applied forces. In each case the physical displacement of the structure is a linear combination of the modes used and is consistent with the following example. Additional equations of constraint may be required to ensure satisfaction of the appropriate boundary conditions.

Consider a simple example of spring mass system coupled to a modal model as in Figure 5.



$$x = q_1 \phi_1 + q_2 \phi_2$$

$$x_1 = a_{11} q_1 + a_{12} q_2$$

$$x_3 = a_{31} q_1 + a_{32} q_2$$

$$\text{Where: } a_{11} = \phi_1(x_1) \quad a_{12} = \phi_2(x_1)$$

$$a_{31} = \phi_1(x_3) \quad a_{32} = \phi_2(x_3)$$

Figure 5. Example of Implicit Coupling.

The matrices for these two components are:

$$M_1 = \begin{bmatrix} M_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & M_3 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} \bar{M}_1 & 0 \\ 0 & \bar{M}_2 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} \bar{k}_1 & 0 \\ 0 & \bar{k}_2 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

where $\bar{M}_1, \bar{M}_2, \bar{k}_1, \bar{k}_2$ are the generalized masses and stiffness of modes ϕ_1, ϕ_2 .

The coupled degrees of freedom may be defined

$$V_c = \begin{bmatrix} x_2 \\ q_1 \\ q_2 \end{bmatrix}$$

where those interface degrees of freedom which are defined by the implicit relationships:

$$\begin{aligned} x_1 &= q_{11} q_1 + q_{12} q_2 \\ x_3 &= q_{31} q_1 + q_{32} q_2 \end{aligned}$$

are eliminated from V_c .

The transformation matrices, then are written

$$T_1 = \begin{bmatrix} 0 & a_{11} & a_{12} \\ 1 & 0 & 0 \\ 0 & a_{31} & a_{32} \end{bmatrix} \quad T_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using Eq. (5) the coupled matrices become:

$$M_c = \begin{bmatrix} M_2 & 0 & 0 \\ 0 & M_1 a_{11}^2 + M_3 a_{31}^2 + \bar{M}_1 & M_1 a_{11} a_{12} + a_{31} a_{32} \\ 0 & M_1 a_{11} a_{12} + a_{31} a_{32} & M_1 a_{12}^2 + M_3 a_{32}^2 + \bar{M}_3 \end{bmatrix}$$

$$K_c = \begin{bmatrix} k_1 + k_2 & -k_1 a_{11} - k_2 a_{31} & -k_1 a_{12} - k_2 a_{32} \\ -k_1 a_{11} - k_2 a_{31} & k_1 a_{11}^2 + k_2 a_{31}^2 + \bar{k}_1 & k_1 a_{11} a_{12} + k_2 a_{31} a_{32} \\ -k_1 a_{12} - k_2 a_{32} & k_1 a_{11} a_{12} + k_2 a_{31} a_{32} & k_1 a_{12}^2 + k_2 a_{32}^2 + \bar{k}_2 \end{bmatrix}$$

While transformation matrices of this type are somewhat more complex to form automatically, this can be done in an unambiguous and routine manner.

Modeling of Components

Physical Components - The only restriction on the modeling of components is the requirement that the transformation from the coupled system degrees of freedom to the component degrees of freedom be constant. This means that, for example, if the coordinates of a rotor hub were written in the rotating coordinate system, the rotor could not be coupled to the fuselage in a fixed coordinate system. This is no handicap, however, since it is normal for rotor hub degrees of freedom to be written in the fixed system. As long as transformations of this type are made within the component model, no problems will occur at the interfaces. When the transformations are made in this fashion, the resulting coefficient matrices may have periodic elements.

As a simple illustration, consider a two-bladed, hinged rotor, with a flapping and feathering degree of freedom for each blade. Also, the hub has a vertical and pitch degree of freedom. Writing the equations with the blade coordinates in the rotating system and the hub coordinates in the fixed system, the resulting mass matrix may be written as follows:

$$M = \begin{bmatrix} I_\beta & I_x & 0 & 0 & S_\beta & (I_\beta + e S_\beta) \cos \Psi_1 \\ I_x & I_\theta & 0 & 0 & S_\theta & (I_x + e S_\theta) \cos \Psi_1 \\ 0 & 0 & I_\beta & I_x & S_\beta & (I_\beta + e S_\theta) \cos \Psi_2 \\ 0 & 0 & I_x & I_\theta & S_\theta & (I_x + e S_\theta) \cos \Psi_2 \\ S_\beta & S_\theta & S_\beta & S_\theta & M_R & 0 \\ (I_\beta + e S_\beta) \cos \Psi_1 & (I_x + e S_\theta) \cos \Psi_1 & (I_\beta + e S_\theta) \cos \Psi_2 & (I_x + e S_\theta) \cos \Psi_2 & 0 & I_R - \frac{1}{2} I_0 \sum_i \cos 2\Psi_i \end{bmatrix}$$

where I_β , I_θ , I_x are the blade flapping, feathering and product moments of inertia; S_β , S_θ are the respective static moments; M_R , I_R and I_0 are the mass and moments of inertia associated with the complete rotor. Ψ_1 and Ψ_2 are the azimuth angles of blades 1 and 2, respectively.

The control system representation will include the transformation from the nonrotating to the rotating system through the swash plate. The rotating end of the control rod, then, will interface properly with the rotating blade, and the nonrotating portion of the control system will interface properly with the airframe.

It should be noted that there are no restrictions on the nonlinear effects which may be included in any component representation.

Airmass - The interface between the airmass and the structure differs in a significant fashion from the interface between two physical components. The physical coordinates move together at their common coordinates. The aerodynamic effects are due to relative motion between the airmass and the structure.

Components which will be subjected to aerodynamic forces, i. e., the rotors and airframe, will contain a call to a subroutine which will compute the aerodynamic forces. The subroutine will be an external subroutine having a standard argument list. When the user sets up a problem and specifies the particular component modules to be used, the airmass subroutine to be used will also be specified. The argument list will transmit all the local surface displacements and velocities (transverse and angular) and the subroutine will return the local forces and moments. The subroutine will have access to all the global information necessary and will not be restricted in its ability to use the most advanced techniques available. Rotor-fuselage interference and advanced wake analysis methods are within the scope of this concept. The local velocities will, in the case of the rotor, require a transformation which is a function of time. Since this is performed by an arbitrary FORTRAN algorithm within the component representation, this does not violate the constant transformation requirement. The local blade in-plane velocity, for example, will be of the form

$$\dot{x} = (r - e) \dot{\xi} + (\dot{X}_H + V) \sin \Psi + \dot{Y}_H \cos \Psi$$

where ξ is the lag velocity and X_H , Y_H are the hub fore and aft and lateral velocities, respectively and V is the forward velocity of the helicopter. Such a transformation may not be used in a T matrix, but may be performed in the technical modules prior to the aerodynamics subroutine call statement.

Solution of Equations - In the more general cases, where the coefficient matrices are not all constant, there cannot be a complete set of differential equation coefficients in storage. In addition, the forcing functions will generally have to be evaluated at each time increment. The methods of numerically solving sets of differential equations start with the initial conditions (all variables and derivatives below the highest). These conditions (the state variables) are sufficient to evaluate the highest derivatives at the same point in time by simply evaluating the terms in the differential equation. The mathematical integration algorithm then obtains all the lower derivatives at the next point in time ($t + \Delta t$). The highest derivatives are then recomputed and the cycle continues. Many methods of solving differential equations treat sets of first-order equations. The transformation from second-order to first-order is routine and trivial and this will be performed, when appropriate, within the mathematical technical module. Most of the operations described above are independent of the particular mathematical algorithm or the component representations and are thus performed under the control of the Application Executive. This process is discussed in detail in the following section.

When a differential equation with constant coefficients has been formed, this problem may be converted into a frequency domain formulation and an eigen-solution may be obtained. In its most general form, the eigenproblem is simply

$$(K_c - \omega_i^2 M_c - i \omega_i C_c) \phi_i = 0$$

where ω_i , ϕ_i are the set of eigenvalues and eigenvectors and may be obtained by appropriate mathematical algorithms which will be included as part of the System. Eigensolutions for nonlinear and periodic systems are treated as postprocessing applications where the differential equations are solved first (either to trim or for a set of perturbed initial conditions), and the resulting data are then processed by use of an appropriate algorithm which will be included as part of the System.

System Operations for Solution of Equations - The technique of modeling the individual physical components and coupling and solving an arbitrary configuration of those components requires the System to perform certain initialization and control functions. Figure 6 illustrates the major logic flow from problem initialization to the end of the problem solution. The System first initiates a "set up phase" which performs problem initialization operations (such as the setting up and calculation of the coefficient matrices) prior to the actual problem solution. The System then determines the type of problem that is to be solved (differential equation or eigenproblem).

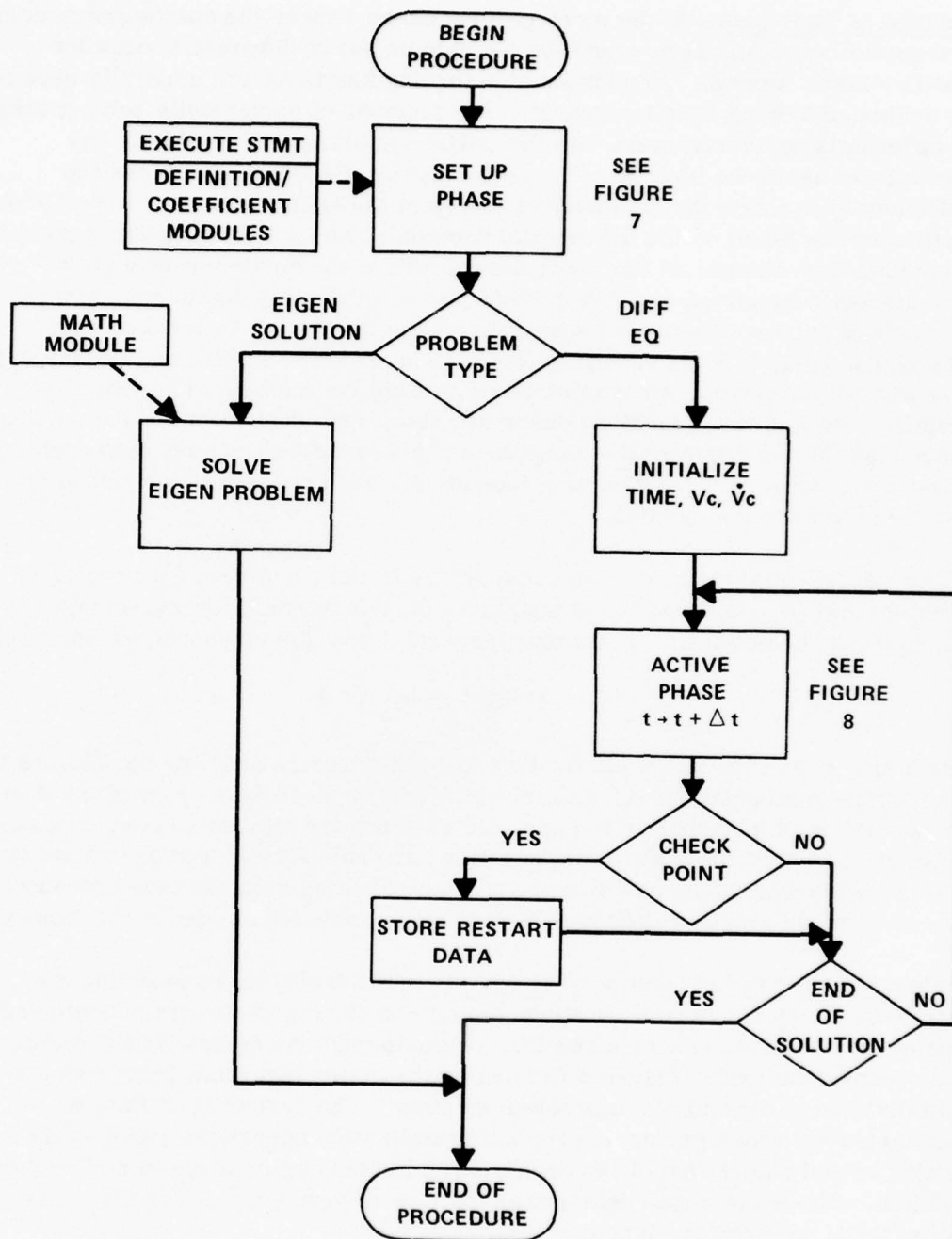


Figure 6. Logic Flow of Execute Statement.

The differential equation solution is no different in principle than present stand-alone programs but is organized so as to achieve maximum efficiency. The first step is to initialize the time and state variables, either to a default option or through specified input. The "active phase", which is described in more detail below, progresses the solution through one time step.

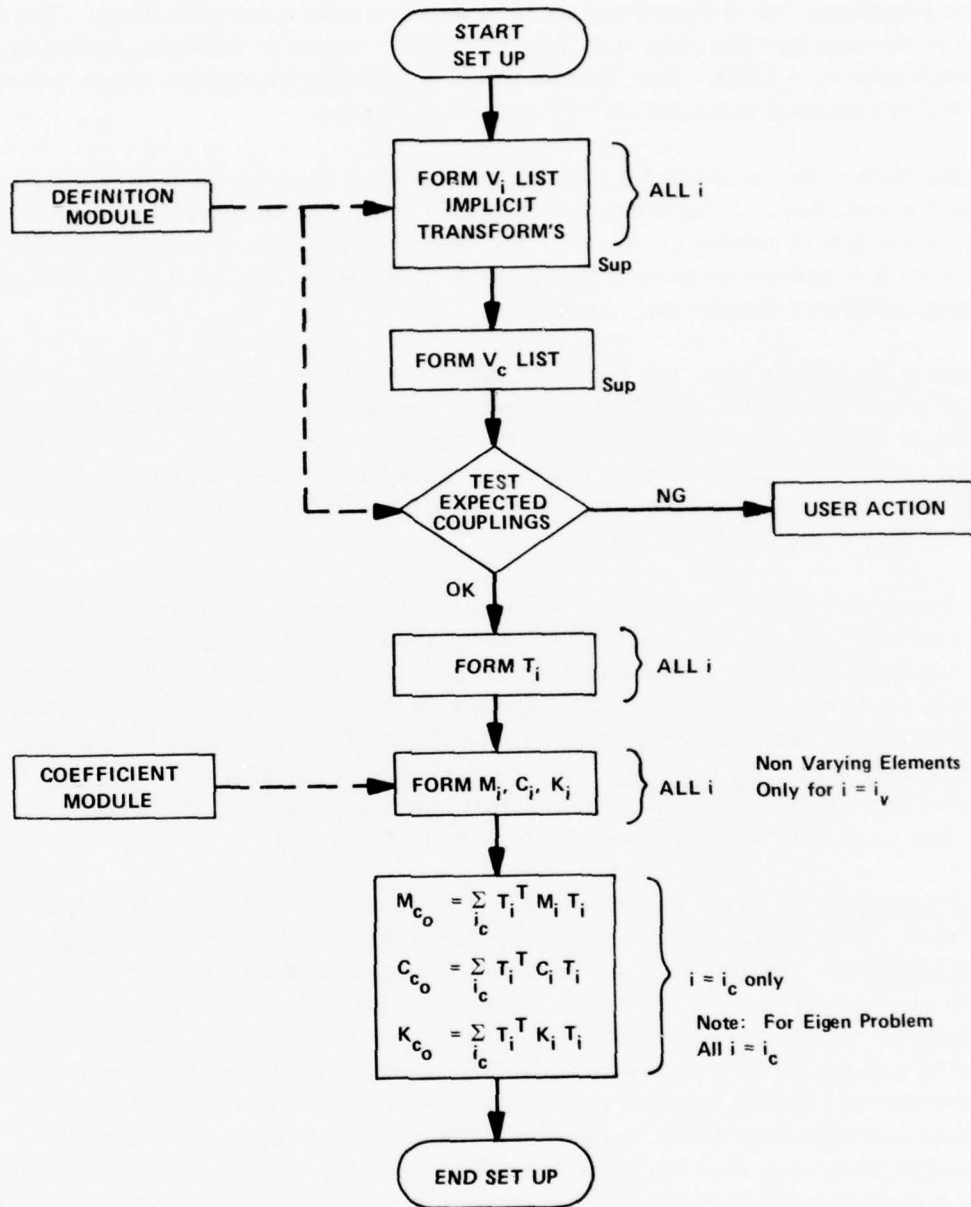
At this time a test is made for checkpoint based on specific input or a standard default parameter. If the condition is satisfied, all the data necessary to perform a restart operation is stored. The end-of-solution test will be a function of the type of problem and may test a number of rotor revolutions, elapsed time, or may test for a steady-state condition.

External to the problem, the user will often specify a criterion judgment test which may check for a specified trim condition and compute new controls and return to the "active phase" to repeat the problem solution. Other noncriteria modules will be available which perform such functions as perturb the initial conditions to produce a Floquet matrix, obtain derivatives for an External Model, and introduce damage parameters.

For an eigenproblem, the System will use a mathematical module specified by the user and process the matrices to obtain the eigenvalues and eigenvectors. The eigenproblem is a frequency domain application. Other linear frequency domain applications may be performed in a postprocessing situation once the coupled M, C, and K matrices have been obtained (see the section entitled: "Implementation Considerations"). The nonlinear frequency domain problem is treated as a postprocessing of time-history data by mathematical modules which perform such functions as harmonic or Fourier analyses.

Set-Up Phase - Figure 7 is a more definitive logic chart of this phase referred to above. The Executive first accesses the Definition Modules (see "Technical Modules" in the section entitled "Technical Components and Relationships") which contain pertinent information regarding each component and forms a list of variable names. Using this information for each component and the descriptions of any implicit relationships a list of names of the coupled system degrees of freedom is formed. At this point, the Executive will examine additional information contained in the Definition Modules to determine if all expected couplings have been formed. For example, the rotor module will indicate that all 6 hub degrees of freedom should be coupled to an airframe and the pitch horn or blade pitch should be coupled to a control system.

When these tests have been satisfied, or a user override instruction is received, all the transformation tables are set up. This will take only a small amount of storage, since the 0's and 1's of the T matrices will not be stored.



Note: i refers to helicopter component
 i_c refers to subset with constant coefficients
 i_v refers to subset with varying coefficients

Figure 7. Logic Flow of Set-Up Phase.

The Coefficient Modules for each component are executed in sequence to compute all the M, C, K matrices which are specified as constant (Definition Module information) and all the constant elements of the variable matrices (these will be the only ones defined in the Coefficient Module).

M_{C_0} , C_{C_0} , K_{C_0} (the constant terms of the coupled system coefficient matrices) are then formed by summing the contributions of each of the constant matrices.

Active Phase - Figure 8 illustrates the logic of the Active Phase which actually solves the differential equations. The specified mathematical module performs its major function which is to convert

$$\ddot{V}_c(t), \dot{V}_c(t), V_c(t) \text{ to } \dot{V}_c(t+\Delta t), V_c(t+\Delta t)$$

The local component variables, V_i , are then obtained through the transformations represented by T_i .

The Active Modules are now executed in sequence to perform any matrix element changes. Intermediate variables, VI, are computed (such as local velocities and displacements for aerodynamic computations) and the F functions are obtained for each component. These will include aerodynamics (using the user-specified subroutine) and any other forces and nonlinear effects that are programmed into the Active Module.

The coupled coefficient matrices and forcing function are then obtained as indicated and \ddot{V}_c at $t + \Delta t$ is obtained.

SYSTEM DESIGN CONCEPT AND ARCHITECTURE

Overview

Design Concept - The overriding technological issue affecting the system design is the requirement for arbitrary coupling of individual rotorcraft components into a complete helicopter configuration. All other considerations such as computational efficiency, user convenience, accuracy, and maintainability have been designed into the system within the framework established to meet this basic technological requirement. The System has been designed as two major components (see Figure 9): the "technology modules," which model individual helicopter components; and the "Application Executive," which provides the dynamic coupling capabilities in addition to managing data and all aspects of problem execution.

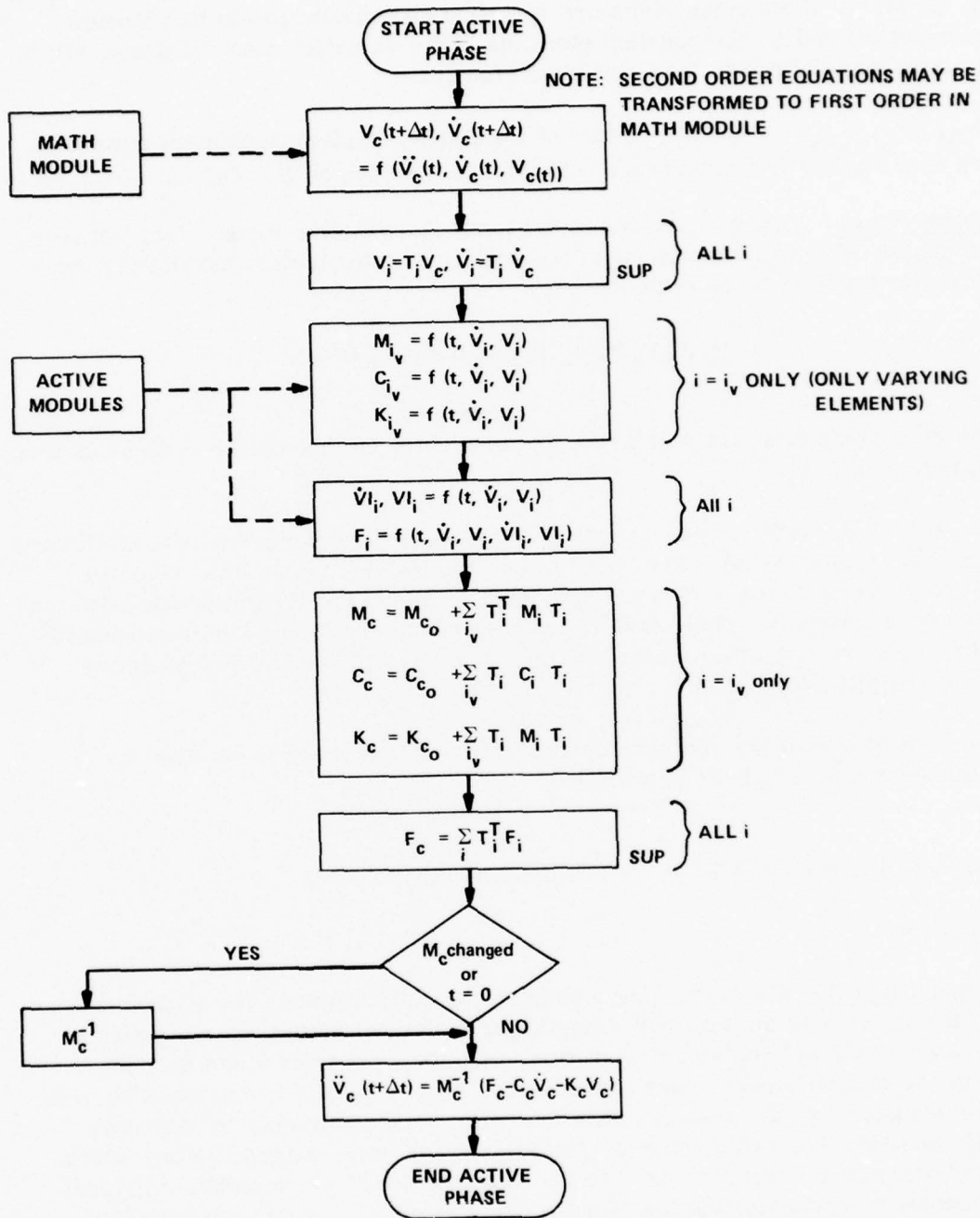


Figure 8. Logic Flow of Active Phase.

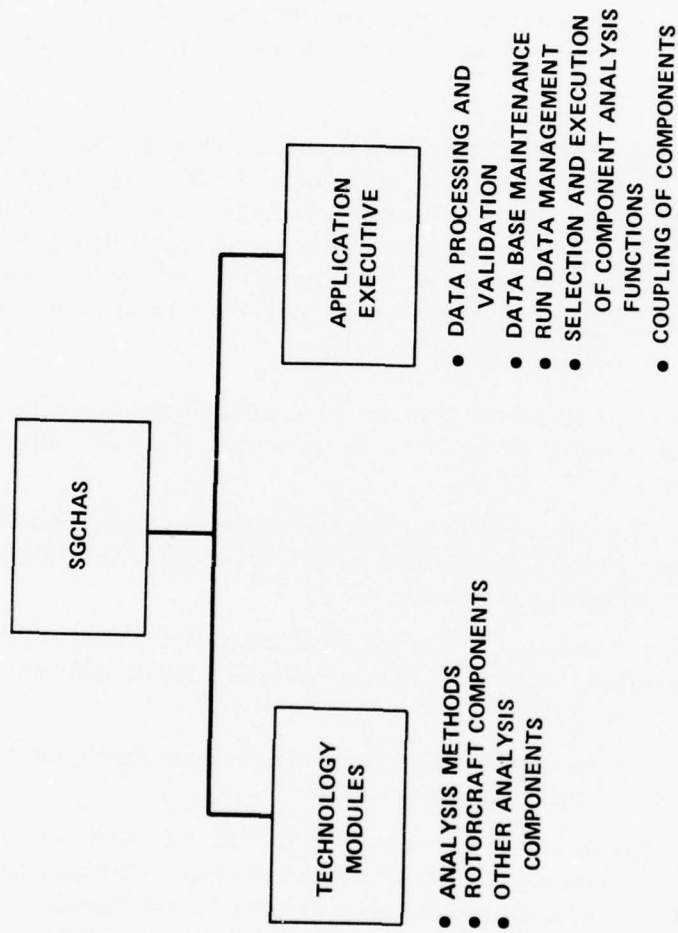


Figure 9. System Concept and Architecture.

Application Executive - The Application Executive provides a system that may be used to control and support modeling of any rotorcraft or component analysis configuration. The System is sufficiently generalized to provide for continuous development and integration of additional technology capabilities without modification of the Application Executive. The software concept of the Application Executive is similar to that of an "interpreter" in that it accepts control inputs and data from a user, validates those inputs and translates them into data that is more meaningful to the System, determines the operations that are to be performed, and sets up and executes the processes required to perform those operations.

A set of control inputs, termed the System Control Language (SCL), has been defined for the SGCHAS. Although few in number, the SCL inputs provide the System user and technical module developer with data definition and maintenance capabilities, helicopter model definition and execution capabilities, technical module definition and installation capabilities, and the capability to store and invoke procedures corresponding to the Particular Functional Capabilities described in the Type A System Specification.

Technical Modules - The technical Modules are separate program entities that represent individual physical or analysis components. They fall into four distinct categories:

1. Differential Equation - These modules represent individual physical or analysis components from which a coupled set of differential equations is to be formulated and solved;
2. Eigenvalue - These modules represent individual physical components where a coupled, linear, constant coefficient set of differential equations is desired;
3. Sequential - These modules perform stand-alone functions that are not dynamically coupled; and
4. Criteria - These modules are used to modify the sequence of problem execution and include such functions as iterative trim algorithms, formulation of Floquet matrices by varying initial conditions, computation of quasi-linear stability derivatives by perturbations of a trimmed system, and interruption of a solution to introduce damage effects.

In addition, there are two classes of technical module subroutines that are specified by the system user: air mass and engine performance subroutines. Each of the air mass subroutines can be used with each of the engine/drive system technical modules.

System Operation

As discussed previously, the System has been designed as two major components: the Technology Modules and the Application Executive. These components operate in a unified fashion to provide the capabilities required by the Baseline Type A System Specification. The inputs, processes, and outputs of the System are illustrated in Figure 10.

Inputs - There are four categories of inputs required by the SGCHAS:

1. Initialization Parameters, which describe the initial conditions for system operation, such as units of measure, and diagnostic level
2. System Control Language statements, which direct system operation
3. Data Base Files on which the user's data and standard system data will be stored
4. Sequential Files, which will be used to input correlation data and restart data
5. Executable Module Files, which will contain the Application Executive and Technology Modules for dynamic loading and execution.

Processing - The engineer initiates the System using the Host Operating System's Job Control Language. The SGCHAS Executive Supervisor will be initiated by the operating system and will perform the initialization functions required by the SGCHAS. During initialization, the user's initialization parameters will be input to be made available to all subsequent processes.

Following initialization, the Executive Supervisor will determine if the user has requested a System Restart. If restart is required, the Executive Supervisor will load and execute the Restart Facility CPCI using the Dynamic Loader. When restart processing has been completed, the Restart Facility will be deleted from memory.

The Executive Supervisor then determines the mode of operation and initiates the appropriate subsystem (Batch or Interactive).

The Batch Subsystem will input the user's SCL statements and validate and process them by functional group (Data Base Definition, Data Base Maintenance, and Processing Sequence Control). Using the Data Base Maintenance statements, the user will introduce new data or modify existing data (or both) in the data base files; thus describing the physical characteristics of the analysis to be performed. Using the Sequence Control statements, the user will invoke stored procedures to perform standard analyses or describe specialized analyses for execution. In the Batch mode, all SCL statements will be validated extensively and any processing is performed.

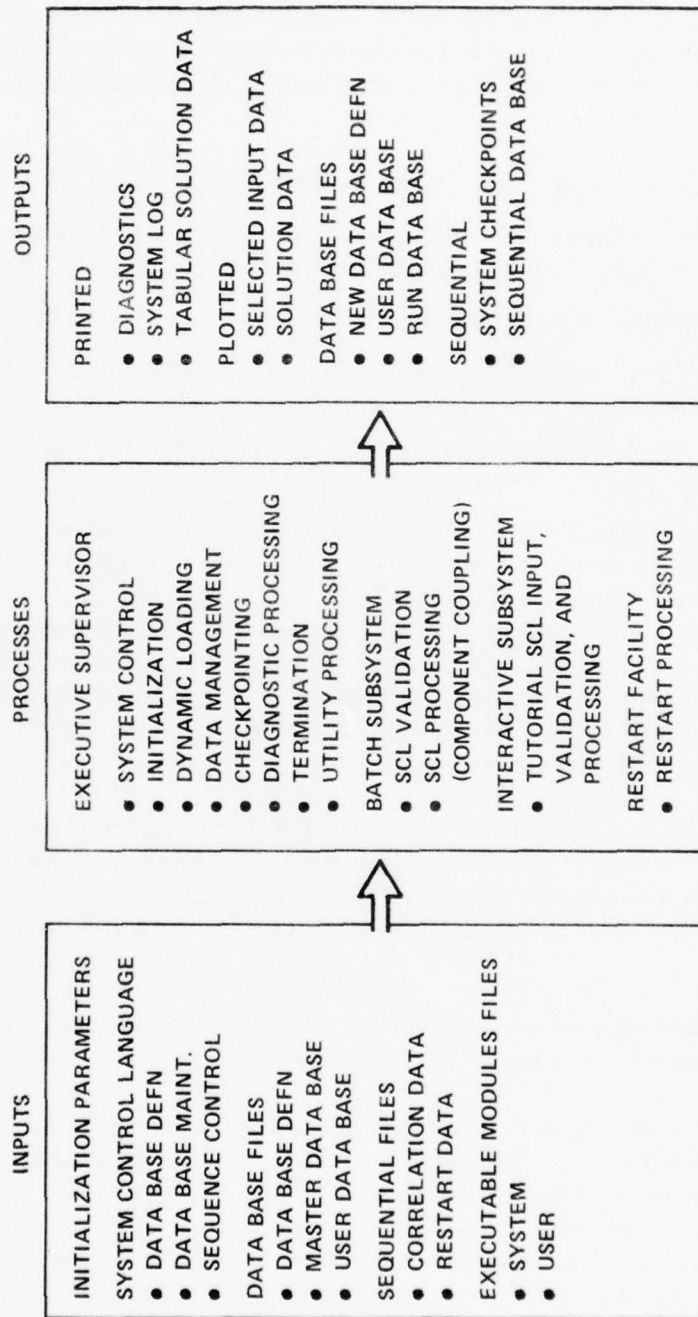


Figure 10. System Processing.

The Interactive Subsystem will interact with a user in a conversational manner: inputting, validating, correcting, and executing the user's SCL statements in an immediate, rather than queued, fashion.

Outputs - Regardless of the operating mode, the System will generate four categories of outputs.

- a. Printed outputs will be generated which will be either output directly to an interactive user or placed on a sequential file for subsequent batch or remote batch printing. Included in this category will be diagnostic, system log, and tabular solution data.
- b. Plotted outputs will be generated in both an interactive and deferred mode. Included in this category will be graphable input and solution data.
- c. Updated data base files will also be produced by the System. These will include a new Data Base Definition file, modified user data base files, and a temporary Run Data Base file.
- d. Two types of sequential outputs will be provided by the System. The first is a System Checkpoint file that will be used as an input to the Restart Facility. The second is a Sequential Data Base file that is used to transfer data base entry records between computer systems.

Engineering Data Management

The management of system data before, during, and after system operation is a major concern for any generalized engineering analysis or simulation program. The problems that must be addressed include: processing efficiency, efficient resource utilization, master data security, and multiple simultaneous usage of master data. These issues have been addressed by the Control Data/Kaman system design by providing three levels of data management.

User Data Management - In any system it is necessary to provide a vehicle for the user to supply data for processing. In the SGCHAS the user supplies data through the System Control Language. This language uses English-like statements to direct data base management processes and explicit assignment of data values (e.g., I=5) to provide the user with a meaningful human-machine interface. However, to provide execution efficiency, all data will be converted to an internal unit of measure (metric) and to the internal hardware representation (binary) prior to storage in the data base.

System Data Base Management - The techniques that are used for data base file management will determine the security and usability of both user and master data. If an integrated generalized data base management system is selected, often the processing efficiency and multiple user capabilities are impacted. On the other hand, when the standard, fixed format, record approach is taken, the maintainability of the system suffers. The solution is to develop a specialized data base manager that provides data independence, master data security, and availability to multiple users.

The SGCHAS Secondary Storage Manager has been designed to specifically address these needs. It provides for management of multiple user data base files, read-only access to the Master Data Base file, and management of multiple variable format entry records through use of a Data Base Definition file. Since the Master Data Base file is a read-only file it provides for simultaneous access by multiple users. The provision for multiple user data base files permits the user to store data for use in subsequent runs, permits an installation to define a "standard" set of data, and encourages individual and organizational exchange of data.

Two categories of data will be stored in the SGCHAS data base: Executive data and Rotorcraft Component data.

- a. There are three record formats defined for use by the Application Executive (Figure 11):
 - (1) Stored Procedure Definitions (SPD), which will be used to store a set of sequence control statements for later recall and execution.
 - (2) Helicopter Model Definitions (HMD), which will be used to store rotorcraft analysis configurations for later execution.
 - (3) Technical Module Definitions (TMD), which describe the data requirements of technology modules and their coupling requirements.
- b. There will be many different Rotorcraft Component data record formats that will be defined for the storage of analysis data. Specifically, they will correspond to the data requirements of the various technology modules developed for the System.

The logical relationships of these records are illustrated in Figure 12. Although the SPD is not explicitly identified, the hierarchy actually begins when the user invokes a stored procedure or executes a helicopter analysis model. If an SPD is invoked at some time during the processing of the procedure, an HMD will be referenced and the resultant helicopter analysis model assembled and executed.

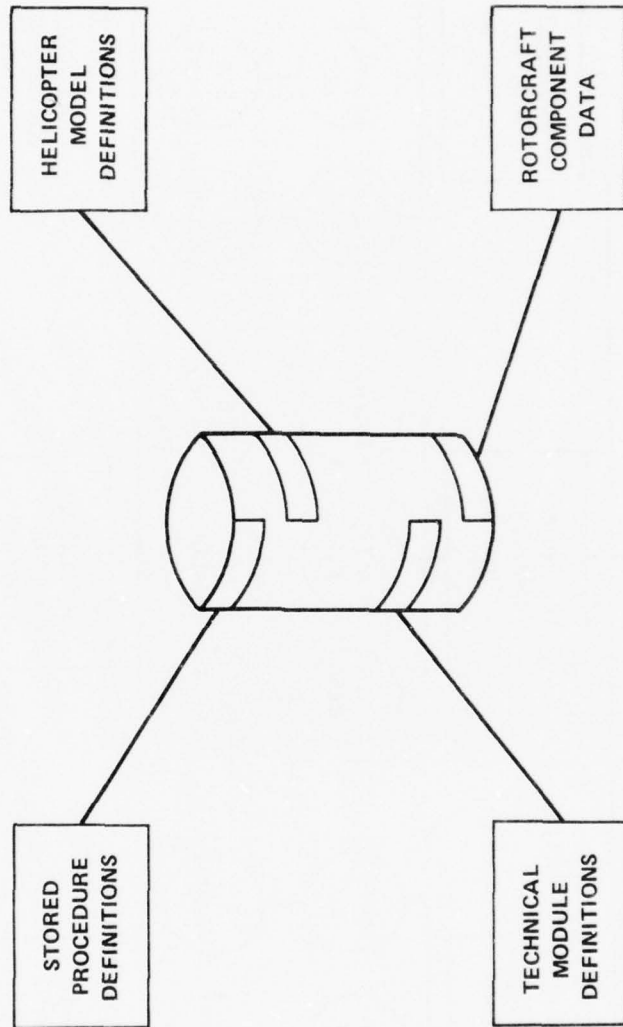


Figure 11. Data Base Record Formats.

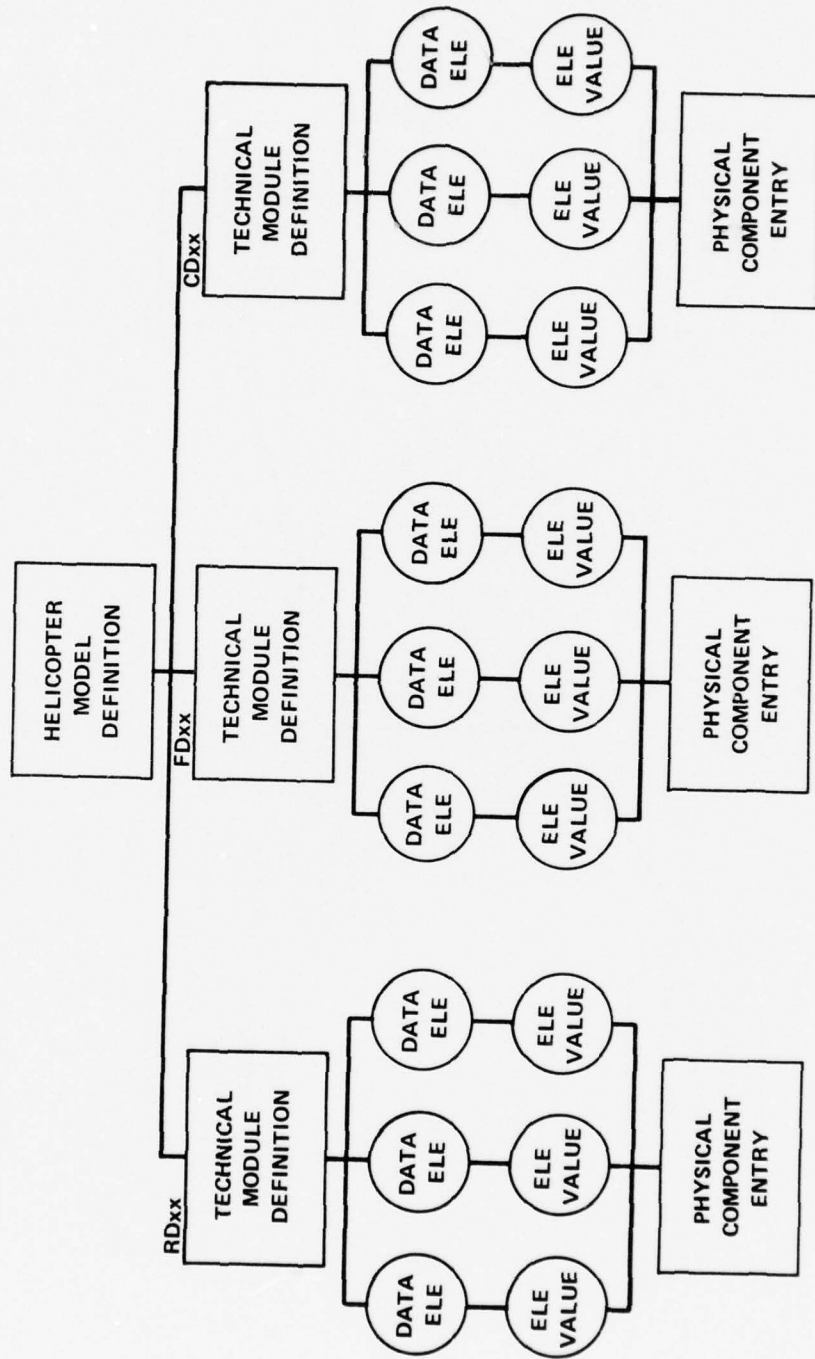


Figure 12. Data Base Record Relationships.

The HMD describes an analysis model by listing an analysis method and several component technical modules. Using these names as keys, the Application Executive retrieves the TMD's describing the selected technology modules and identifies the specific data requirements. The Application Executive will then locate the corresponding physical component and other analysis component data in memory or on the Run Data Base.

The values contained in these data base records are then linked by name to the input data requirements. Tables of valued variables will be built in memory along with FORTRAN argument lists for each technology module which, when executed, can use the data directly and efficiently.

System Memory Management - The only way to insure efficient memory utilization for the variety of problems that can be solved using the SGCHAS is to permit the System to manage its own memory. The SGCHAS Primary Storage Manager will provide comprehensive memory management capabilities to minimize memory requirements during operation.

Application Executive Components and Relationships

The Application Executive has been designed to provide all of the functional, restart, interactive, and graphic capabilities required by the Type A System Specification. These requirements have been allocated to five major functional areas (Figure 13), which have been designated as Computer Program Configuration Items (CPCIs) as follows:

Executive Supervisor - The Executive Supervisor CPCI (Figure 14) provides the centralization of major control functions, system utility functions, and machine-dependent operations that is necessary for insuring the maintainability and manageability of the System.

- a. The Executive Supervisor executes the Initiation Component module to perform the following functions for system initiation:
 - (1) determination of the System's operation mode (batch or interactive) for processing,
 - (2) determination of the type of processing (initial or restart) that is to be performed,
 - (3) identification of data conversion requirements (English/metric), and
 - (4) initialization of all System tables and control structures;

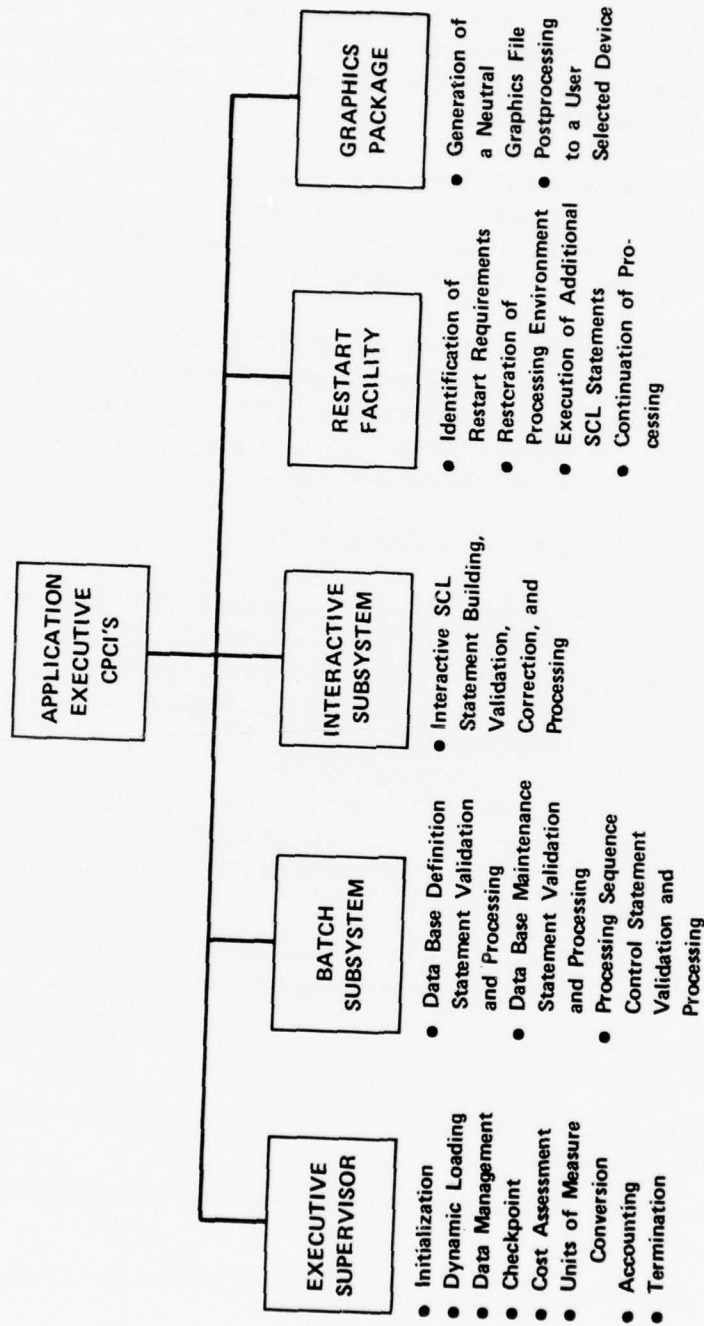


Figure 13. Application Executive CPCIS.

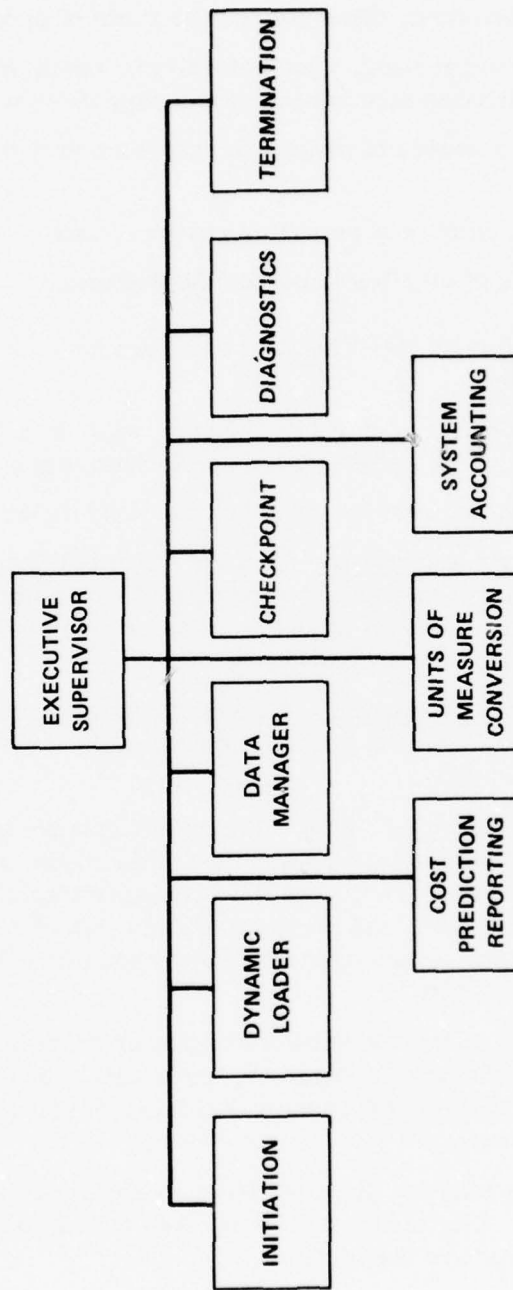


Figure 14. Executive Supervisor Components.

- b. The Executive Supervisor then loads and executes the Restart Facility CPCI if restart processing is called by the user;
- c. The Executive Supervisor then loads and initiates either Batch or Interactive Subsystem, depending on the mode of processing; and
- d. The Executive Supervisor, upon return from batch or interactive processing, initiates termination processing which will include:
 - (1) the postprocessing of diagnostic information to a user-specified medium,
 - (2) the execution of cost reporting routines, and
 - (3) the closing of all files in use by the System.

The following utility and machine-dependent functions have also been assigned to the Executive Supervisor.

- a. The Dynamic Loader permits the loading, execution (with parameter passing), and eviction of executive and technical modules.
- b. The Data Manager provides the following capabilities:
 - (1) Manage Primary Storage - Primary storage is considered to be main memory; the Data Manager provides definition of free memory, stages data to and from primary storage and manages intermediate memory usage.
 - (2) Input/Output Operations - All input/output operations are controlled by the Data Manager; access methods include sequential, random, indexed and direct.
 - (3) Manage Secondary Storage - Secondary storage is considered to be storage medium of disk, tape, and other media external to the central processing unit; the Data Manager identifies data sets, creates data sets, and maintains catalogues of data sets and their characteristics; data base file maintenance is provided through the Data Manager.
- c. The Units of Measure Conversion function provides modules to convert from English to metric on input and from metric to English on output. All helicopter analysis processing will be performed using the metric system of measure.
- d. The Checkpoint function provides the data required by the Restart Facility CPCI. Checkpoint will be invoked during processing by the Batch and Interactive Subsystems.

- e. The System Accounting function is required to collect data for use in cost prediction and for timing studies. The Batch and Interactive Subsystems will invoke the System Accounting modules at each major processing step.
- f. The System will provide Cost Prediction and Reporting modules which will estimate the a priori cost of a proposed computer run and will report the cost of a run at completion.
- g. The Diagnostic function of the System will provide centralized diagnostic processing for all CPCIs and through this centralization, user control of the diagnostic output medium. Statistics will be kept on error frequency by the particular diagnostic that was issued, thus permitting identification of deficiencies in user training, system documentation, and error message content.

Batch Subsystem - The Batch Subsystem (Figure 15) controls System Control Language processing in a noninteractive environment. There are three types of SCL statements which must be validated and processed:

- a. Data Base Definition statements which define the format and content of data base entry records;
- b. Data Base Maintenance statements which create, modify, and delete entry records; and,
- c. Sequence Control statements which specify the sequence in which helicopter analyses are to be performed.

The Data Base Definition statements are intended for use by system development and maintenance personnel. They provide an easy method to define the format and content of data base entry records and, thus, permit the addition and modification of data base entry data. In addition, the definition statements provide a vehicle for assigning the validation criteria for the data elements within an entry and for identifying the units-of-measure for each data element for conversion purposes.

The Data Base Maintenance statements are used to create, modify, and delete entry records from the data base files. These statements will permit the assignment of data element values on an element-within-entry basis. There will be several types of entries defined in the data base (Helicopter Model Definition, Rotorcraft Physical Characteristics, Technical Module Definition, etc.). Therefore, entry records within the data base files will be identified by a unique entry name and type.

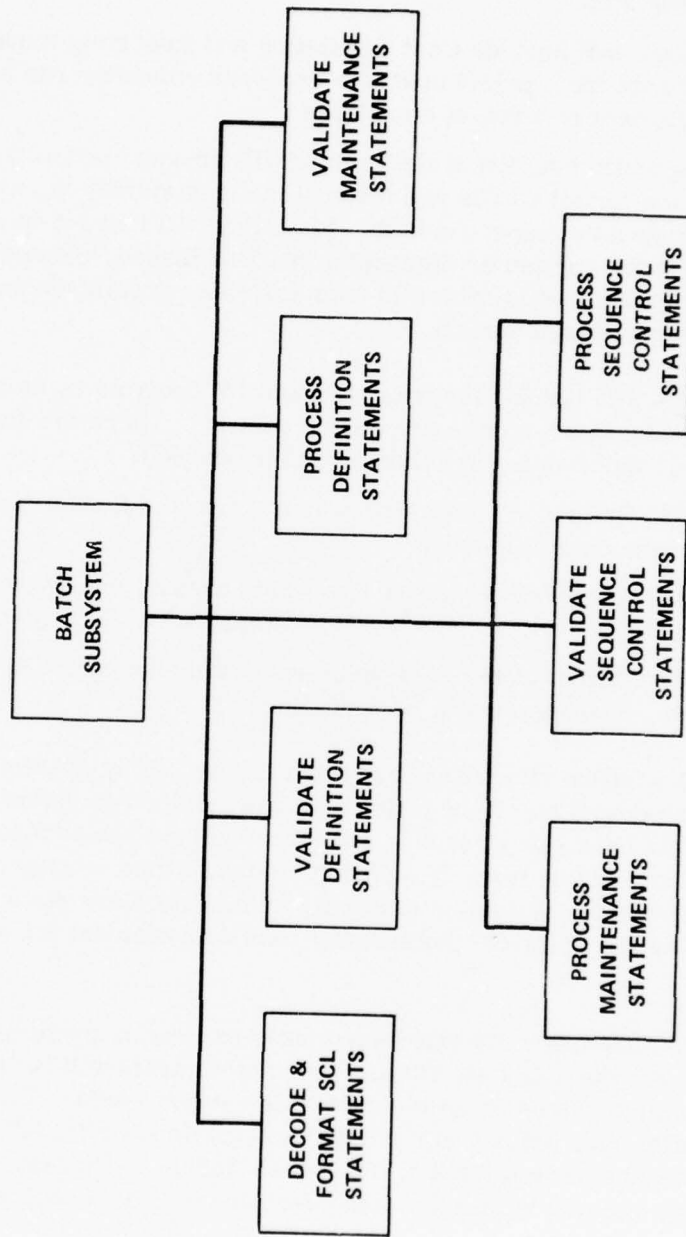


Figure 15. Batch Subsystem Components.

The Sequence Control statements will provide the ability to invoke Stored Procedures from the data base (CALL statement), execute Helicopter Models that are defined in the data base (EXECUTE statement), branch to a particular statement (GOTO statement), perform conditional operations (IF statement), temporarily change values stored in a data base entry record (CHANGE statement), establish and modify values for parametric data (SET statement), and interrupt processing (STOP statement).

The SCL statements will be grouped according to type. The Data Base Definition statements will begin with a DEFINE statement which will identify the file on which the Data Base Definition data is to be written. The Data Base Maintenance statements will begin with an ACCESS statement which will define the files that are to be accessed. The Sequence Control statements will begin with a PROCESS statement which will define the data base files that are to be used in processing helicopter analyses.

Validation and processing of the statements will occur by group. All statements in a particular group will be validated and then processed before another group is begun. In this way, all changes to the data base files will occur before the processing of helicopter analyses which might use those files.

Interactive Subsystem - The Interactive Subsystem (Figure 16) provides control of System Control Language input, validation, and processing in an interactive tutorial fashion. The Interactive Subsystem will input the same three types of SCL statements that are input to the Batch Subsystem:

- a. Data Base Definition statements
- b. Data Base Maintenance statements
- c. Sequence Control statements

However, the Interactive Subsystem will converse with the user and provide interactive diagnosis of errors in language syntax, set up, and processing. The Interactive Subsystem also provides a tutorial capability which will permit the user to obtain a brief description of SCL statements, Technical Modules, and Helicopter Models.

The design of the Interactive Subsystem is substantially different from the Executive Subsystem. This is because the subsystem must be capable of prompting the user to provide input, identifying the input and diagnosing errors, and transferring control to subcomponent modules which will further validate the input, request corrections from the user, and process the specified SCL statement.

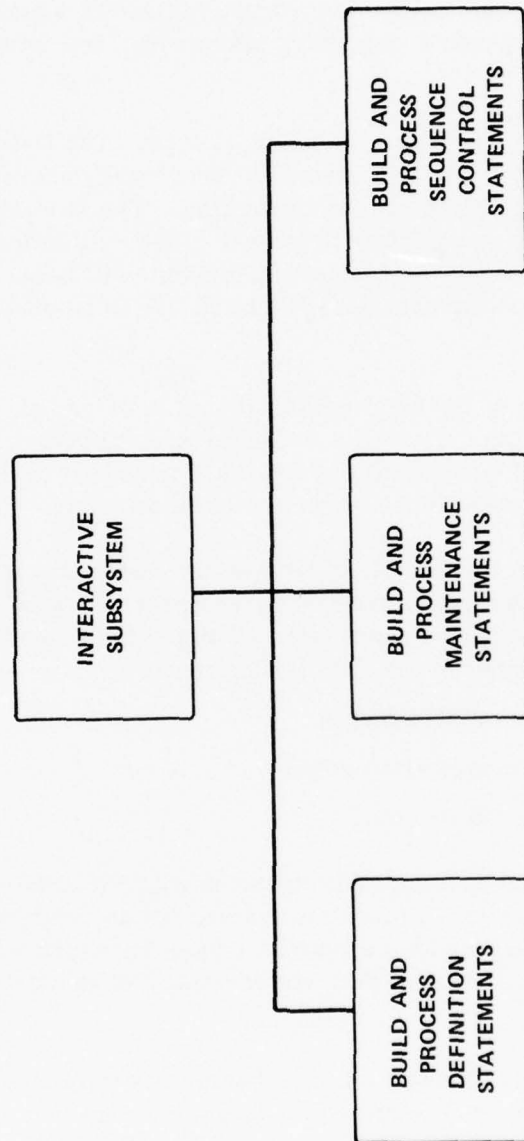


Figure 16. Interactive Subsystem Components.

Restart Facility - The Restart Facility provides for recovery of system interruptions, whether the interruption was user-invoked (via the STOP SCL statement), caused by data errors, or caused by a computer system malfunction. Restarts will be permitted whenever system checkpoint data has been retained.

Graphics Package - The Graphics Package CPCI will be procured from a graphics hardware/software vendor. The package will provide all the functions necessary for the generation of plotted outputs.

TECHNICAL COMPONENTS AND RELATIONSHIPS

All the technical functions of the System reside in a collection of "technical modules". The run time loading of these modules, the sequential and dynamic couplings, the inputs and outputs, and all similar functions are performed by the Application Executive.

There are several general types of technical modules and each technical module is partitioned into functional modules for ease of programming and for economical operation. A complete description of these components and certain operational considerations are described in this section.

Technical Capabilities

The System has been designed with the engineering user in mind. The System will provide the user with the greatest possible flexibility in modeling and analyzing helicopter problems. For standard and production analyses, the user will have the convenience of extremely simple control inputs.

The engineering user can use a library of technical modules (or CPCIs), each of which will be an analytical representation of one or more aircraft components or a method of analysis or a numerical algorithm. Within the scope of the available CPCIs, the user will be able to specify any combination of (compatible) component representations, any method of analysis, and any numerical processing of the resulting data.

The System will be delivered with a set of validated PFCs: however, the System is oriented toward the GFC capability and the PFCs are simply special cases consisting of a prescribed set of control inputs which can be addressed. Additional PFCs can be constructed at each user's installation to provide particular needs, by storing the appropriate set of control statements and giving this set a unique name for future reference. When a problem is prescribed (including solution method and the required data) with System Control Language statements,

the user can designate and name all or part of this problem as a PFC which can be accessed at any time in the future. When it is desired to reexecute this PFC, the name is referenced along with any desired changes in the model, physical data or condition, analysis method, or numerical processing.

The EMFC's are performed by executing a GFC or a PFC (possibly a number of times with perturbations) and performing an analysis of the resulting data using the appropriate mathematical CPCI. Any EMFC can itself be a PFC.

The capability of the System to economically, conveniently, and accurately perform as described above is dependent on the technical CPCI concepts described below. These capabilities have been verified by sample "walk-throughs" which helped define many of the conceptual details of the System.

Couplings of Analyses

There are basically two types of couplings for which provisions must be made. The first can be considered to be a sequence of analyses; where each analysis is independent of the others except that it can use data provided by a prior analysis and it can supply data for a subsequent analysis. The second type contains analyses which are coupled in such a way that a simultaneous solution must be performed. This type is crucial to the success of the System and has been previously discussed in detail.

Run Time Coupling Considerations

As stated above, virtually all of the couplings are automatic, with no input required by the user because of the unique variable names that are used throughout.

In a sequence of analyses where the output of one analysis is to be processed by a general mathematical algorithm, it will be necessary for the user to specify which data is to be analyzed. A harmonic analysis, for example, can be performed on many sets of data and the user will be required to specify which are to be processed. On the other hand, a rotor dynamic analysis which uses normal modes and follows a model analysis requires no special information from the user since the coupling is unique and unambiguous.

An ambiguity may occur when more than one component of the same type is used in a coupled analysis, as for example, when two rotors are used. Even if two different representations are used for the two rotors, the same variable names will appear in both. This ambiguity will be resolved by having the user add a

character to each of the rotor CPCI names when the CPCI's are specified to be used. For example, if the rotor CPCI's names are RD00 and RD22, the user can specify these as:

RD00 (1)
RD22 (2)

or to use the same CPCI for both, the user could specify them as:

RD22 (1)
RD22 (2)

The executive would then add these arguments as suffixes to all the variable names in the two modules, thus making them unique. For example, if BETA (NB) is the flapping angle of blade NB, the new variables would be:

BETA1 (NB)
BETA2 (NB)

The input variables would also be distinguished so that separate inputs could be made to the two rotor representations.

Any fuselage which interfaced with two or more rotors would have to have this provision built-in along with inputs specifying location and orientation. A method of implementing this requirement is to have the user specify the respective rotor numbers in an argument list. A set of component CPCI's could be designated by the user at run time as:

FD33(1,2)
RD22(1)
RD22(2)

Similar considerations apply to control system CPCI's which may control one or two or more rotors and/or aerodynamic control of lifting surfaces on the fuselage.

Technical Modules

There are four distinct categories of technical modules. Each of these technical modules consists of at least two functional modules. (These are distinguished from a programmable module which must conform to the requirements of the Type A System Specification. Each functional module can contain a number of programmable modules).

The four categories of technical modules and the functional modules associated with each are listed below:

TECHNICAL MODULE TYPES

		Diff. Eq.	Eigenvalue	Sequential	Criteria
Functional Modules	Active	X			
	Processing			X	X
	Coefficient	X	X		
	Definition	X	X	X	X

Purpose of the Technical Module Types - The four technical modules are briefly described as follows:

- a. Differential Equation - These modules represent individual physical or analysis components where a coupled set of differential equations is to be formulated and solved.
- b. Eigenvalue - These modules represent individual physical components where a coupled, linear, constant coefficient set of differential equations are desired. An eigenvalue analysis is to be performed on the coefficient matrices.
- c. Sequential - These modules perform stand-alone functions as described previously. The algorithms which perform numerical solutions of differential equations and use "active modules" of components below are included in this classification.
- d. Criteria - These modules are used in controlling overall problem logic and are included in a generalized "IF" type statement. They will include such functions as iterative trim algorithms, formulation of Floquet matrices by varying initial conditions, and computation of quasi-linear stability derivatives by perturbations of a trimmed system.

Definition Module - The definition module must be a part of all technical modules. It is not an executable program but supplies necessary information to the Executive. It is, in effect, a kind of documentation of the CPCI, and appears in the System data base. The information contained in this module is listed below with definitions and examples.

- a. Name of CPCI - This will be the name of the complete technical module as would be specified by the user in defining the problem within the control language. A convenient format which identifies the general scope to the user is given below.

1st. character defines the component or a type of analysis:

R - rotor
C - control system
E - engine/drive system
F - airframe
A - airmass
H - more than one component
M - mathematic algorithm
P - mathematical postprocessing
J - criteria judging

2nd. character defines the type of problem

D - differential equations
E - eigenproblem
S - sequential
G - general - independent of type of problem

3rd. character (0-9) defines the level of complexity

0-2 - preliminary design
3-6 - detailed design
7-9 - research

4th. character (0-9) defines the level of technology

0-2 - prior
3-6 - present
7-9 - advanced

5th.-8th. character - xxx, arbitrary, optional designation

For example: RD88-FT

- b. Narrative - This data will be a concise description of the function and methods of the CPCI. When the System is requested to produce a description of the problem being solved, it will list the narrative data of all the CPCIs which have been specified by the user.
- c. Input Data List - This is a list of names of variables required as input. This is data which does not normally change with time. The System Control Language validation processing will determine whether each item is new input, available in the data bases, or will have been produced by a previously executed analysis. The format will be similar to a FORTRAN specification statement with variable dimensions. The variable names will conform to the standardized nomenclature.

- d. Output Data List - This is a list of output data computed by the "active" or "processing" modules. This data may be used for direct output, may be processed by a routine which modifies it for plotting, or may be used as an input for another CPCI. User selectable options will be available for those items which are required for printed or plotted output.
- e. Degree of Freedom List - This is a list of the variables which are degrees of freedom of the modeled component. For example, a rigid rotor CPCI may specify BETA(NB) where there is a flapping angle degree of freedom for each blade. This is used only in differential equation or eigenvalue problems.
- f. Implicit Coupling Relationships - These relationships are in tabular form and are used by the Executive Supervisor in forming the transformation matrices and the coupled degrees of freedom vector. This information is required when degrees of freedom are not explicit degrees of freedom of the component. This is used only in differential or eigenproblems.
- g. Expected Coupled Variables - This is a subset of the degrees of freedom or implicit variables (see f., above) which would normally be coupled to another component, such as the 6 hub degrees of freedom of a rotor analysis. It is used for checking validity of the overall model definition. This is used only in differential or eigenproblems.
- h. Variability of Coefficient Matrices - An indicator to inform the Executive Supervisor which, if any, of the M, C, K matrices will vary. This is used for improving efficiency of computation.

Coefficient Module - These modules are used in the differential equation and eigensolution problems. After the Executive has used all the data in the definition modules and established tables of variables, allocated core, formed transformation matrices, and the other necessary functions, the coefficient modules are called upon to actually compute the constant matrix coefficients and any other coefficient data required by the active modules.

Active Module - These modules take the same part in the differential solution process as do the user-supplied subroutines commonly required in present differential equation solution algorithms. These active modules perform whatever analyses are required to compute the highest derivative vector in the equations, given all the lower derivatives. They use the constant coefficients already generated, and can include any time varying or periodic functions, table look-ups, and nonlinearities of any kind. Active modules for rotor, fuselage, or engine/drive system will contain call statements to an aerodynamic or engine performance subroutine.

Processing Modules - These modules used in sequential or criteria technical modules are, in effect, ordinary routines which perform specified computations.

Technical Subroutines

Most of the technical functions of the System are performed by the Technical Modules described above. There are certain of these functions, however, which are performed by ordinary FORTRAN subroutines. Both modeling and utility functions are performed by these subroutines. The modeling subroutines also will have definition modules associated with them while the utility subroutines will not.

As previously described, the airmass computations which are performed during the active phase of the differential equation solution are performed by one of a set of subroutines. In addition, in the Engine/Drive System technical modules, the engine performance computations are carried out in a similar manner by a user-selected subroutine. These subroutines are developed and validated in a manner identical to the technical modules and are also considered to be CPCIs.

This capability allows for the flexibility of the user to choose a rotor analysis and a fuselage analysis and to independently select airmass analyses as appropriate. The same flexibility exists in selecting drive system dynamics and engine performance analyses.

The user, when specifying the rotor or airframe technical modules, must also specify the selection of the appropriate subroutines. This can be accomplished as an argument to the technical module name. The example given previously may be expanded to the following form:

```
FD33 (A02, 1, 2)
RD22 (A34, 1)
RD22 (A02, 2)
```

These names identify the rotor and airframe analysis methods, the order of location of the rotors on the fuselage (specific locations are defined by fuselage input parameters), and the specific airmass algorithms to be used.

It should be noted that there are also "stand-alone" airmass technical modules which perform such functions as setting up a rigid wake distribution.

Utility Subroutines

In addition, a set of subroutines is identified as CPCIs which perform a number of utility functions and may be called by technical modules, technical subroutines or the executive CPCIs. They include such functions as matrix operations and data checking.

SYSTEM CAPABILITIES

SATISFACTION OF REQUIREMENTS

The Type A System Specification describes the functional capabilities and operational concepts of the System. The Control Data/Kaman SGCHAS capabilities which directly address these requirements are summarized in Table 1.

AVAILABILITY OF CAPABILITIES

Executive Area

In accordance with the Control Data/Kaman design concept, an Executive System was conceived to be the nucleus for the processing of rotorcraft analysis. Executive functions are identified as the management of software processing, and control and management of data to be manipulated by the System. After identifying functions, classifying requirements that are unique to the executive area, and allocating them for Computer Program Configuration Item (CPCI) identification, the following Executive CPCIs and schedule of capabilities emerged:

Executive Supervisor - The management of software, management of software processing, and control and management of data to be manipulated can best be performed from a central control point. For this purpose a nucleus of programmable functions was designed to be loaded and executed as needed to support the System. Major functions of the Executive Supervisor are:

(FIRST LEVEL)

- Initiate the System (System start-up), determine mode of processing, determine type of processing (initial start-up or restart), initialize common work areas and set indicators to direct processing.
- Dynamically load the programs to be executed, pass parameters and control to the loaded program, and delete the program from main storage when it is no longer needed.
- Manage all data that will be manipulated by the System, manage primary storage, control input/output operations, and manage secondary storage.
- Convert units of measure from English to metric or metric to English units.

TABLE 1. SATISFACTION OF REQUIREMENTS

<u>Type A Requirement</u>	<u>SGCHAS Capability</u>
1. Operational Concepts	
a. Multiple Independent Users	<ul style="list-style-type: none"> ● Data Base and Executable Module files will be read only within the System. Provision has been made for User Data Base files to provide storage of problem related data.
b. Minimized Resource Utilization for rapid job turnaround	<ul style="list-style-type: none"> ● Dynamic management of central memory via the Data Manager and Dynamic Loader.
2. General Functional Capability	
a. Inputs	
1. Program Logic Input	<ul style="list-style-type: none"> ● Sequence Control Statements.
2. Aircraft Physical Data and other Analysis Components	<ul style="list-style-type: none"> ● Component Data Base Entries and Data Base Maintenance Statements.
3. Coupling of Components	<ul style="list-style-type: none"> ● HMD and TMD Data Base Entries.
4. Maneuvers, Conditions, Operating Regime, and Failure/Damage	<ul style="list-style-type: none"> ● Developer defined Data Base Entries.
b. Processing	
1. Aircraft Technical Characteristics, Life Cycle Phases, Aircraft Components, and other Analysis Components	<ul style="list-style-type: none"> ● Technical Modules and Subroutines
2. Coupling of Components	<ul style="list-style-type: none"> ● EXECUTE Sequence Control Statement with HMD and TMD data.
3. Maneuvers, Conditions, Operating Regimes, and Failure/Damage	<ul style="list-style-type: none"> ● Technical Modules (Criteria)
c. Outputs	
1. Input Data	<ul style="list-style-type: none"> ● Application Executive
2. Basic Vehicle Characteristics	<ul style="list-style-type: none"> ● Data Checking Algorithms
3. Flight Conditions	<ul style="list-style-type: none"> ● Data Checking Algorithms (if input), or, Air Mass and Trim Technical Modules and Subroutines
4. Performance Data	<ul style="list-style-type: none"> ● Data Checking Algorithms (if input), or Rotor, Airframe, Air Mass, and Engine Performance Technical Modules and Subroutines

TABLE 1. SATISFACTION OF REQUIREMENTS (Continued)

<u>Type A Requirement</u>	<u>SGCHAS Capability</u>
5. Stability and Control Data	<ul style="list-style-type: none"> ● Trim, Mathematical Postprocessing, Airmass, and Control Technical Modules and Subroutines
6. Loads Data	<ul style="list-style-type: none"> ● Rotor, Airframe, Airmass, and Mathematical Postprocessing Modules
7. Acoustics Data	<ul style="list-style-type: none"> ● Acoustics Postprocessing Modules
8. Aeroelastic Stability Data	<ul style="list-style-type: none"> ● Airframe and Mathematical Post-processing Technical Modules
3. Particular Functional Capabilities	
a. Inputs	
1. Program Logic Inputs	<ul style="list-style-type: none"> ● Stored Procedures invoked by CALL SCL statement
2. Other Inputs same as GFC	<ul style="list-style-type: none"> ● Same as GFC
b. Processing - Same as GFC	<ul style="list-style-type: none"> ● Same as GFC but invoked via CALL
c. Outputs - Same as GFC, but limited by Life Cycle and Technical Characteristics	<ul style="list-style-type: none"> ● Same as GFC
4. Detailed Functional Capabilities	<ul style="list-style-type: none"> ● Same as GFC and PFC
5. External Model Functional Capabilities	<ul style="list-style-type: none"> ● Same as PFC
6. Diagnostic Capability	<ul style="list-style-type: none"> ● Diagnostic Component of the Executive Supervisor CPCI
a. Inputs	
1. Fatal Error Level	<ul style="list-style-type: none"> ● Input to Initialization Component of the Executive Supervisor CPCI
2. Inputs to GFC, PFC, EMFC, DFC	<ul style="list-style-type: none"> ● Validation components of the Batch and Interactive Subsystems
b. Processing	<ul style="list-style-type: none"> ● Internal consistency checks via processing/setup validations
c. Output	<ul style="list-style-type: none"> ● Via PRINT and PLOT SCL statements
7. Accuracy Assessment	<ul style="list-style-type: none"> ● Specialized Sequential Technical Module
8. Cost Assessment	<ul style="list-style-type: none"> ● Cost Prediction Module of the Executive Supervisor CPCI

- Generate diagnostics at three levels of severity: (1) informative, (2) warning and (3) fatal; and provide diagnostics to aid in debugging the System.
- Terminate the System in an orderly fashion insuring that all data sets are properly closed, produce audit trails, and load the termination routines necessary to dispose of diagnostic messages and cost reports.

(SECOND LEVEL)

- Provide checkpoint data necessary for continuity whenever restarts are necessary.
- Perform System Accounting, gathering data pertinent to estimating cost of proposed runs and reporting cost at the completion of the runs.
- Predict cost of a proposed computer run using the user's installation algorithm and at the completion of a computer run report the cost of the run using the user's algorithm.

Restart - The Restart capability is a Type A System Specification requirement. The development of this capability was determined to be a major effort and because it was programmable as an entity, it was given a designated CPCI status. Major functions of Restart are:

(SECOND LEVEL)

- Dispose data for analysis prior to restart.
- Determine if an effective restart can be performed.
- Restore the System to the condition that existed when a checkpoint was taken during processing.
- Perform data maintenance to a data set prior to restarting.
- Re-execute all processing steps affected by input modifications subsequent to restart.
- Restart the System in close proximity to the interrupt point.

Batch Subsystem - The functions of the SGCHAS Batch Subsystem are:

(FIRST LEVEL)

- Input, validate, and process Data Base Definition statements which will define: the types of entries in the SGCHAS Data Base files, the particular data elements assigned to those entries, and the characteristics and validation criteria for each data element.

- Input, validate, and process Data Base Maintenance statements which store, retrieve, and maintain data element values within uniquely named data base entries in accordance with the entry types defined by the Data Base Definition statements.
- Input, validate, and process Sequence Control statements which will specify the functional capabilities of the system to be executed and the sequence of their execution.

(SECOND LEVEL)

- Print Data Base Definition information and generate a sequential, machine-independent file from the System Data Base.

Interactive Subsystem - The functions of the SGCHAS Interactive Subsystem are:

(SECOND LEVEL)

- Provide an interactive tutorial capability to aid the System user in the selection and employment of System Control Language statements.
- Provide interactive input, validation, correction, and processing of Data Base Definition statements which will define: the types of entries in the SGCHAS Data Base files, the particular data elements assigned to those entries, and the characteristics and validation criteria for each data element.
- Provide interactive input, validation, correction and processing of Data Base Maintenance statements which store, retrieve, and maintain data element values within uniquely named data base entries in accordance with the entry types defined by the Data Base Definition statements.
- Provide interactive input, validation, correction and processing of Sequence Control statements which specify the functional capabilities of the system to be executed and the sequence of their execution.

Graphics Package - The Graphics Package provides:

- a. definition of the beginning of a graphical output and its size
- b. plots of points, lines, and arcs
- c. positioning of cursor/pen and determination of the position of the cursor/pen
- d. termination of a graphical output

The graphic data shall be output to a file in a "neutral" form that can be post-processed to interactive or offline graphics devices or to a printer.

Data Dictionary and Cross Reference - The functions to be performed by the SGCHAS Data Dictionary and Cross-Reference CPCI are as follows:

(SECOND LEVEL)

- Input component entry definitions from the SGCHAS Data Base file and corresponding Technical Module Definitions from the Master Data Base file.
- Output, by component and data element within component, a cross-reference listing of the technical modules using the data elements defined for the component entry.

Helicopter Module Documenter - The functions to be performed by the SGCHAS Helicopter Model Documenter are as follows:

(SECOND LEVEL)

- Input Stored Procedure entries from the Master and User Data Base files.
- Output a graphical representation of the logic flow of each stored procedure with a textual list of the Helicopter Models, stand-alone Technical Modules, and Criteria Judgement Modules referenced by the procedure.
- Input Helicopter Model Definition entries from the Master and User Data Base files.
- Output a hierarchical representation of each Helicopter Model with a textual description of the component and analysis Technical Modules specified for the Helicopter Model.

Technical Area

The following list is a summary of the Technical CPCIs. These CPCIs are arranged in 17 groups. Each group corresponds to a single Type B5 Development Specification and each of the CPCIs within a group is a deviation from the most complex. Most of the CPCIs are Technical Modules. In addition, certain of these are subroutines and are so identified.

Group 1 - Rotor (Modal)

(FIRST LEVEL)

- RD00 Actuator disk representation of rotor, propeller, ducted fan. Simplified equations for thrust, drag, and power.
- RD22 One out-of-plane mode for all hubs (hingeless, bearingless, semi-articulated, articulated, teetering or gimballed). Blade feathering included. Arbitrary distribution of airfoil section geometric, inertial and structural blade properties. Rigid hub. Kinematic coupling. Conventional rotor, circulation control, servo flap, reaction drive, propeller, ducted fan.
- RD45 Linear model representation, out-of-plane, in-plane and torsional degrees of freedom. All hubs except teetering and gimballed. Arbitrary distribution of blade properties. Kinematic coupling and Coriolis forces. Pre-cone, pre-lag of feathering axis. Sweep, droop with respect to feathering axis. Feathering axis offsets, blade chordwise and flapwise offsets. Lag dampers. Conventional rotor, circulation control, servo-flap, reaction drive, propeller, ducted fan.
- RD45-T Same as RD45 except for teetering and gimballed hub. Teetering rotor undersling.

(SECOND LEVEL)

- RD77 Same as RD45 with addition of general nonlinear effects and airfoil warpage. Unequal shaft axial and azimuthal spacing. Elastomeric lag dampers. Plastic flap and lag stops. Blade dissimilarities. Complex hub representation. Blade and hub vibration control devices, control load reduction devices.
- RD77-T Same as RD45 except teetering and gimballed hub.

GROUP 2 - Rotor (Finite Element)

(SECOND LEVEL)

- RD44-F Finite element representation of rotor blade. Rigid, bearingless, semi-articulated, articulated hub. Linear analysis including out-of-plane, in-plane and torsional degrees of freedom. Lag dampers. All features of RD45.
- RD44-FT Same as RD44-F except for teetering, gimballed hub.
- RD88-F Same as RD77 except finite element representation.
- RD88-FT Same as RD88-F except for teetering and gimballed hub.

Group 3 - Rotor (Eigensolution)

(FIRST LEVEL)

- RE23 Flapping degree of freedom, all hub types. Multi-blade coordinates including simplified linear aerodynamics.
- RE34 Flapping, lagging, torsion modes, all hub types. Multi-blade coordinate analysis including simplified linear aerodynamics.

Group 4 - Rotor (Stand Alone)

(FIRST LEVEL)

- RS44 Determination of natural frequencies and normal modes for uncoupled or coupled motions including out-of-plane, in-plane and torsional directions. All hub types.

Group 5 - Airframe (Modal)

(FIRST LEVEL)

- FD22 Rigid body and static elastic analysis of airframe including fuselage, pylons, landing gear, external stores, rigid aerodynamic surfaces. Arbitrary number and location of rotor interfaces. Infinite impedance and specified motions included.
- FD55 Airframe model analysis including all degrees of freedom, including fuselage, lifting surfaces, pylons, external stores, suspended cargo, vibration control devices, rotor isolation, and landing gear. Arbitrary rotor locations. Arbitrary suspension and applied vibratory forces.
- FE33 Same as FD22 with addition of linear aerodynamics.

Group 6 - Airframe (Stand Alone)

(FIRST LEVEL)

- FS45 Determination of natural frequencies and mode shapes for uncoupled, coupled motions. Segmented beam formulation.

(SECOND LEVEL)

- FS66 Finite element matrix representation of airframe (e. g., reduced NASTRAN analysis). Including fuselage, lifting surfaces, pylons, landing gear, external stores, suspended cargo, internal cargo, vibration control devices, rotor isolations. Arbitrary rotor locations. Arbitrary suspensions and applied vibratory forces.

Group 7 - Airframe (Finite Element)

(SECOND LEVEL)

- FD56 Use of vibration test data representation of fuselage impedances.
- FD77 Finite element matrix representation with the addition of dynamic landing gear representation, including wheel skid or flotation devices and possibility of asymmetric retraction. Fuel representation including weight, inertia, distribution, fuel consumption and slosh effects. Internal cargo including departure from aircraft. Fuselage mounted stores, weapons and sensors with dynamic representation. Externally suspended cargo including towing of devices and winch down capabilities with prescribed motions. Ground plane or deck capability.

Group 8 - Engine/Drive System

(FIRST LEVEL)

- ED22 Torsional analysis using rigid or static elastic representation of engine/drive system.
- ED33 Torsional, transverse bending analysis of engine/drive system. Clutches, flywheels and mounting devices including effects of mounting system. Gearboxes, gear losses.

(SECOND LEVEL)

- ED66 Finite element representation of engine/drive system, torsion, transverse bending. Dynamic with slop, vibration isolation devices, gearboxes, gear losses, fuel and fuel control systems, circulation control and reaction drive components.

Group 9 - Engine Performance

(FIRST LEVEL)

- E11 Subroutine - steady-state engine performance analysis using simplified curve fit or generalized curve fit.
- E22 Subroutine - steady-state engine performance analysis. Generalized thermodynamic analysis or thermodynamic analysis for specific engines according to SAE APR 681C. Fuel and geometry control system.

(SECOND LEVEL)

- E33 Subroutine - transient engine performance, first-order representation, including fuel and geometry control system.
- E44 Subroutine - transient engine performance, higher order representation, including fuel and geometry control systems.

Group 10 - Control System

(FIRST LEVEL)

- CD22 Simple representation of control system. Rigid or static elastic representation.
- CD33 Primary control system represented by constant gear ratio with linear force feel, control mixing including mixing of fixed-wing controls using constant coupling matrix. Passive simulated pilot. Simple, linear automatic flight control system.

(SECOND LEVEL)

- CE43 Control system with linear simulated pilot and linear automatic flight control system.
- CD66 Primary control system represented by nonlinear gearing, second-order dynamics including lost motion. Secondary controls. Active pilot responsive to accelerations and stick forces, including map of the earth flight. Automatic flight control system represented as nonlinear with limits.

Group 11 - Air Mass

(FIRST LEVEL)

- A01 Subroutine - steady aerodynamics. Linearized representation for C_l , C_d , C_m . Fixed stall. Lift, drag pitching moment of rotating, lifting surfaces.
- A02 Subroutine - steady aerodynamics. Lift, drag, pitching moment, vertical drag of lifting surfaces and bodies, including interference effects.
- A23 Subroutine - steady aerodynamics. Representation by tabular data including bivariant (α , M), trivariant (α , Λ , M), and quadrivariant (α , Λ , M , R_e) or (α , Λ , M , C_μ). Effects of flaps, spoilers, three-dimensional tip effects, circulation control and radial flow. Lift, drag pitching moment of rotating, nonrotating lifting surfaces and bodies.

AD-A064 131

CONTROL DATA CORP HAMPTON VA
PREDESIGN OF THE SECOND-GENERATION COMPREHENSIVE HELICOPTER ANA--ETC(U)
OCT 78

F/G 1/3

DAAJ02-77-C-0058

USARTL-TR-78-43

NL

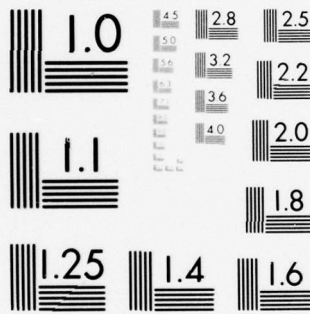
UNCLASSIFIED

2 OF 2

AD
A064131



END
DATE
FILMED
4-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

- A34 Subroutine - steady aerodynamics. Analytical methods such as Theodorsen, Loewy, vortex lattice.
- AS33 Sets up uniform and nonuniform inflow distribution, including methods such as Glauert with and without time delay for rotating components.

Group 12 - Air Mass

(SECOND LEVEL)

- A35 Subroutine - dynamic stall. Analytical methods including (α , A, B) and time delay.
- A45 Subroutine - nonuniform inflow, for nonrotating lifting surfaces, including interactions between and/or among rotating and nonrotating components.
- A45-1 Subroutine - dynamic stall, semi-empirical, empirical, including drag effects.
- A46 Subroutine - rotor nonuniform inflow, prescribed wake analysis including interactions between and/or among rotating and nonrotating components.
- A77 Subroutine - nonuniform inflow. Free wake analysis including interactions between and/or among rotating and nonrotating components.
- A88 Subroutine - advanced analytical techniques including potential flow with separation for analysis of aircraft or aircraft component. Drag, vertical drag, lift, pitching moment, for combination of aircraft components, external stores, suspended cargo, doors, ramps and other components. Wind tunnel representation including effects of tunnel walls and ground plane.
- AS21 Sets up gust geometry with penetration in the vertical and horizontal directions. Also, vortex encounter.

Group 13 - Complete Aircraft

(FIRST LEVEL)

- HS22-S Coleman Ground Resonance Analysis
- HS26-S Aeroelastic Stability
- HS02-P Complete Aircraft Performance Analysis

Group 14 - Post Processing - Acoustics

(FIRST LEVEL)

- PS32 Summation of noise spectra, noise index calculations
PS54 Engine noise prediction; piston or turbine.
PS55 Rotor system noise prediction

(SECOND LEVEL)

- PS33 Interior noise prediction
PS35 Transmission noise prediction
PS68 Advanced technology rotor noise prediction
PS77 Advanced technology interior noise prediction

Group 15 - Post Processing - Math

(FIRST LEVEL)

- MS22-D Differential equation solution (simple)
MS11-E Eigensolution; real, symmetric matrices
MS20-F Harmonic analysis
MS55-S General statistical analysis (auto, cross correlations;
 regression analysis)
MS13-I System identification (simple; external model use)

(SECOND LEVEL)

- MS53-D Differential equation solution (intermediate with error checks)
MS55-A Accuracy assessment algorithm
MS23-E Eigensolution; nonsymmetrical, complex matrices
MS46-E Floquet analysis
MS45-F Fast Fourier transform
MS88-D Differential equation solution (advanced)
MS76-F Moving block randomdec
MS46-I System identification (intermediate; external model use)
MS77-I System identification (advance; external model use)

Group 16 - Criteria

(FIRST LEVEL)

JG10-T Simple trim procedure

(SECOND LEVEL)

JG35-T Generalized trim procedure

JG55-E Perturbs condition for external model analysis

JG62-F Simple failure/damage criteria; stops execution at specified time

JG41-S Perturbs conditions for development of Floquet matrix

JG67-T Generalized trim procedure; response surface generation

JG86-F Failure/damage; tests for specific loads, stops program execution, changes parameters and continues execution.

Group 17 - Utility

(FIRST LEVEL)

MAT 1 General matrix routines including inversion

DAT 1 Simple data check for reasonableness

UNIT Units conversion

FOR Data format conversion

INT Integration

(SECOND LEVEL)

MAT 2 Advanced matrix inversion method

DAT 2 Advanced data check

SYSTEM USAGE

SYSTEM CONTROL LANGUAGE

The System Control Language (SCL) provides the user with a comprehensive set of helicopter modeling capabilities and data base management. These include the ability to control execution sequence in the analysis of a variety of aircraft configurations, define data base entry formats and their content, and retrieve and update values in these data base entries. The basic components of data base entry definitions are scalar data elements and array data elements. Values corresponding to these elements are stored in data base entry records. Array data elements can be fixed or variable in length with the size of variable length arrays determined by values assigned to scalar elements.

The execution and control of helicopter analyses is provided through a set of Sequence Control statements. Employing these statements, a user of the system can invoke sets of stored Sequence Control statements that are termed "Stored Procedures", execute Helicopter Models and associate specific data sets with the model, modify the sequence of execution, assign values to parametric data, perform conditional operations, and interrupt processing.

The data required by the Helicopter Models will be stored, either temporarily or permanently, in the data base. This data will be defined by the user and identified when a Helicopter Model is executed.

Helicopter analysis models are defined and stored in the data base for subsequent execution. These models contain a list of the analysis components and the analysis method modules that will be used for the analysis. When the helicopter analysis model is executed, this list directs the System to a set of "Technical Module Definitions" which describe the input, output, and coupling requirements for each of the analysis component modules. This data will then be used in setting up the analysis problem for solution.

LEVELS OF USE

The SGCHAS provides three levels of user interface permitting system usage with a minimum of training while simultaneously providing extended features to the experienced user. The three levels of use are as follows:

- a. Basic System Usage
- b. Intermediate System Usage
- c. Advanced System Usage

Basic System Usage

The basic level of system usage provides the engineer with the ability to introduce rotorcraft physical component data to the system and invoke standard analysis procedures. The SCL statements which provide these capabilities are of two types: Data Base Maintenance and Sequence Control.

Basic Data Base Maintenance - There are five Data Base Maintenance Statements which are required for basic system use.

- a. ACCESS - This statement initiates data base maintenance processing and, optionally, identifies a file on which the user's data is to be stored. If the user does not specify a file for data storage, the System will place the data on a temporary "Run Data Base" file.
- b. ADD - This statement is used to add a new entry record to the data base. Through the ADD statement the user will identify the specific entry record type (i. e. , main rotor, passive pilot, airframe, etc.) and assign values to specific named data elements (e. g. , number-of-blades) within the record. For example:

ADD ROTOR = MY-R-A4B

TYPE = A, (Articulated Rotor)

NB = 4, (4 blades)

NS = 5, (5 blade stations)

(etc.)

- c. MODIFY - This statement is used to modify values assigned to specific named data elements within an entry record.
- d. DELETE - This statement is used to remove entry records from the user's data base file.
- e. END* - This statement indicates to the System that the user has completed data base maintenance processing.

Basic Sequence Control - There are three Sequence Control statements that are required during basic system usage.

- a. PROCESS - This statement initiates Sequence Control processing and optionally identifies a user data base file from which component data will be retrieved during processing. If the file is not specified, the System will utilize data stored in the Run Data Base and Master Data Base files.

- b. CALL - This statement invokes standard and user defined analysis procedures that are stored in the Master Data Base or User Data Base files.
- c. END* - This statement identifies the end of Sequence Control statement processing.

Intermediate System Usage

The intermediate level of system usage provides the engineer with the ability to define specialized rotorcraft analysis configurations and procedures. An expanded set of SCL statements and two specialized data base records are used to implement these capabilities.

Intermediate Data Base Maintenance - Three additional data base maintenance capabilities are defined for intermediate system use.

- a. COPY - This statement provides the user with the ability to transfer data from one data base file to another.
- b. PRINT - This statement provides the user with the ability to examine information from the data base file.
- c. PLOT - This statement provides the user with the ability to obtain a graphic representation of data contained in the data base files.

Intermediate Sequence Control - The set of intermediate sequence control statements includes the basic capabilities and adds seven additional ones:

- a. CHANGE - This statement provides the ability to temporarily change the data being used in an analysis. Its operation is similar to the MODIFY data base maintenance statement except the changed record is not stored on a user data base file, but is placed on the temporary Run Data Base file.
- b. EXECUTE - This statement is used to invoke an arbitrary or predefined rotorcraft analysis configuration which is defined in the data base as a Helicopter Model Definition. It can also be used to execute sequential (stand-alone) technical modules representing complete simple analyses, acoustics analyses, and other non-simultaneous processes.
- c. GOTO - This statement provides simple branching within a Sequence Control statement set.
- d. IF - This statement provides simple selection capabilities and, when used with the GOTO statement, iterative capability within a Sequence Control statement set. It can be used with criteria technical modules to form iterative trim procedures, to introduce damage effects, or both, and can test user and system run-time data through simple conditionals.

- e. SAVE - This statement provides the ability to stage data from the Run Data Base file to a User Data Base file, thus retaining the information for use in subsequent runs.
- f. SET - This statement provides the user with simple arithmetic capabilities within a set of Sequence Control statements.
- g. STOP - This statement permits the user to arbitrarily interrupt system processing and restart with the next statement. Data generated prior to the STOP statement will be reported and a system checkpoint will be performed.

Data Base Records - The intermediate level user will have available two special record formats in the data base files in addition to the component data record formats used for the basic level.

- a. Helicopter Model Definitions (HMD) - The HMD is used to describe an arbitrary rotorcraft analysis configuration to the System. An HMD will identify the mathematic technique, analysis method, and component analysis technical modules and subroutines that are to be used.
- b. Stored Procedure Definition (SPD) - The SPD is used to describe a set of Sequence Control statements for storage in the data base for subsequent recall via the CALL statement.

Advanced System Usage

The advanced level of system usage provides the research engineer with expanded Data Base Maintenance and Sequence Control capabilities and with the ability to install new technical capabilities.

Advanced Data Base Maintenance - Two additional Data Base Maintenance statements are provided for the advanced level user.

- a. LOCK - The LOCK statement is used to permanently protect a data base file from further system output operations.
- b. UNLOAD - The UNLOAD statement is used to extract records from the data base, create card image records, and output the data to a sequential file. It permits data interchange between the SGCHAS and other systems.

Advanced Sequence Control - In addition to the statements already defined, the advanced level user has an expanded form of the EXECUTE statement which permits modification of Helicopter Model Definition records dynamically before execution.

Data Base Records - An additional data base record format is available to the advanced level user. This record provides dynamic installation of technical modules and subroutines. Termed the Technical Module Definition (TMD), the record will identify input, output, degrees of freedom, and coupling relationships for a specific technical module or subroutine and makes the functional portion of that module logically available for use.

ILLUSTRATIONS OF USE

In this section several annotated examples are given to demonstrate the use of the System. Some necessary details such as complete data record references, output option specification, and JCL have been omitted for the sake of clarity. It should also be noted that precise syntax has not yet been defined.

Preliminary Design Performance

In this example, it is assumed that the Master Data Base contains a preliminary design performance which was delivered with the System and is designated PFC1.

The only user input required to execute this "stored procedure" is a CALL statement, followed by the names of the records that contain the proper input data.

The user simply inputs the following in order to perform the complete analysis:

```
CALL PFC1, ROTD = RUH2(1), FUSD = . . .
```

where ROTD, FUSD, . . . are dummy names of records containing rotor, fuselage, . . . data and RUH2, . . . are the specific record names. The subscript for RUH2 indicates that the data is to be used for rotor 1. The establishment and maintenance of these records and files is discussed elsewhere.

Previous to this usage, a Helicopter Model Definition (HMD) and a Stored Procedure Definition (SPD) were formulated and placed in the Master Data Base by the System Developer. The HMD and SPD for this example are discussed in the paragraphs below.

Helicopter Model Definition - The helicopter model definition contains the appropriate physical component technical module names and the mathematical algorithm module as follows:

- (1) HMD = PRELP
- (2) METH = MS22-D

- (3) COMP = RD22(A01, 1)
- (4) RD00(2)
- (5) ED22(E11, 1, 2)
- (6) FD22(A03, 1, 2)

These inputs may be described as follows:

- (1) The helicopter model has been named PRELP for preliminary performance
- (2) The method specified is MS22-D which is a simple numerical solution method for differential equations (see definitions of all technical modules under the SYSTEM CAPABILITIES section of this report).
- (3) Rotor 1 (main rotor) is to be represented by a single out-of-plane mode and feathering degree of freedom for each blade. The aerodynamic forces and moments are computed using linear steady aerodynamics with fixed stall.
- (4) Rotor 2 (tail rotor) is an actuator disk representation with thrust, drag, and power capabilities.
- (5) The static elastic engine/drive system drives both rotors and uses a steady engine performance method.
- (6) The airframe is a rigid or static elastic representation, coupled to both rotors and uses a single aerodynamic analysis.

Information such as the number of blades, airfoil section, locations and orientation of rotors, and location and geometry of aerodynamic surfaces on the fuselage is passed to the technical modules through the input data records.

Since no control system has been specified, control inputs are passed directly to the rotors and aerodynamic surfaces. Note that no data is referenced in the helicopter model definition.

Stored Procedure Definition - The stored procedure which in our example was assumed to be supplied with the System may have been defined as follows:

- (1) SPD = PFC1
- (2) A, EXECUTE PRELP
- (3) USE ROTD, FUSD, . . .
- (4) IF JG10-T GOTO A.
- (5) EXECUTE MS20-F, USE BM1.

These inputs may be described as follows:

- (1) The stored procedure has been named PFC1 and is now addressable through a CALL statement.
- (2) This statement causes the helicopter model previously defined to be executed (see details under Mathematical Basis of System in this report). Note that any statement may have an arbitrary label as shown.
- (3) The USE statement defines dummy data records which are specified in the CALL statement.
- (4) The IF statement uses the criteria module JG10-T which tests for trim. If the vehicle is not trimmed within specified tolerances, new control inputs are computed by the module and the processing returns to statement A.
- (5) When the trim conditions are satisfied, a harmonic analysis algorithm is executed on the data representing the bending moments of rotor 1 (BM1).

Linear Stability Preliminary Design

A simple model, similar to the above is presented for a stability analysis. The components all are named with an "E" as the second character indicating a pure linear analysis (no "active functional module"), and the rotor and fuselage have self-contained linear aerodynamics. The method is an eigenvalue analysis.

For this problem it is necessary to define or use a predefined helicopter model in the data base. Such a model may be formed as shown:

```
HMD = LSPD
METH = MS11-E
COMP = RE23(1)
      RE00(2)
      FE33(1, 2)
      CE43(1, 2)
```

Note that the control system is specifically modeled in this problem. The input required to execute the module is as follows:

```
EXECUTE LSPD, USE . . .
```

The input describing the physical components would be different from that required by the previously described problem since this problem requires control inputs.

Loads, Acoustics, Detailed Design

For this analysis the helicopter model and stored procedure may be defined as follows:

Helicopter Model Definition -

```
HMD = LDD
METH = MS22-D
COMP = RD45(A23, 1)
      RD00(2)
      ED22(E11, 1, 2)
      FD55(A02, 1, 2)
      CD33(1, 2)
```

In this case a more complex rotor model is used employing a blade modal representation and a more comprehensive airmass subroutine. The airframe is also represented in modal form.

Stored Procedure Definition - The stored procedure required is similar to above:

```
SPD = LDDTRM
A, EXECUTE LDD
USE RDAT, . . .
IF JG10-T GOTO A
```

The function of this procedure is to trim the above helicopter model.

User Input - Once the model and procedure, above, have been defined, the user may supply the following input for his particular problem:

- (1) EXECUTE AS33, USE INDAT
- (2) EXECUTE RS44 (1), USE RDAT1
- (3) CALL LDDTRM
- (4) EXECUTE MS20-F, USE BM1, CLD
- (5) EXECUTE PS55, USE RPO1

- (1) This step produces a rigid nonuniform wake field for use by the aerodynamic subroutine.
- (2) A modal analysis of the blades of rotor 1 is performed for use by the rotor technical module.
- (3) The helicopter is trimmed (see SPD above).
- (4) Bending moments (BM1) and control loads (CLD), which were output from the trim analysis, are analyzed.
- (5) A rotor noise prediction analysis is performed using the computed rotor pressure distribution (RPD1).

It should be noted that the names of the data sets that are output by each step are listed in the Definition Modules of the Technical Modules used for the analyses.

Nonlinear Aeroelastic Stability

This example illustrates some of the convenient features of the System Control Language which allow maximum user control of the System.

Define a new stored procedure, similar to the user-defined problem, above:

- (1) SPD = ASDD1
- (2) EXECUTE AS33, USE INDAT
- (3) EXECUTE RS44, USE RDAT1
- (4) CALL LDDTRM, RD45 = RD77
- (5) SET INIT (1) = INIT (1) + 1
- (6) EXECUTE LDD, RD45 = RD77, USE . . .
- (7) EXECUTE MS76-F

Lines (2), (3), and (4) are identical to the previous example, except that in line 4 the rotor analysis, RD45, is replaced by a more advanced analysis, RD77.

When (4) is completed, the helicopter is trimmed. Step (5) is a user data input which perturbs an initial condition. In step (6) the same helicopter model is executed but not trimmed. In step (7) a randomdec analysis is performed on the transient data.

The statement CALL ASDD1 is required to actually perform the analysis described by the SPD.

Rerun for Stability Check - In a practical problem situation, it is possible that the engineer may have some doubts as to the validity of the analysis. If he should desire to check the results by using a more accurate and better behaved integration algorithm, he may input

CALL ASDD1, MS22-D = MS88-D

This one input will repeat the entire analysis with no changes except that a more accurate (and more costly) method of integrating the equations is used.

Damage Effects

A last example is given in which the helicopter is trimmed, the stiffness of rotor 1, blade 1, station 5 is changed, the helicopter is then retrimmed, and the bending moments of the good and damaged blades are harmonically analyzed.

CALL LDDTRM

SET E11 (1, 5) = .5 * E11 (1, 5)

CALL LDDTRM, RD45 = RD77, MS22-D = MS53-D

EXECUTE MS20-F USE BM1, BM2

Summary

The preceding illustrations were intended to display the ease of use and great flexibility of the System. Features such as the capability to predefine a virtually unlimited set of helicopter models and problem definitions and the capability to specify changes in these definitions at run time are believed to be of great benefit to the engineering user.

The engineer will have the capability to use the precise model and procedure required for the problem being studied. He will have the capability to easily modify the problem to obtain engineering information relative to sensitive flight conditions and to perform an analysis to optimize the cost and accuracy characteristics of the solution prior to production running.

DEVELOPMENT AND MAINTENANCE AIDS

Normal system development and maintenance activities often result in the introduction of errors in existing, tested processes. These activities encompass the installation of new capabilities and modification and deletion of existing capabilities. Often new capabilities require new or modified record formats and thus changes impact other processes and proliferate throughout the system.

These problems have been alleviated in the system design through use of a structural modular design approach and emphasizing functional data independence.

Structured Modular Design

The design technique employed for the SGCHAS combines two structured design techniques to ensure that the design meets the user's needs and is highly maintainable. The first technique, design by objectives, identifies the long-range system objectives and the specific processing required to meet those objectives. The second technique, functional hierarchical refinement, is then used to align these requirements into functional categories and in a step-wise manner define detailed operations until a programmable system is designed. The resultant system is highly modular and maintainable.

Functional Data Independence

One of the main expenses in system modification and maintenance results from the changing and addition of record formats when the system components directly interface with the data records. Such data dependence has been largely eliminated by modern data base management systems, such as IBM's IMS and Control Data's DMS-170, which allow program components to access data by referencing the name of a particular field within a record. The main disadvantage of these systems has been the memory required for their usage (125K bytes for IMS, 21K words for DMS-170).

This design reduces the memory required while providing data independence by designing a specialized engineering data base management system. It provides the maintenance team with two types of System Control Language statements as maintenance aids:

- a. Data Base Definition statements which describe new record formats; and,
- b. Data Base Maintenance statements which manipulate data within the data base.

Data Base Definition - Using five Data Base Definition statements, the development and maintenance teams can perform the following operations.

- a. DEFINE - This statement initiates Data Base Definition processing and identifies a file which is to receive the new definition.
- b. ENTRY - This statement introduces a new entry record format and assigns a record type identifier to it.
- c. ELEMENT - This specification describes the characteristics of a particular data field in a record including: dimensions, type, output format, validation criteria, and units of measure.
- d. PRINT - This statement requests the printing of an entry record format.
- e. END* - This statement terminates Data Base Definition statement processing.

Data Base Maintenance - The Data Base Maintenance statements described under "Levels of Use" are supplemented by extensions to the UNLOAD statement which provide for transferring of system data between computer systems. These Data Base Maintenance statements will be used to define standard Helicopter Model Definitions and Stored Procedure Definitions to provide the PFCs described in the Type A System specification.

RESOURCE UTILIZATION

Executive Overhead

The modular design of the System and extensive use of dynamic loading throughout the Application Executive to control the residency and nonresidency of system components results in the minimization of executive memory overhead. Although memory utilization will vary during system execution, it is estimated that a typical analysis problem can be solved in less than 95K bytes of memory, including host operating system overhead and the Application Executive requirements listed below.

Executive Supervisor CPCI - During normal system execution, many of the components of the Executive Supervisor are nonresident. The components that will be resident are expected to have the following memory requirements:

- a. Dynamic Loader - 500 bytes
- b. Data Manager -
 - (1) Primary Storage Manager - 500 bytes
 - (2) Secondary Storage Manager - 5000 bytes
- c. Checkpoint - 500 bytes
- d. Diagnostic Processor - 200 bytes
- e. Units-of-Measure Conversion - 300 bytes
- f. System Accounting - 150 bytes

Thus making the total Executive Supervisor overhead 7150 bytes.

Batch Subsystem CPCI - During the solution processing for the typical analysis problem the majority of the components of the Batch Subsystem CPCI will be nonresident. The components and subcomponents that will be resident are as follows:

- a. Batch Subexecutive - 250 bytes

- b. Sequence Control Statement Processor - 250 bytes
- c. Execute Statement Processor - 1500 bytes

Making the total Batch Subsystem CPCI memory overhead during problem solution 2000 bytes.

Interactive Subsystem CPCI - During interactive solution processing, the overhead is expected to be largely the same as the Batch Subsystem except for the addition of about 1500 bytes for interactive set up and diagnostic functions.

Graphics Package CPCI - The Graphics Package is expected to require approximately 3 - 4 kilobytes of memory.

Host Operating System Overhead - The Host Operating System is expected to require 30 - 35 kilobytes of memory for normal FORTRAN system routines.

SGCHAS Executive Data - The Application Executive is expected to require a maximum of 10 kilobytes for tables and other data.

Small Problem Efficiency

In this System small problems are solved as small problems and not as large problems filled with zeroes. The size of the matrices in differential equation solutions is precisely equal to the particular number of degrees of freedom in the problem.

The core allocations for the problem are set up dynamically by the Application Executive at run time and there is no storage overhead as is associated with FORTRAN programs using fixed COMMON blocks even when variable dimensions are used in subroutines.

The executable code will be developed and controlled during development so as to be as efficient as possible.

The overhead associated with the Application Executive will be the minimum consistent with necessary functions (see previous section).

It is anticipated that all problems will be considerably more efficient in both time and computer resources than presently used programs.

Large Problem Capability

The efficiency of large problems benefits from the same considerations mentioned in the previous paragraph, i. e. , minimum case and maximum computational efficiency.

In addition, there are several features associated with the solution of large problems (time domain solution of dynamically coupled systems) which have considerable impact on the efficiency of solution.

The separation of the coding of the technical component modules into "coefficient" and "active" modules assures maximum case utilization for instruction storage. Those operations which are performed only once during a problem solution (such as computation of constant coefficients and formulation of constant summed matrices) are not retained in memory but are executed and then eliminated. The memory region is then used for other functions. Only coding which represents operations that are to be repeated during the problem solution are retained in main memory. This is a considerable advancement over most present state-of-the-art programs.

Another feature which increases efficiency is the capability of the engineer to customize the specific problem to be solved to his specific needs (see "Illustrations of Use," above). This capability assures that the problem will not be treated in a more complex manner than necessary and thus waste resources and time, and that an inadequate analysis will not be performed and waste the entire effort. The capability of the engineer to easily test several levels of complexity prior to production running assures that this extremely important aspect of efficiency is realizable.

The memory requirements for the matrices described above are quite straightforward. N_i is the number of degrees of freedom of component i and N_c is the number of coupled system degrees of freedom.

The M_i , C_i , K_i , F_i matrices for each component require $N_i (3N_i + 1)$ words of storage. When these coefficients are constant there is no need to retain these after M_{c_0} , C_{c_0} , K_{c_0} are formed. Thus the storage for all the component matrices is:

$$\sum_{i_v} N_i (3N_i + 1) + \sum_{i_c} N_i$$

where i_v is the subset of i corresponding to variable matrices and i_c is the subset corresponding to those which are constant.

The requirement for the constant coupled matrices (M_{c_0} , C_{c_0} , K_{c_0}) is $3N_c^2$.

It is also necessary to allocate storage for M_c , C_c , K_c , F_c , M_c^{-1} , V_i , V_i , \bar{V}_c , V_c , V_c . Thus the total matrix storage required is:

$$3 \sum_{i_v} N_i (N_i + 1) + 3 \sum_{i_c} N_i + N_c (7N_c + 4)$$

The memory requirements for the storage of the coefficient matrices will be less than that indicated above by implementing an algorithm which recognizes that in most analyses the equations for each blade will be identical (except for periodic terms), even though a different solution may be associated with each blade.

SYSTEM DEVELOPMENT

ORGANIZATIONAL RESPONSIBILITIES

The organization to be used during the Development Phase should be a project-oriented organization designed to maximize the utilization of resources but still provide all the necessary functions for successful system development. The basic reasons for establishing a formal organization include clarity of job assignment, minimizing unnecessary interactions, controlling change and establishing responsibilities and direction. These reasons extend to the Development Contractor, Subcontractors and the Government.

The Development Contractor must provide the management and control for a successful conclusion of the Development Phase under the auspices of the Government and within the Statement of Work (SOW). Management and control features must extend beyond personnel and subcontract management and into the areas of analysis, design, programming, testing and documentation to ensure that the resultant products for the system are acceptable, reliable, and standardized to be transportable and maintainable. The Development Contractor will utilize a portion of his resources to effect this management and control.

To ensure that the above areas are defined for management and control responsibilities, specific statements and assignments must be made. Also, management and control must necessarily be in proportion to the size and complexity of the project.

Based upon estimates of the size of project and division of the effort into areas where the best specific knowledge and experience can be applied, the format of the organization is partitioned into two areas: (1) management, control and the executive portion of the system; and (2) the technical portion. The Development Contractor will effect the first area, and subcontractors and CPI contractors will effect the second area.

DEVELOPMENT CONTRACTOR

The Development Contractor must be administratively as well as technically qualified to support large software development projects such as the SGCHAS. If the contractor lacks support in the areas of contract administration, financial accounting, personnel, EEO, and similar administrative functions, then the burden for these functions will fall on the Project Development Team and will detract from the development effort.

Based upon the detail of the System design, division of effort, documentation that will be produced, necessity for comprehensive tests, further necessary effort of design and programming, and management and control, a suggested project organization is shown in Figure 17 for the first 6 months' effort. A suggested project organization is shown in Figure 18 for the peak organization. The project organization for the Second-Level Release is shown in Figure 19.

It is suggested that the responsibilities of the Development Contractor include, but not be limited to, the following items:

- (1) Developing a detailed viable schedule for the completion of activities to include measurable units of work (documentation, each formal review, each programmable module's specifications, each programmable module's design, each programmable module's code, each programmable module's test, each program integration, CPCI test, CPCI integration and test, CPCI acceptance test). The schedule can be used by the Development Contractor to control and report activities.
- (2) Designing the System.
- (3) Providing walk-through reviews to assist the Government with its monitoring function in system development.
- (4) Recommending revisions to the Type A System Specifications.
- (5) Recommending revisions to the Development Plan.
- (6) Preparing the System Specification to specify system design concepts.
- (7) Preparing the Acceptance Test Plan.
- (8) Preparing system hierarchical charts.
- (9) Identifying CPCIs.
- (10) Preparing Subsystem Specifications for CPCIs.
- (11) Implementing quality assurance techniques defined in the Development Plan.
- (12) Recommending those CPCIs to be developed by the Development Phase contractor, those CPCIs for subcontractors, and those CPCIs to be Government furnished.
- (13) Developing those CPCIs approved by the Contracting officer.
- (14) Determining that each CPCI in design meets the functional requirements and quality assurance provisions via the Preliminary Design Reviews and Critical Design Reviews.

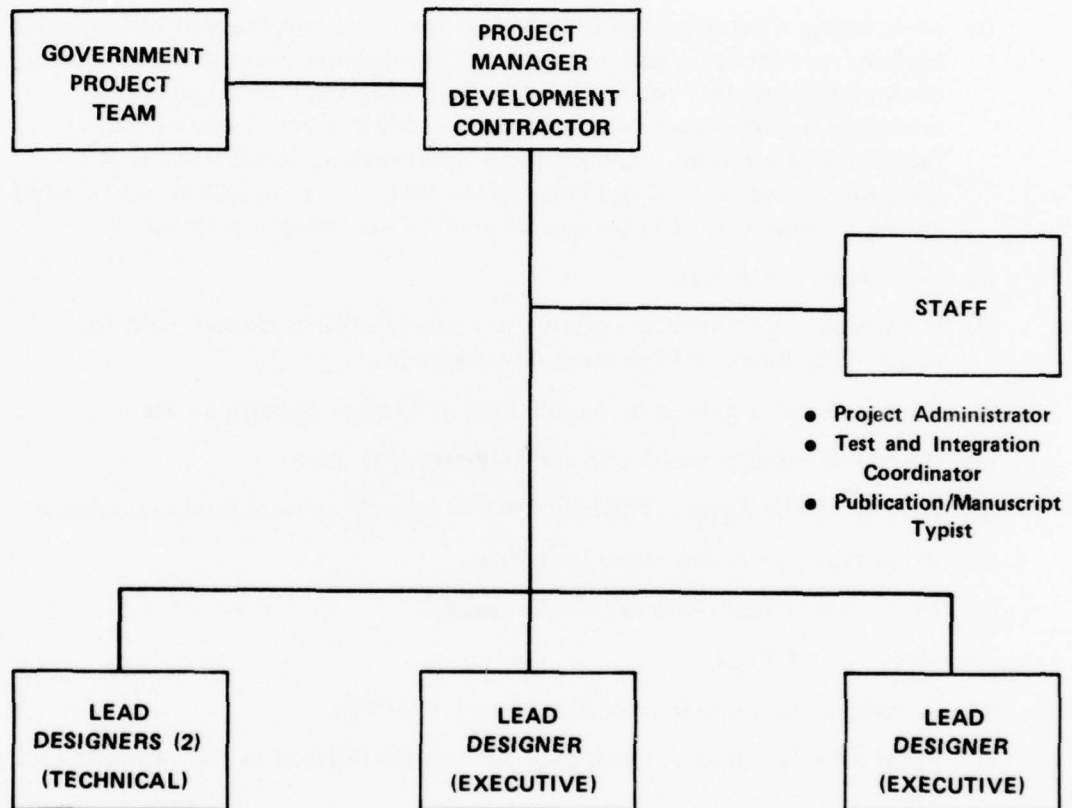


Figure 17. Analysis and Design Organization - 6 months - Control Data/Kaman.

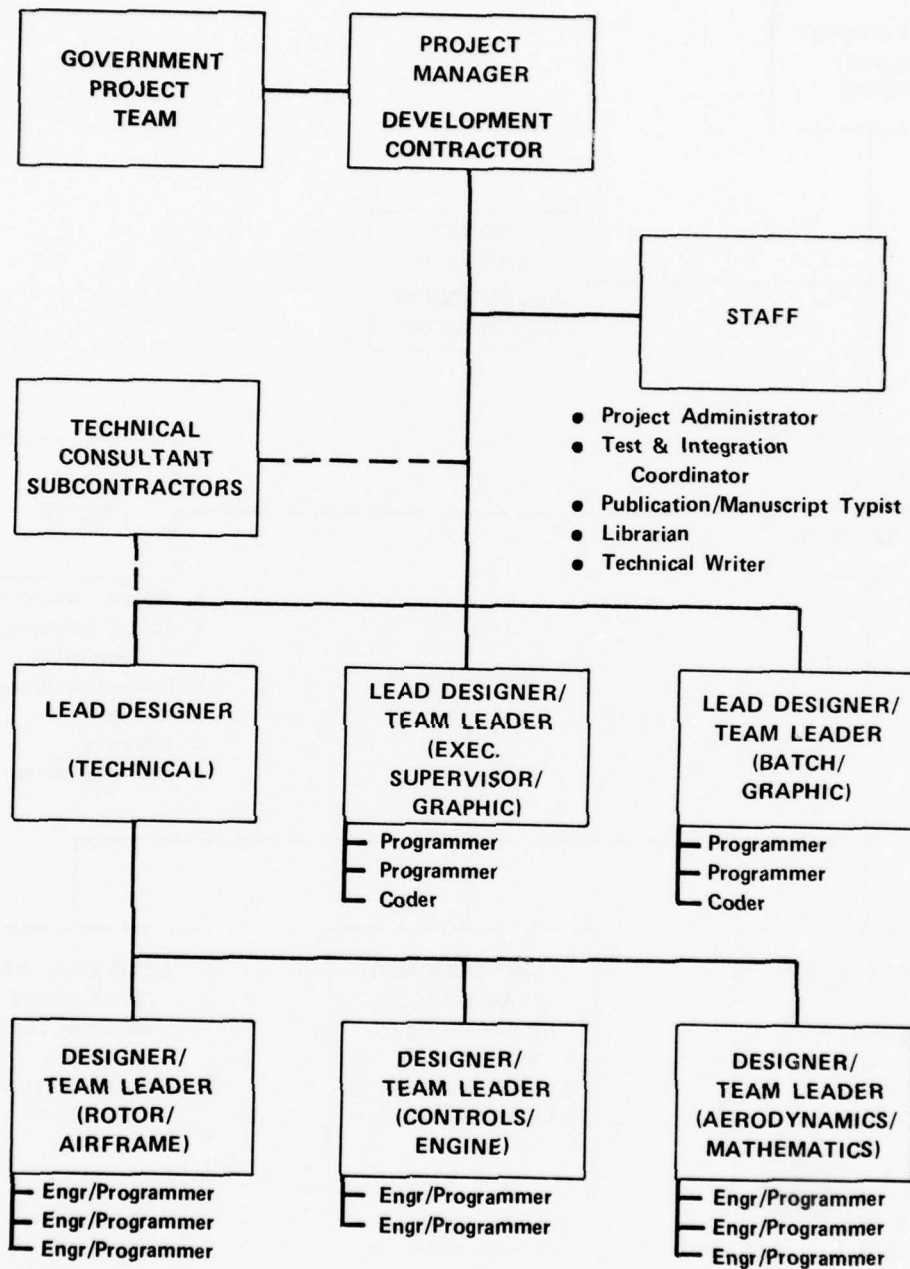


Figure 18. First-Level Release Peak Organization.

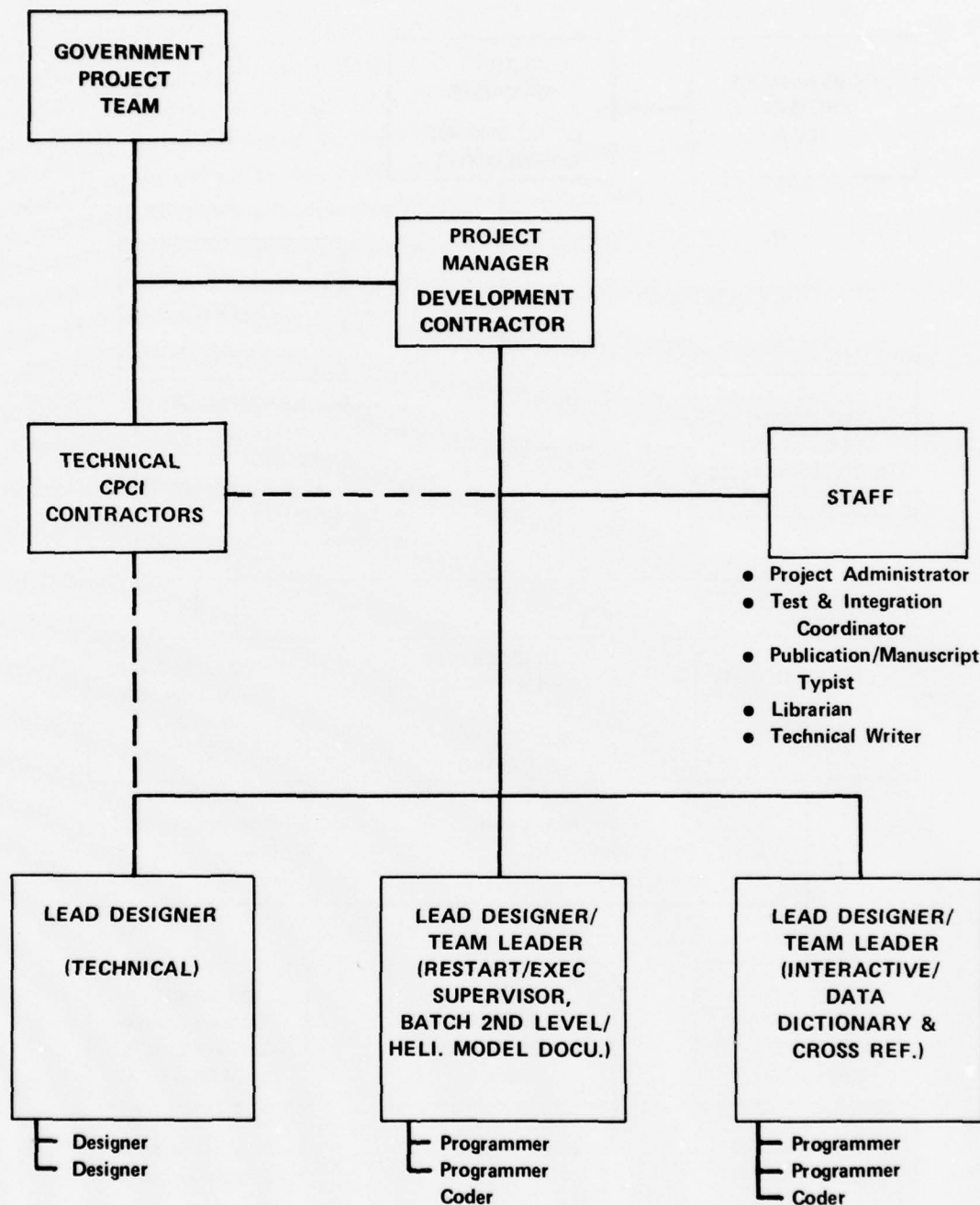


Figure 19. Second-Level Average Organization.

- (15) Producing specifications for CPCIs.
- (16) Assisting the Government's participation in the evaluation of contractors for CPCI development.
- (17) Preparing an Acceptance Test Plan.
- (18) Expanding the Development Plan to define and implement all techniques and standards for commonality of all deliverable products which will include a unified documentation approach and a configuration management plan.
- (19) Preparing to implement each CPCI that is selected and approved by the Government.
- (20) Insuring that the detail design provides the logic to meet the functional requirements and quality assurance provisions.
- (21) Producing structural flow design and source code for the modules to meet standards.
- (22) Insuring that programmable modules adhere to programming standards.
- (23) Conducting modular tests.
- (24) Conducting CPCI tests.
- (25) Determining that each CPCI, through tests, meets the functional and accuracy requirements of its specification.
- (26) Producing test cases for the System against specifications.
- (27) Having Tests performed and evaluated by personnel other than the producers of the coded instructions.
- (28) Conducting integration tests for developed modules that are placed into the system via the Test and Integration Plan.
- (29) Assisting in finalizing requirements for experimental data necessary to determine CPCI and System Level accuracy.
- (30) Building and testing deliverable program packages in preparation for Governmental acceptance tests.
- (31) Assisting the Government's participation to be actively involved in tests of the System.
- (32) Conducting functional demonstrations of the System to demonstrate to the Government and industry that the System meets the functional requirements and quality assurance provisions of the Type A System Specification via the Acceptance Test Plan.

- (33) Preparing and controlling developmental libraries.
- (34) Preparing Test Analysis Reports.
- (35) Preparing for and providing training and maintenance support to Government and industry users during the initial portion of the Validation Phase.
- (36) Producing a User's Manual, Maintenance Manual and Theoretical Manual for releases of the System.
- (37) Delivering all products to the Contracting Officer.

Subcontractors

A technical subcontractor should be utilized by the development contractor for the technical area to provide the expertise that is required for rotorcraft technology. It is suggested that outside technical consultants (research organizations, educational institutions, and rotorcraft manufacturers) be given consideration as possible subcontractors for consultation and development to provide additional rotorcraft expertise particularly for the First-Level. The Technical Subcontractor should be a definite integral part of the Development Phase team. Utilizing the concept of Development Phase Contractor and Technical Subcontractor, definitive statements of effort can be made.

The Technical Subcontractor will be issued a Statement of Work at the onset of the Development Phase. This Statement of Work will be a subset of the Government's Statement of Work and contract provisions and will establish the overall objectives, assignments and expectations of the work to be performed by the Technical Subcontractor. The Statement of Work will be oriented to work in helicopter technology and technical CPCIs as the Development Phase Contractor will be working in the Executive area. The Technical Subcontractor, as well as the Development Contractor, will be required to adhere to those standards as defined in sections of the Development Plan to ensure that the final delivered products for the SGCHAS are standardized.

All types of formal communication to the Government that are stated in the Development Plan will be the responsibility of the Development Contractor. However, the Technical Subcontractor shall have the responsibility to adhere to activities (communications, progress and cost reports, formats, standards, etc.) of the Development Plan with the Development Contractor in the same manner as the Development Contractor will adhere to the plan with the Government.

Technical Subcontractor - First-Level Release - The Technical Subcontractor will have the following prime responsibilities for the First-Level Releases.

- (1) Participate in the design of the overall System.
- (2) Assist in the preparation of the System Specification and Acceptance Plan for coordination and inclusion of the Technical CPCI's.
- (3) Participate in the Functional Design Review.
- (4) Participate in the System Design Concurrence Review.
- (5) Develop, submit and update a detailed schedule to the development contractor for the completion of activities to include measurable units of work (documentation, each formal review, each programmable module's specification, each programmable module's design, each programmable module's code, each programmable module's test, CPCI test and CPCI integration and test).
- (6) Prepare technical CPCI's preliminary Subsystem Specification and Test and Integration Plan for the Preliminary Design Review.
- (7) Participate in the Preliminary Design Reviews for CPCI's.
- (8) Assist in additional consultant/developer subcontracting direction.
- (9) Finalize the technical CPCI's preliminary Subsystem Specification and Test and Integration Plan for the Critical Design Review.
- (10) Participate in the Critical Design Reviews for CPCI's.
- (11) Design, code and test the technical CPCI's programmable modules.
- (12) Test the programmed technical CPCI for CPCI tests.
- (13) Prepare integration test data to prove system compatibility and assure accuracy.
- (14) Integrate and test the technical CPCI.
- (15) Produce acceptance test data according to Government-approved specifications provided in the Acceptance Test Plans for the technical CPCI.
- (16) Demonstrate that the final products meet the standards of the Development Plan and the Subsystem Specification.
- (17) Provide assistance to resolve CPCI errors discovered during and after technical CPCI integration.
- (18) Prepare documentation for integration into the User's Manual, Program Maintenance Manual and Theoretical Manual.

- (19) Coordinate, as required, with technical representatives of the Government in conjunction with the Development Contractor.
- (20) Deliver all completed products (programs, documentation, test data) to the development contractor.
- (21) Participate in the Functional Configuration and Formal Qualifications review.

Technical Subcontractor - Second-Level Release - During the Development Phase for the Second-Level Release, the Technical Subcontractor will continue with the responsibilities as defined for the First-Level Release for those CPCIs that are not contracted by the Government. In addition, the Technical Subcontractor will have the following responsibilities for those CPCIs that are contracted by the Government in the Second-Level Release.

- (1) Prepare draft technical CPCI preliminary Subsystem Specifications for Government contracting.
- (2) Participate in the Preliminary Design Reviews for CPCIs.
- (3) Recommend technical CPCI contracting direction.
- (4) Coordinate with the Government-sponsored technical CPCI contractor in the finalization of the CPCI Subsystem Specification.
- (5) Participate in the Critical Design Reviews for CPCIs.
- (6) Monitor the development of the technical CPCIs.
- (7) Check and validate technical CPCI design for System compatibility.
- (8) Validate the technical CPCI contractor's test cases.
- (9) Assess the validity of the technical CPCI contractor's acceptance criteria for CPCI integration into the system.

Technical CPCI Contractors - Second-Level Release - Technical CPCI contractors can be utilized to provide specialized rotorcraft expertise. As generally known, the various participants in the rotorcraft industry have specialized talents and expertise that may not be industry wide. These specialized talents and expertise will be required to develop technical CPCIs, particularly the more advanced technical CPCIs. A premise of the Statement of Work for the Predesign effort was that few, if any, Second-Level System CPCIs will be developed by subcontractors.

Technical CPCI contractors can be obtained by the Government through two methods: competitive bid and sole source. As the development schedule for a CPCI is lengthy due to the necessary reviews that are associated with quality

assurance and contract production efforts, it is suggested that the majority of technical CPCI contracts be awarded by the sole source method. A method that should be investigated is the open-ended contract (time and materials/level of effort). This method could produce a shorter time frame for a CPCI but must be very rigidly controlled.

The technical CPCI contractors will be issued a Statement of Work as well as the provisions in the prime contract by the Government. It is suggested that the Government-sponsored technical CPCI contractors be required to adhere to the standards defined in the Development Plan to ensure that the final delivered products are standardized. (Note: Items that are not to be included in the System but are required to be delivered can be the contractor's format.)

It is suggested that the responsibilities of a technical CPCI contractor shall include, but not be limited to:

- (1) Develop, or coordinate the development of, and submit a detailed schedule to the Government for the completion of activities to include measurable units of work (documentation, each formal review, each programmable module's specification, each programmable module's design, each programmable module's code, each programmable module's test, each program integration, CPCI integration and test, CPCI acceptance test). The schedule will be used to control and report activities.
- (2) Develop the detailed technical CPCI Subsystem Specification from the initial CPCI Subsystem Specification and Integration and Test Plan in consonance with the Second-Level Release Technical Subcontractor.
- (3) Submit the detailed technical CPCI Subsystem Specifications and Integration Test Plan for the Critical Design Review.
- (4) Participate in the Critical Design Review.
- (5) Finalize technical CPCI Subsystem Specification for the baselined documentation.
- (6) Design, code, and test the CPCI's programmable modules.
- (7) Test the programmed CPCI for CPCI tests in a stand-alone or system test-bed environment.
- (8) Prepare integration test data to prove system compatibility and assure accuracy, and coordinate with the Development Contractor's Test and Integration Coordinator.

- (9) Provide assistance to the Government for the conductance of stand-alone tests or system test-bed tests for acceptance of the CPCI in accordance with the quality assurance provisions of the development specifications.
- (10) Provide assistance to the Development Contractor in repeating the same acceptance tests prior to formally integrating the CPCIs into the System.
- (11) Integrate and test the CPCI into the System.
- (12) Provide assistance to resolve CPCI errors discovered during and after CPCI integration.
- (13) Produce system acceptance test data according to Government-approved specifications provided in the Acceptance Test Plans for the technical CPCI.
- (14) Prepare documentation for integration into the User's Manual, Program Maintenance Manual and Theoretical Manual and coordinate with the Development Contractor's Technical Writer.
- (15) Demonstrate that the final products meet the standards of the Development Plan and the Subsystem Specification.
- (16) Deliver all completed products (programs, documentation, test data) to the Government and Development Contractor.

Government

This Final Report does not attempt to define the responsibilities of the Government with respect to the Second Generation Comprehensive Helicopter Analysis System. However, Table 2 summarizes relationships and responsibilities of the various organizational categories and how they can apply to the SGCHAS Project.

Level of Effort

Figure 20 is a summarization of detailed CPCI schedules and personnel assignments, and shows the estimated manning level for the Development Contractor, Technical Subcontractor and Government-sponsored CPCI contractors on a bi-monthly basis. The peaks and valleys of the manning level were the result of scheduling Second-Level Release CPCIs in the Second-Level time period. These efforts can be smoothed by rescheduling to begin the Second-Level CPCIs in the First-Level time frame. It has been estimated through detailed examination of activities and products that are required for SGCHAS that the level of effort will be approximately 1100 man-months.

TABLE 2. RELATIONSHIPS OF SYSTEM DEVELOPER, TECHNICAL
SUBCONTRACTOR, CPCI CONTRACTOR, GOVERNMENT AND TAG/GIWG.

	<u>System Developer</u>	<u>Technical Subcontractor</u>	<u>CPCI Contractor</u>	<u>Government</u>	<u>TAG/GIWG</u>
1. Objectives and Requirements				Prepares	
2. System Design					
a. Revisions to Type A System Spec.	Prepares	Prepares		Approves	Critiques
b. Revisions to Development Plan	Prepares	Assists		Approves	Critiques
c. System Specification	Prepares	Prepares		Approves	Critiques
d. Acceptance Plan	Prepares	Prepares		Approves	Critiques
3. Executive CPCIS - (1) Exec. Supr. Subsystem, (2) Batch Subsystem (3) Restart Subsystem (4) Inter-active Subsystem					
a. Subsystem Specification and Test and Integration Plan	Prepares	Reviews		Approves	Critiques
b. Modules' Spec., Design, Code, Test	Produces			Reviews	
c. CPCI Test	Performs			Approves	
d. Functional Config. Audit	Assists			Performs	
e. Integration and Test	Performs			Reviews	
f. Acceptance Test	Assists			Performs	
g. Manuals - User and Prog. Maint.	Prepares			Approves	Critiques

TABLE 2. RELATIONSHIPS OF SYSTEM DEVELOPER, TECHNICAL
SUBCONTRACTOR, CPCI CONTRACTOR, GOVERNMENT AND TAG/GIWG. (Continued)

	<u>System Developer</u>	<u>Technical Subcontractor</u>	<u>CPCI Contractor</u>	<u>Government</u>	<u>TAG/GIWG</u>
4. Graphics Package					
a. Subsystem Specification	Prepares			Approves	Critiques
b. Buys or Lease	Accomplishes			Approves	
c. CPCI Test	Performs			Approves	
d. Functional Config. Audit	Assists			Performs	
e. Integrate and Test	Performs			Reviews	
f. Acceptance Test	Assists			Performs	
g. Manuals - User and Prog. Maint.	Prepares			Approves	Critiques
5. Technical CPCIs					
First Level - 14 Specifications 39 CPCIs					
a. Subsystem Specification and Test and Integration Plan	Reviews	Prepares		Approves	Critiques
b. Modules' Spec., Design, Code, Test	Approves	Produces		Reviews	
c. CPCI Test	Reviews	Performs		Approves	
d. Functional Config. Audit	Assists	Assists		Performs	
e. Integration and Test	Assists	Performs		Reviews	
f. Acceptance Test	Assists	Assists		Performs	
g. Manuals - User and Prog. Maint.	Integrates Drafts and Produces	Writes Drafts		Approves	Critiques

TABLE 2. RELATIONSHIPS OF SYSTEM DEVELOPER, TECHNICAL
SUBCONTRACTOR, CPCI CONTRACTOR, GOVERNMENT AND TAG/GIWG. (Continued)

	<u>System Developer</u>	<u>Technical Subcontractor</u>	<u>CPCI Contractor</u>	<u>Government</u>	<u>TAG/GIWG</u>
6. Technical CPCIs					
Second Level - 11 Specifications 42 CPCIs					
a. Prelim. Subsystem Specifications and Test and Integration Plan	Reviews	Prepares		Approves/ Contracts	Critiques
b. Detailed Subsystem Specification and Test and Integration Plan	Reviews	Critiques	Prepares	Approves	Critiques
c. Modules' Spec., Design, Code, Test	Approves		Prepares	Reviews	
d. CPCI Test	Approves	Reviews	Performs	Approves	
e. Functional Config. Audit	Assists		Assists	Performs	
f. Integration and Test	Assists	Reviews	Performs	Reviews	
g. Acceptance Test	Assists	Assists		Performs	Critiques
h. Manuals - User and Prog. Maint.	Integrates Drafts and Produces		Writes Drafts	Approves	Critiques

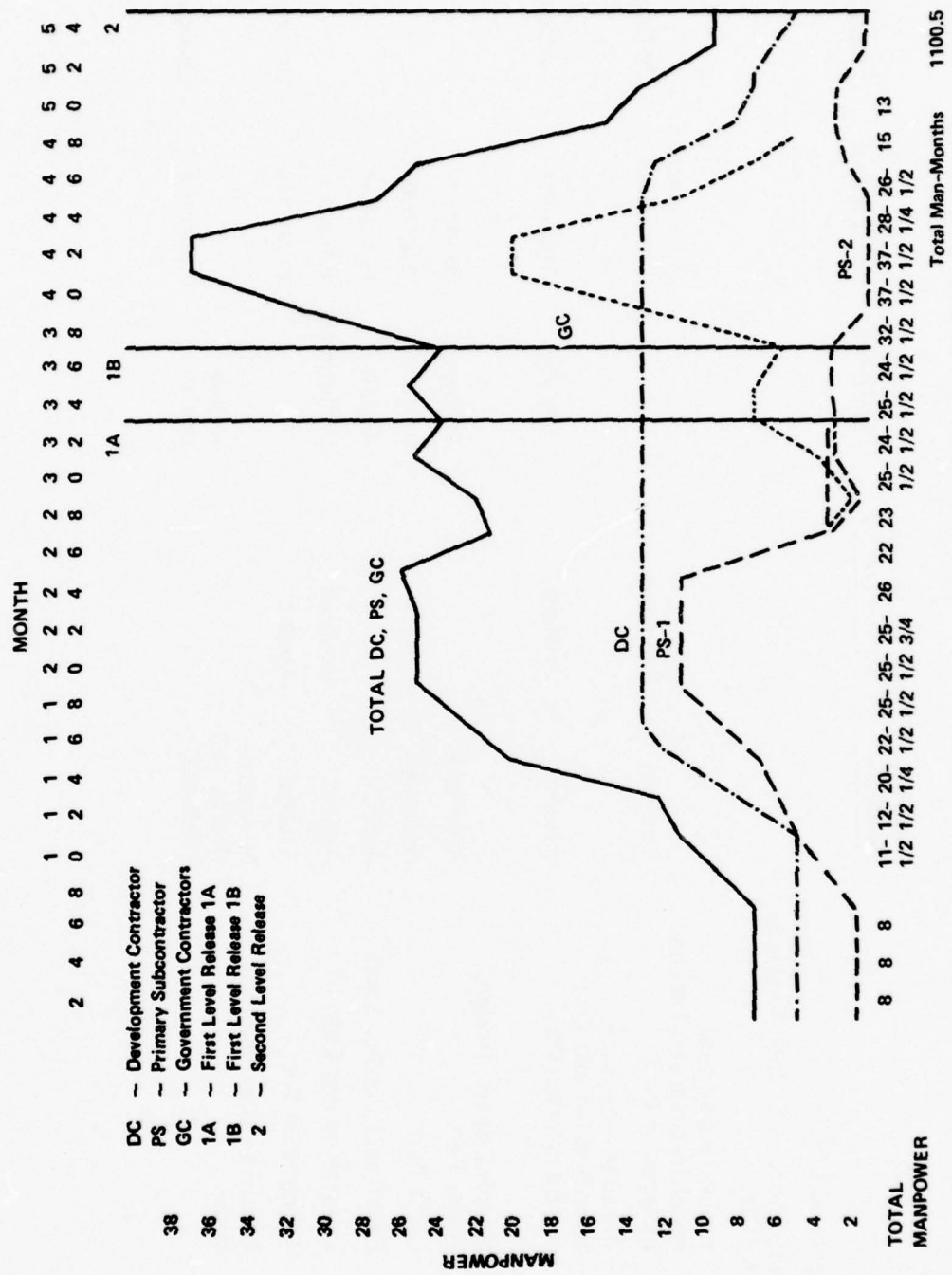


Figure 20. Bi-Monthly Levels of Manpower.

Developmental Facilities

For expediency, long-range cost effectiveness, and pseudo-operational environment for expected users, the software can be developed on commercially available facilities (or similar facilities which utilize IBM equipment and CDC equipment). As these types of facilities are often linked into nationwide services, development of software (particularly CPCI integration testing) by the Development Contractor, Technical Subcontractor, other subcontractors, and Second-Level CPCI contractors will be enhanced through the common facilities and system data base. (A CPCI Government contractor may develop on the facilities of his choice before CPCI integration into the system data base.)

DEVELOPMENT SCHEDULE

The activities and events, based upon the Type A System Specification, Statement of Work in the Predesign effort, the system design concept, program hierarchical structure, and military standard documents, have been established to form the schedule for the First- and Second-Level Releases. Table 3 is a summarization of the technical capabilities discussed in previous sections.

First-Level Release

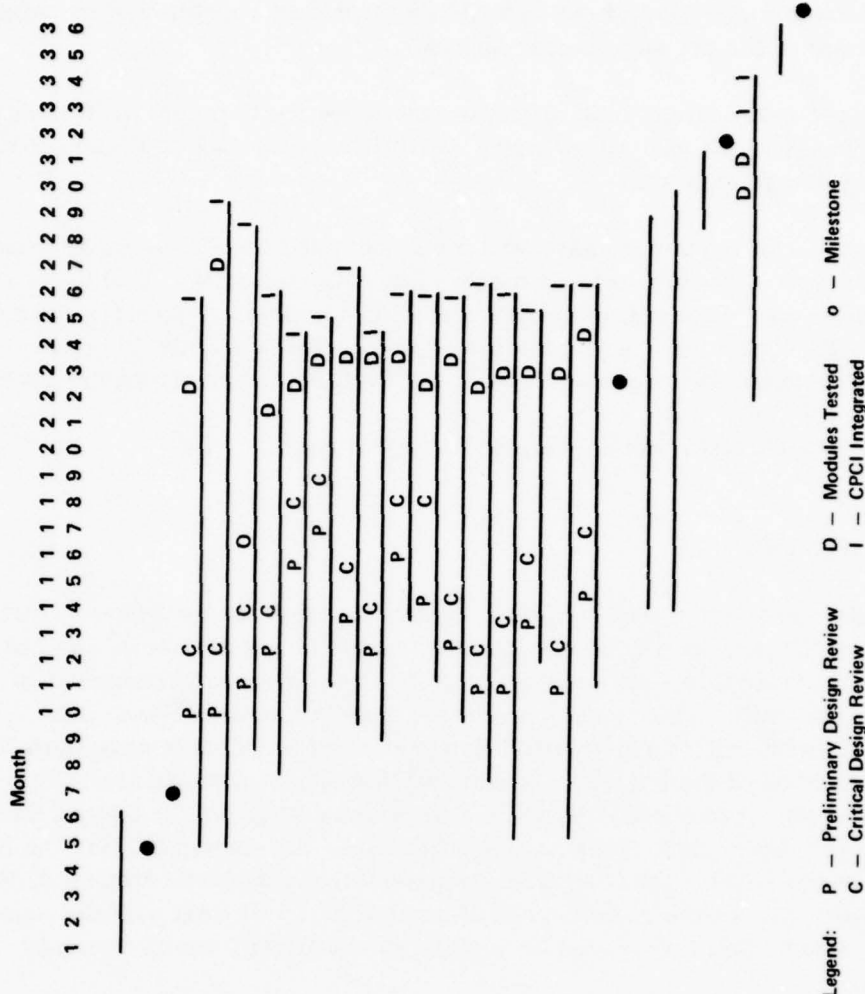
The objective of the First-Level System Release will be to produce a system making extensive use of state of the art rotary-wing technology and software techniques. The First-Level Release is expected to contain an executive program and technical modules that will provide the level of sophistication and capabilities comparable to that used currently in the helicopter industry. The First-Level Release is an approach to provide an early leadtime implementation of the system capabilities for validation and user acceptance. The First-Level System Release shall provide current state of the art technology and software techniques without sacrificing the potential of the Second-Level and Long-Range System capabilities. The First-Level (1A) System Release for IBM equipment is scheduled for release 32 months after beginning the Development Phase. The First-Level (1B) System Release for CDC equipment is scheduled for delivery 36 months after the beginning of the Development Phase. Figure 21 is a schedule for the First-Level Releases.

Second-Level Release

The objectives of the Second-Level System Release will be to provide more advanced rotary-wing technology and software techniques than the First-Level System. It will incorporate corrections for errors and deficiencies which will

TABLE 3. TECHNICAL CPCI LEVEL SCHEDULE

<u>Group</u>	<u>First Level</u>	<u>Second Level</u>
1 - Rotor (Modal)	RD00, RD22, RD45 RD45-T	RD77, RD77-T
2 - Rotor (Finite Element)		RD44-F, RD44-FT, RD88-T, RD88-FT
3 - Rotor (Eigensolution)	RE23, RE34	
4 - Rotor (Stand Alone)	RS44	
5 - Airframe (Modal)	FD22, FD55, FE33	
6 - Airframe (Stand Alone)	FS45	FS66
7 - Airframe (Finite Element)		FD56, FD77
8 - Engine/Drive System	ED22, ED33	ED66
9 - Engine Performance	E11, E22	E33, E44
10 - Control System	CD22, CD33	CE43, CD66
11 - Air Mass	A01, A02, A23 A34, AS33	
12 - Air Mass		A35, A45, A45-1, A46, A77, A88, A21
13 - Complete Aircraft	HS22-S, HS26-S, HS02-P	
14 - Post Processing - Acoustics	PS32, PS54, PS55	PS33, PS35, PS68, PS77
15 - Post Processing - Math	MS22-D, MS11-E, MS20-F, MS55-S, MS13-I	MS53-D, MS55-A, MS23-E, MS46-E, MS45-F, MS88-D, MS76-F, MS46-I, MS77-I
16 - Criteria	J610-T	JG35-T, JG55-E, JG62-F, JG41-S, JG67-T, JG86-F
17 - Utility	MAT1, DAT2, Unit for Int	MAT2, DAT2



1. System Design Documents
2. Functional Design Review
3. System Design Concurrence
4. Exec. Supervisor (1st Level)
5. Batch Subsystem (1st Level)
6. Graphic Package
7. Tech. CPCI Group 1
8. Tech. CPCI Group 3
9. Tech. CPCI Group 4
10. Tech. CPCI Group 5
11. Tech. CPCI Group 6
12. Tech. CPCI Group 8
13. Tech. CPCI Group 9
14. Tech. CPCI Group 10
15. Tech. CPCI Group 11
16. Tech. CPCI Group 13
17. Tech. CPCI Group 14
18. Tech. CPCI Group 15
19. Tech. CPCI Group 17
20. Tech. CPCI Integration Build
21. User's Manual (1st Level - IBM)
22. Prog. Maint. Manual (1st Level - IBM)
23. Acceptance Testing (IBM)
24. FIRST LEVEL RELEASE 1A (IBM)
25. Development to CDC (Prog., Manuals)
26. Acceptance Testing (CDC)
27. FIRST LEVEL RELEASE 1B (CDC)

Legend: P - Preliminary Design Review D - Modules Tested o - Milestone
 C - Critical Design Review I - CPCI Integrated

Figure 21. First-Level Release Schedule.

have been identified after the First-Level System release. It will complete the executive system and incorporate additional functional capabilities by using advance state of the art engineering analysis.

The Second-Level System release will maintain a growth potential in the overall system concept which will allow future capabilities to be added in the course of its Long-Range existence.

The Second-Level System release will occur near the end of the Development Phase, which is approximately 54 months after its beginning. The Second-Level System will be operable on IBM and CDC equipment. Interim releases have not been scheduled. It is possible and suggested that interim releases be made to provide on-site evaluation of the System at more frequent intervals.

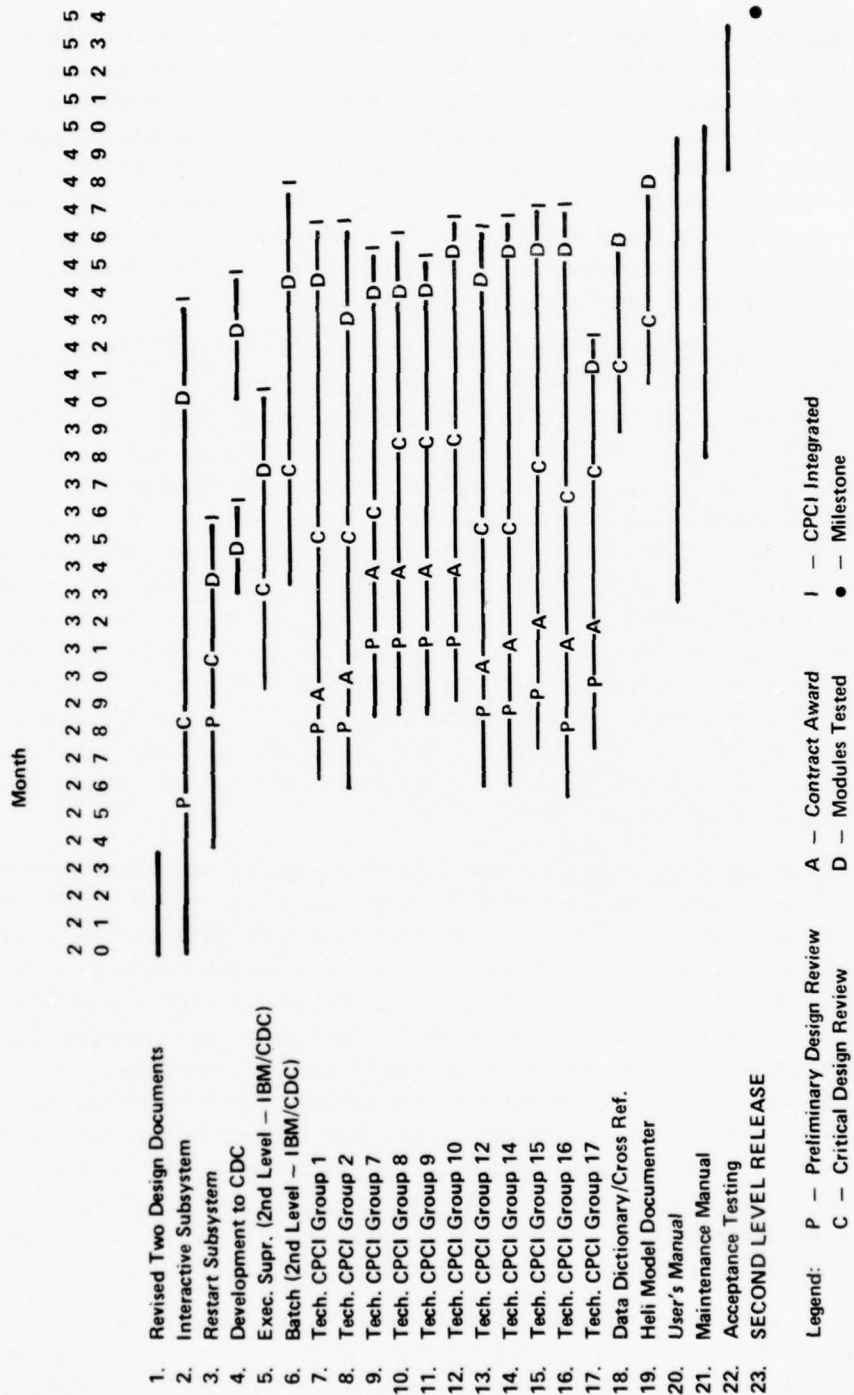
Figure 22 is a schedule for the Second-Level Release.

DOCUMENTATION

Documentation for large and complex systems can take many different forms for the same purpose. Documentation has many uses but is primarily used to (1) provide developers with documents that can be reviewed at significant developmental milestones to determine that requirements are met and (2) record technical information to allow coordination of later development and use/modification of the system. Documentation should provide uniformity of format and content particularly within a project as large as the Second Generation Comprehensive Helicopter Analysis System. The documents for the SGCHAS project can conform to the DOD ADS Documentation Standards Manual 4120.17-M with exceptions. As the standards in Manual 4120.17-M deal with the communication of information that cannot be rigidly standardized, modification is recommended.

The Development Contractor should expand the Documentation Plan at the earliest practical time of the Development Phase to provide a unified and practical guide to be followed for writing, punctuation, editing, formatting and publishing the SGCHAS documents. This section will contain the detail statements required to promote uniform standards. All detail for publishable documents will be contained in the Development Plan to provide the commonality and standardization for all agencies of SGCHAS.

It is emphasized that the specifications are the vehicles that dictate the capabilities that will be produced for the System. As such, all input from the various interested agencies and users are necessary and required before the specifications are baselined.



Legend: P - Preliminary Design Review
 A - Contract Award
 I - CPC1 Integrated
 C - Critical Design Review
 D - Modules Tested
 • - Milestone

Figure 22. Second-Level Release Schedule.

The documents envisioned for SGCHAS are discussed in the following paragraphs.

- Type A System Specification - The Baseline Type A System Specification as provided from the Predesign Phase will be used as the document that defines the system requirements and operational capability that are to be developed and will be the major control for System Development. If the requirements defined by the Type A System Specification are changed during project development, the Type A System Specification shall be updated as controlled by Configuration Management. Use of the Type A System Specification is in addition to the Standards Manual 4120.17-M.
- System Specification - One of the first objectives of the Development Phase is for the Prime Contractor and Technical Subcontractor(s) to produce a System Specification for the overall design of the System. The System Specification will be produced during system design to identify CPCIs, allocate requirements of the Type A System Specification to CPCIs, and specify the complete overall design for the System. The rationalization for design decisions will be chronicled to provide an historical trail. The System Specification will contain an executive summary that can be used by nontechnical personnel to obtain a comprehensive understanding of the capabilities of the System. The System Specification will be the second control for Government approval before proceeding to the more detailed Subsystem Specifications for CPCIs. The System Specification should be prepared for review and approval at the Functional Design Review to direct the more detailed design effort.
- Subsystem Specifications - The Subsystem Specifications (comparable to Type B5 Development Specifications) will be provided for CPCIs or a group of similar CPCIs. The Subsystem Specification will contain enough information so the detailed program design can be developed. The Subsystem Specification is a technical document prepared for systems personnel. It is to be as detailed as possible concerning the environment and design elements in order to provide maximum guidance for the CPI modular design effort. The document also defines system/subsystem interfaces and provides a logical flow in the form of macro flowcharts so coding modules can be specified. Subsystem Specifications for Executive CPCIs will be prepared by the Development Contractor. Subsystem Specifications for Technical CPCIs will be prepared by the Technical Subcontractor(s).

As the system development progresses, the System Specification may have to be updated to remain current. Any changes in the requirements specified by the document will be controlled through the Configuration Management process.

The Subsystem Specification for a CPI will be a control for Government approval before proceeding to the coding phase.

- Program Maintenance Manual - The Program Maintenance Manual presents general and specific information on the System for use by the personnel who will be responsible for the maintenance of the System. It will describe the System in a detailed, technical presentation to assist the maintenance programmer in his functions.

The Program Maintenance Manual shall be prepared by the Development Contractor during the coding and system test phases of the Development Phase. A Program Maintenance Manual should be produced for each approved release of the System.

- User's Manual - The primary purpose of the User's Manual is to serve the needs of the user group with documentation sufficient to utilize both the executive system capabilities and technical modules. As the User's Manual must provide the detailed information for operation of user system's capabilities, the format in Standards Manual 4120.17-M will be modified. The User's Manual will be prepared by the Development Contractor and Technical Subcontractor and will be updated for each approved release of the System.
- Theoretical Manual - The purpose of this manual is to provide a concise description of the methods that can be employed in the solution of problems. It will be prepared by the Technical Subcontractor under supervision of the Development Contractor. It will describe analytical methods, modeling techniques, component characteristics, complexity levels, etc.
- Test and Implementation Plans - It is recommended that two types of test plans be implemented: (1) Acceptance Test Plan for the System and (2) CPCI Test and Integration Plans for the Computer Program Configuration Items. These plans are tools for directing the different types of tests, and contain the orderly schedule of events and lists of materials necessary to effect delivery of the System. The Acceptance Test Plan should be prepared by the Prime Contractor's test and integration coordinator during the system design phase for approval with the System Specification at the Functional Design Review, thereby assuring that all requirements are included for testability. To produce the Acceptance Test Plan at a later date could introduce omissions.
- Test Analysis Reports - Test Analysis Reports will describe the status of the computer program system after tests and provide a presentation of deficiencies for review by Government staff and management personnel. The reports will be prepared by the organization which conducts the testing of the CPCIs and the system.

- Development Plan - The Development Plan is in addition to Standards Manual 4120.17-M and will be prepared by the Development Contractor. It is suggested that a Development Plan be included to define the development effort. The purpose of the Development Plan is three-fold: (1) it is a planning document that describes the development effort; (2) it is a master document to organize and contain the planning effort; and, (3) it is a standardization document for projecting standards and techniques for commonality. The Development Plan should contain at least the following plans: organization plan, quality assurance and control plan, test plan, configuration management plan, documentation plan, training plan, review and reporting plan, maintenance plan, and a resources and deliverables plan.
- Computer Program Documentation - The information and data that are commonly written into a programming specification should also be written for placement into the program source listing. This information applies to design, data and flow charts for each program module. As the program modules are designed from the Subsystem Specifications, functions, data design, tables, common areas and flow charts will be created. This information can be transformed by the programmer into a special module prologue section and into regular comment statements interspersed among the executable statements of the program module.

QUALITY ASSURANCE AND CONTROL

Quality assurance and control should begin with the initiation of the project and continue until its completion. Figure 23 illustrates the quality assurance loop and shows how reviews, walk-throughs and testing assures compliance with the specifications developed for the System. Quality control begins with the definition of testable system objectives which lead to the identification of functional requirements for the objectives. These system objectives and requirements are documented in a Type A System Specification which is reviewed and modified by the Development Team and baselined at a Functional Design Review. The Type A System Specification becomes the basis for subsequent system design activity, the results of which will be documented in a System Specification (see DoD Manual 4120.17-M) and Subsystem Specifications. These design specifications are reviewed at the System Design Concurrence (SDC), Preliminary Design Review (PDR), and Critical Design Review (CDR) to assure compliance with the stated objectives and requirements of the System. The System will then be programmed and each program module will be thoroughly tested. To ensure that no logic path within a module has been overlooked in testing, the module tests will be reviewed, using the walk-through technique, by the programmer and the responsible team leader. When a CPCI has been programmed, it will be tested to determine compliance with its design as

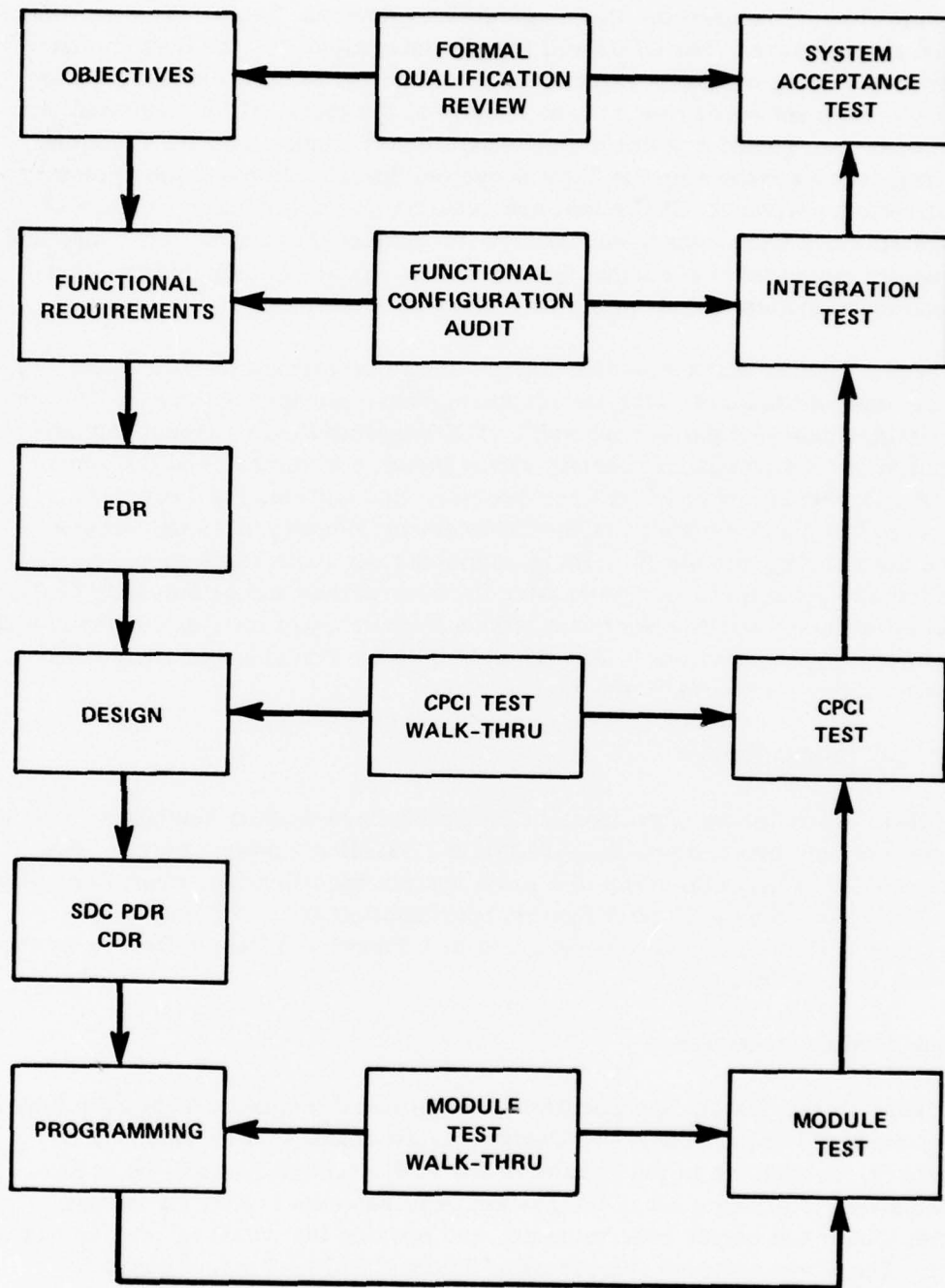


Figure 23. Quality Assurance Loop.

documented in a Subsystem Specification. These tests will then be reviewed by the team which developed the CPCI and the Development Team's Test and Integration Coordinator. The CPCI will then be integrated into the System and integration testing will be performed by the Test and Integration Coordinator under Government supervision. The results of the tests will be reviewed at a Functional Configuration Audit to demonstrate compliance with the functional requirements as defined by the Type A System Specification and the System Specification. When all CPCIs that are required for a particular release of the system have been integrated, System Acceptance Testing is performed and the results reviewed at a Formal Qualifications Review to demonstrate that the developed system achieves the objectives defined for it.

It is recommended that a Baseline Review Board be formed to review and critique design products in the Development Phase for approval by the contracting agency of the Government. The Baseline Review Board can be composed of Government personnel, development contractor, and (as appropriate) subcontractors and CPCI contractors. In addition, the Technical Advisory Group and members of the Government/Industry Working Groups can be contributing members. Their comments and criticisms should be solicited and considered in determining the desirability and practicality of the system design and the adequacy and probable accuracy of analysis methods and mathematic representations. The Baseline Review Board would review and approve products at formal reviews.

Functional Design Review

The initial effort for the Development Phase involves system synthesis, analysis, risk assessment and trade-offs to establish a baseline system design. The effort results in the publication of a draft System Specification, draft Acceptance Test Plan, revisions to Type A System Specification and revisions to the Development Plan which will be reviewed at a Functional Design Review by the Baseline Review Board.

System Design Concurrence

The drafts of the System Specification (design) and Acceptance Test Plan that were produced for the Functional Design Review (FDR) will be revised, if necessary, to conform to the results of the FDR. The drafts will be refined and expanded to allocate all of the system requirements to CPCIs, define system inputs and output requirements, and provide the detail to develop each CPCI. The System Design Concurrence by the Baseline Review Board is required to ensure that the final system concept and direction of the system design and test plan meets with the approval of the Government.

Preliminary Design Review

Each subsystem (CPCI) identified and defined within the System Specification will undergo a preliminary subsystem design. Basically, this activity involves the identification of subsystem components, allocation of processing and performance requirements to those components, and preliminary definition of subsystem test and integration procedures. The results of these efforts will be documented in a preliminary Subsystem Specification and a preliminary Subsystem Test and Integration Plan and will be provided to members of the Baseline Review Board for review and approval at a Preliminary Design Review (PDR). A PDR will be held for each CPCI to ensure that the design approach for the CPCI will meet the requirements as defined in the Type A System Specification and the System Specification.

Critical Design Review

The detailed subsystem design occurs after preliminary design approval and before programming for a CPCI. The activity involves further development of the preliminary Subsystem Specification into a detailed Subsystem Specification from which the subsystem will be programmed. A final Subsystem Test and Integration Plan will also be prepared. The results of this activity will be reviewed by the Baseline Review Board at a Critical Design Review. The purpose of this review is to assure the accuracy and adequacy of CPCIs prior to their actual programming and to ensure that the developing system continues to meet all requirements that are placed upon it.

It has been estimated that for the First-Level Release and Second-Level Release, the Baseline Review Board would convene 13 times over a 15-month period and 14 times over a 14-month period, respectively.

Program Production

Quality assurance for program production will be based on the techniques of hierarchical structured concepts, analysis and design walk-throughs, Chapin logic flow charts, pseudo code prologues for programmable modules, standardized FORTRAN coding techniques, four levels of tests and scheduled units of work.

Hierarchical Structured Concepts - The Second-Generation Comprehensive Helicopter Analysis System should be developed utilizing "structured concepts" for design, programming and test. Structured concepts involve the construction of the software in terms of well-defined techniques of top-down reasoning processes for modular structure and modular interfaces.

The software should be structured in a hierarchical manner dependent upon an analysis of the problem, the flow of data through the problem, and the subdivision of the problem into modules that will perform the transformations on that data. This process involves an iterative division, identification and definition of programmable modules to solve the problems. A graphic representation of a module and subordinate modules that depict the top-down reasoning or structural process will result in a hierarchy of modules.

Walk-throughs - Reviews should be conducted at all levels of the effort by the software development team with recommended Government attendance. The walk-throughs are conducted for the purpose of controlling the quality of the analysis, design and coding because: (1) all personnel become knowledgeable of each area; (2) functional requirements can be aligned early in the process; (3) design flaws can be detected early; (4) modifications can be easily achieved; (5) communication can be facilitated and (6) on-going guidance can be received.

The walk-throughs are conducted during periods of analysis to determine the significance, criteria, and desired implementation of each stated functional requirement. These walk-throughs are to provide assurance of the completeness and placement of the functional requirements implementation into the emerging design, alignment of the requirements into functional areas and general modules, and refinement of any generalities of the initial concept. The walk-throughs can be used as the primary method of communication to encourage the exchange of information to expedite the formalized review and approval of specifications.

Design walk-throughs are conducted for the purpose of reviewing developed hierarchy charts which depict the modular structure and accompanying documentation in the form of external specifications, HIPO charts, etc. Design walk-throughs are also conducted to review the logic within a module after development. The logic of the modules will be developed along strict guidelines in top-down control structures known as "Chapin charts."

Chapin Charts - A "Chapin-style" chart is drawn for each module depicted on the hierarchy chart. The hand-drawn Chapin chart will detail the internal logic of the module using simple logic control structures of (1) simple sequence, (2) If Then/Else, (3) Do While, (4) Do Until, and (5) Case or variations of the structures.

The logic flow of the control structures begins at one top entry point and flows to a single bottom exit point. The execution flow within a module is sequential from one logic structure to the next logic structure. Thus, the flow of a module is controlled from the top-down through strict adherence to the Chapin-style charts and the five logic structures. Any kind of processing, any combination of decisions, and any sort of logic can be accommodated with these control

structures or a combination of the control structures. Utilizing these control structures in the top-down design of a module eliminates arbitrary and capricious branching, results in a precise flow of data, and simplifies the modular testing process.

The Chapin charts can contain other attributes of the module for ease of understanding. The chart will contain the title, name, function, input and output data, external effects, parameters, tables, partial code, test data, etc. The chart usually will be drawn on one or two pages. The attributes of the module can be placed on separate pages to produce a module package. The information from the Chapin charts will be delivered in the form of "prologue" and pseudo code.

Pseudo Code - After the design walk-through of the module and approval, the Chapin chart can be converted into an indented logic structure pseudo code which shall constitute listable and deliverable program documentation.

Pseudo code provides a word description of the module which is derived from the Chapin chart and is concerned primarily with the flow of the control structure. The indented pseudo code will act as a bridge between the design and coding phases. It aids in the transformation of the highly graphic, parallel-vision Chapin charts into a top-down, straight-line, final source code for placement into the program listing. It is kept up to date by the programmer as the source code changes.

Guidelines should be contained in the Quality Assurance and Control Plan for the standardized use of the pseudo code.

Coding Techniques - To control quality for ease of understanding, maintaining, and interchanging code, standards should be imposed for documentary comments and specific language statements.

Comment statements are both source code and documentation. Various kinds of descriptive information which normally appear in publishable design and programming documentation should be written as comments in the source code. In this way design and documentary information are brought together and placed in the program source listing. This information is contained in a special module prologue section at the beginning of a module and in regular comment statements interspersed among the executable statements of the module. The prologue section explains the purpose and functioning of the module as well as the flow of control within the module. The interspersed comments are supplementary in nature for additional explanations.

The first part of the prologue contains the attribute information from the Chapin chart, i.e., title, name, function, input, output, parameters, etc. The second part of the prologue contains the converted indented pseudo code from the Chapin chart. The statements in the pseudo code are labeled with statement numbers to appropriately correspond with the statement numbers in the FORTRAN listing. Any deviations from standards are noted in the prologue code and supplementary comments ensure that adequate documentation is contained in the source listing. The Quality Assurance and Control Plan should specify strict guidelines for converting the Chapin charts into indented pseudo code.

Standard FORTRAN (ANSI X3.9 - 1966) should be utilized to provide the source language for the Second-Generation Comprehensive Helicopter Analysis System. FORTRAN, a high-level compiler language, is relatively machine independent. However, ANSI X3.9 - 1966 FORTRAN has not been implemented by the same or different manufacturers to be completely independent. To minimize possible future implementation and conversion problems to next-generation machines and to enhance the control for quality, non-ANSI constructs should not be employed. The "Programming Standards for the Second-Generation Comprehensive Helicopter Analysis System", Fort Eustis, should be followed to maximize machine independence.

Test Level Technique - Testing takes FORTRAN language statements and removes compiler statement errors, modular interface errors, input and output formatting errors, and program structural, logic, and calculation errors. The testing activities for quality assurance should be composed of (1) program module testing, (2) CPCI testing, (3) integration testing, and (4) acceptance testing. Testing activities are discussed in greater detail in the following sections.

Work Units - Quantity of work control can be achieved by utilizing the concept of work units. Each activity in the development effort can be identified and defined as a work unit. The structured concept allows the work unit to be the control element against which progress can be visibly measured with the specification, design, code, and test of each individual module as well as publishable documentation. The work units can be scheduled, resources can be allocated, and a "hierarchical work unit status log" can be maintained. The status log can be updated regularly to reflect status changes of the work units. The status log will provide progress information for project reporting requirements. The items for which status can be reported are:

- a. the scheduled work for each module
- b. the actual work on each module
- c. the number of modules that were scheduled and completed for the activities of specification, design, code, module test, CPCI test and integration test

- d. cumulative totals and percentages
- e. the schedule and accomplishment of each individual person
- f. an estimate of the number of days that the team is ahead or behind schedule
- g. forthcoming activities
- h. past and potential problems

This process of control is especially useful for monitoring, reporting, and directing the production of time-critical configuration items.

TESTING REQUIREMENTS

The testing that must be performed for the Second Generation Comprehensive Helicopter Analysis System should be detailed, comprehensive, and structured to provide the quality that is required to verify the accuracy of the code and adequacy of the design. Tests of the programmed system should begin at the lowest programmable level and continue through successively higher levels in a meaningful test hierarchy. An Acceptance Test Plan should be written for the system during the early stages of the Development Phase to be reviewed at the Functional Design Review. Test and Integration Plans for CPCIs should be written at the time the CPCI is designed to be reviewed at the Preliminary Design Review for the respective CPCI.

Four levels of testing should be utilized as illustrated in Figure 24.

- (1) Module Testing - Upon the completion of the pseudo code, FORTRAN and/or assembly-coded statements for a module, the product is reviewed with the Chapin chart for completeness and accuracy. The product is compiled and all compiler-generated errors are removed to obtain an error-free compilation. Module testing exercises the module through its full range of inputs and outputs and evaluates its performance for any necessary correction. Each and every path, decision, and code of a module is exercised during module testing.
- (2) CPCI Testing - CPCI testing exercises the CPCI to validate that it is correctly interpreting input data, successfully performing its processing tasks, providing arithmetic and logical accuracy, as well as statistics for storage utilization and CPU timing. Each CPCI is tested with a full range of data. The responsibility for the test of the CPCI should be with the developer of the CPCI with coordination from the Test and Integration Coordinator. All CPCI test results will be analyzed by the Test and Integration Coordinator and representatives of the Government with assistance from the Technical Subcontractor for the Second-Level technical contractors.

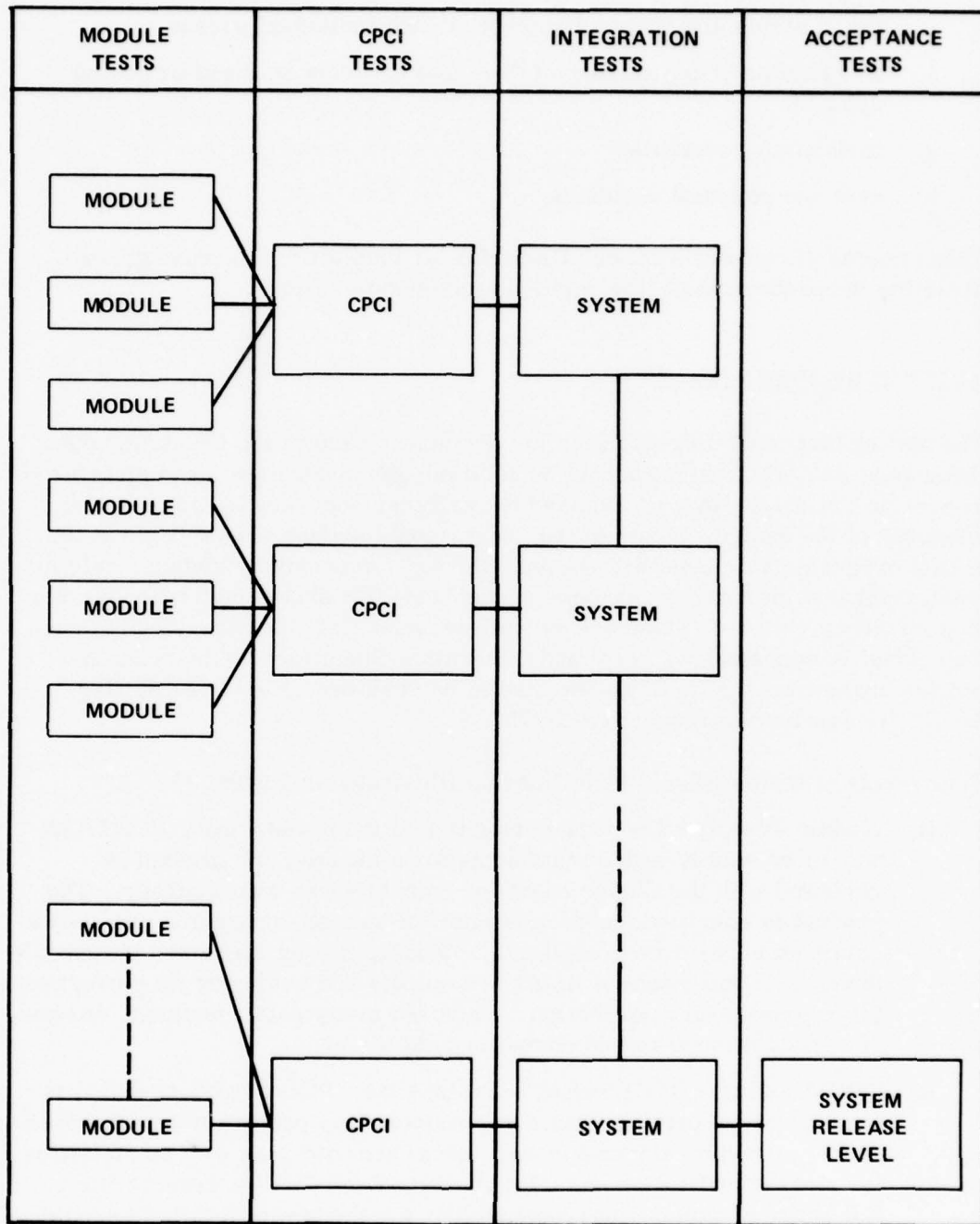


Figure 24. Levels of Tests.

The CPCI test procedures will be defined in a CPCI's Test and Integration Plan that is produced when the CPCI is specified and designed. It is recommended that CPCI tests be stand-alone tests for the CPCIs that are an entity to themselves. However, other CPCIs can be tested in a system test-bed program to utilize the produced system capabilities. This procedure will effect an overlap of CPCI test and CPCI integration and test. For Government acceptance of a contracted CPCI, one method can be specified.

Each CPCI will be specifically tested for arithmetic and logical accuracy and limitations. Test data for CPCI testing is built at the time the CPCI is designed and is specified in a manner that is readily input to the CPCI. The test data for a CPCI will be designed and produced to force the execution of all module invocations, assure the adequacy of a CPCI design, collect storage utilization and CPU timing statistics. Test data for CPCI testing will be utilized for CPCI integration testing wherever possible.

The test data for a CPCI will be deliverable in a form to duplicate or repeat the test and stored in the System test data file. CPCI test results will be analyzed to determine whether the derived results are consistent with the inputs. All tests results will be forwarded to the Test and Integration Coordinator for analysis and approval by him and representatives of the Government.

- (3) *Integration Tests* - The objective of integration test is to add a tested module or CPCI into the system, exercise it as thoroughly as possible, determine the adequacy of analysis for technical CPCI modules upon which the System is based and prove that the CPCI performs all of its processing tasks.

The responsibility for the test of the integration should be with the CPCI developer after CPCI test approval and will be directed, controlled, coordinated or performed by the Test and Integration Coordinator.

The CPCI integration test procedures will be defined in a CPCI's Test and Integration Plan that is produced when the CPCI is specified and designed. Briefly, this plan will require the testing of every functional and performance requirement of the CPCI (including any coupling requirements of technical CPCIs), along with the requirements of already accepted CPCIs. Data used in testing the various CPCIs will be maintained by the Test Coordinator for subsequent integration, acceptance, and installation tests. The Test Coordinator will oversee the testing and will provide a Test Report to the Baseline Review Board. The Baseline Review Board will accept or reject the CPCI. Should the CPCI fail to meet with approval, the Baseline Review Board will define the deficiencies and suggest specific remedial actions.

Integration testing starts with the topmost module in the functional hierarchy. Other modules and/or CPCI's are then added to the program. A module or CPCI is never integrated into the program unless it is subordinate to a previously integrated module or CPCI. Tests are then conducted to exercise the integrated unit. The effort is one of interfacing programmed units and verifying interaction and adequacy rather than retesting the previously tested units. Consequently, the structured integration testing predisposes most system testing.

Test data for integration testing is built at the time that the module or CPCI is designed and is specified in the same manner as the CPCI tests. As the CPCI tests are designed to exercise only a particular module or set of modules, the integration test is designed to exercise the System along the paths to and through the module or CPCI and ensure that all interfaces between CPCI's are tested. If it is not possible to test all of the module or CPCI at the time of integration, the tests will be shown as acceptance tests and identified as such.

The technical CPCI's represent routines with generalized coupling capabilities. When the CPCI's have been individually tested, the ability of the System to perform the proper coupling functions will be tested.

Adequacy of analysis will be determined through correlation with engineering data if available. In addition, a set of component simulations will be executed which isolate the effects of each physical component as much as possible so that a comparison can be made with corresponding data from physical tests; e.g., nonrotating blade shake test, rotating rotor in vacuum, etc. Tests to demonstrate adequacy of analysis will be identified in the CPCI Test and Integration Plan.

- (4) Acceptance Tests - The objective of the acceptance test is to demonstrate and verify that the programmed system operates according to the specifications and is correctly installed. Acceptance testing is the final quality assurance provision for a particular level of the system. It will be performed under Government supervision in accordance with the Acceptance Test Plan.

The Government should have the responsibility to finalize requirements for, and sponsor acquisition of, additional experimental data necessary to determine further CPCI and System level adequacy and accuracy if the test cases from CPCI tests and integration tests are considered inadequate. The test for acceptance will be coordinated by the Test and Integration Coordinator.

The definition of acceptance test cases should begin during the production of system specifications. This definition will develop the cases to test the specified functional correctness and accuracy.

Tests to demonstrate the adequacy of the analysis will be identified in the Acceptance Test Plan. The definition will effect the certainty that all specified requirements are testable and are therefore usable. An untestable area of the program can be considered an ill-designed area subject to the production of system errors.

Due to the concept of structured programming and subordinate module integration, as the last module and CPCI are integrated and tested, the complete program will, theoretically, have been tested. Acceptance test cases will begin at the time of CPCI development in a manner similar to the integration test cases. In actuality, most of the test cases used in integration tests will be used for acceptance tests. Acceptance tests should include a random sampling of the physical systems described by the Detail Functional Capabilities in the Type A System Specification. Data from physical tests and existing analysis program should be specified and available for correlation with the tests from the Second-Generation Comprehensive Helicopter Analysis System.

Error Resolution

A Program Trouble Report (PTR) form can be generated by the Development Contractor to be used to report programmed and documentation errors within the areas of subcontractor responsibilities, CPCI integration and program deliveries. The PTR forms can be utilized either directly or indirectly by users to report all errors. The PTR identification and number sequence would be controlled by the configuration control personnel. The PTR can be the formalized error reporting, correction, and dispensation vehicle as all PTRs would be answered.

Program Trouble Reports would contain at a minimum the Configuration Item, release, version, reporting agency, date, type of error, error classification, effects, reproducible, degradation, etc.

The maintenance of programs for system correction becomes standardized during system testing for the future maintenance effort.

A corrected version of the baseline product release can be distributed based upon the type and classification of the error(s). System failure errors would have precedence for maintenance, and corrected versions would be distributed as soon as possible. Corrections to less critical errors would be made on a periodical or numerical basis.

TRAINING

Government Internal Training

Training for Government personnel must be complete, structured, and formalized and encompass concepts through usage. This training will prepare the Government to assume maintenance of the system and to provide subsequent training to users. Documentation on the system (i. e., Users Manual, Maintenance Manual and Theoretical Manual) will be provided such that training for the Government and for users can be accomplished by reading, since one of the main objectives of the system is simple usage. However, it is often more cost effective to provide classroom training for this type of system. The types of training are discussed below.

Understanding the System Concept - This training provides an overview of the system concepts and the functions of the CPCIs (both executive and technical). System usage is discussed as well as examples and theory of problems that the system can be used to solve. Direction on more detailed information on all portions of the system will be provided. Appropriate attendees are technical management, senior designers and engineers, and potential users of the system.

Module and Structured Concepts - The purpose of this training is to provide the attendee with an understanding of how a system and a program are developed using the modern structured techniques. System modules and hierarchical structure will be presented as background for understanding the system as well as modifying or adding to the system. Emphasis will be placed on the design and structure of programmable CPCIs to be added to the system. Appropriate attendees would be senior designers and engineers, senior programmers and implementers of technical CPCIs for the system. This training would be provided during the design phase for CPCI developers and continued for the life of the system.

System Installation - System Installation Training will provide the Government personnel with the knowledge required to install the system onto different host computers. The software contractor will be responsible for initial installations of the system. This training will permit the Government personnel to become familiar with the deck and tape files required for system installation. It is suggested that the Government personnel assist the software contractor in system installation so that actual experience, as well as theory, will be gained. Subjects for system installation training include technique and language for generation of files for installation, file requirements (e. g., source and binary), installation decks, test decks, verification techniques.

Modifying the Software System - The emphasis in this training will be modification for the purpose of adding or changing technical CPCIs. However, modification techniques for changing or adding to the executive portion of the system will also be provided. Training will include module and system interfaces, interface techniques, data base concepts, data base file content and interface, usage of utilities, and library maintenance. Emphasis is placed on the "mechanical" aspects of system modification rather than the conceptual. Appropriate attendees are those personnel who are expecting to modify or add to the system. This training would be provided during the programming phase for CPI developers and continued for the life of the system.

System Usage - This training provides the potential system user with the knowledge required to enter the system, process data, checkpoint if required, and evaluate output results. The training will be slightly different for the various releases of the system and will differ in emphasis depending on whether the attendees were oriented to batch or interactive usage of the system. Control statement usage will be discussed in detail. The training would be provided near the first release of the system and continued for the life of the system.

User Training

Normally, the user community will need only the system usage training described in the above paragraph for modifying the software system. However, as the system gains wide acceptance and usage, many users may want to modify the system (at their own risk) and keep a standard copy of the system at their site as well as the modified version. When this is the case, all the training will be of value. Consideration should be given to establishing this training on a periodic basis for the life of the system.

Interactive Aids for Tutoring

The interactive version of the system will have the capability to guide the user in the use of the system. For example, the user can enter a control statement and the parameters required for the statement, or he can enter a segment of the control statement and the system will tutorially guide the user in its operation. The interactive capability will provide the user with descriptive information about the operation of system control statements. In addition, commands will be provided that display descriptive information about technical modules, helicopter modules, and problem descriptions contained in data base files to help the user learn how to work with the system.

RISK ASSESSMENTS

DEVELOPMENTAL

Any system, large or small, always has the potential for developmental risk. A system such as the Second-Generation Comprehensive Helicopter Analysis System (SGCHAS) will have a potential for developmental risk for the following reasons:

- a. Underestimating and scheduling
- b. Excessively rigorous development schedule
- c. Inadequate specifications at all levels
- d. Inadequate communication and interface
 1. Government/Development Contractor
 2. Development Contractor/Technical Subcontractor
 3. Government/CPCI Contractors/Development Contractor
 4. Designers/Programmers
- e. Integration of contracted CPCIs
- f. Inadequate tests and analysis of test results
- g. Failure to identify critical areas (e.g., Data Management) which require special design and consideration for completion to interface with system
- h. Failure to follow development standards (i. e., engineers may not be as disciplined to follow standards for programming as software development persons may be)
- i. Multiple developmental agencies
- j. Geographical location of development agencies

The Baseline Development Plan (Reference 12) that has been presented defines, in detail, procedures and standards for development of the SGCHAS, and alleviates the potential risks. This Baseline Development Plan was written from experience gained from developing other systems similar to the SGCHAS. If the Government elects to employ this Baseline Development Plan for the SGCHAS, and administers it with little deviation, there will be little, if any, development risk involved with the development of the SGCHAS.

¹² Control Data Corporation; Baseline Development Plan for the Second Generation Comprehensive Helicopter Analysis System (in response to Task IIIa, CDRL A009, contract DAAJ02-77-C-0058), Control Data Corporation, Hampton, Virginia 23666, and Kaman Aerospace Corporation, Bloomfield, Connecticut 08002; January 27, 1978.

HARDWARE

Hardware will only be a potential risk whenever central processing units of different computer manufacturers are involved. Peripherals do not appear to be a potential risk except in the areas of graphics and plotting. Input and output peripherals should not present a risk because advances in technology should support the current recording and access techniques; however, to fully realize the capabilities of new technology it may become necessary to perform a data set conversion from the current device to the new device. This conversion should be performed using utility software provided by the host operating system and should not require a significant amount of computer time.

Whenever a using installation upgrades its central processing unit there could be some potential risk in that the new software may not fully support the SGCHAS as written for the current central processing unit (this will be discussed in more detail under SOFTWARE).

The SGCHAS design concept presented herein will prevent costly reprogramming to keep the System in step with hardware advances.

SOFTWARE

Software risks are hard to assess and predict whenever a system has a life expectancy of the SGCHAS. Software developed for the SGCHAS should be expected to have a long life, depending on the advancements in computer software technology during the life of the System. Regardless of software advancements, it is expected that the SGCHAS software will be upward compatible. However, this is not necessarily true and some reprogramming to maintain compatibility and employ new technology should be anticipated and planned for at the one-quarter stage of the life expectancy.

Software to run on multiple computer manufacturer's central processing units is not within the capability of the System design concept. However, the design concept does isolate noncompatibilities into the Executive area. Therefore, with the modular concept, a minimum programming effort is needed to make the System operational on multiple vendor computers.

To summarize, software risks are to be expected during the life (15 years) of the SGCHAS. The System design is one that has allowed for this type of risk and, if employed, will prove to be very cost effective in the development and maintainability of software.

TECHNICAL RISK ASSESSMENT

The risks related to advanced technology development and application involves questions of adequacy, cost of development, cost of use and overall cost effectiveness. These considerations are discussed below.

Adequacy of Advanced Technology Development

A list of technological areas where present methods are inadequate or unproven may be readily formulated by anyone familiar with the field. A small sampling of the more important items in such a list would include: hub structures with complex load paths, vibratory response of fuselage, interior noise prediction, aerodynamic interference effects, aerodynamic wake and dynamic stall.

There is little doubt that industrial, government, and university researchers will make significant advances in these areas during the time period of the Development Phase of the SGCHAS. The important question to be asked is not: "Will the SGCHAS development lead to the complete solution to all these technological problems?" (The answer is, of course, "no".) A more appropriate question should be, "Will greater advances be made in these fields if the SGCHAS is developed?"

The answer to the second question is assuredly "yes", providing that the System developed has technical characteristics identical to those presented in this report. The pertinent features of this concept in this regard are as follows:

- a. The analysis of each component is independent of the analysis methods used for other components.
- b. The method of coupling component analyses is exact.
- c. Users may conveniently incorporate their own methods into the system on a temporary or permanent basis.
- d. A particular component representation may be executed either as a single component or coupled to any combination of other components.
- e. The System operates in an efficient manner in regard to both cost and computer resources.

Such a System will allow the method developer to test a new approach in an environment which will allow the evaluation of the method as a single independent analysis and to gradually progress to the point of comparing the changes in solution to a complete problem, with the full confidence that these changes are due solely to the new analysis method.

Cost Effectiveness of Advanced Technology Methods

There are two aspects to the cost effectiveness of advanced technology methods. The first is whether it is cost effective to devote the effort required to develop a particular analysis method. This, in itself, is not a question which can be answered except through the use of sound engineering judgement based on a thorough familiarity with the principles involved and experience regarding deficiencies in the correlation of prediction and experiment.

Once an advanced analysis has been developed, the question arises as to whether it is cost effective to use this method if its use results in greater costs. The System capabilities described in the previous section can be used to give the user and developer specific information regarding the impact of the new analysis on a complete problem solution. It is possible and convenient to carry out the solution to a given problem two or more times, where the solutions are identical in every way except that one uses the new analysis and one uses the old. Comparison of the results and costs by an experienced engineer will allow the establishment of a set of guidelines regarding the appropriate uses of each of the advanced methods.

This capability will have a considerable impact on the cost of routine analyses, since it will be possible to include only the level of complexity which is actually needed. No existing program has such a capability and it is presently impossible to ascertain if a particular analysis is being performed with an inadequate level of detail or is being performed with an unnecessarily complex and costly method.

Operational Costs

In addition to the cost considerations discussed above, which are dependent on the user's choice of individual component levels of representation, there are several other observations to be made related to the cost of operation and the main storage resources required as discussed below.

Sizes of Programs - The main memory required for the program code for the SGCHAS is anticipated to be significantly less than that required by a current state-of-the-art program having the same capabilities. The principal reason for this is the required division of the technical modules into the Coefficient and Active functional modules. The Coefficient modules associated with each component are executed only once and are not retained in the core. The Active modules contain only that portion of the code which is actually required to remain in main storage during the problem solution.

Current state of the art programs do not, in general, rigidly maintain this division of code and it is not uncommon for reasonably large regions of main storage to be occupied by code which is used only once. The Coefficient/Active module concept results in a minimization of core requirements for the executable code.

Storage of Matrices - The core storage required for the storage of transformation and coefficient matrices is worthy of discussion.

The transformation matrices derived and discussed previously consist mainly of null and unit elements. As previously discussed, it is not intended to actually occupy storage areas with these matrices. It is quite possible and feasible to store this information in a very concise and efficient manner. The specific algorithms are not presented here; however, a quite similar problem has been implemented in the CHIANTI program (see Reference 10 in this document) recently developed at Kaman Aerospace Corporation and has resulted in a very efficient process in terms of core requirements and execution times.

One aspect of the matrix formulation of the differential equations requires further consideration. Each blade in the more advanced rotor representations is represented by a set of differential equations. When the blades are identical the sets of constant coefficients are identical from blade to blade, even though the forcing and interfaces with the other components will be dependent on the respective azimuth positions. At present, the sets of identical coefficients are stored separately and even though each rotor blade has its own differential equations, a reduction in core storage (but not processing time) would be achieved if this effect were taken into account. During the Development Phase an analysis will be performed to evaluate the potential reduction in core requirements as compared with a slightly increased complexity in code.

Inversion of Changing Mass Matrices - One of the most time-consuming analytical processes is the inversion of matrices. When the mass matrix is a function of time, no method of solving differential equations can avoid the necessity of repetitive mass matrix inversions. It is important that the user only make use of technical CPCIs with time-dependent mass matrices when they are warranted in light of the problem being solved. It is important that the inversion algorithm(s) selected for inclusion in the System be the most efficient possible. One other aspect of this problem must be considered: It will be a relatively common occurrence that only a small portion of the coupled system mass matrix varies with time. Thus, the problem may often be of the following form: Find $(M + \Delta M)^{-1}$ when M^{-1} is known. There are algorithms available which are exact or approximate for solving this problem. The appropriate algorithms must be determined and implemented during the Development Phase.

REFERENCES

1. Control Data Corporation; Baseline Type A System Specification for the Second Generation Comprehensive Helicopter Analysis System (in response to Task IIIa, CDRL A008, contract DAAJ02-77-C-0058), Control Data Corporation, Hampton, Virginia 23666, and Kaman Aerospace Corporation, Bloomfield, Connecticut 08002; January 27, 1978.
2. Anon.; Statement of Work, Predesign of the Second Generation Comprehensive Helicopter Analysis System, DAAJ02-77-C-0058, Section F, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia 23604; September 8, 1977.
3. Hurty, W. C., "Dynamic Analysis of Structural Systems by Component Mode Synthesis," Jet Propulsion Lab., Technical Report 32-530, Jan. 15, 1964.
4. Benfield, W. A. and Hrudá, R. F., "Vibration Analysis of Structures by Component Mode Substitution," AIAA Journal, Vol. 9, No. 7, July 1971.
5. Goldman, R. L., "Vibration Analysis by Dynamic Partitioning," AIAA J. 7 (6), 1152, 1154, 1969.
6. Hou, Shou-nien, "Review of Modal Synthesis Techniques and a New Approach," Shock and Vibration Bulletin No. 40, Dec. 1969.
7. Benfield, W. A., Bodley, C. S., and Morosow, G., "Modal Synthesis Methods," Presented at the Space Shuttle Dynamics and Aeroelasticity Working Group Symposium on Substructuring, Marshall Space Flight Center, Alabama, Aug. 30-31, 1972.
8. Hurty, W. C., Collins, J. D., Hart, G. C., "Dynamic Analysis of Large Structures by Modal Synthesis Techniques," Computers and Structures 1 (4), 535-563, Dec. 1971.
9. Przemieniecki, J. S., Berke, L., "Digital Computer Program for the Analysis of Aerospace Structures by the Matrix Displacement Method," AFDL report No. FDLTDR64-18, April 1965.
10. Berman, A., "Vibration Analysis of Structural Systems Using Virtual Substructures," The Shock and Vibration Bulletin 43, NRL, Washington, D.C., June 1973.

REFERENCES (Continued)

11. Berman, A., Giasante, N., "CHIANTI - Computer Programs for Parametric Variations in Dynamic Substructure Analysis," The Shock and Vibration Bulletin 47, NRL, Washington, D. C., Sept. 1977.
12. Control Data Corporation; Baseline Development Plan for the Second Generation Comprehensive Helicopter Analysis System (in response to Task IIIa, CDRL A009, contract DAAJ02-77-C-0058), Control Data Corporation, Hampton, Virginia 23666, and Kaman Aerospace Corporation, Bloomfield, Connecticut 08002; January 27, 1978.

GLOSSARY OF TERMS

Active Module	One of three types of functional modules defined for a technical module. It is used in differential problems to compute the highest derivative vector in equations given all the lower derivatives.
Application Executive	The part of the SGCHAS which performs all support processing for the system, including data base and run data management, and the assembly, setup, and execution of the technical processing.
Chapin Chart	A hand-drawn chart displaying the internal logic of a program module using simple logic control structures of (1) Simple Sequence, (2) If Then/Else, (3) Do While, (4) Do Until, and (5) Case or variations of the Structures.
Coefficient Module	One of three types of functional modules defined for a technical module. It is used in eigensolution and differential problems to compute the constant matrix coefficients and other coefficient data.
Component	A part of a helicopter that has been identified for analysis within the System (i. e., rotor, engine, drive, controls, fuselage, etc.).
CPCI	A subprogram or a group of functionally related modules that is necessary to provide the functional capabilities and technical characteristics defined by the Type A System Specification and which will be developed by the Development Phase Contractor or subcontractors, will be furnished by the Government, or will be procured from a software vendor.
CPCI Component	A major functional subdivision of a Computer Program Configuration Item (CPCI).
CPCI Testing	A testing procedure for a CPCI to prove that it interprets its input correctly and performs all its processing tasks before it is integrated into the System.

GLOSSARY OF TERMS (Continued)

Data Manager	A program component that will manage all data used for the System to include input/output operations, internal core storage and external recording of data.
Definition Module	See Technical Module Definition.
Diagnostic File	A data set containing all syntax and diagnostic messages used in the System. The message will have a count field to record the usages.
Dynamic Loader	A program that will permit dynamic loading of executable code during the actual execution of the System.
External Model Functional Capability (EMFC)	The ability of the System to provide analysis results for input to other computer programs or computer simulations.
Functional Module	One of the four components that may make up a technical module. Functional module types are: Active Module, Coefficient Module, Definition Module, and Processing Module. Two or three types of functional modules are required to form a technical module.
General Functional Capability (GFC)	The ability of the System to model any arbitrary helicopter configuration.
Helicopter Model (also: Helicopter Analysis Model)	A user-specified rotorcraft and other analysis component configuration which is to be analyzed by the System.
Helicopter Model Definition (HMD)	A special record format used by the engineer to describe an arbitrary rotorcraft configuration to the SGCHAS for subsequent analysis.
Host Operating System	The operating system that controls program initiation/termination and general utilities for a particular computer central processing unit (i. e., OS, NOS, etc.).

GLOSSARY OF TERMS (Continued)

Master Data Base	A file containing physical rotorcraft characteristics that will be provided by the Government for use at all System user sites.
Module Testing	Testing performed by the programmer while developing a module.
Particular Functional Capability (PFC)	The ability of the System to provide predefined standard procedures to analyze particular rotorcraft configurations.
Primary Storage	Internal core storage.
Programmable Module	The definitions provided by the Government in the Language-Independent Programming Standards for Modular Characteristics will apply to a Programmable Module when used in the Control Data/Kaman System reference material.
Prologue	A group of comment cards that usually precede the module code, written in English-like statements that describe in detail the module data sets, module interfaces, methods, and other pertinent data.
Pseudo Code	English phrases derived from a Chapin Chart which describes the flow of the program module. The code provides a bridge between the design and the coding phases.
Second Storage	Storage for data that is external to central processing unit.
Sequence Control Table	A table built from user inputs (SCL) which is used to direct the logical sequence of system operation.
Sequential Module	A stand-alone technical module composed of processing and definition modules.
Stand-Alone	Capable of independent operation within the System.

GLOSSARY OF TERMS (Continued)

Stored Procedure, Stored Procedure Definition (SPD)	A set of SCL statements which direct problem solution and are stored in the data base for later recall and execution. (See PFC)
Structured Walk-Through	Reviews conducted at all levels of design, analysis, and programming by the software development team with Government attendance. Walk-through, control quality of the design and coding.
System Acceptance Testing	Testing for final quality assurance performed under Government supervision in accordance with the Acceptance Test Plan prepared during the Design Subphase.
System Accounting	Recording of System run statistics (core used, time used, I/O counts, etc.).
System Control Language (SCL)	A simple language through which the user will supply data to the System and direct the sequence of System operation.
Technical Module	A program entity composed of two or three functional modules which performs a specific analysis function. There are four types of technical modules: Differential Equation, Eigenproblem, Sequential, and Criteria (see "Technical Components and Relationships" in this report).
Technical Subroutine	A program entity used by Technical Modules to perform special analysis or utility functions such as airmass or engine performance analysis or matrix operations.
User Data Base	A set of one or more files on which the System stores a user's data and from which data is retrieved during system operation.

DEPARTMENT OF THE ARMY

Applied Technology Laboratory
U.S. Army Research and Technology
Laboratories (AVRADCOM)
DAVDL-EU-TSD
Fort Eustis, Virginia 23604

OFFICIAL BUSINESS

PENALTY FOR PRIVATE USE, \$300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD-314



THIRD CLASS