AFHRL-TR-74-87

AIR FORCE HUMAN RESOURCES LABORATORY

COMPUTER-AIDED TECHNIQUES FOR PROVIDING OPERATOR PERFORMANCE MEASURES

By

Edward M. Connelly
Francis J. Bourne
Diane G. Loental
Quest Research Corporation
6845 Elm Street
McLean, Virginia 22101

Patricia A. Knoop

ADVANCED SYSTEMS DIVISION
Wright-Patterson Air Force Base, Ohio 45433

12

December 1974
Final Report for Period July 1972 – August 1974

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235

NOTICE

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the 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.

This final report was submitted by Quest Research Corporation, 6845 Elm Street, McLean, Virginia 22101, under contract F33615-72-C-2094, project 6114, with Advanced Systems Division, Air Force Human Resources Laboratory (AFSC), Wright-Patterson Air Force Base, Ohio 45433. Miss Patricia A. Knoop, Advanced Systems Division, was the project engineer and contract monitor.

This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DoDD 5230.9. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

GORDON A. ECKSTRAND, Director
Advanced Systems Division

Approved for publication.

HAROLD E. FISCHER, Colonel, USAF
Commander

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(12) 84p.

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (18) AFHRL-TR-74-87	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) (6) COMPUTER-AIDED TECHNIQUES FOR PROVIDING OPERATOR PERFORMANCE MEASURES,	5. TYPE OF REPORT & PERIOD COVERED (9) Final rept. Jul [redacted] 72 – Aug [redacted] 74
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Edward M. Connelly, Patricia A. Knoop, Francis J. Bourne, Diane G. Loental	8. CONTRACT OR GRANT NUMBER(s) (15) F33615-72-C-2094 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Quest Research Corporation, 6845 Elm St., McLean, Virginia 22101 (16) AF-6114	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ~~62703F~~ (17) 61141 2 [redacted]
11. CONTROLLING OFFICE NAME AND ADDRESS Hq Air Force Human Resources Laboratory (AFSC), Brooks Air Force Base, Texas 78235	12. REPORT DATE (11) December 1974
	13. NUMBER OF PAGES 84
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Advanced Systems Division, Air Force Human Resources Laboratory, Wright-Patterson Air Force Base, Ohio 45433	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

performance measurement
flying training
adaptive training
multiple regression algorithms
performance measurement processor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents the theory, structure, and implementation of a performance measurement processor (written in FORTRAN IV) that can accept performance demonstration data representing various levels of operator's skill and, under user control, analyze data to provide candidate performance measures and validation test results. The processor accepts two types of information: (1) Sample performance data on magnetic tape, and (2) User information reflecting knowledge about features of the performance that are considered to be important to measurement. The sample performance data input is smoothed by the processor in order to remove or reduce noise factors in accordance with information provided by the user. Criterion performance functions are, optionally, provided by the user or are computed by the processor using skilled performers' data. The processor then develops a cont on p 1473B

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

389 460

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20 Continued:
(fr p 1473A)

discrete representation of the continuous performance data based on observed deviations from the criterion functions. This discrete representation, in turn, is used to model each performance using state-space techniques. The processor operates on the state-space model to compute vectors which form generators of various conceivable measure spaces. Candidate performance measures are then generated by operating on the vectors with multiple regression algorithms. Empirical validation tests of several types are applied to the candidate measures for assessment of their validity-likelihood.

The processor can be applied to measurement problems where the human operator working with his equipment obtains demonstrations of various levels of performance. These potential applications include those situations where criterion performance cannot be quantitatively predefined and/or the existing definitions are ambiguous.

Demonstration of some portions of the processor was accomplished using limited flight demonstration data from an instrumented T-37B aircraft for five undergraduate pilot training (UPT) maneuvers: (1) Barrel Roll, (2) Lazy 8, (3) Cloverleaf, (4) Split S, and (5) Normal Landing.

1473B

SUMMARY

Problem

The problem was to develop and implement standardized techniques for deriving and validating measures of operator performance. Traditional techniques involve hand-selecting measures which appear to have content validity, then testing the measures against other validation criteria using operator performance data. This usually results in a resource-consuming iterative research process that is often unsuccessful, because: (1) it is never known at the onset whether or not the most useful measures have been overlooked, (2) the number and potential validity of measures investigated are limited by and vary with the researcher's ingenuity and the time he has available for the study, and (3) the research process and all associated manual effort must be repeated for each new measurement task.

Approach

The approach was to develop and implement computer-aided techniques for deriving and validating operator performance measures. A "universal" set of potential measures was defined which possesses characteristics encompassing many traditionally selected measures. The set also inherently contains a myriad of other measures whose characteristics render them reasonable candidates. Vectors were then identified which constitute generators for the set of measures (i.e., the vectors span the defined measure space). Computational algorithms were developed which generate and operate on the constituent vectors using multiple regression techniques. Several empirical validation methods were developed for testing candidate measures thereby generated. All techniques were implemented in a computer-aided measurement processor which: (1) accepts sample performance data and various user inputs, and (2) generates and tests candidate measures, computes' statistics for assessing their validity likelihood, and prints results for user analysis.

Results

The developed measurement processor was successfully implemented on a Sigma 5 computer. Demonstrations of the operation of the software were performed using a limited amount of pilot performance data recorded on a T-37B aircraft. The processor performed necessary data smoothing, automatically segmented the flight maneuvers for measurement, and developed criterion functions from the skilled operator data provided. Actual generation and validation of measures was not demonstrable due to nonavailability of originally anticipated data. However. correct software performance of all parts of the processor was verified.

Conclusions

The theoretical concepts and computational techniques underlying the developed measurement processor are unique and have great potential for operator performance measurement research. The applied concept of developing a set of vectors which span a conceived measure space and operating on it with regression techniques to generate candidate measures is itself suggestive of a new and extremely powerful measurement tool. The processor operation can be largely independent of user intervention; however, it is also capable of accepting user inputs reflecting his knowledge about specific measurement problems. It represents a truly interactive research system wherein user tasks as distinguished from processor tasks are logically defined, and the outcomes of each are integrated.

Evaluation of the adequacy of the spanned measure set, the generating vectors, and the computational mechanisms for generating and testing measures could not be performed as originally planned due to nontechnical problems which prevented the collection of required data. This was extremely detrimental to the study because: (1) many of the techniques could not even receive preliminary test prior to their incorporation in the processor, and (2) the contributions made by this study to the general technology can only be suggested instead of exemplified.

Follow-up research should include derivation of the basis of the defined measure set using the implemented processor as an aid to empirical studies. This is, in essence, the real crux of the operator performance measurement problem.

PREFACE

This study was initiated by the Advanced Systems Division, Air Force Human Resources Laboratory (AFSC), Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. The research was conducted by Quest Research Corporation, McLean, Virginia, under Contract F33615-72-C-2094. The work is in support of Project 6114, Simulation Techniques for Aerospace Crew Training, with Mr. Carl F. McNulty serving as Project Scientist; and Task 611412, Automated Simulator Instruction and Performance Evaluation in Air Force Training, with Miss. Pat Knoop as Task Scientist.

Quest's principal investigator was Mr. E. M. Connelly. Mr. F. J. Bourne was chief programmer and Mrs. Diane G. Loental contributed significantly to the analysis and documentation work on the program. Other Quest personnel contributing to this program were Mr. D. E. Gausvik, Mr. J. E. Whelchel, Jr., and Mrs. Joanne C. Oliver.

Miss. Patricia A. Knoop was Project Engineer for the Air Force and participated in the program in an active and significant manner. The work performed by Mr. Steve Hogue, formerly of the Advanced Systems Division, is gratefully acknowledged. Mr. Hogue developed and performed initial tests of the applied data smoothing techniques.

The authors thank the personnel of the Resources and Instrumentation Branch, Advanced Systems Division, for their support during the preliminary data collection and software implementation phases of this effort. This Branch operates the Simulation and Training Advanced Research System (STARS) facility on which the extensive software developed in this study was debugged and implemented and without which much of the work relying on preliminary data calibration and smoothing could not have been performed. In particular, the authors thank Mr. Robert Cameron for his effective management of the facility and procurement of special purpose data calibration hardware, Mr. William Schelker for his professional interface and consultation with contractor personnel on systems analysis problems, and frequent long hours devoted to their solution; and Mr. Robert Roettele for his technical support of the STARS hardware during the debug and implementation phases of the program.

Finally, the authors sincerely thank Mr. William Welde of the Personnel and Training Requirements Branch, Advanced Systems Division, for his dedicated and capable management of the aircraft instrumentation and data collection effort designed in part to support this program. It is putting it mildly to say it is unfortunate that the bulk of the data collection effort did not materialize as planned. However, the cause was nontechnical, and the development work performed by Mr. Welde is commendable for having provided at least some preliminary data for use in this effort, and a reliable, first-of-it's-kind technique for objective in-flight data acquisition in Air Force undergraduate pilot training.

This report documents research work performed from July 1972 to August 1974.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

List of Illustrations (*Continued*)

LIST OF TABLES

List of Tables (*Continued*)

COMPUTER-AIDED TECHNIQUES FOR PROVIDING OPERATOR PERFORMANCE MEASURES

I. INTRODUCTION

This report describes underlying theoretical concepts and computer-implemented techniques for deriving valid and objective operator performance measures. The original impetus for the work came from requirements for measures of pilot performance; however the techniques and de.elopment-concepts are equally applicable to general assessment of operator performance on continuous control tasks. Therefore, the basic mathematical and computer techniques will be described in a general context, while (limited) example data are presented for selected pilot performance tasks.

A Tale to Illustrate Basic Concepts

Suppose we are faced with the relatively simple task of deriving and validating measures of performance on the terminal portions of a ground controlled approach (GCA). Let us further simplify the problem by restricting it to measurement of the pilot's ability to maintain proper altitude during descent to the runway. Following a typically employed course of action, we might begin by specifying an intuitive (accepted) notion of "ideal" performance and perhaps sketch some hypothetical performance profiles, such as those shown in Figure 1.

The next step would typically be to identify candidate measures which, singly or in combination, and in whole or in part, we expect will solve the problem. Thus, we might logically pick: (1) RMS Glideslope deviation, (2) Maximum glideslope deviation, and (3) Time in tolerance (glideslope $\pm \Delta$) as measures to compute and examine for validity.

We would then probably perform some initial data collection, and study the behavior of these selected measures for various pilots (perhaps some novice and some at various other stages in the range from novice to experienced). We might discover (again assuming a typical case) that one of the selected measures tends to discriminate between some of the novice and highly experienced performers, but not in all cases; and that none of the measures say anything conclusive or consistent about performers whose experience level (and/or subjectively judged skill level) lies between the two extremes.

"Aha!" says our colleague. "The reason your RMS doesn't work well is because glideslope deviations close to the ground are more critical than deviations at higher altitudes. You need to take altitude into consideration and weight the deviations accordingly."

"And I know why maximum deviation didn't work out," says another. "The deviation doesn't matter as much if it is above the glideslope as if it is below. You should take deviation direction into account."

"Your time in tolerance looks like it might be OK if you would just change the tolerance value to be more in line with the way our good pilots actually perform. And maybe the tolerance should be variable – perhaps a function of altitude – because tighter control is critical as you near the threshold."

Well, no wonder things looked so bad on the initial study! As a result of this first iteration, we might be well advised to plot some of the actual performances and reconsider the problem altogether. We could discover, for instance, that one thing unique about the 5 least experienced pilots for whom we have some data is that they oscillate about the glideslope considerably more than the skilled performers do. (Maybe the number of glideslope crossings would be a good measure!) We might also observe that the more experienced pilots, when they do deviate significantly from the glideslope, make very gradual corrections, whereas the novice performers tend to correct more rapidly, and they often overshoot. (Maybe rate of error correction would work!) Finally, we may see that the good pilots (except for 3) never descend below the glideslope, even though deviations above it are sometimes rather large. (Maybe whether or not descents below the glideslope occur at all would provide at least part of the answer .... or might this just be a characteristic of a cautious pilot?)

At this point, our original list of 3 potential measures has tripled. We have now identified the following 9 measures for investigation:

1. RMS glideslope deviation – unweighted

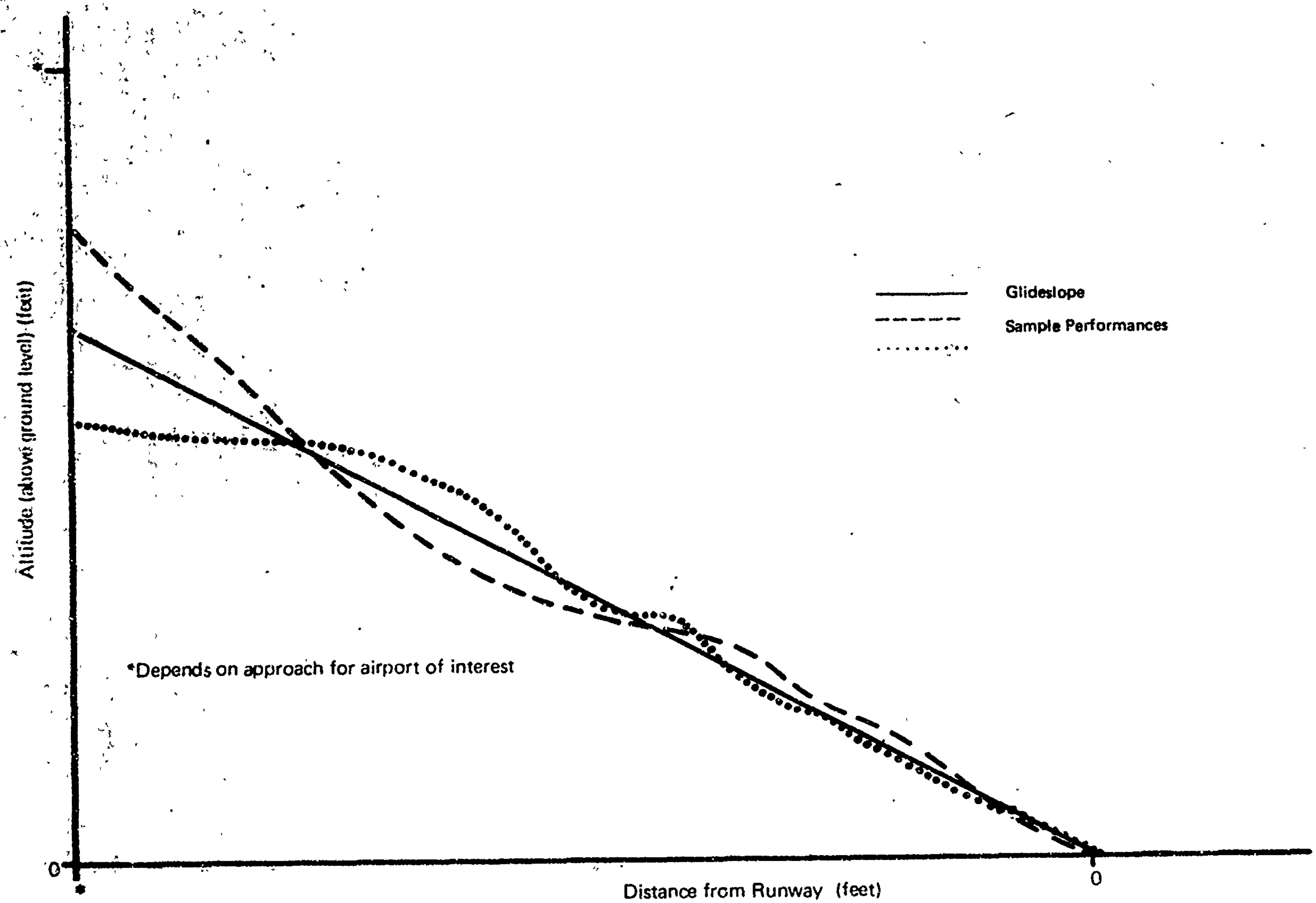


Figure 1. **Hypothetical performances**

2. RMS glideslope deviation – weighted by altitude
3. Maximum glideslope deviation – unweighted
4. Maximum glideslope deviation – weighted by direction of deviation
5. Time in tolerance (Tol. = ± Δ)
6. Time in tolerance (Tol. = f (altitude))
7. No. of glideslope crossings
8. Rate of error correction
9. Whether or not descents below the glideslope occur

Iterations 2 through K of the study would be similar in nature to that described above, and typically, the number of candidate measures would vary and grow multiplicatively in direct proportion to the number of iterations we are able or willing to conduct. The concluding actions of the study, again if typical, would be one or more of the following:

(a) Documentation of the work performed and a recommendation that further study be conducted.

(b) Determined selection and use of a few of the best-looking measures, with reluctant acceptance of the fact that they lack sensitivity and reliability (but by golly they have to work because they have content validity!)

(c) Reconsider whether we really need performance measurement techniques at all.

(d) Use some other method of assessing performance, perhaps one that seemed to work OK in the 1940's or '50's (although if it really did work, there would have been no need for this ground controlled approach (GCA) study in the first place!).

The purpose of the preceding tale was twofold. First (although certainly not enlightening to the readers experienced in this area), it illustrates on a comprehensible scale some of the complexities and problems inherent in measurement work, at least as it is commonly approached. To put it simply, the researcher is faced with assessing the performance of the most incredibly complex "black box" conceivable, and many times without even the benefit of knowing the standards that should be expected of it as distinguished from those that appear, in practice, to be expected of it. (Certainly, much progress has been made in rendering the human black box white; however considerably more is required before measurement of human performance on real-world tasks can be considered straightforward.)

The second purpose was to lay the groundwork for a description of the concepts underlying the work reported herein.

Why attempt to identify and laboriously investigate a few hand-selected candidate measures, repeating the process for each new measurement problem, and never knowing whether or not the measures most suitable have been simply overlooked or unconceived? Why not, instead, define a "universal" measure-set which encompasses at least the characteristics represented by the so-called classical measures (and then some), and assign to computers the task which are logically theirs; i.e., information search and retrieval? In other terms, what is suggested is that a measure-set be designed which is, in effect, inclusive of measures we typically select for investigation, and, moreover, contains the power to generate a myriad of other potential measures which have either not yet been conceived and/or are too numerous to list for purposes of hand-selecting those that seem appealing. This is feasible, and the measures in such a set are reasonable to investigate if the characteristics of the set are defined rationally.

II. APPROACH

The approach is to develop a trial measure-set encompassing characteristics common to many of the classical measures; and to develop a computer program which generates candidate measures from the set, executes various empirical validation tests, and prints results for analysis.

Measure Set Summary

The devised measure set is partitioned into three subsets, each of which represents measures with different characteristics. One subset generates candidate measures which assess performance as characterized by unique patterns of performance variables and their frequency characteristics ("Absolute" measures). The second generates measures that assess performance as characterized by simultaneous (or non-simultaneous) occurrence of unique events ("Relative" measures). The third generates measures that assess performance as characterized by unique successions of events or system states ("State Transfer" measures) and deviations from standards (state frequency measures), where the standards are either defined by the user or computed from user-provided performance data.

Introductory Example

In way of example, each of the above mentioned types of measures will be illustrated, using where possible the previous example of a GCA approach. An attempt will be made to demonstrate that the three types of measures comprising the defined set not only encompass the specific measures of the previous example, but conceivably most other measures commonly (or uncommonly) selected for pursuit in measurement efforts as well as a host of previously untried ones.

First it is necessary to mention (with details presented later in the report) that the measures computed are based on a discrete representation of the performance data, derived through a transformation process. The transformation results in a representation of the data in terms of the number of units by which the value of each variable (e.g., roll, pitch, altitude) is displaced from some reference level or reference function. (The size of the unit-displacements is determined partly as a function of performance range and variance.) Thus altitude, in the GCA example, may be represented by several Boolean functions, each of which denotes whether or not altitude lies in a specific band around the glideslope (e.g., a band 30′ wide located 100′ above (or below) the glideslope may be represented by one Boolean function).

States of the (pilot/vehicle) system are represented by the collective states of the various Boolean functions over time and, in turn, are represented simply by numbers. Thus the number 6 (binary 110), depending on the Boolean functions being investigated, may tell us that at that sampling instant, the pilots altitude was 100′ ± 15′ above the glideslope (first binary digit (1)), his airspeed was 120 knots ± 3 knots (second binary digit (1)), and his roll angle was *not* equal to zero ± 2° (3rd binary digit (0)).

It is this state representation which allows us to efficiently generate and test measures of the 3 types described. Any measures of deviation from the reference function (including time in tolerance, for instance) are inherent in the collective frequencies with which the various defined states are acquired in performance of the maneuver. Any measures of error correction or its rate are inherent in the transfers that occur between various states over time. Measures of frequency content of the data (including the number of glideslope crossings in the previous example) are inherent in the state transfers that occur and/or in the "absolute" type of measure that is investigated. Finally, measures which relate various key events (e.g., smaller glideslope deviations at lower altitudes) are inherent in the "relative" type of measure that is explored.

Consider, first, the measures of RMS glideslope deviation in the previous GCA example. Mathematically, this is represented as

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2}$$

where X is the actual altitude and Y the glideslope altitude. Equivalently this relationship may be correlatively represented by mean square $= RMS^2 \doteq \frac{1}{n} \left(\sum_{i=1}^{k} f_i D_i^2 \right)$

where D_i are specific deviations from the glideslope and f_i are the frequencies with which the associated deviations are encountered. Similarly, a weighted mean square error would be $RMS^2 \doteq \frac{1}{n} \left(\sum_{i=1}^{n} W_i \, Fi \, D_i^2 \right)$; where W_i are weights assigned to each deviation. In the measure-subset based on state frequencies and state transitions, the D_i are represented by the number of units displacement from the glideslope associated with the Boolean functions representing altitude. The F_i values are the computed state frequencies depicting the number of observed samples in which the respective Boolean function is true (1).

Since the referenced measure-subset encompasses F_i, D_i, and related vectors, it may be viewed as a vector set which spans the space of measures represented by RMS glideslope deviations, weighted or unweighted, in addition to any other conceivable measures attending to deviations from a reference function (i.e., average deviation, maximum deviation, integrated error, etc). Therefore, the referenced measure-subset contains a basis for the space of measures of this type, and any measure of the type mentioned may be approximated by some linear combination of the vectors of the defined subset. It is maintained that since this is true, all measures of the type spanned by this subset may be explored by a computational mechanism which can generate the defined vectors and perform multiple regression analyses.

Next, consider the time in tolerance measures and the measure of whether or not descents below the glideslope occur. The former is represented simply by the frequency of occurrence or one or more states, formed by Boolean functions involving the desired tolerance value. A variable tolerance value is automatically included in the analysis because the different Boolean functions themselves (and associated states) represent different tolerances. The latter is also represented by state frequencies, in that states representing descents below the glideslope would all be zero if no descents below the glideslope occurred, and non-zero otherwise. Therefore, these two types of measures are included in the computational mechanism that explores the measure-subset based on state frequencies and state transitions.

Finally, consider the measures (in the GCA example) of number of glideslope crossings and rate of error correction. The first is represented by the number of transitions that occur between states corresponding to aircraft positions below and above the glideslope. The second is also represented by state transitions which; (1) distinguish error growth from decay by the identity of the states between which transitions are occurring, and (2) assess the rate of growth or decay by the relative frequencies of between-state transitions and within-state transitions. Therefore, these types of measures, too, are included in the state frequency and state transition measure subset and related computational mechanism.

This single measure-subset therefore covers all of the specific measures "selected" in the previous GCA example and much more – it covers the general types of measures that are suggested by any considerations of deviation from a reference function, steady-state or transitive positions with respect to it, and movement or rate of movement toward or away from it. The potential power, flexibility, and utility of a computational mechanism exploring this variety of measures is significant. A recent unique application of the state transition concept in measurement and analysis of performance is described in Connelly and Loental (1974).

We have yet to discuss the other two measure-subsets ("Absolute" and "Relative"). The "Absolute" subset and its respective computational mechanism assesses performance characteristics related to the repetative frequencies, periodicities, and associated patterns of changes between various states. This is accomplished in an overall manner similar to that described previously; i.e., vectors which span these types of measures are generated and various measures are explored using regression analyses. Examples of measures that would be included here are the extent and type of "control diddle" used by an operator; frequency characteristics of an operator's ballistic response to, say, a step input; and measures related to the number of control reversals used in performing a segment of some task.

The "relative" subset and computational mechanism fills an identifiable void in the system as thus far described. It takes into consideration the proximity in time with which various events take place and the conditional probabilities of certain events occurring, given that others have occurred. Again, the approach is to generate vectors which span these types of measures and employ regression analysis. Examples of measures thereby addressed are whether or not a pilot achieves and maintains straight and level flight whenever he is within a specified distance from the threshold on a GCA approach; whether or not he begins to roll at the same time (not before or after) he acquires maximum pitch in an aerobatic maneuver; and whether or not he characteristically achieves a specific (criterion) airspeed at key points in performance of, say, a lazy 8 maneuver.

Summary

The preceding subsections describe the fundamental concepts of the approach. The intent is to define a "universal" set of measures, not by enumerating all measures in the set, but by defining their characteristics. Three major characteristics have been defined, and the evolved measure set is represented accordingly by three subsets. It has been shown that measures generated from these various subsets encompass a host of typically selected measures (e.g., those of the GCA example) as well as many others possessing the subset characteristics. Mathematically, this may be viewed as developing vectors which span various measure-spaces, and it is proposed that the measures thereby spanned may be explored using multiple regression analysis.

III. BACKGROUND AND STUDY OBJECTIVES

Many of the basic concepts and mathematical techniques fundamental to this study were explored on a trial basis in previous feasibility studies and are documented in the references (Connelly, Schuler, & Knoop, 1969; Connelly, Schuler, Bourne, & Knoop, 1971). However, efficient computer techniques for exploring the various types of measure-subsets were never fully developed in previous efforts; and the data transformation techniques as well as the measure subsets themselves have been altered and refined for this study on the basis of earlier experience with the approach.

The purpose of the present study was originally to: (1) refine the previously explored technqiues, (2) develop efficient computer implementation methods, (3) validate and demonstrate performance of the software, and (4) apply the techniques thereby implemented to derive and validate performance measures for five training maneuvers flown in T-37B aircraft as part of the Air Force UPT program. Due to non-technical difficulties encountered in collecting the required student and instructor-pilot data, part 4 of the original objectives had to be abandoned, and the objective substituted in its place was to implement and demonstrate the developed software on the Simulation and Training Advanced Research System (STARS). (The STARS system is located at the Advanced Systems Division, Air Force Human Resources Laboratory (AFSC) Wright-Patterson Air Force Base, Ohio. The associated digital computer is a Xerox Data Systems (XDS) Sigma 5.) Therefore, this report documents the computer software developed and the related computational algorithms implemented for exploring selected types of measure-subsets; however, since only a very small amount of data was able to be collected for the study, it was not possible to develop and validate any specific measures. The extensive data collection and reduction machinery developed for use (but unfortunately not applied in this study) is described in Knoop and Welde (1973) and Gregory and Cavanagh (1973).

Scope of Study

The study includes the development and implementation of 3 different computational mechanisms for generating candidate measures from the defined subsets. These are the relative, absolute, and state transfer measures previously discussed. A separate computational mechanism for state frequency measures was not included, partly because the state transfer mechanism itself generates the state frequency data that is needed. Original plans were to develop and independently test a separate state frequency mechanism using this generated data and then, depending on results, interface it with the other elements of the processor. Due to the previously mentioned change in program objectives and associated lack of performance data, however, this was not able to be pursued beyond the planning stage. Emphasis in the study, therefore, was on developing and implementing efficient computer techniques for the 3 developed computational mechanisms and the overall computer-aided processor as described next.

IV. COMPUTER-AIDED PROCESSING TECHNIQUES AND SUBSYSTEMS

The computer-aided automated pilot performance measurement processor is a FORTRAN IV program which generates candidate performance measures through various operations on actual performance data. These operations include:

a. Develop performance criteria,

b. Determine the significance of deviations from criteria,

c. Transform sample performance data into a compact form for processing,

d. Conduct a systematic standardized search for candidate performance measures,

e. Perform validation tests, and

f. Provide data management processes.

A generalized flow diagram of the processor appears in Figure 2.

A primary task in developing performance measures is the determination of standards or reference functions. Performance standards should define the unique manner in which the operator should perform the task. Often, however, there are a number of satisfactory ways to properly accomplish a task and there may exist a family of reference functions representing criterion performance. As a result, the reference function forms employed by the processor may accept parameters provided by the user or estimated from sample performance data.

Multi-variable regression is used in formulation of reference functions from sample data. The idea is to extract from demonstrations of superior performance functions which uniquely represent that performance. Evaluation of the function fit is accomplished through analysis of residues. A small residue value indicates a convenient clustering of all superior performance data, while a large residue value indicates that the regression formulation is not appropriate or that other parameters are required.

An additional test of the candidate functions is made by comparing residues obtained from the superior performance category data with those obtained from other performance category data such as good, fair, and poor. The difference between the residues obtained is an indication of the potential performance discrimination capability of a measure developed from that criterion.

A second important step in the development of performance metrics is the determination of the relevance of deviations from the reference performance. It should be noted that the importance of operator errors is generally not constant over the entire problem state space. Thus, some systematic means must be provided to test various types of deviations and patterns of deviations as to their relevance to performance measurement. Table 1 shows various ways that deviations from the criterion or reference might be related to performance measurement. The processor's capability to assess the significance of a wide variety of relationships such as these is automatically assured due to the types of performance measures it is designed to generate and test.

The processor has four main portions:

1. Input and preparation of data, including
   a. Data management
   b. Smoothing
   c. Maneuver Sectoring
2. Generation of criterion functions via regression analysis
3. Processing of data by adaptive mathematical models
4. Testing and specification of performance measures.

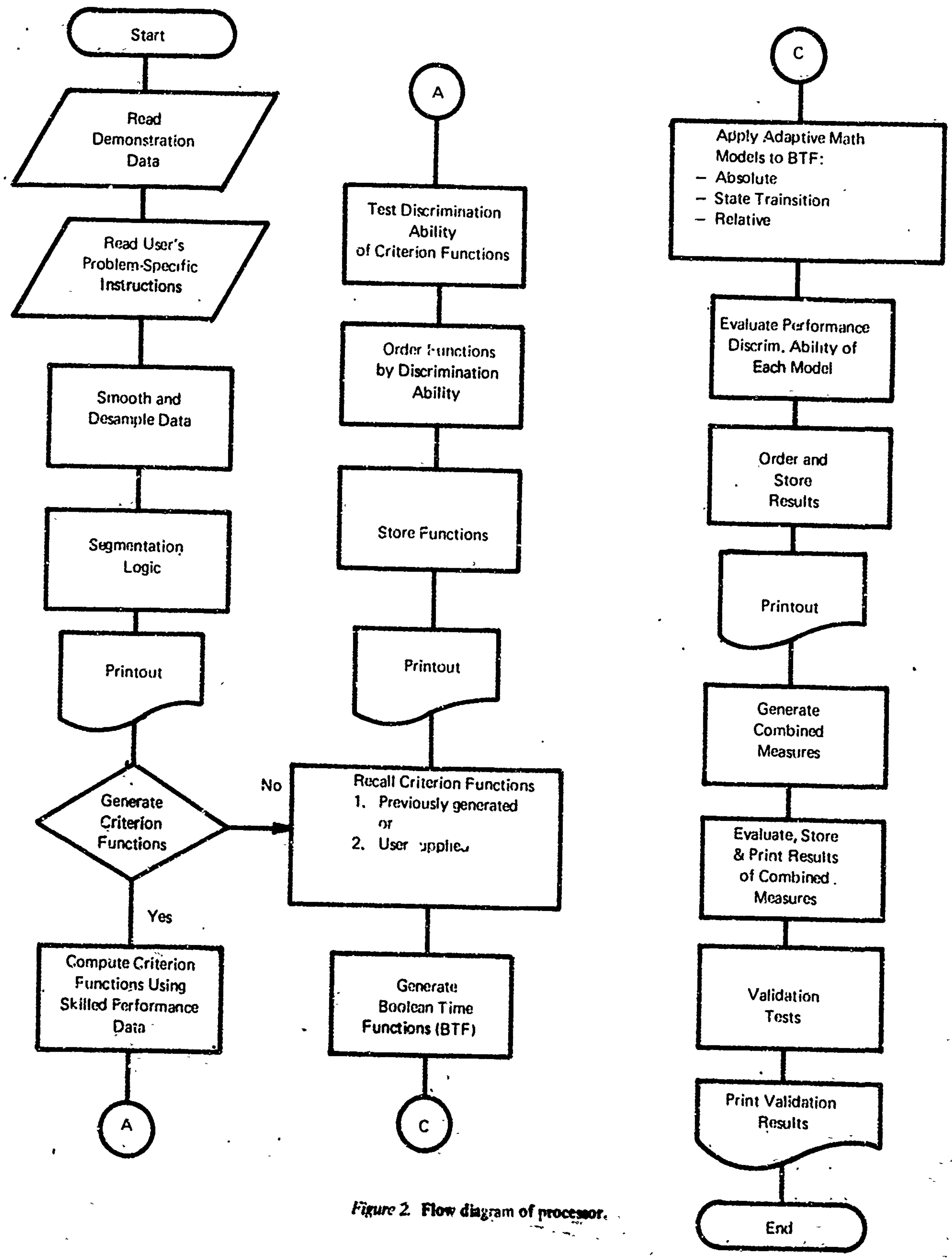


Figure 2. **Flow diagram of processor.**

Table 1. Some Possible Criterion and Performance Measure Factors

Type of Criterion	Possible Ways Deviation (Error) Is Related to Performance
Functions Relating Problem Variables (Reference Path)	o Amount of deviation from path
	o Max deviation
	o Time in a tolerance band
	o Convergence/divergence
	o Similarity to reference path
	o Shape of deviation
	o Time significant deviation occurs
	o Frequency of significant deviations
	o Rate of error correction
	o Way error is corrected
	o Number of errors that occur simultaneously
Differential Reference (where criterion is specified by differential or difference equations)	o Error in differential
	o Critical variable values exceeded
	o Time critical variables values are exceeded
	o Convergence/divergence to reference point on path trajectories
	o Shape of trajectory
Fixed (variable) tolerance at a specific time or at a specific value of another variable	o Variable out of tolerance
	o Amount variable is out of tolerance
	o Time variable is out of tolerance
Sequence of Operation	o Number of errors in sequence
	o Number of critical errors in sequence

Data Management

Due to the great volume of data that must be handled by the processor, systematic data management is of great importance. This is basically a housekeeping operation which controls the coding of data and its efficient storage and retrieval.

Data Smoothing

Examination of recorded flight data shows occasional noise "glitches" on the data samples. These glitches occur at random times and must be removed prior to processing. Noise glitches are assumed to be pulses applied to the filters that smooth data prior to sampling. Thus, the noise pulse appears as a pulse with an exponential decay as shown in Figure 3. The resulting sampled values show a large sample to sample delta change between the samples before and after the noise pulse.

Detection of the noise is accomplished by comparing the sample to sample (delta) change with a pre-established criterion value as follows:

$$|a_{i+1} - a_i| \leq c, \quad i = 1, 2, \ldots \; n \tag{1}$$

where a_i is a sample value and c is a delta criterion value. If the inequality is not satisfied, a noise pulse is assumed to exist.

Once a noise pulse is detected, the time duration of the disturbance must be determined. Experience has shown that the nominal disturbance duration can be expected to be .1 seconds (10 samples at a sampling rate of 100/sec.) for the recorded T-37 flight data. The duration of the disturbance is computed

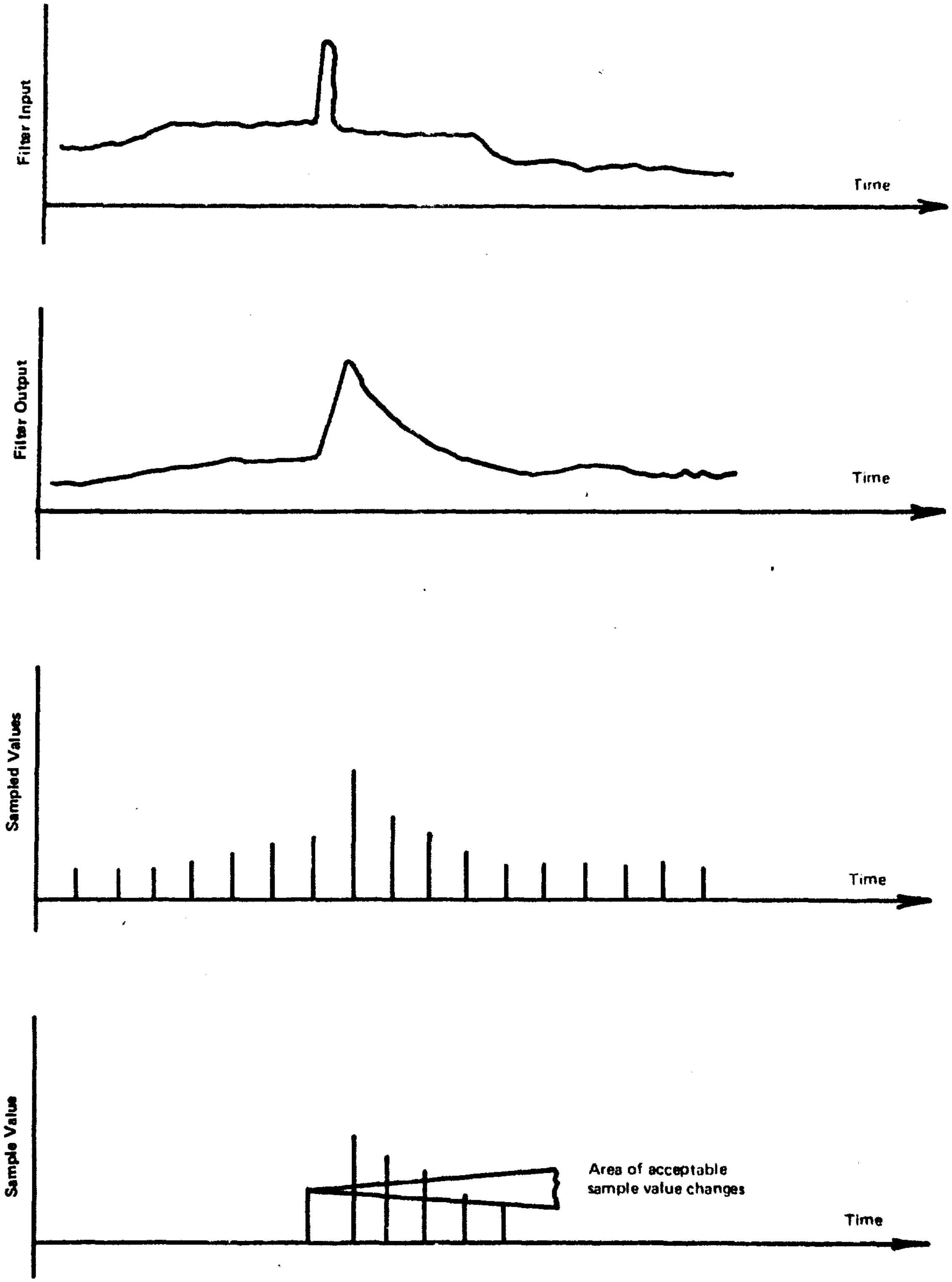


Figure 3. **Smoothing operation.**

by examining the value of future samples one-by-one to determine which sample value first falls within the acceptable region. The test is given by the inequality:

$$|a_{i+j} - a_i| \leq c^*j, \quad j=1,2,\ldots, M \qquad (2)$$

The smallest value of j for which the above inequality is true, is one greater than the number of samples included in the glitch.

When the samples included in the noise glitch have been determined, those sample values are replaced by interpolated values. These values are determined by:

$$a_{i+j} = a_i + \left(\frac{a_{i+n} - a_i}{n}\right)^* j, \quad j=1,2,\ldots \; n-1 \qquad (3)$$

where n is the value of j for which inequality 2 is true.

Maneuver Sectoring

a. *Introduction.* Automatic performance assessment normally involves the computation of several different measures, each of which corresponds to different aspects of (or skills used in) the task of interest. In most flight maneuvers, the skills required (and measured) vary from segment to segment of the maneuver. For example, different skills (and measures) are required during the downwind portion of an approach and landing than are required during the turn to final or during the flare and touchdown. Therefore, it is necessary to segment the maneuver by identifying natural breakpoints which delineate portions requiring computation of different measures. Once these breakpoints have been identified, algorithms and computer techniques are needed for automatically detecting them on the basis of recorded pilot performance data. This section describes the development of such techniques for inclusion in the processor and their trial application to several undergraduate pilot training flight maneuvers. It is appropriate to point out that automatic segmentation of performance also has utility in a number of advanced simulator training capabilities. For example, an increasing number of requirements for and applications of automatic malfunction insertion are emerging in recent and current flight simulator developments. To automatically insert a malfunction at the point in a mission that is realistic for the malfunction and at which the highest training value is expected, it is first necessary to automatically detect the desired point (thus, the utility of automatic segmentation). Other advanced training capabilities such as reinitialization of the simulator and subsequent playback of a portion of the performance also can make use of segmentation techniques (for automatically detecting the point from which playback is desired). Finally, the distinct trend toward the use of cathode ray tube (CRT) displays (rather than or in addition to aircraft repeater instruments) at simulator instructor stations suggests another application of automatic segmentation. Present display techniques are to: (1) always display everything the instructor/operator may ever need to see during the entire mission, or (2) allow various CRT "pages" to be manually selected. The first technique is objectionable due to the number of displays required and subsequent load on the instructor/operator information sorting and processing requirements. The second is equally objectionable due to the instructor/operator information retrieval load. Automatic segmentation techniques could be used to assure that display contents always suit the instructor/operator needs based on what the student is practicing. The techniques described herein could be usefully employed for any of the above applications.

b. *Approach.* The approach was to develop techniques for generating a mathematical representation of the state of pilot/aircraft performance which could be applied to any maneuver. Using this state-representation, segmentation logic was developed for detecting specific states corresponding to the desired breakpoints within each maneuver. (The breakpoints themselves were identified largely on the basis of maneuver analyses performed as a part of other performance measurement studies (Connelly, Bourne, Loental, Migliaccio, Burchick, & Knoop, 1974). This section describes the state representation techniques and the basic segmentation logic that was developed.

(1) Maneuver State Representation

The technique for representing maneuver-performance states was to model significant aspects of the various performances using Boolean functions. The specific Boolean functions used differed from maneuver to maneuver as applied in various combinations to represent desired states. However, since many of the functions were found to be applicable to more than one maneuver, the approach used was to develop a universal function-set, and associated Boolean notation which could be computer-implemented and, collectively, would satisfy state-representation requirements for all the maneuvers.

The devised Boolean functions and the associated notations included indications that a specific condition (Boolean Function) is presently true or that the condition was previously true at least once. Furthermore, functions and notations were developed to indicate when a condition is first true, last true (i.e., when it becomes false), and the interval of time during which the condition is true.

The notation used for representation of states is summarized in Table 2. As shown in the table, the Boolean notation "A = 1" is used to indicate that condition A is presently true, and "A = 0" is used to indicate that condition A is presently false. In addition, "$\widetilde{A}$" indicates whether or not A has been true during the maneuver. Initially, $\widetilde{A}$ is set equal to zero; if condition A becomes true at least once during the maneuver, then $\widetilde{A}$ is set equal to 1 for the remainder of that maneuver. This provides Boolean notation with a "memory" and allows a logic function to be written in terms of present, as well as, previous events.

Table 2. **State Representation Notation**

Notation	
$A = 1$	Condition A is presently true.
$A = 0$	Condition A is presently false.
$\widetilde{A} = 0$	Condition A is not and has not been true during this maneuver.
$\widetilde{A} = 1$	Condition A is, or has been true during this maneuver.
$t(A)$	Time A became true.
$t(\widetilde{A})$	Time A first became true.
$t(\bar{A})$	Time A became false.
$\{ A = X < Z \}$	Defines logic variable as $A = 1$ if $X < Z$ $A = 0$ if $X \geqslant Z$

The time that events take place is also important. Thus, $t(A)$ represents the time that condition A became true, and $t(\bar{A})$ is the time that condition A became false. This is illustrated in Figure 4 where condition A is true for a period of time and then false. Note that the symbol $t(\widetilde{A})$ indicates the time that condition A first became true and is always equal to some corresponding $t(A)$. However $t(\widetilde{A})$ itself may vary over the maneuver if the associated condition (A) changes from false to true more than once.

The devised functions and notation can be used to detect the sequence in which events occur. Consider the time plots of two aircraft variables, pitch (θ) and roll (ϕ), shown in Figure 4. In plot 1 in the figure, pitch reaches zero first, whereas in plot 2 roll reaches zero first. In cases 1 and 3, accordingly, the Boolean conditions B and C indicate when the two variables of interest are zero. In case 2 and 4, the Boolean conditions with "memory" ($\widetilde{B}$ and $\widetilde{C}$) remain true for the subsequent time samples after they first become true. (In cases 5 through 8 and 9 through 12, AND/OR combinations of the Boolean variables are illustrated, respectively.)

This notation provided a concise and easily applied framework within which all desired states could be defined, logically manipulated, and tested. Although simple in appearance (Table 2), the notation is powerful enough for use in developing logic which detects the nature as well as the sequence of various

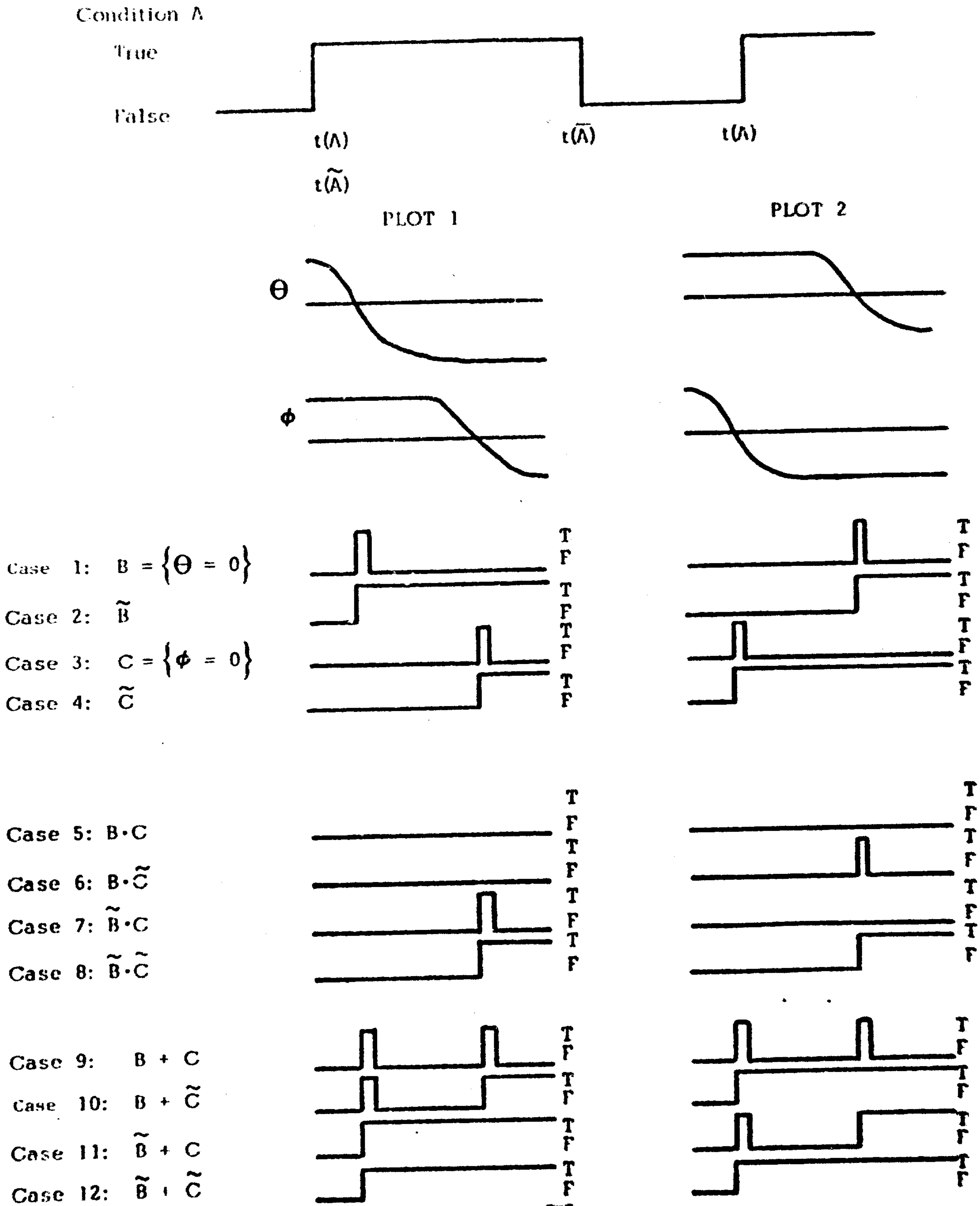


Figure 4. **Examples of logic notation.**

pilot actions, as observed through their affect on aircraft performance. The following paragraphs describe the Boolean function set that was developed and the segmentation logic used for five flight maneuvers.

(2) Segmentation Logic

Table 3 lists the set of individual Boolean functions identified for use in developing the segmentation logic. The function numbers shown in Table 3 are used in Table 4 to summarize the segmentation logic developed for each of five maneuvers. Note that only 17 of the Boolean functions were required for the five maneuvers that were investigated. Based on this observation and the representativeness of the maneuvers tested, it appears that the Boolean function set is more than adequate for developing segmentation logic for conceivably any basic flight training maneuver.

Tables 5 through 9 present the segmentation logic for each maneuver separately, including a descriptive title of each segment and the Boolean functions used for detecting the states determining the desired breakpoints between segments.

c. *Trial Applications.* To perform initial tests of the logic, sample pilot performance data recorded on a T-37B aircraft were used. (The basic data acquisition system is described in (Knoop & Welde, 1973). For this data collection effort, some revisions were incorporated including the addition of stick force sensors and the increase of both range and reliability of aircraft attitude sensing. Revisions made to the original data acquisition system are documented in (Gregory & Cavanagh, 1973).) The performances were flown by instructor pilots who purposely demonstrated examples of how a novice might perform each maneuver as well as examples of skilled maneuver performance. Four flights of each of the 5 maneuver types were used, two of which were rated excellent by the performing pilot and two rated poor. This provided examples of both performance extremes for testing the segmentation logic.

(1) Cloverleaf

The cloverleaf maneuver consists of a pattern of four consecutive loops, or leaves, all identical except for heading. For purposes of explanation, only the first leaf is discussed.

The leaf is begun after the start condition (level flight) is satisfied. Figure 5 is the computer printout of one leaf of an excellent cloverleaf as processed by the logic (segmentation) program. Sector 2 begins when the pilot pitches up above T_2. He then begins to roll (sector 3) until he reaches a maximum roll value (C_{M1}). Although excellent pilots generally roll to 180°, poor pilots often do not achieve 180°; hence, roll maximum is used to trigger the start of sector 4 because the logic must work on all types of flights. In sector 4, the pilot rolls back and pitches down until he reaches a minimum pitch (B_{M2}). Most pilots, regardless of their proficiency, begin to roll out before a pitch of −90° is attained; therefore, sector 5 triggers on minimum pitch. The pilot levels off his pitch (sector 6) prior to entering the next leaf, then begins the leaf by pitching up again (sector 2).

(2) Split S

The split s is an evasive type maneuver in which the pilot effects a 180° heading change by pitching up, rolling over, and pulling out. The plot from the logic program is shown in Figure 6. Initially, sector 5 triggered on pitch = −90°. However, as with the cloverleaf, most pilots, excellent and poor alike, roll out as they pitch down and never reach −90°. Therefore the condition was changed to minimum pitch (θ_{M2}).

(3) Lazy 8

This maneuver consists of two halves, each of which are identical except for heading and direction of roll. The start condition (level flight) is a function of pitch and roll, while the subsequent sectors are identified solely on pitch angle. Figure 7 shows logic program output for the first half (sectors 1–5) of a lazy 8.

(4) Normal Landing

The landing maneuver is made up of five sectors. A sample logic program output is shown in Figure 8. On the sample flights examined in this study, the pilot did not land; instead he performed a touch and go maneuver. In either case, the maneuver is logically terminated when the touch down condition is detected.

Table 3. **Boolean Functions**

No.	Function	No.	Function
0	$B_1 \cdot C_1$	22	$E_2 = \{Rs = \text{Full left}\}$
1	$B_1 = \{\theta = \theta_1 \pm T_1\}$	23	$E_3 = \{Rs = \text{Full right}\}$
2	$B_2 = \{\theta > T_2\}$	24	$E_4 = \{Rs = \text{Reversed}\}$
3	$B_3 = \{\theta < T_3\}$		$= \{E_2 \widetilde{E}_3 + E_3 \widetilde{E}_2\}$
4	$B_4 = \{\theta = +45^\circ \pm T_4\}$	25	$F_1 = \{Dt = \text{Neutral} \pm S_1\}$
5	$B_5 = \{\theta = -90^\circ \pm T_5\}$	26	$F_2 = \{Dt = \text{Full forward}\}$
6	$B_{m1} = \{\theta = \theta_{m1}\}$	27	$G_1 = \{T = \text{Idle}\}$
7	$B_{m2} = \{\theta = \theta_{m2}\}$	28-32	Future Expansion
8	$B_{m3} = \{\theta = \theta_{m3}\}$	33	$\overline{C}_1$
9	$B_{m4} = \{\theta = \theta_{m4}\}$	34-39	Future Expansion
10	$B_{m5} = \{\theta = \theta_{m5}\}$	40	C2 + C3
11	$C_1 = \{\phi = 0. \pm I_1\}$	41	B1 • C1
12	$C_2 = \{\phi > I_2\}$	42	B2 + C2 + C3
13	$C_3 = \{\phi < I_3\}$	43	B1 + C1
14	$C_4 = \{\phi = +180^\circ \pm I_4\}$	44	CM1 • C1
15	$C_5 = \{\phi = +90^\circ \pm I_5\}$	45	CM2 • C1
16	$C_6 = \{\phi = -90^\circ \pm I_6\}$	46	CM3 • C1
17	$C_{m1} = \{\phi = \phi_{m1}\}$	47	C5 + C6
18	$C_{m2} = \{\phi = \phi_{m2}\}$	48	E2 + E3
19	$C_{m3} = \{\phi = \phi_{m3}\}$	49	Future Expansion
20	$D_1 = \{As < Tas_1\}$	50	B1 • C1 Stop Condition
21	$E_1 = \{Rs = \text{Neutral} \pm R_1\}$		

Notes: 1 TAS_1, T_n, and I_n, are tolerance factors defined as follows for the five maneuvers:

	Cloverleaf	Split S	Lazy 8	Landing	Barrel Roll
TAS_1	0	0	0	90	0
T_1	7	7	5	4	10
T_2	5	5	8	0	2
T_3	–5	0	–8	0	–5
T_4	5	5	0	0	5
T_5	5	5	0	0	5
I_1	5	5	5	8	5
I_2	10	9	8	10	10
I_3	–10	–9	–8	–10	–10
I_4	5	5	5	5	5
I_5	5	5	5	5	1
I_6	5	5	5	5	1

[2] Computation of successive maximum or minimum values such as θ_{MX} and θ_{MX+1} requires that an intermediate null condition (B_1) be true. Thus the following sequences must occur in order for successive extreme values to be established: $\theta_{MX}, B_1, \theta_{MX+1}, B_1, \theta_{MX+2}, \ldots$

$\phi_{MY}, C_1, \phi_{MY+1}, C_1, \phi_{MY+2}, \ldots$

[3] R_1 = general tolerance on rudder position
S_1 = general tolerance on stick position

[4] θ - pitch
ϕ - roll
θ_{MI}, ϕ_{MI} - ith local maxima
As - airspeed
Rs - rudder position
Dt - longitudinal stick position
T - throttle position

Table 4. **Summary of Segmentation Logic**

		Segments																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Cloverleaf	1	0	2	40	17	7	1	50															
	2							2	40	17	7	1	50										
	3												2	40	17	7	1	50					
	4																	2	40	17	7	1	50
Split S	1	0	2	40	3	7	50																
Lazy 8	1	0	42	6	1	7																	
	2						43	8	1	9	50												
Normal Landing	1	0	40	44	40	45	20																
Barrel Roll	1	0	3	2	47	43	50																

Table 5. **Maneuver State Logic for a Cloverleaf**

Sector Number	Sector Name	Condition	Boolean Function	Function* Number
1	Entry	Pitch = $0° \pm T_1$ and Roll = $0° \pm I_1$	$B_1 \cdot C_1$	0
2	Climb	Pitch > T_2	B_2	2
3	Roll	Roll < I_3 (Left) or Roll > I_2 (Right)	$C_2 + C_3$	40
4	Pitch to 90°	Roll ϕ_{M1}	C_{M1}	17
5	Pull Thru	Pitch = θ_{M2}	B_{M2}	7
6	Final (entry to next leaf)	Pitch = $0° \pm T_1$	B_1	1

*From Table 3.

Table 6. **Maneuver State Logic for a Split S**

Sector Number	Sector Name	Condition	Boolean Function	Function* Number
1	Start	Pitch $= 0° \pm T$ and Roll $= 0° \pm I_1$	$B_1 \cdot C_1$	0
2	Entry	Pitch $> T_2$	B_2	2
3	Inversion	Roll $> I_2$ or Roll $< I_3$	$C_2 + C_3$	40
4	Pull thru to 90°	Pitch $< T_3$	B_3	3
5	Pull thru to 0°	Pitch $= \theta_{M2}$	B_{M2}	7
	Stop	Pitch $= 0° \pm T_1$ and Roll $= 0° \pm I_1$	$B_1 \cdot C_1$	50

*From Table 3.

Table 7. **Maneuver State Logic for a Lazy 8**

Sector Number	Sector Name	Condition	Boolean Function	Function* Number
1	Entry	Pitch $= 0° \pm T_1$ and Roll $= 0° \pm I_1$	$B_1 \cdot C_1$	0
2	1st Quarter	Pitch $> T_2$ or Roll $> I_2$ or Roll $< I_3$	$B_2 + C_2 + C_3$	42
3	2nd Quarter	Pitch $= \theta_{M1}$	B_{M1}	6
4	3rd Quarter	Pitch $= 0° \pm T_1$	B_1	1
5	4th Quarter	Pitch $= \theta_{M2}$	B_{M2}	7
6	1st Quarter	Pitch $= 0° \pm T_1$ or Roll $= 0° \pm I_1$	$B_1 + C_1$	43
7	2nd Quarter	Pitch $= \theta_{M3}$	B_{M3}	8
8	3rd Quarter	Pitch $= 0° \pm T_1$	B_1	1
9	4th Quarter	Pitch $= \theta_{M4}$	B_{M4}	9
Stop	End	Pitch $= 0° \pm T_1$ and Roll $= 0° \pm I_1$	$B_1 \cdot C_1$	50

*From Table 3.

Table 8. **Maneuver State Logic for a Normal Landing**

Sector Number	Sector Name	Condition	Boolean Function	Function* Number
1	Start	Pitch = 0° ± T_1 and Roll = 0° ± I_1	$B_1 \cdot C_1$	0
2	Pitch Out	Roll > I_2 or Roll < I_3	$C_2 + C_3$	40
3	Downwind	Roll = 0° ± I_1 after Roll = ϕ_{M1}	$C_{M1} \cdot C_1$	44
4	Final Turn	Roll > I_2 or Roll < I_3	$C_2 + C_3$	40
5	Final Approach	Roll = 0° ± I_1 after Roll = ϕ_{M2}	$C_{M2} \cdot C_1$	45
Stop	(Touch Down or Touch and Go)	Airspeed < T_{AS1}	D_1	20

*From Table 3.

Table 9. **Maneuver State Logic for a Barrel Roll**

Sector Number	Sector Name	Condition	Boolean Function	Function* Number
1	Start	Pitch = 0° ± T_1 and Roll = 0° ± I_1	$B_1 \cdot C_1$	0
2	Entry	Pitch < T_3	B_3	3
3	1st Quarter	Pitch > T_2	B_2	2
4	2nd Quarter	Roll = 90° ± I_5 or Roll = −90° ± I_6	$C_5 + C_6$	47
5	3rd Quarter	Roll = 180° ± I_4	C_4	14
6	4th Quarter	Roll = −90° ± I_6 or Roll = 90° ± I_5	$C_5 + C_6$	47
7	End	Pitch = 0° ± T_1 or Roll = 0° ± I_1	$B_1 + C_1$	4
Stop		Pitch = 0° ± T_1 and Roll = 0° ± I_1	$B_1 \cdot C_1$	50

*See Table 3.

REFERENCE NO. 224 NO. OF SAMPLES READ= 88 TOTAL SAMPLES READ= 88 CLOVERLEAF RIGHT EXCELLENT

CLOVERLEAF SEGMENT #	SAMPLES START	SAMPLES STOP	SAMPLES TOTAL	TIME START	TIME STOP	TIME TOTAL
1	0	23	24	0.0	11.50	12.00
2	24	34	11	12.00	17.00	5.50
3	35	39	5	17.50	19.50	2.50
4	40	50	11	20.00	25.00	5.50
5	51	59	9	25.50	29.50	4.50
6	60	73	14	30.00	36.50	7.00

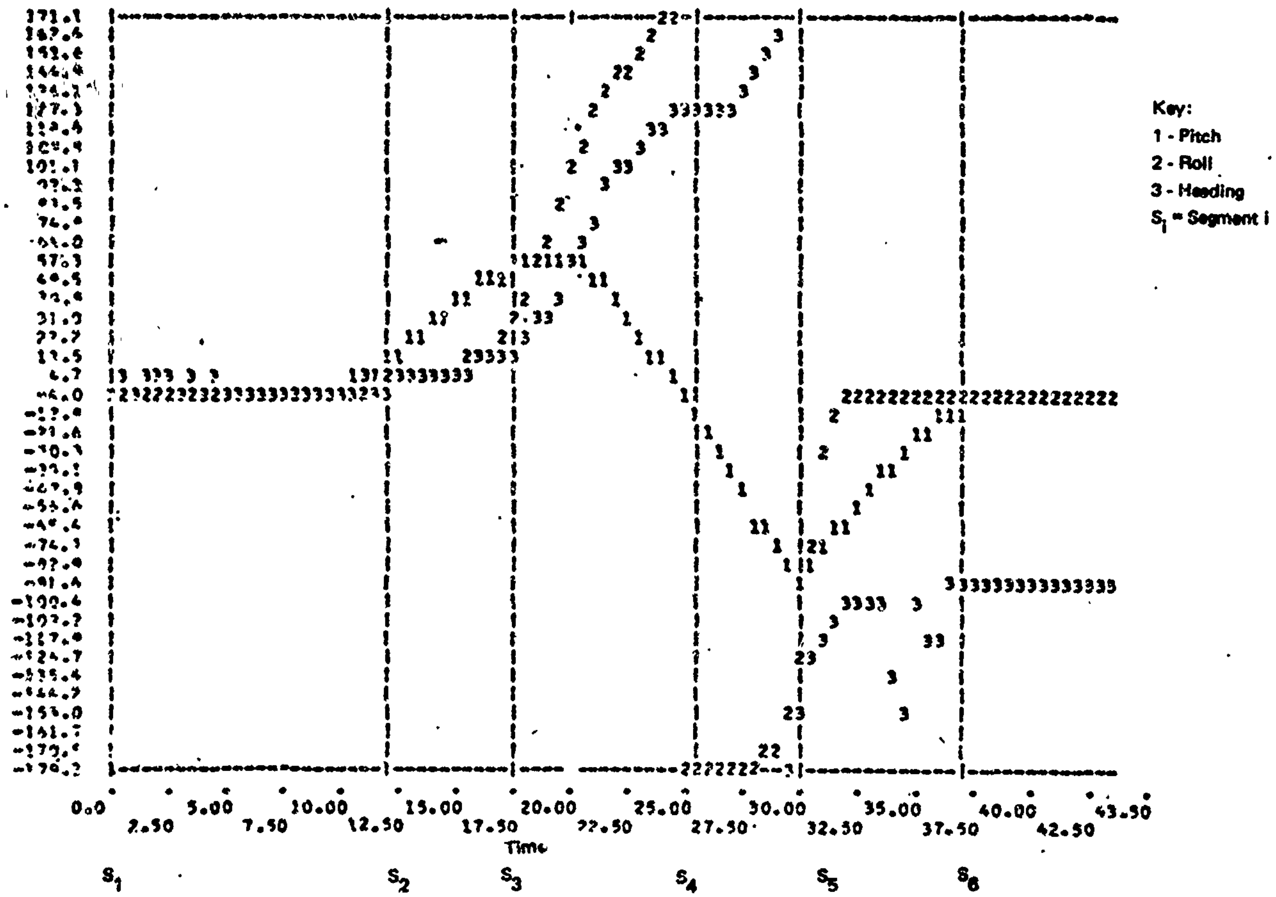


Figure 5. **Logic plot of one leaf of a cloverleaf.**

REFERENCE NO. 126 NO. OF SAMPLES READ= 76 TOTAL SAMPLES READ= 76, SPLIT S RIGHT EXCELLENT

SPLIT S SEGMENT #	SAMPLES START	SAMPLES STOP	SAMPLES TOTAL	TIME START	TIME STOP	TIME TOTAL
1	0	2	3	0.0	1.00	1.50
2	3	29	27	1.50	14.50	13.50
3	30	37	8	15.00	19.50	4.00
4	35	47	10	19.00	23.50	5.00
5	48	58	11	24.00	29.00	5.50

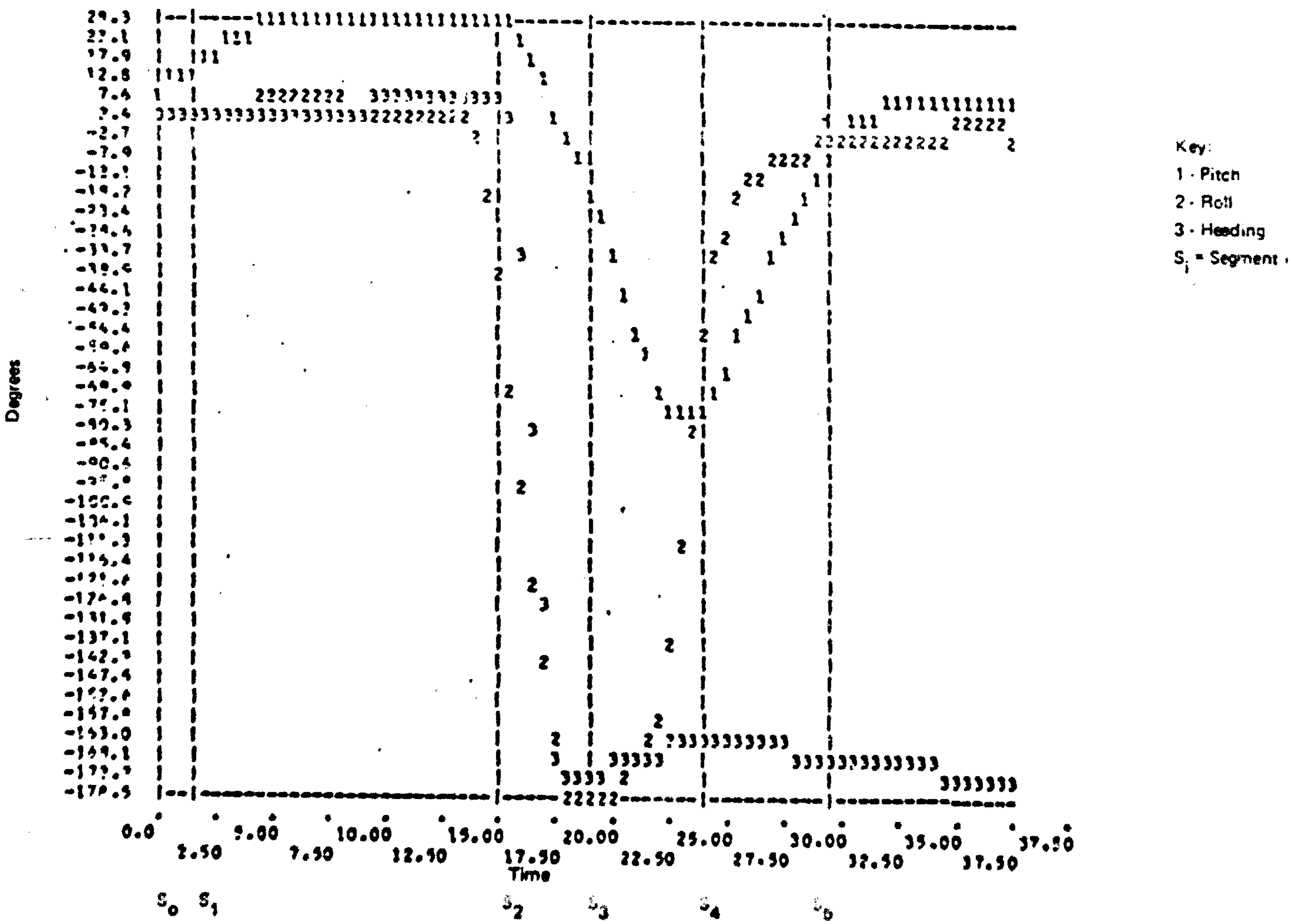


Figure 6. **Logic plot of a split S.**

LAZY EIGHT SECTOR #	SAMPLES START	SAMPLES STOP	SAMPLES TOTAL	TIME START	TIME STOP	TIME TOTAL
1	0	7	8	0.0	3.50	4.00
2	8	30	23	4.00	15.00	11.50
3	31	41	11	15.50	20.50	5.50
4	42	56	15	21.00	28.30	7.50
5	57	73	17	28.50	36.50	8.50

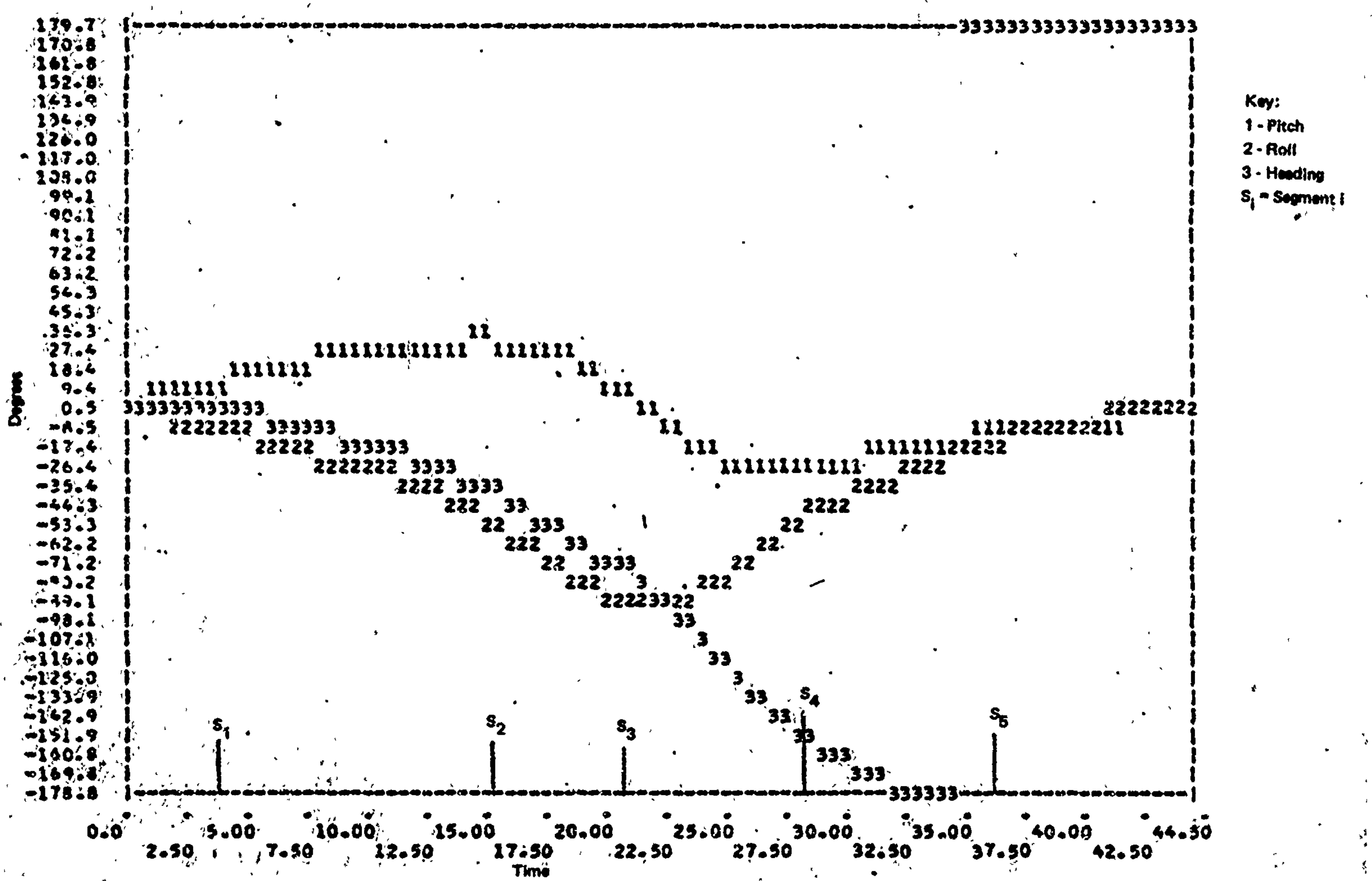


Figure 7. **Logic plot of one-half of a lazy 8.**

REFERENCE NO. 121 NO. OF SAMPLES READ= 95 TOTAL SAMPLES READ= 95 MANEUVER TYPE= NORMAL LANDG

NORMAL LANDG SECTOR #	SAMPLES START	SAMPLES STOP	SAMPLES TOTAL	TIME START	TIME STOP	TIME TOTAL
1	0	17	18	0.0	34.00	36.00
2	18	28	11	36.00	56.00	22.00
3	29	38	10	58.00	76.00	20.00
4	39	57	19	78.00	114.00	38.00
5	58	69	12	116.00	138.00	24.00
6	70	70	1	140.00	140.00	2.00

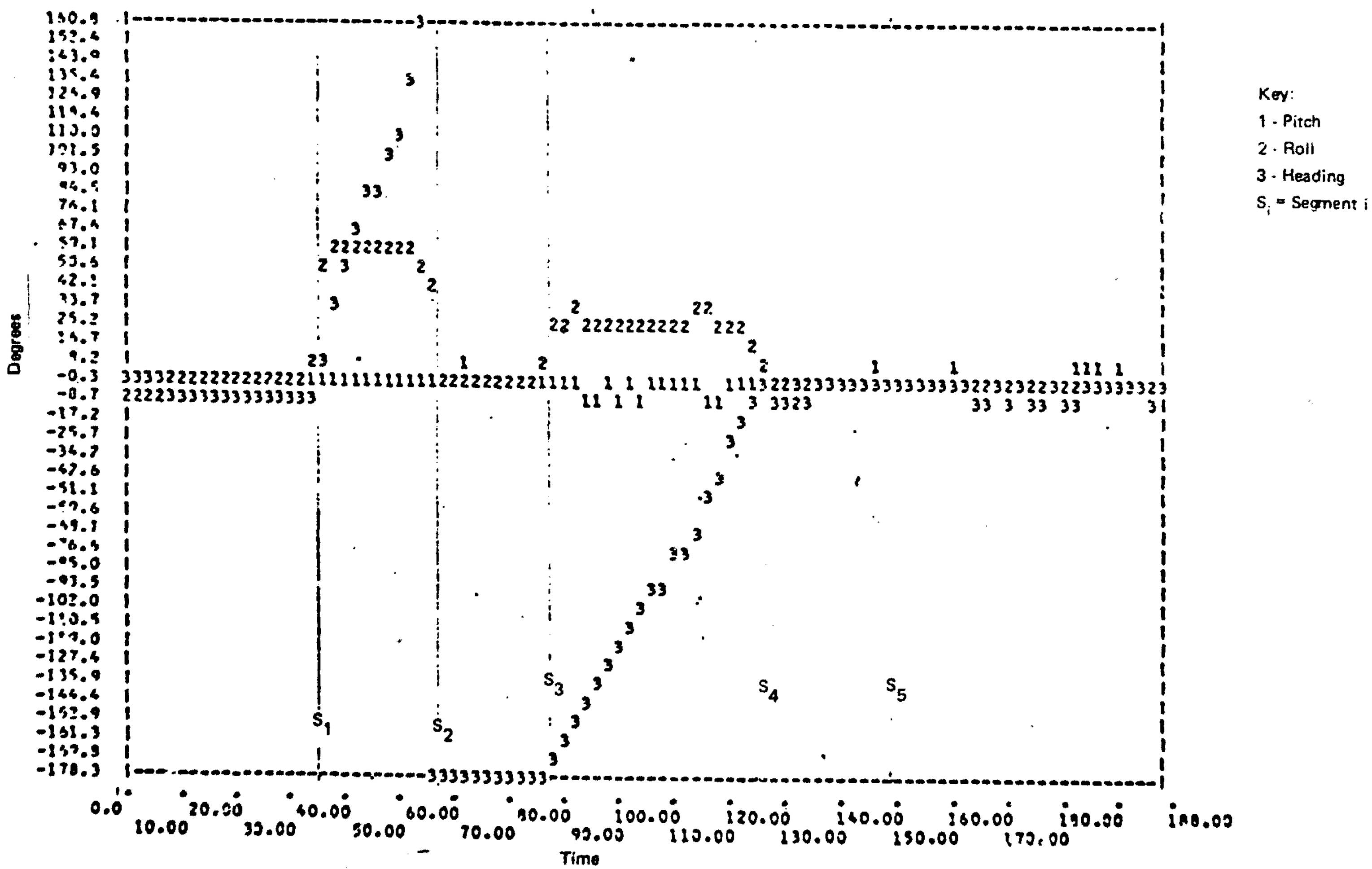


Figure 8. **Logic plot for a normal landing.**

(5) Barrel Roll

This is an acrobatic maneuver in which the pilot rolls through 360°. A sample logic program plot appears in Figure 9. The sample data indicates that inexperienced pilots have difficulty attaining level flight ($B_1 \cdot C_1$) at the end of the 4th quarter (sector 6); consequently, sector 6 contained a long tail of data not representative of the specific pilot task we were trying to isolate. A seventh sector was added to allow termination of sector 6 if either pitch or roll is level ($B_1 + C_1$). The maneuver terminates when the pilot attains level flight.

Regression Analysis

a. *Introduction.* A major purpose of the regression analysis is to generate reference functions which are representative of excellent performances. These reference functions are automatically generated by the processor for use in deriving performance measures.

A number of reference functions are constructed for each sector of each maneuver type. Each function is a mathematical representation of certain parameter relationships characteristic of that sector. Deviations of an actual flight from this function are computed. A standard set of operations on these deviations are performed, and results are tested for performance discrimination content. The techniques of measuring and interpreting these deviations are discussed in detail in later sections of this report.

A useful reference function must be consistent, in that it produces small deviations with data from excellent performances; at the same time, it must be able to provide discrimination in tests among various performance levels. In the processor, reference functions are generated, then tested for consistency and discrimination capability. This procedure is discussed in the following sections.

b. *Theory.* Several reference functions are generated for each sector of each maneuver by performing a least squares regression analysis on selected skilled performance data for the specified sector. In our initial analysis, four candidate reference functions are generated for each maneuver sector by using data from two available excellent-rated flights of each maneuver type. The technique is illustrated in Figure 10.

(1) Regression Computation Method

In applications, many sample flights of several performance categories will be used to form the reference functions and update them. The data are initially arranged on tape by maneuver type and it is not feasible to store all data or to reread the tape for each maneuver sector. Therefore, it is desirable to use a technique which allows updating of the regression coefficients without having to store all previous raw data. The method, discussed in detail in Connelly et al., (1969, pp. 179-181), represents the data in a compact summary form. Briefly, the problem is stated as:

$$Y = A + \sum_{j=1}^{N} B_j X_j \tag{3}$$

where Y is a factor of interest (dependent variable) and X_j is a combination of the system variables (independent variables). Given T samples or experiments, the method of least squares minimizes P:

$$P = \sum_{i=1}^{T} \left(-Y_i + A + \sum_{j=1}^{N} B_j X_{ji}\right)^2 \tag{4}$$

and gives solution values for the coefficients:

$$-\sum_{i=1}^{T} Y_i X_{ki} + \sum_{i=1}^{T} \sum_{j=1}^{N} B_j X_{ji} X_{ki} + \sum_{i=1}^{T} X_{ki} \left(1/T \sum_{i=1}^{T} Y_i - 1/T \sum_{i=1}^{T} \sum_{j=1}^{N} B_j X_{ji}\right) = 0$$

BARREL ROLL SECTOR #	SAMPLES START	SAMPLES STOP	SAMPLES TOTAL	TIME START	TIME STOP	TIME TOTAL
1	0	5	6	0.0	[illegible]	3.00
2	6	37	32	3.00	18.50	16.00
3	38	51	14	17.00	25.50	7.00
4	52	61	10	26.00	30.50	5.00
5	62	70	9	31.00	35.00	4.50
6	71	79	9	35.50	39.50	4.50
7	80	81	2	40.00	40.50	1.00

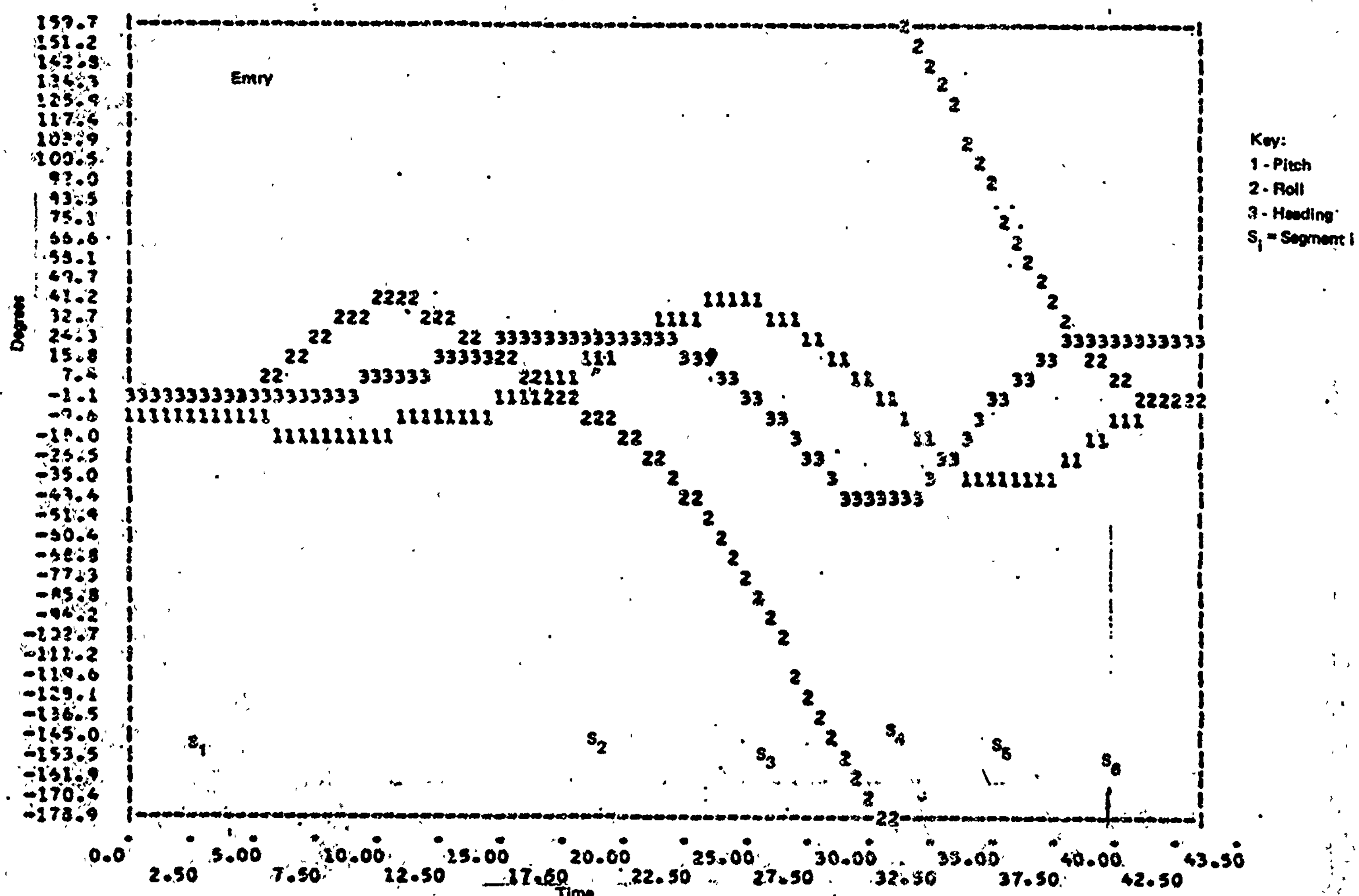


Figure 9. Logic plot for a barrel roll.

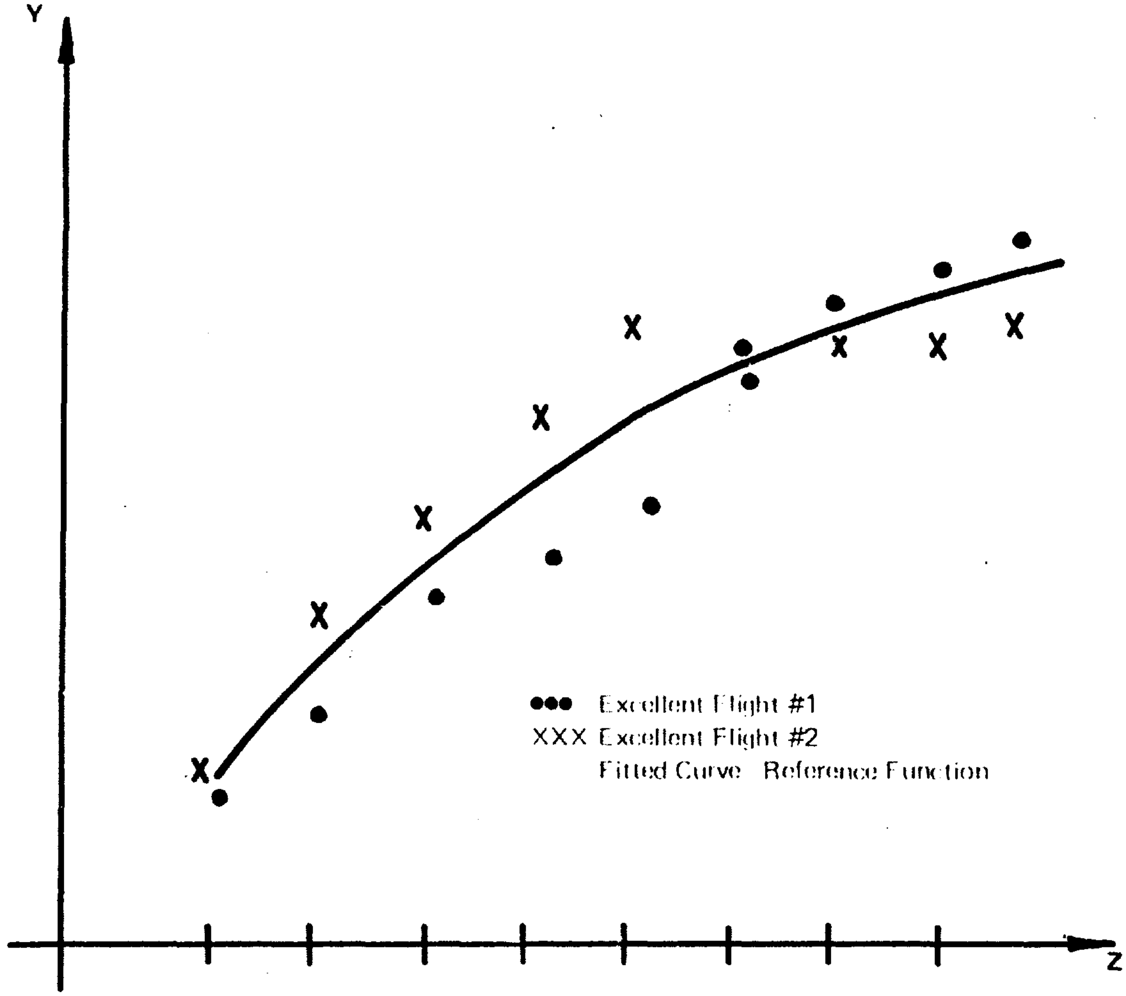


Figure 10. **Generation of a reference function by a least squares regression analysis.**

where

$$K = 1, N \tag{5}$$

$$A = 1/T \left(\sum_{i=1}^{T} Y_i - \sum_{i=1}^{T} \sum_{j=1}^{N} B_j X_{ji} \right) \tag{6}$$

Defining the following sums:

$$SUMYX(K) = \sum_{i=1}^{T} Y_i X_{ki} \tag{7}$$

$$SUMXX(J, K) = \sum_{i=1}^{T} X_{ji} X_{ki} \tag{8}$$

$$SUMX(K) = \sum_{i=1}^{T} X_{ki} \tag{9}$$

$$SUMY = \sum_{i=1}^{T} Y_i \tag{10}$$

allows representation of the problem as:

$$0 = -SUMYX(K) + \sum_{j=1}^{N} B_j\, SUMXX(J, K) + SUMY * SUMX(K)/T - SUMX(K)/T * \sum_{j=1}^{N} B_j\, SUMX(J)$$

where

$$K = 1, N \tag{11}$$

Solving for the B's is done via a matrix approach:

$$\text{Let } B = \begin{bmatrix} B(1) \\ B(2) \\ \cdot \\ \cdot \\ \cdot \\ B(N) \end{bmatrix} \tag{12}$$

$$R = \begin{bmatrix} SUMYX(1) - SUMY * SUMX(1)/T \\ SUMYX(2) - SUMY * SUMX(2)/T \\ \cdot \quad\quad \cdot \\ \cdot \quad\quad \cdot \\ \cdot \quad\quad \cdot \\ SUMYX(N) - SUMY * SUMX(N)/T \end{bmatrix} \tag{13}$$

$$Q(J,K) = SUMXX(J,K) - SUMX(K) * SUMX(J)/T \tag{14}$$
(known as correlation coefficients)

$$SUM = \begin{bmatrix} Q(1,1) & Q(2,1) & \dots & Q(N,1) \\ Q(1,2) & Q(2,2) & \dots & Q(N,2) \\ \vdots & \vdots & & \vdots \\ Q(1,N) & Q(2,N) & & Q(N,N) \end{bmatrix} \tag{15}$$

(an NxN matrix)

The problem is now written as:

$$SUM \cdot B = R \tag{16}$$

and the solution is:

$$B = SUM^{-1} \cdot R \tag{17}$$

Now A can be computed from equation 6:

$$A = 1/T \left(SUMY - \sum_{j=1}^{N} B(J)\, SUMX(J) \right) \tag{18}$$

For ease in programming, the matrices *SUM* and *R* can be further broken down:

$$SUMXX = \begin{bmatrix} SUMXX(1,1) & SUMXX(2,1) \cdots & SUMXX(N,1) \\ \vdots & \vdots & \vdots \\ SUMXX(1,N) & SUMXX(2,N) \cdots & SUMXX(N,N) \end{bmatrix} \tag{19}$$

$$SUMXJ = \begin{bmatrix} SUMX(1) & SUMX(2) & \dots & SUMX(N) \end{bmatrix} \tag{20}$$

$$SUMXK = \begin{bmatrix} SUMX(1) \\ SUMX(2) \\ \vdots \\ SUMX(N) \end{bmatrix} \tag{21}$$

Then $SUM = SUMXX - 1/T * SUMXK * SUMXJ$ (22)

Let

$$SUMYX = \begin{bmatrix} SUMYX(1) \\ \vdots \\ SUMYX(N) \end{bmatrix} \tag{23}$$

Then $R = SUMYX - SUMY/T * SUMXK$ (24)

Therefore

$$B = \left[SUMXX - 1/T\, SUMXK * SUMXJ\ \right]^{-1} \cdot \left[SUMYX - SUMY/T \cdot SUMXK\right] \qquad (25)$$

This technique provides for efficient computer usage by compact storage of all previous data that is needed for updating of the regression coefficients.

(2) Candidate Reference Functions

In the processor, reference functions are based on data from several excellent maneuvers. Initially, one excellent maneuver sector is read, and a least squares regression is performed on it. The data are read sector by sector and maneuver by maneuver from tape; when another excellent maneuver sector is encountered it is used to update the regression.

Four candidate reference functions are generated for each maneuver sector:

$$\hat{h} = F_{\hat{h}}(X, \theta_{Max}, \phi_{Max})$$

$$\hat{\psi} = F_{\psi}(X, \theta_{Max}, \phi_{Max})$$

$$\widehat{AS} = F_{AS}(X, \theta_{Max}, \phi_{Max})$$

$$\hat{\theta} = F_{\theta}(X, \theta_{Max}, \phi_{Max})$$

$$F(X, \theta_{Max}, \phi_{Max}) = B_0 + B_1 X + B_2 X^2 + B_3 \theta_M + B_4 \phi_M$$

The above variables are: θ_M = maximum pitch in the sector, ϕ_M = maxiumum roll, h = altitude, ψ = heading, AS = airspeed, θ = pitch, and the "$\wedge$" means "estimate". X, the independent variable, can be roll, pitch or normalized time. Roll (ϕ) is always selected first if roll is monotonic in that sector. If roll data is not monotonic over the sector, pitch (θ) is selected; and the fourth reference equation then becomes:

$$\psi = B_0 + B_1\theta + B_2\theta^2 + B_3\, \theta_M + B_4\, \phi_M$$

If neither pitch nor roll is monotonic, normalized time (t) is selected as the independent variable, and in that case the fourth reference function can have either θ or ϕ as the dependent variable.

(3) Performance Discrimination

The purpose of a reference function is to specify a standard flight path for use in developing performance measures. It is necessary, therefore, for a refence function to give consistent results for all excellent performances, and at the same time provide a basis for discriminating performance that are other than excellent. The measure used for a preliminary test is the mean absolute residual error:

$$\epsilon = 1/T \sum_{i=1}^{T} |Y_i - \hat{Y}_i|$$

where T is the number of samples in the maneuver sector; Y_i is the actual value of the dependent variable; and $\hat{Y}_i$ is the prediction of Y_i by the reference function. The value of ϵ gives an indication of how much a sample flight deviates from the reference function. If the reference function possesses good performance discrimination capabilities, then ϵ would be expected to be small for excellent flights, and to increase as the performance level worsens. A graphical interpretation of ϵ appears in Figure 11.

In the processor, candidate reference functions are generated from several excellent maneuvers. These maneuvers can be considered to be a "training set" for the processor and this set possesses some mean residual error, e_T, with respect to each reference function. As a test of the consistency and discrimination ability of the reference function, "test sets" are formed consisting of one set of excellent maneuvers not included in the training set, and one set of "poor" maneuvers. The test sets also have a mean residual error,

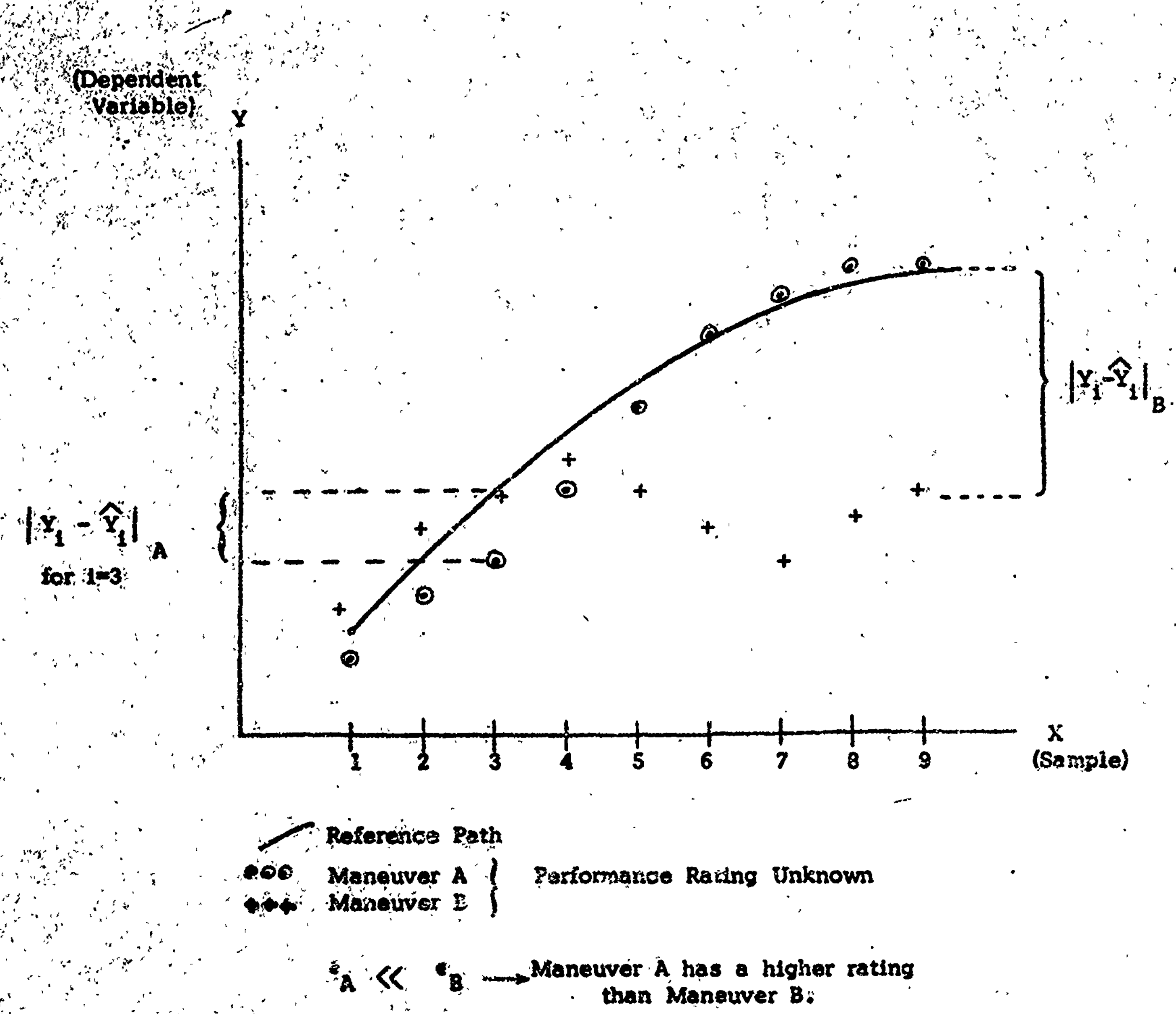


Figure 11. **Performance discrimination by a reference function.**

Table 10. **Independent Variable Selected for Each Maneuver Sector**

Maneuver Type	Sector 1	2	3	4	5	6	7
Cloverleaf	Time	Pitch	Roll	Pitch	Pitch		
Split S	Time	Time	Roll	Roll	Pitch		
Lazy 8	Time	Roll	Roll	Pitch	Roll		
Normal Landing	Time	Time	Time	Time	Time		
Barrel Roll	Time	Time	Roll	Roll	Roll	Roll	Time

Table 11. **Regression Analysis Results for Sector 1 of the Cloverleaf**

Dependent Variable	Errors				
	Excellent			Poor	
	ϵ_C	125(L)	124(R)	127(L)	128(R)
h	25.44	25.05	25.73	3794.50	3478.60
Ψ	0.49	.09	0.79	.68	2.79
AS	1.11	1.14	1.09	3122.40	2730.00
θ	0.87	.79	.93	170.35	146.99
ϕ	0.34	.36	.32	17.32	15.76
#/samples		17	23		
θ_M		–7.0	–7.7	5.4	6.3
ϕ_M		–6.0	–1.6	–.4	3.1

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

Variable	Reference Functions B_0	B_1	B_2	B_3	B_4
h	2240.00	–631.20	220.50	318.10	–141.80
Ψ	–.40	.44	0.08	–0.02	–.29
AS	1890.00	28.70	–11.78	254.90	–175.70
θ	90.58	–23.55	30.65	14.05	–8.48
ϕ	10.22	–2.81	2.56	1.37	–0.09

ϵ_S, with respect to the reference function. If the reference function is a suitable basis for performance discrimination, then ϵ_S should be approximately equal to ϵ_T for a test set comprising excellent maneuvers, and significantly greater than ϵ_T for a test set made up of poor maneuvers. A simple test is to compare the distribution of residuals, ϵ_{Ti}, from the training set to the distribution of residuals, ϵ_{Si}, from the test set via the rank sum statistic (see Rank Sum Statistic section). This statistic tests the hypothesis that the two sets (training and test) are equal; i.e., that the reference function in question produces similar results for two different sets of maneuvers. The rank sum test is used because it requires no knowledge about the distribution of residuals, and it can compare two data sets of different lengths.

c. *Results*. A total of 19 maneuvers were run. The independent variables chosen for each maneuver sector, as discussed in Regression Analysis section, are shown in Table 10. Table 11 shows the actual reference functions for sector 1 of a cloverleaf. The functions are written in the bottom table below the functional form. The residual errors ($\in$) appear in the top table. Column 1 is the combined error of the two excellent maneuvers and is computed as follows:

$$\in_c = \frac{N_1 \cdot \in_1 + N_2 \cdot \in_2}{N_1 + N_2}$$

where N_1 and N_2 are the number of samples in the two excellent maneuvers and $\in_1$ and $\in_2$ are their associated errors. Columns 4 and 5 show the errors for the two poor maneuvers.

Regression functions for other cloverleaf sectors and other maneuvers appear in Appendix A. (The three digit numbers for the maneuvers were assigned for maneuver identification. A summary of maneuvers used is shown in Table 12.) Some sectors contained too few data points to obtain a significant regression; consequently, no data appears for them.

Table 12. **Maneuver Identification Code for Computer Processor Printouts**

Maneuver	Mode	Proficiency Rating	ID Code
Cloverleaf	Right	Excellent	124
	Left	Excellent	125
	Left	Poor	127
	Right	Poor	128
Split S	Left	Excellent	126
	Left	Poor	129
	Left	Poor	130
Lazy 8	Right	Good Plus	102
	Right	Poor	106
	Left	Excellent	108
	Left	Poor	113
Normal Landing		Excellent	121
		Excellent	122
		Poor	123
		Poor	124
Barrel Roll	Left	Excellent	101
	Left	Excellent	107
	Left	Poor	110
	Left	Poor	114

On certain maneuvers, very small differences in θ_{Max} and ϕ_{Max} seem to cause large differences in the residual errors. For example, in sector 6 of the normal landings (Appendix A), the coefficient of ϕ_{Max} for the heading regression is extremely large (B_4 = −7990). Although ϕ_M of maneuver 124 differs from ϕ_M of the excellents by only about two degrees, the difference in $\in$'s is huge. Based on the processing of four maneuvers, it would appear that an excellent maneuver with a slightly different ϕ_M could significantly increase the $\in$'s for the excellent maneuvers.

Further processing is required to determine if this is true in general. If so, one alternative would be to change the form of the regression; e.g., $Y = B_0 + B_1 X + B_2 X^2$. However, lacking the necessary data to test many more maneuvers, as required, the regression functions in the processor were implemented as documented in this report.

Adaptive Math Models

This section describes the experimental techniques for generating candidate measures for subsequent validation-testing. The models for so doing are called adaptive mathematical models because the candidate measures which they generate are derived recursively and adaptively in accordance with the success encountered with various measure-types. Since much of the underlying mathematics has been documented in earlier referenced studies, emphasis here will be on a brief description of each model and those areas where refinements were incorporated as a part of this study.

The purpose of the adaptive math models (AMM) is to systematically search Boolean time sequences (BTS) for various characteristics and determine if the characteristics are related to performance measurement. A block diagram of this process is shown in Figure 12. Smoothed flight data is directed to Boolean logic which processes the data and develops Boolean functions designed to succinctly represent critical performance-related information contained in the data. The output from the Boolean logic is a set of Boolean time sequences which are directed to three processes: relative, absolute, and state transfer. Each of these processes searches for different types of measures, as discussed in the Data Smoothing section. The processes are two-step operations in which first, the Boolean time sequences are systematically searched for characteristics and relationships potentially related to performance measurement. When useful characteristics are detected, a test is conducted to determine their significance to measurement. Finally, the outputs of each of the three processes, which are intermediate performance measures, are combined in a weighted sum to provide an overall performance measure for evaluation.

To establish the notation used in following sections, consider a Boolean time sequence where a single bit of the sequence is represented by BTS_{ij}^{k}. The first subscript (i) represents the Boolean function which generates the BTS, and the second subscript (j) identifies the jth element of that sequence. Thus, Boolean time sequence i is given by:

$$BTS_{ij}^{k}; j = 1, M_k$$

where M_k is the number of elements in the sequence. The superscript (k) is used to indicate the flight event or flight maneuver number associated with the Boolean sequence. It is seen that M_k is a function of k only and not i, because every Boolean sequence generated with data from flight event k has the same length. When reference is made to a total BTS for a specified flight event, the notation BTS_i^k is used.

a. *Boolean Function Data Representation.* A special transformation of the data is performed to simplify its analysis and to permit the user to interact with the processor by adding to it his knowledge of the problem. The transformation results in representation of the data in the form of BTS produced by applying a sequence of performance demonstration data samples to Boolean functions (BF). Two types of BF are constructed and will be discussed in turn:

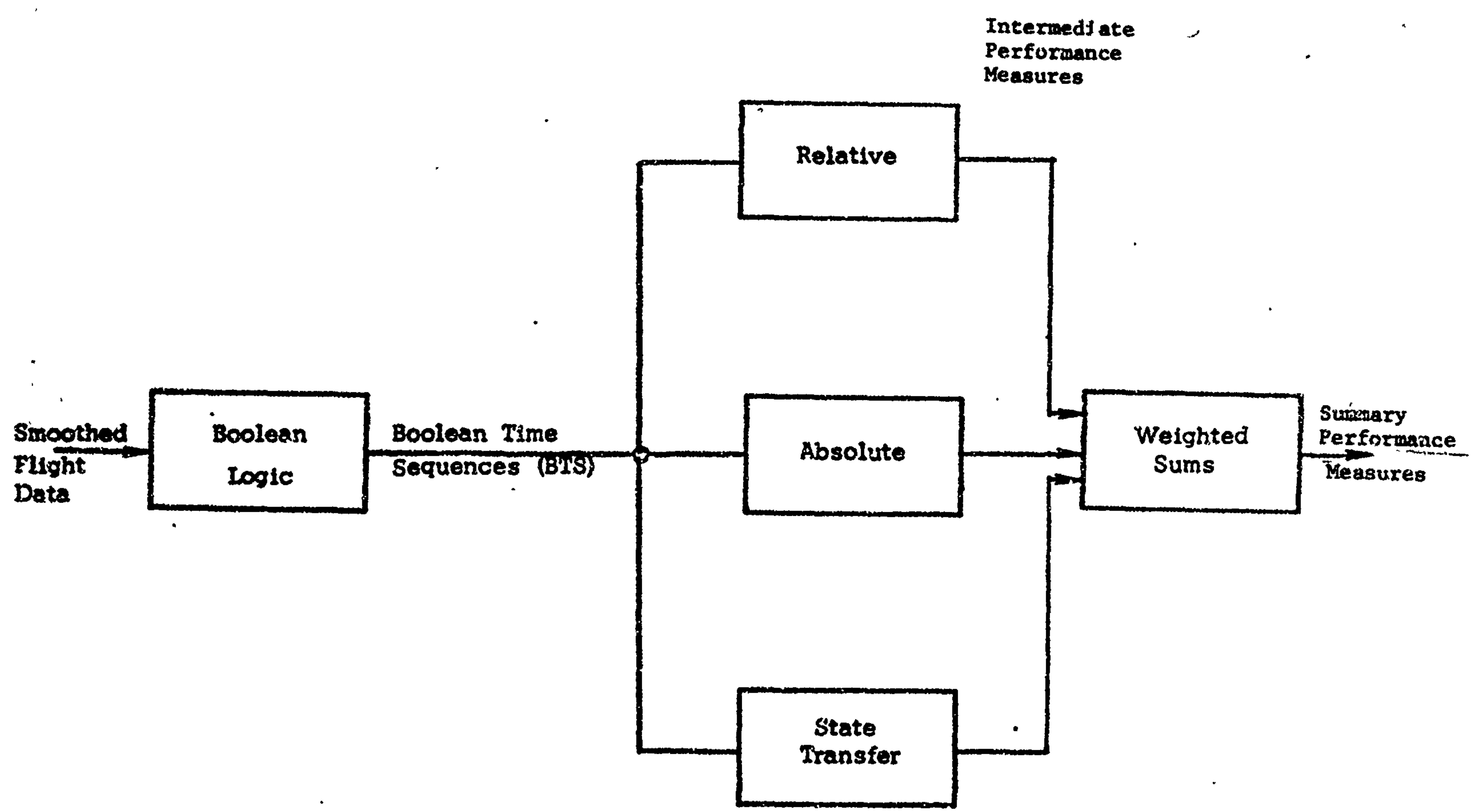


Figure 12. **Block diagram of adaptive math model search process.**

- Functions representing the raw data itself, as expressed in terms of discretized deviations from criterion (or reference) functions; and
- Functions which are demanded by the user and reflect his knowledge about features of the performance that are considered by him to be relevant to measurement.

(1) Functions Representing the Raw Data

Without relying on guidance from the user, BF are constructed by modeling the deviation of the various parameters from standard profiles or time-histories. This is accomplished by, first, spanning the envelope of performance with amplitude test bands, as illustrated in Figure 13. The test bands (labeled 1 through 5 in Figure 13) are determined empirically and consist of multiples of standard deviations around the reference function, computed using performances of skilled subjects. BF are constructed to represent activity in each test band. The BF are set true only when the actual performance data sample is within the limits of the respective test bands.

(2) Functions Demanded by the User

The user may construct special BF by asking pertinent yes-or-no questions about the performance. The answers (1 or ϕ) then form the values of the associated BTS. For instance, the user may have reason to believe that whether or not a pilot's turns are consistently coordinated is particularly relevant to measuring his performance on a given maneuver. Therefore, he may ask whether

$$\text{Roll} = f(\text{pitch, rate-of-turn}, \ldots) \pm \delta$$

where the function f is designed to model a coordinated turn. If indeed this information is relevant to measurement, then the level of activity (percentage of time true) of the associated BTS would probably be a good performance measure.

As a second example, there may be reason to believe that performance at critical points in the maneuver is particularly relevant. To augment the processor with this information, the user might pose the question, "is the performance currently at a critical point (like Pitch = Max Pitch ± 10°)?" The resulting BF then identifies that point in the maneuver. In this particular case, the associated BTS itself may not be relevant; rather, its logical relationship with other BTS (i.e., what is happening at the critical point) would probably be of most value.

b. *Absolute Measures.* The absolute computation mechanism consists of a correlation of each BTS against a fixed set of functions or sequences (MacDonald Codes). This results in the transformation of a long sequence (BTS) into a new set of non-Boolean variables which in turn can be examined to determine if they are relevant to performance evaluation.

Correlation against an absolute reference allows a search for measurement-significance of particular sequences or patterns as they are generated by the Boolean functions. If it is found that some BTS pattern is likely to be predictive of skilled operator performance, this information can serve as a basis for specification of automatic scoring systems as well as provide clues about the operator techniques used in achieving superior performance. The absolute measure also allows analysis employing multiple BTS as well as a single BTS via a regression computation. This provides the tools required for a systematic study of which Boolean function and combinations thereof are relevant to measurement.

The absolute computation is defined as:

$$C_{fi}^{k} = \frac{1}{N_2 - N_1} \sum_{j=N_1}^{N_2} BTS_{ij}^{k} \times Z_{fj}, \; N_2 - N_1 \leq M_k$$

where Z_{fj} is an element in a reference sequence. The subscript (f) indicates which reference sequence is being used. Note that the summation is not conducted over the total length of the BTS; rather, it is computed over a short interval of the BTS. There are two factors that lead to this approach. First, every performance does not require the same length of time and as a result M_k is not a constant. Thus, the

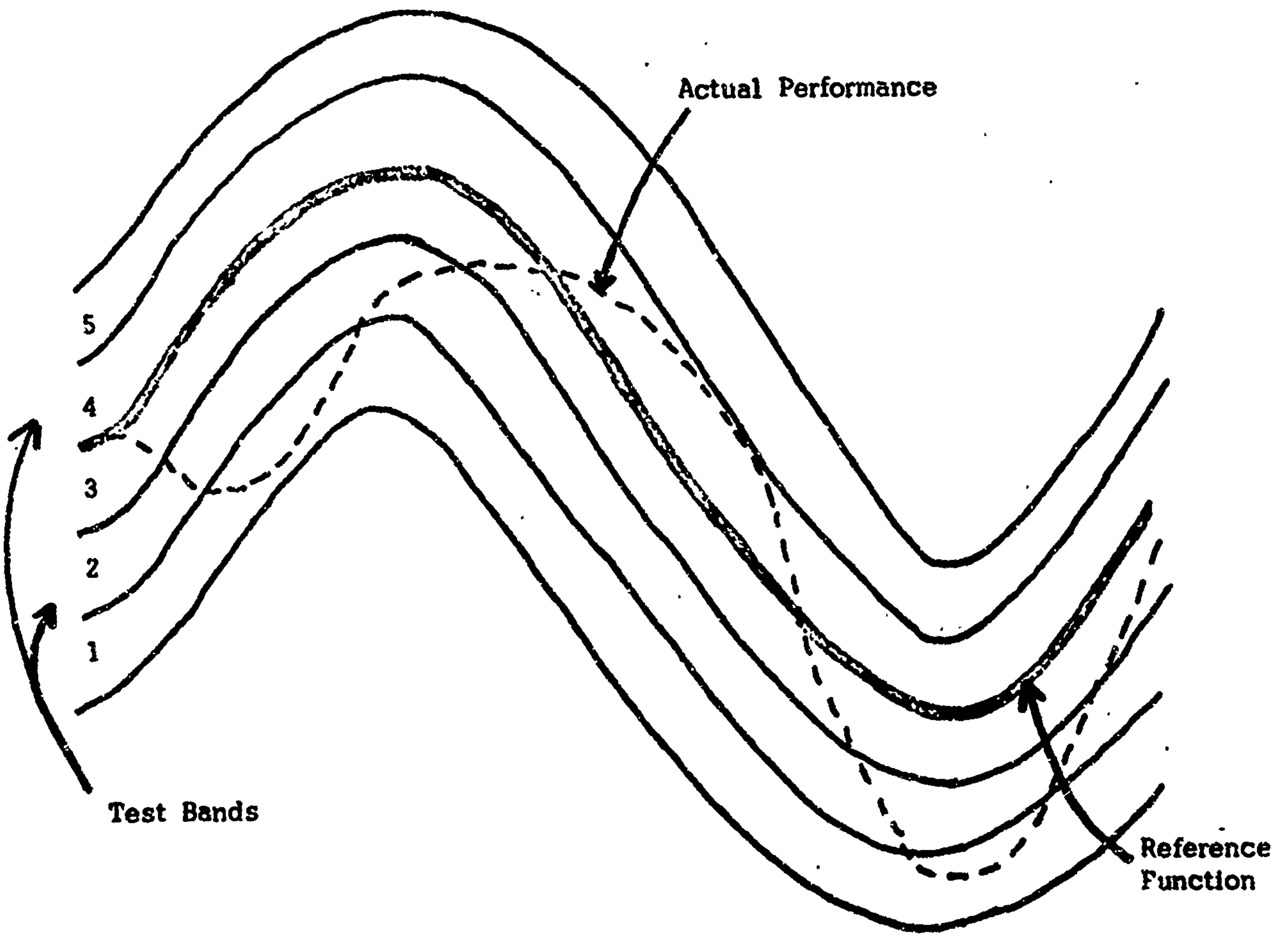


Figure 13. **Amplitude test bands for boolean function construction.**

summation must not be over an interval greater than the smallest value of M_k. Second, and far more important, is that the set of reverences fj should be a set of orthogonal binary sequences in order to have an efficient reference set. (It can be shown that there are N such sequences N elements long; i.e., fj is a square matrix.) The absolute measure can also operate on threshold variables (see Adaptive Mathametrical Models section) instead of the BTS. In that case, the computation is defined as:

$$C^k_{fi} = N_2 - N_1 \sum_{j=N_1}^{N_2} R^k_{ij} \times Z_{fj}, \quad N_2 - N_1 \leq M_k$$

The process implemented in the computer program uses a Hadamard transform producing coefficients from all reference sequences in one efficient operation rather than N operations. However, the process is described by the equivalent correlation operation with a reference sequence.

The length of the reference sequence (and therefore the number of reference sequences) is taken as a varaible 2^M, M = 2, 3, 4, 5, 6, in order to facilitate generation of the Hadamard transform. As stated previously, the reference sequence length must be adjusted to accommodate the length of the BTS; however, the length of the BTS is not the only factor of interest. The optimum length of the reference sequence required to produce sensitive performance measures is not known. It is known that the BTS pattern can be searched in several ways. Two possible ways are shown in Figure 14. Method A shown in the figure requires a correlation of the 4 bit reference sequence to 4 bits (in general 2^M) of the BTS followed by correlation of the reference sequence with BTS bits 5-8, etc., until all BTS bits have been processed. Since there are 4 reference sequences (of 4 bits each) the process is a multi-pass operation. The equivalent transform operation requires one pass. It can be shown that the values of the 4 coefficients for each shift uniquely specify the BTS and no information is lost by the correlation (transform) operation. (Preservation of information may or may not be necessary or a sufficient requirement in performance measurement. In fact, it is easily seen that performance measurement is an operation in which information is discarded systematically, thus reducing a great volume of data to a few variable values representing performance.)

Method B employs a correlation (transform) operation followed by a shift of one bit, followed by a second correlation, etc. This method allows examination of each sequence of 2^M bits and for that reason is preferred. Various length reference sequences can be processed without risk of an incomplete correlation at the end, due to a BTS length not equal to a multiple of 2^M. Thus, the processor is designed to employ reference lengths 2^M, M = 2, 3, 4, 5, 6, and use the shift pattern shown as Method B in Figure 14.

Each correlation operation produces one value of the correlation coefficient and there are N + 1 correlations, where N is the number of shifts. If the BTS has L bits and the length of the reference sequence is 2^M, there are $L-2^M + 1$ correlation coefficients (C) for each BTS and reference sequence combination.

Detection of "patterns" in each BTS is accomplished by analysis of the distribution of the correlation coefficient (C) values obtained from each channel (BTS and reference sequence combination). A fundamental question is determining the C distribution that might result from a random BTS (i.e., without consistent patterns). Consider a random BTS where each bit of the sequence has a probability of 0.5 of being a 1 or −1. This population has a mean and variance of:

$$\mu = 0$$
$$\sigma^2 = (-1)^2 \times .5 + 1^2 \times .5 = 1$$

The correlation operation can be considered as a summation of N elements of that population and the distribution of summation has a mean and variance

$$\mu_N = N\mu = 0$$
$$\sigma_N{}^2 = N\sigma^2 = N$$

BTS	1	-1	-1	1	-1	-1	1	1	1	1	-1	-1	1	1	-1	-1
METHOD A																
Start	1	-1	1	-1												
1st Shift					1	-1	1	-1								
2nd Shift									1	-1	1	-1				
3rd Shift													1	-1	1	-1
METHOD B																
Start	1	-1	1	-1												
1st Shift		1	-1	1	-1											
2nd Shift			1	-1	1	-1										
3rd Shift				1	-1	1	-1									
. . .								.	.	.						
Nth Shift													1	-1	1	-1

Figure 14. **Methods of pattern search.**

The distribution of σ_N^2 is of specific interest since a computed sample variance from flight data is to be used to detect the existence of data patterns. Thus, the .90 and .95 probability points of the cumulative distribution for σ_N^2 are desired for an automatic decision threshold so that the number of false pattern detections (Type I error) can be controlled. Analysis of this problem resulted in the identification of appropriate decision thresholds for incorporation in the processor.

c. *Method of Constructing System States for State Transfer Measures.* Previous efforts in constructing Boolean functions for the purpose of proficiency measurement, employed one or more threshold states for each flight variable (Connelly et al., 1969 & 1971). It was observed in those studies that improved results could be obtained using a new variable derived from deviations from reference functions. For example, in the lazy 8 maneuver, a function relating pitch and roll angles was used to provide a reference. Threshold states were constructed as $K\sigma$ displacements (i.e., multiples of standard deviations) from this reference function. These threshold states, sequences of threshold states, and patterns of threshold states are used to obtain estimates of the system performance.

While this method works well, it is desirable to extend the method such that threshold states for more than one reference function can be considered collectively. For example, a state transition measure should be more effective where the states reflect collective deviations from more than one reference. However, the number of combinations of threshold states can be large which leads to problems in computation, data storage, and data collection.

It is possible to combine several threshold state combinations into performance states with what is believed to be a reasonable way. Benefits from such an approach include simplified computation while maintaining a "physical" interpretation of the performance states. It is necessary to compare each sector of actual data to this reference path to obtain an indication of how much deviation exists between the two (Figure 15). Instead of listing the sequence of the residuals ($e_i = \theta_i - \hat{\theta}_i$) for each sample, a new sequence can be written as follows. If $|e_i|$ is less than 1σ, define $R_\theta = 0$. If e_i is between 1σ and 2σ, let $R_\theta = 1$, etc. Now, the sequence of residuals is reduced to a sequence which contains values 0-4 which are states. The sequence R_θ can be considered as a function state since it shows the progression of states followed by the flight. This process can be generalized as follows:

Consider a set of reference functions as follows:

$$\hat{\theta} = f_\theta (\phi, \theta\text{Max}, \phi\text{Max})$$

$$\widehat{AS} = f_{AS} (\phi, \theta\text{Max}, \phi\text{Max})$$

$$\hat{h} = f_h (\phi, \theta\text{Max}, \phi\text{Max})$$

$$\hat{\psi} = f (\phi, \theta\text{Max}, \phi\text{Max})$$

where

ϕ is roll angle

θ is pitch angle

AS is airspeed

h is altitude

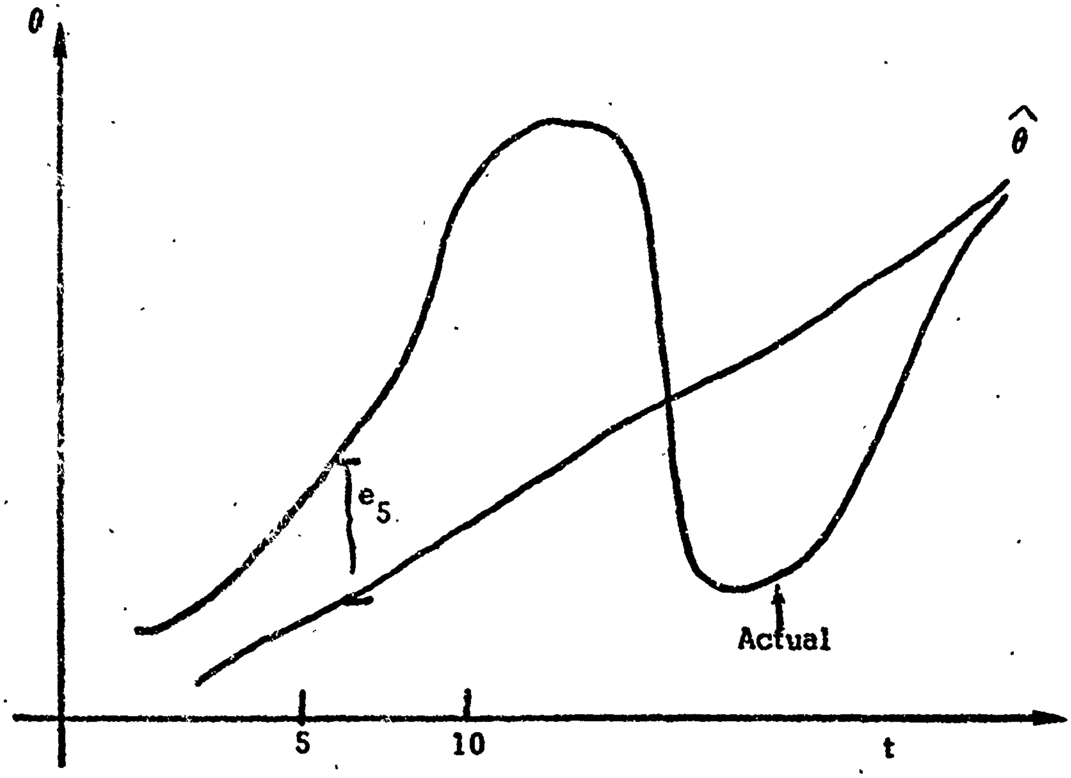
ψ is heading

$\hat{x}$ is the estimate of variable x.

The error function for θ is given by:

$$E_\theta = \theta - f_\theta = E_\theta (\theta, \phi, \theta \text{ Max}, \phi\text{Max})$$

$$E_{\theta\sigma} = E_{\theta\sigma} (\phi, \theta\text{Max}, \phi\text{Max})$$



Sequence of residuals: $\{e_1, e_2, e_3, e_4, \ldots e_T\}$

Sequence R : $\{1, 1, 1, 2, 2, 3, 3, 4, 4, 3, 1, \ldots\}$

Boolean Time Sequence:

$$\text{BTS} = \begin{cases} 0 & R \leq 2 \\ 1 & R > 2 \end{cases}$$

$$= \{0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, \ldots\}$$

Figure 15. **Deviation between actual data and reference path.**

where the value of $E_{\theta\sigma}$ is one sigma deviation function for the distribution of θ given (ϕ, θMax, ϕMax).

In a similar way, additional error and one sigma deviation functions can be formed

E_{AS} (AS, ϕ, θMax, ϕMax)

$E_{AS\sigma}$ (ϕ, θMax, ϕMax)

E_h (h, ϕ, θMax, ϕMax)

$E_{h\sigma}$ (ϕ, θMax, ϕMax)

E_ψ (ψ, ϕ, θMax, ϕMax)

$E_{\psi\sigma}$ (ϕ, θMax, ϕMax)

Define threshold variables as:

$R_X = 0$ when $|E_X| < k\, E_{X\sigma}$

$R_X = 1$ when $KE_{X\sigma} \leq |E_X| < 2k\, E_{X\sigma}$

$R_X = 2$ when $2kE_{X\sigma} \leq |E_X| < 3k\, E_{X\sigma}$

$R_X = 3$ when $3kE_{X\sigma} \leq |E_X| < 4k\, E_{X\sigma}$

$R_X = 4$ when $4kE_{X\sigma} \leq |E_X|$

where X = θ, AS, h, ψ.

A system state can be formulated as:

$$S = R_\theta + R_{AS} + R_h + R_\psi$$

Thus, S values range from 0 to 16. The following diagram illustrates the translation between threshold regions activity and the system state representation.

S	R	R_{AS}	R_k	R
0	0	0	0	0
1	1	0	0	0
1	0	1	0	0
1	0	0	1	0
1	0	0	0	1
2	1	1	0	0
2	0	1	1	0
2	0	0	1	1
2	2	0	0	0
.	.			
.	.			
.	.			
16	4	4	4	4

This method provides a summary performance state which renders several individual threshold states equivalent. The method allows a compact 17 state representation of a system that contains many more states.

In summary, there are several types of system states available. These are:

- **Binary Threshold states**

  A binary threshold state is defined by a binary valued function indicating if the present BTS sample (residual) exceeds a specified level, i.e.,

$$BTS_{ij} = 1 \text{ if } R_{ij} > k_i$$
$$= 0 \text{ if } R_{ij} \leq k_i$$

where R_{ij} is the threshold variable and k is the associated threshold factor.

- **Threshold states**

  A threshold state is the value of the threshold variable $R_X = 0, 1, 2, \ldots$

- **System states**

  A performance state is the value of the system variable(s) given by the sum of the threshold variables $S = R_\theta + R_{AS} + R_h + R_\psi$

The BTS is a sequence with 1/0 valued elements. In a strict sense, only binary threshold states can be represented in a BTS; however, for computation of transition measures threshold and system variables are useful. In the state transfer computational mechanism, the multi-valued variables R and S are used to replace the binary valued BTS.

d. *State Transfer Measures.* The state transfer computation mechanism is a means for determining if performance (or score) information is related to the sequence of operator actions. It may be that operator's performance is partially or totally a function of how he corrects for errors, where he may or may not have caused the errors initially.

In order to implement this computation and also provide a convenient means for compactly representing long Boolean time sequences, a transition matrix is formed which identifies how the sequence moves from state to state. Also, a composite transition matrix can be formed to represent transition patterns from all demonstration data sets (DDS) of a given performance level.

A Boolean state is defined by the set of binary values associated with the set of selected BTS as described in the previous section. Each DDS can be viewed as a sequence of Boolean states or, alternatively, as a sequence of state transitions termed "transtates." The state transition measure seeks to relate the frequency of use of each transtate to performance measurement. This is accomplished by associating with each transtate a score value stored in an incremental score matrix (ISM). A performance measure value is produced by summing the score values from the element of the ISM corresponding to each transtate used in the DDS. The final performance measure value is obtained by dividing the sum by the number of transitions. The transition matrix is formed using the system state values

$$S = 0, 1, 2, \ldots, 16$$

A 17 x 17 transition matrix is required to store all transition probabilities.

If we assume that the DDS state sequence can be described as a Markov Process, the sequence can be represented by its transition matrix. Under this assumption, the performance measure can be computed in another way. A fundamental theorem for Markov Chain processes states that if π_o is the initial probability vector (probability distribution density), then π_n (the probability vector after n trials) is given by

$$\pi_o = \pi_o T^n \qquad (1)$$

where T is the process transition matrix. Proof of the results given here can be found in Connelly et al., 1969.

Now, we assume that the process is a regular Markov process. Such a process is identified by a transition matrix (T) where for some value of n, T^n has no zero elements. This implies that the system could be in any state after N trials, independent of the initial state.

The assumption that the system is a regular Markov process lets us state that there exists a unique probability vector (a) such that

$$\text{as } n \to \infty, \text{ limit } \pi_n = a \qquad (2)$$

Elements of this probability vector correspond to the probability of the system being in the associated state.

Furthermore, it is seen that

$$aT = a \qquad (3)$$

must hold due to Equations (1) and (2). This limiting probability vector gives the first state distribution desired. Note that a is the probability distribution of finding the process in each state, given that we have not observed the process previously.

The distribution of transtates can be determined by imagining an ensemble of many adjustment systems with first states distributed according to a. The second states for each system are determined according to the transition matrix T. This yields new states, also with a distribution a (according to Equation 3). The probability that transition i-j is used in the operation is the probability of the transition (T_{ij}) given the first state times the probability of being in state i (a_i).

Let a' be an N by N matrix with zero value elements off the main diagonal. Also, the elements (a'_{ii}) of a' are given by

$$a'_{ii} = a_i$$

A new N x N matrix (D) is defined as

$$D = a' T$$

Elements of D ($a_i\, T_{ij}$) are the probabilities of the system being in each transtate assuming the first (or any other) state is not known.

The probability matrix (D) can be used to establish the equivalent population statistics. Elements of D (i,j) may be considered normalized weighting factors, and elements of ISM (i,j) provide the population values. The population mean (P) is

$$P = \sum_{i=1}^{N} \sum_{j=1}^{N} D(i,j)\, ISM(i,j)$$

P is equivalent to the performance measure computed from state transition as described previously. In addition, we now have the tools for computing values for the ISM. The method is to form a representative transition matrix for two or more performance levels of the DDS. In this way, one transition matrix represents excellent performance and another represents another performance level, etc. Once these composite transition matrices are available, the elements of ISM can be adjusted (trained) to improve the performance measure discrimination capability between (or among) the demonstrated performance categories. The method is to sequentially adjust the ISM using one transition matrix at a time and to continue the process until a measure with stable discrimination capability is obtained. There are alternative methods of adjusting the ISM, but this iterative method converges rapidly and allows introduction of new data as it is obtained. The iterative method requires computation of the amount each element in ISM should be changed in order to modify the score (measure value) by a specified amount.

The probability matrix D (i,j) can be used to compute the expected value of score change by means of the adjustment process. For each transition, the probability that transtate (i,j) is used (assuming we do not know the present state) is D (i,j). Thus, the expected change in the associated incremental score, ISM (i,j), at each transition is the product

$$\Delta \cdot D(i,j)$$

where Δ is the amount added to the ISM increment. Since n_2 transitions are used in the sequence, the total expected change is given by:

$$n_2 \cdot \Delta \cdot D(i,j)$$

If we use ISM $(i,j)'$ to represent the expected updated incremental score matrix, we see that:

$$ISM\,(i,j)' = ISM\,(i,j) + n_2 \cdot \Delta \cdot D(i,j)$$

The expected mean score is

$$P' = \sum_i \sum_j D(i,j) \cdot ISM(i,j) + n_2 \cdot \Delta \cdot (D(i,j))^2$$

and the expected change in mean score (CM) is

$$CM = n_2 \cdot \Delta \cdot \sum_i \sum_j (D(i,j))^2$$

The summation term can be considered as the system gain G, such that

$$G = \sum_i \sum_j (D(i,j))^2$$

(Note that G will be less than one.)

Thus, the amount that must be added to each element of ISM is $n_2 \cdot \Delta \cdot D(i,j)$ in order to change the measure value by amount CM.

e. *Relative Measures.* The relative computation technique operates on up to four Boolean functions simultaneously to determine if logical relationships exist and, if so, how the relationships are associated with performance. Thus, as opposed to the absolute computation where operations are performed on a single BTS, the relative computational technique uses a "trainable logic" concept to detect possible relationships among BTS. The approach is to select a set of base BF channels and form all combinations of these sequences for each data sample. For example, if three base BF channels are selected, eight combinations can be produced. Only one combination is true for each sample. Correlation of one combination (say combination j) with an additional BF channel (say BF_i) yields the conditional probability that BF_i is true given that combination j is true.

Each combination can be expressed as:

$$C_m(j) = \left(BTS^k_{cj}\, BTS^k_{bj}\, BTS^k_{aj} \right)$$

where the first subscript indicates the BTS, and the superscript indicates the DDS. The second subscript (j) indicates the element in the sequence. The AND operation is conducted on a bit-by-bit basis. Therefore, the combination $C_m(j)$ has a binary value corresponding to each element in the BTS; i.e., $C_m(j)$ is a Boolean sequence itself.

As an example, we determine combination 5. It is convenient to represent m (or 5 in this case) by its binary form (101). Thus, using the binary form of m to code the combination, we find

$$C_5(j) = BTS^k_{cj}\ \overline{BTS^k_{bj}}\ BTS_{aj}$$

where $\overline{BTS}$ is the NOT function. Note that the second subscript need not be the same for each term; i.e., we can compute using relative time positioning by selecting (for example) second subscripts (j, j+2, j+3, j−1). While this flexibility is available it has not been extensively explored to date.

The analysis consists of the following operations. First, a number, SUM_N, is computed which is a count of the number of times each combination occurs during the flight event under study. Thus,

$$SUM_N = \sum_{j=1}^{M_k} C_n(j).$$

A second count is formed which represents the number of times a BTS variable is true, given that combination C_n is true. Thus,

$$SUM_C = \sum_{j=1}^{M_k} C_n(j)\, BTS^k_{ej}$$

Normally, the first subscript e would have a different value than those of the base variables. Next, the conditional probability that BTS^k_{ej} is true given that combination n has occurred is determined as

$$P_c(n) = \frac{SUM_C}{SUM_N}$$

This conditional probability identifies the relationship between each combination of the base variables and the predicted variable. These conditional probabilities are candidate performance measurement variables and are tested for validity, as described in Validation Tests section.

f. *Summary Description of Measures.* Figure 16 illustrates a 2-dimensional state-space defined by roll and pitch angles, and the approximate trajectory outline of one quarter of a lazy 8 maneuver. The small arrows in the figure depict alternative directions in which a roll/pitch trajectory might move in a given performance. The State Transfer type of measure is based on probabilistic assessments of this direction of movement. To compute the probability values, the state-space is gridded into discrete states. (For instance, each cell or rectangle in Figure 16 may be considered an individual state.) By computing the frequency with which each state is acquired and the state sequence, the probability of transfer from state to state is calculated.

The State Transfer computational mechanism can operate on up to four flight variables at a time. To minimize the number of states to be handled simultaneously and the associated computational complexity of the problem, threshold states are used which represent the sum of the deviation units from each criterion function. For example, in Figure 16 the shaded cells might be recorded as threshold state number 1 (depending on computed performance variance and resulting cell sizes used to model each performance), because they are located one deviation unit from the reference trajectory. This state representation not only reduces problem complexity, but permits ready interpretation and assessment of divergence from or convergence on criterion terminal performance.

The Relative measure is based on conditional probabilities of various states being acquired simultaneously with the acquisition of other states. For instance, consider the user-defined BTF's of (1) Pitch = maximum pitch $\pm\Delta_1$, and (2) airspeed = $A_K \pm \Delta_2$, where A_K is a criterion airspeed value. Analyzing

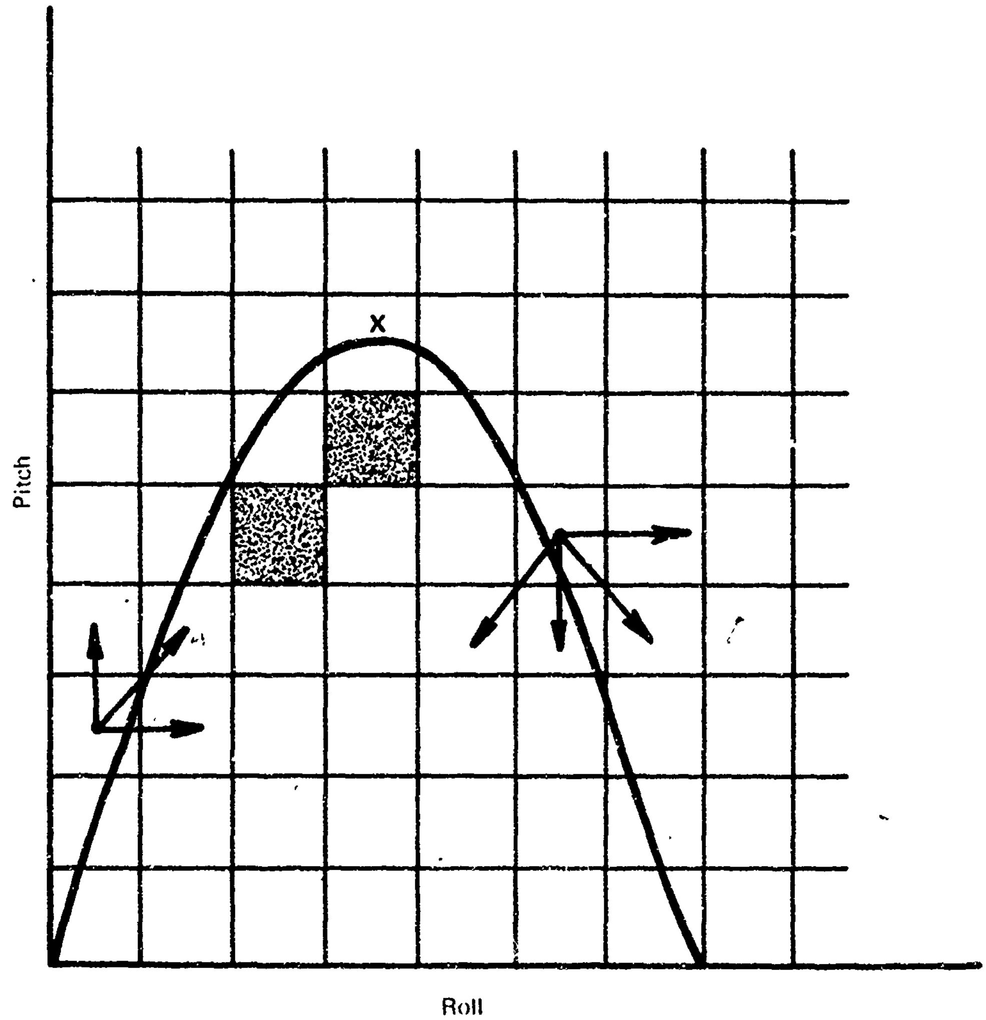


Figure 16. Representative roll/pitch state space.

these BTF's along with the information in Figure 16, the Relative computational mechanism would examine as a potential measure the probability that pitch is maximum whenever airspeed $\doteq A_K$ and state X in Figure 16 is active. (This is precisely a significant factor of criterion lazy 8 performance as suggested by Air Training Command flight manuals and as substantiated empirically in previous studies (Knoop & Welde, 1973)). This is only one example of hundreds where simultaneity of significant events bears on successful performance, and the role of the Relative computational mechanism is to explore the relevance to performance measurement of conditional probabilities. When one considers the plausible theory that much of performance on continuous control tasks can be modeled by discrete successive acquisitions of key states, the value of the Relative type of measure and its role in the processor becomes clear.

The Absolute computational mechanism essentially amounts to computing a discrete version (Hadamard Transform) of the Fourier Transform, wherein the power of various frequency components of the signal (BTS) is assessed. Each transform operation produces a number of coefficients (correlation values), which are summarized by their mean and variance. Previous work (Connelly, et. al., 1969) has shown that the variance of the Hadamard coefficients is most useful in discriminating performances of various skill levels. The Absolute mechanism generates these variances for subsequent validation testing, described next.

Validation Tests

a. *Description.* For the vast majority of performance tasks, there is no single necessary and sufficient test that can be applied to candidate measures to assess their validity. Measures which appear to have content validity often fail to reliably discriminate even between novice and highly experienced performers. Measures which appear to have concurrent validity may or may not satisfy other validation criteria, depending on the reliability and sensitivity of the metric used as a basis of comparison.

The approach in this study was to develop three empirically-based validation tests to be applied by the measurement processor. Collectively, the tests are used to determine the likelihood that each candidate measure is valid. Final analysis and assurance of the measure's content validity is performed by the user of the processor, based on the evidence accrued by it and printed out for his consideration.

The first test assesses the measure's potential contribution to discriminating between performances at opposite ends of the skill continuum. The data employed for this test are selected by the user. For the T-37 pilot performance tasks that were to have been addressed here, the following two types of data would have been investigated:

(1) Flights flown by instructor pilots to demonstrate their best performances and simulated novice performances of each maneuver.

(2) Flights flown by students at the neophyte stage and at the successful completion of training.

The techniques implemented to apply this first test include: (a) comparison of residues from regression analyses, and (b) the rank sum statistic (see Validation Tests section).

The second test assesses the measure's functional relationships with variables such as number of trials and time in training. A measure which demonstrates that learning has occurred from neophyte to experienced levels of performance would posses a higher likelihood of validity than one which consistently does not, for example. Again, the data to be employed for this test are specifiable by the user. For the T-37 pilot tasks, the following data would have been experimented with: (1) time in training, (2) Number of practice sorties on the maneuver, and (3) number of practice trials on the maneuver. The technique used to apply this test consists of developing and analyzing a multi-variable regression function. (An alternative technique based on the use of Markov learning models was conceived, but due to lack of data, has not yet been developed to the point of implementation.)

The third test assesses the measure's functional relationships with subjectively derived ordinal scale measures of performance. Measures which tend to reinforce the subjective ordering of performances are considered more likely to be valid than those which consistently fail to do so. The data employed for this test, as with the other tests, are specified by the user. For the T-37 tasks, instructor pilot ratings would have been investigated for use. The technique for applying the test is to develop and analyze multi-variable regression functions, as in the second test described in the preceding paragraph.

The regression techniques used for applying some of the above validation tests were described previously. The rank sum test is described next.

b. *Rank Sum Statistic.* The computer-aided generation of performance measures requires the systematic generation and evaluation of many candidate measures. It is necessary to assess these measures' potential contribution to overall performance measurement. One aid in accomplishing this using the rank sum statistic was developed for investigation in this study.

Consider a process where data are available from two performance classes (e.g., flights produced by instructor pilots and flights produced by neophyte student pilots). Candidate proficiency measures will yield two sets of quantitative variable values when applied to the data from these two performance classes. It is possible to test these sets to determine if they come from different parent distributions. If they do come from different parent distributions and there is little overlap in the distribution functions, then the candidate measures are highly likely to be useful in proficiency assessment; i.e., there is a high probability that the tested measures will satisfy other validation criteria. On the other hand, if these two distributions have considerable overlap, then the measures would probably not be very useful.

A test for determining the degree of similarity of the two parent distributions can be selected from statistical analysis hypothesis testing where one can initially assert that the two distributions are the same (null hypothesis). One method of testing the hypothesis that distributions $F_1(X)$ and $F_2(X)$ are equal is the rank sum test. This is a simple non-parametric test which indicates the likelihood that two sets of data, which may be of different sizes, come from the same distribution.

Test of the hypothesis that $F_1(X)$ is equal to $F_2(X)$ is developed as follows:

Let $X_1, X_2, \ldots, X_{n_1}$, and $Y_1, Y_2, \ldots, Y_{n_2}$

denote random samples of two sizes, n_1 and n_2, taken from populations with continuous density functions $F_1(X)$ and $F_2(X)$, respectively. Let these two sets of samples be ordered in increasing magnitude and combined to a single ordered set where a possible arrangement might be as follows:

$Y_1, Y_2, X_1, Y_3, X_2, Y_4$, etc.

Of special interest is the sum of the ranks of the smaller set n_1 (where $n_1 \leq n_2$). (For example, the ranks of the X's are 3, 5, etc.) The sum of these ranks is a statistic of known distribution for given values of n_1 and n_2. Therefore, the statistic value can be used as an indicator that the hypothesis $F_1(X) = F_2(X)$ is valid. Table 13 gives the critical values or limits of a 95 percent confidence interval for small values of n_1 and n_2. (The significance level of the rank sum test is not preserved if the two populations differ in dispersion or shape. Whether or not they differ in this way is expected to depend on the measure under test. Plans to analyze this empirically for the various measures on the T-37 problem and, as required, develop techniques to account for observed effects did not materialize due to inability to collect required data.) This table is taken from (Hoel, 1962) and applies for values of n_1 and n_2 less than 10. For larger sample sizes the distribution is approximated closely by the normal distribution with a mean and variance given as follows:

$$\text{Mean} = n_1 (n_1 + n_2 + 1)/2$$

$$\text{Variance} = \frac{n_1 n_2 (n_1 + n_2 + 1)}{12}$$

V. SUMMARY AND CONCLUDING REMARKS

A computer-aided system has been developed and implemented for use in deriving and validating measures of operator performance. Its uniqueness is characterized by: (1) a logical division of human and computer-processor functions, integrated through an interactive man/machine systems approach to measurement research; (2) an experimental approach to deriving measures by generating vectors which span various conceivable measure spaces and operating on the vectors using multiple regression analysis; and (3) a systematic empirical approach to validation-testing of candidate measures to assess their likelihood of contributing to overall performance measurement.

One of the most important features desired in the processor is its ability to automatically generate and test candidate performance measures with a minimum of inputs from the user. The processor successfully implements this desired feature in that it reads in raw performance data and prints out tested performance measures. To do this, it first automatically performs data smoothing; i.e., removal of noise in the data. It then performs logical sectoring in which maneuvers are automatically divided into sectors that can be conveniently analyzed in subsequent processing. Next, the processor automatically applies a regression analysis procedure to establish criterion performance in the form of simple regression functions. The independent variables for each function were selected based on their being monotonic over the maneuver sector of interest. Finally, the processor applies adaptive mathematical models to the data, and based on types of deviations (BTS) from the criterion performance functions, generates and tests for validity a variety of performance measures. It is truly an automatic processor.

Table 13. **Rank Sum Critical Values***

The sample sizes are shown in parentheses (n_1, n_2). The probability associated with a pair of critical values is the probability that $R \leq$ smaller value, or equally, it is the probability that $R \geq$ larger value. These probabilities are the closest ones to .025 and .05 that exist for integer values of R. The approximate .025 values should be used for a two-sided test with $\alpha = .05$, and the approximate .05 values for a one-sided test.

	(2, 4)			(4, 4)			(6, 7)	
3	11	.067	11	25	.029	28	56	.026
	(2, 5)		12	24	.057	30	54	.051
3	13	.017		(4, 5)			(6, 8)	
	(2, 6)		12	28	.032	29	61	.021
3	15	.036	13	27	.056	32	58	.054
4	14	.071		(4, 6)			(6, 9)	
	(2, 7)		12	32	.019	31	65	.025
3	17	.028	14	30	.057	33	63	.044
4	16	.056		(4, 7)			(6, 10)	
	(2, 8)		13	35	.021	33	69	.028
3	19	.022	15	33	.055	35	67	.047
4	18	.044		(4, 8)			(7, 7)	
	(2, 9)		14	38	.024	37	68	.027
3	21	.018	16	36	.055	39	66	.049
4	20	.036		(4, 9)			(7, 8)	
	(2, 10)		15	41	.025	39	73	.027
4	22	.020	17	39	.053	41	71	.047
5	21	.061		(4, 10)			(7, 9)	
	(3, 3)		16	44	.026	41	78	.027
6	15	.050	18	42	.053	43	76	.045
	(3, 4)			(5, 5)			(7, 10)	
6	18	.028	18	37	.028	43	83	.028
7	17	.057	19	36	.048	46	80	.054
	(3, 5)			(5, 6)			(8, 8)	
6	21	.018	19	41	.026	49	87	.025
7	20	.036	20	40	.041	52	84	.052
	(3, 6)			(5, 7)			(8, 9)	
7	23	.024	20	45	.024	51	93	.023
8	22	.048	22	43	.053	54	90	.016
	(3, 7)			(5, 8)			(8, 10)	
8	25	.033	21	49	.023	54	98	.027
9	24	.058	23	47	.047	57	95	.051
	(3, 8)			(5, 9)			(9, 9)	
8	28	.024	22	53	.021	63	108	.025
9	27	.042	25	50	.056	66	105	.047
	(3, 9)			(5, 10)			(9, 10)	
9	30	.032	24	56	.028	66	114	.027
10	29	.050	26	54	.050	69	111	.047
	(3, 10)			(6, 6)			(10, 10)	
9	33	.024	26	52	.021	79	131	.026
11	31	.056	28	50	.047	83	127	.053

* This table was extracted from a more complete table (A-20) in *Introduction to Statistical Analysis*, 2nd edition, by W. J. Dixon and F. J. Massey, with permission from the publishers, the McGraw-Hill Book Company.

The types of measures generated and tested were defined a priori in terms of underlying characteristics which render them suitable candidates. One type (Relative) represents probability measures related to significant event proximities. Another (Absolute) represents measures of system variable frequencies and periodicities. A third (State Transfer) represents measures of state transitions that occur over time, including system divergence from or convergence upon criterion terminal performance. A fourth type (State Frequency) that was identified but not specifically addressed in this effort uses data generated by the State Transfer computational mechanism to address measures of reference function deviations.

To investigate the above measures, vectors were identified which constitute generators of the measure spaces corresponding to each measure-type. The measures thereby spanned are explored by the various computational mechanisms using regression analysis and a number of empirical validation tests. Table 14 summarizes the measure spaces, the components of vectors by which they are spanned, and the basic functions performed by each computational mechanism.

Table 14. **Summary of Measure Spaces**

No. of BTS Processed Per Iteration	Measure Subspace	Major Function of Computational Mechanism	Components of Generating Vectors	Types of Measures Spanned
4	Relative	Compute Conditional Probabilities	Conditional Probabilities	Probability of Simultaneous Occurrence of Significant Events Or System States
1	Absolute	Perform Hadamard Transform	Hadamard Coefficients; Coefficient Distribution Parameters (μ,σ)	State Variable Periodicities and Response Frequency Characteristics
4	State Transfer	Generate State Frequencies, Transtate Frequencies, and Transition Matrices	Transtate Frequencies; State Transfer Measures Derived Via Transition Matrix Model	Operator/System State Transitions; Transitive and Steady State Movements Relative To Criterion Terminal Performance
	State Frequency*		State Frequencies and Corresponding Deviation Units*	Reference Function Deviations*

*Separate computational mechanism not yet implemented in processor.

The success of the automatic maneuver sectoring is a main factor in processor effectiveness. It allows use of simple regression functions for describing criterion performance since small portions of the maneuvers can be treated separately. Had this automatic sectoring not been feasible, then a considerably more complicated regression function would have been required; i.e., it would have been necessary to attempt to model the entire maneuver or large portions thereof with a single regression function. Preliminary evaluation of the automatic sectoring using two excellent and two poor maneuvers, as rated and flown by IP's, indicates that the sectoring will work in a satisfactory way over a range of maneuver demonstrations. However, since the beginning of each sector is detected when a specific variable amplitude exceeds a threshold limit, it is possible that some maneuver demonstrations (especially those produced by neophyte students) presented to the processor will not be processed properly. Thus, before the evaluation of the automatic sectoring can be considered complete, distributions of these key variable values are required over the range of expected performance demonstrations; i.e., student demonstrations from neophyte to skilled.

Another key area in the development of the processor is the use of a simple regression function that uses the "five sums approach" (Connelly et al, 1969, pp. 179-181), so that additional information can be simply added to the processor as it becomes available. Initial evaluation using two excellent and two poor demonstrations of each maneuver type indicates that the mechanism for developing the satisfactory criterion function is available. It should be noted, however, that although the mechanisms for producing the criterion functions exist in the processor, the data itself must be studied using additional demonstrations of flight performance in order to determine if excellent performance data is clustered about the criterion functions. Such clustering is necessary for establishing useful criterion functions. Should clustering of excellent performance data fail to materialize, as evidenced by a large residual value, maneuver parameters such as θ_{Max} and ϕ_{Max} may have to be included in the regression functions.

The adaptive mathematical models developed and experimented with in earlier studies have now been refined and adapted to use in an automatic processor. Future refinements beyond those now implemented may be easily invoked as required due to the modular design of the software. Whether or not further refinements are necessary or desirable could not be determined in this effort due to the previously mentioned unavailability of sample performance data. However, the central features of what is believed to be powerful and highly useful measurement research tool have been successfully implemented; and hopefully the underlying theoretical concepts and the implementation techniques that were developed and documented herein will, as a minimum, serve to inspire further measurement work along these lines.

REFERENCES

Connelly, E.M., Bourne, F.J., Loental, D.G., Migliaccio, J.S., Burchick, D.A., & Knoop, P.A. *Candidate T-37 pilot performance measures for five contact maneuvers.* AFHRL-TR-74-88. Wright-Patterson AFB, OH.: Advanced Systems Division, 1975, in press.

Connelly, E.M., & Loental, D.G. A performance measurement model for non-linear man-machine control problems. *Proceeding of the 10th Annual Conference on Manual Control,* 9-11 April 1974.

Connelly, E.M., Schuler, A.R., Bourne, F.J., & Knoop, P.A. *Application of adaptive mathematical models to a T-37 pilot performance measurement problem.* AFHRL-TR-70-45, AD-726 632. Wright-Patterson AFB, OH.: Advanced Systems Division, Air Force Human Resources Laboratory, March 1971.

Connelly, E.M., Schuler, A.R., & Knoop, P.A. *Study of adaptive mathematical models for deriving automated pilot performance measurement techniques.* AFHRL-TR-69-7, Volumes I and II, AD-704 597 and AD-704 115. Wright-Patterson AFB, OH.: Air Force Human Resources Laboratory, October 1969.

Gregory, F.D., & Cavanagh, R.A. *Student pilot performance measurement system.* Flight Test Report No. ENE FTR 73-37. Wright-Patterson AFB, OH.: Aeronautical Systems Division, 1973.

Hahn, G.J., & Shapiro, S.S. *Statistical models in engineering.* New York: John Wiley and Sons, Inc., 1968.

Hoel, P.G. Introduction to mathematical statistics. New York: John Wiley and Sons, Inc., 1962.

Kemeny, J.G., & Snell, J.L. *Finite markov chains.* New York: D. Van Nostrand Co., Inc., 1960.

Knoop, P.A., & Welde, W.L. *Automated pilot performance assessment in the T-37: A feasibility study.* AFHRL-TR-72-6, AD-766 446. Wright-Patterson AFB, OH.: Advanced Systems Division, Air Force Human Resources Laboratory, April 1973.

MANEUVER TYPE Cloverleaf SECTOR 2

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	125(L)	124(R)	127(L)	128(R)
h	10.63	9.74	11.72	374.[illegible]6	315.94
ψ	0.77	.59	0.99	1.89	1.25
AS		.82		2.40	53.31
θ					
ϕ	2.95	1.68	4.51	3.06	7.38
#/samples		11.0	9.0		
θ_M		44.4	40.4	41.6	22.5
ϕ_M		5.6	9.5	8.9	8.3

FUNCTIONAL FORM: $DV = B_0 + B_1\theta + B_2\theta^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_0	B_1	B_2	B_3	B_4
h	782.40	3.60	0.31	-14.19	-69.44
ψ	-3.24	.19	-0.00	-0.05	0.44
AS	72.30	.32	-.01	2.96	4.09
θ					
ϕ	-20.01	0.09	0.00	0.22	1.44

MANEUVER TYPE Cloverleaf SECTOR 3

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	125(L)	124(R)	127(L)	128(R)
h	31.90	32.22	31.76	1143.00	9074.90
ψ		38.52		559.62	4751.90
AS		2.16		6.22	211.48
θ	1.00	.88	1.11	3.03	122.09
ϕ					
#/samples		17	19		
θ_M		62.9	60.1	49.[illegible]	-52.4
ϕ_M		178.0	181.0	179.0	166.0

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-2588.00	12.03	-.02	77.[illegible]6	-10.04
ψ	1668.00	0.57	0.00	-42.[illegible]6	5.45
AS	150.70	.70	0.00	-2[illegible]7	2.89
θ	-21.68	.70	-.00	1[illegible]4	-0.13
ϕ					

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	125(L)	124(R)	127(L)	128(R)
h	16.54	17.81	15.35	348.85	2926.30
ψ	14.20	13.99	14.60	79.20	332.24
AS	1.14	1.25	1.02	69.60	57.72
θ					
ϕ	6.98	7.74	6.13	5.58	11.51
#/samples		11	10	9	4
θ_M		-86.9	-81.7	-81.2	-86.9
ϕ_M		179.0	178.0	178.0	166.0

FUNCTIONAL FORM: $DV = B_0 + B_1\theta + B_2\theta^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_0	B_1	B_2	B_3	B_4
h	-29680.00	-2.68	-0.11	78.01	214.20
ψ	-2687.00	-2.45	-0.03	8.08	19.48
AS	980.30	0.13	0.00	-3.62	-6.71
θ					
ϕ	336.3	-1.12	-0.01	-0.81	-1.37

MANEUVER TYPE Cloverleaf SECTOR 5

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	125(L)	124(R)	127(L)	128(R)
h	38.28	35.74	41.72	303.15	718.86
ψ	1.51	1.18	1.97	101.35	23.40
AS	1.80	1.71	1.91	54.19	19.66
θ					
ϕ	6.24	5.73	66.93	5.91	8.25
#/samples		19	14	9	14
θ_M		-79.9	-80.9	-78.2	-82.4
ϕ_M		68.5	75.7	66.5	92.0

FUNCTIONAL FORM: $DV = B_0 + B_1\theta + B_2\theta^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	3806.00	-18.33	-.00	34.14	-22.27
ψ	-2355.00	0.25	-0.00	-20.21	8.37
AS	-161.8	0.80	-0.00	-5.13	-0.43
θ					
ϕ	-11.71	1.25	0.02	-0.20	0.15

MANEUVER TYPE Split S　　　　　　　　SECTOR 1

ERRORS

Dependent Variable	Excellent			Poor	
h					
ψ					
AS					
θ					
ϕ					
#/samples		1			
θ_M		6.44			
ϕ_M		0.01			

FUNCTIONAL FORM:

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h					
ψ					
AS					
θ					
ϕ					

MANEUVER TYPE Split S SECTOR 2

ERRORS

Dependent Variable	Excellent			Poor	
		126(L)		129(L)	130(L)
h		20.41		1271.30	178.71
ψ		0.17		17.95	5.04
AS		1.14		1132.50	140.58
θ		0.57		243.35	35.12
ϕ		.72		5.38	1.31
#/samples		28		31	18
θ_M		28.4		23.3	26.2
ϕ_M		-5.6		6.9	-3.9

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-1251.00	1680.00	12.18	19.61	-105.00
ψ	-8.67	-1.33	-5.85	0.02	-1.41
AS	-357.20	-82.27	-16.46	2.16	-90.56
θ	-123.00	62.83	-44.86	0.69	-19.43
ϕ	7.91	-18.50	18.84	-0.12	0.65

MANEUVER TYPE Split S SECTOR 3

ERRORS

Dependent Variable	Excellent			Poor	
		126(L)		129(L)	
h		1.38		137.79	373.67
ψ		2.62		9.69	61.01
AS		.59		36.63	26.35
θ		.31		5.63	11.05
ϕ					
#/samples		7		9	9
θ_M		26.3		19.6	27.2
ϕ_M		161.0		110.0	118.0

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-235.70	2.34	-0.00	45.07	3.77
ψ	-6.38	0.60	0.01	-4.25	0.73
AS	-3.61	0.16	.00	1.07	0.61
θ	5.23	0.14	-0.00	1.48	-.12
ϕ					

MANEUVER TYPE Split S SECTOR 4

ERRORS

Dependent Variable	Excellent			Poor	
		126(L)		129(L)	130(L)
h		49.47		381.73	342.84
ψ		2.83		86.61	105.01
AS		4.32		18.96	45.27
θ		10.48		16.80	99.70
ϕ					
#/samples		12		11	12
θ_M		-76.6		-57.3	-64.7
ϕ_M		178.0		172.0	159.0

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-4173.00	-20.70	0.11	30.61	47.07
ψ	-372.30	-6.16	0.01	0.96	6.34
AS	473.10	1.57	-0.00	-2.79	-3.34
θ	-628.00	-3.60	0.01	3.99	5.87
ϕ					

MANEUVER TYPE Split S SECTOR 5

ERRORS

Dependent Variable	Excellent			Poor	
		126(L)		129(L)	130(L)
h		3.83		45.92	394.26
ψ		5.52		435.99	58.71
AS		.77		73.78	84.06
θ					
ϕ		2.38		11.24	11.88
#/samples		12		10	11
θ_M		-73.2		-56.0	-64.1
ϕ_M		52.7		60.2	73.3

FUNCTIONAL FORM: $DV = B_0 + B_1\theta + B_2\theta^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-2036.60	-4.83	0.15	12.38	5.50
ψ	-3.53	-1.42	-0.02	-4.99	1.05
AS	9.41	0.05	-0.00	-4.58	-2.32
θ					
ϕ	-3.82	0.63	0.01	0.21	0.60

MANEUVER TYPE Lazy 8 SECTOR 1

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	Good Plus 102	108(L)	113(L)	106(R)
h	4.44	3.18	7.20	18.98	26.24
ψ	0.11	0.08	0.18	0.14	.35
AS	0.40	0.42	0.37	13.83	91.07
θ	0.64	0.48	.98	1.17	19.01
ϕ	0.62	0.48	.93	.68	7.07
#/samples		11	5	13	8
θ_M		7.0	7.3	6.9	10.8
ϕ_M		6.80	4.10	3.3	-3.9

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	250.10	-120.60	150.80	-21.94	-13.55
ψ	.85	.99	1.61	-.09	-.01
AS	398.00	5.18	-7.50	-25.05	-2.52
θ	-24.03	5.44	3.12	3.82	-.81
ϕ	-5.38	0.55	5.80	0.68	.00

MANEUVER TYPE Lazy 8 SECTOR 2

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	Good Plus 102	108	113(L)	106(R)
h	63.88	64.96	63.06	483.84	820.75
ψ	0.89	0.81	.96	9.32	11.76
AS	1.42	1.21	1.59	27.81	37.23
θ	0.31	0.27	0.34	6.44	3.20
ϕ					
#/samples		19	25	19	25
θ_M		27.6	31.4	26.6	43.1
ϕ_M		44.4	45.8	40.6	56.5

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-4410.00	32.47	-0.02	53.69	58.31
ψ	-7.80	0.46	0.00	.02	0.08
AS	387.7	-1.18	-0.00	0.26	-4.08
θ	-53.07	1.27	-0.01	.75	.69
$\dot{\phi}$					

MANEUVER TYPE Lazy 8 SECTOR 3

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	Good Plus 102	128(L)	113(L)	106(R)
h	14.83	17.65	11.78	526.14	3010.40
ψ	3.30	3.93	2.63	37.07	42.26
AS	1.63	1.94	1.29	32.04	173.65
θ	1.80	2.03	1.54	12.41	84.63
ϕ					
#/samples		14	13	12	13
θ_M		27.4	31.3	26.2	40.9
ϕ_M		73.9	89.1	76.7	84.9

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-4848.00	38.43	-0.18	328.9	-61.06
ψ	-3.68	2.47	-.00	3.03	-1.81
AS	402.00	-2.94	0.01	-17.70	4.49
θ	-1-4.20	0.91	-0.01	8.50	-1.61
ϕ					

MANEUVER TYPE Lazy 8 SECTOR 4

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	Good Plus 102	108(L)	113(L)	106(R)
h	29.65	28.18	31.46	325.76	1912.90
ψ	3.65	2.83	4.43	33.30	318.12
AS	2.14	2.12	2.17	3.03	602.64
θ					
ϕ	2.82	2.47	3.24	7.96	249.63
#/samples		16	13	14	17
θ_M		-27.2	-25.0	-26.8	-50.7
ϕ_M		75.9	90.0	82.6	84.7

FUNCTIONAL FORM: $DV = B_0 + B_1\theta + B_2\theta^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	390.50	-8.05	-0.51	32.63	26.75
ψ	-394.10	.78	0.03	-13.25	1.65
AS	-732.2	0.48	0.06	-21.96	3.31
θ					
ϕ	331.80	.57	-0.05	7.52	-.64

MANEUVER TYPE Lazy 8 SECTOR 5

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	Good Plus 102	108(L)	113(L)	106(R)
h	24.66	32.17	19.02	563.96	2892.20
ψ	0.35	.40	.31	17.80	22.28
AS	1.65	2.11	1.30	22.98	60.16
θ	0.96	1.30	.69	1.30	7.26
ϕ					
#/samples		18	24	21	19
θ_M		-27.1	-24.7	-26.1	-47.6
ϕ_M		48.6	50.6	50.6	44.7

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-2083.00	13.86	0.14	151.60	126.00
ψ	159.70	0.08	-0.01	-2.07	5.83
AS	-35.85	0.73	-0.00	-3.99	2.88
θ	-14.80	0.68	0.00	0.16	0.34
ϕ					

MANEUVER TYPE Normal Landing SECTOR 1

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	122	121	123	124
h	57.17	39.85	78.58	120.03	100.33
ψ	0.75	.70	.80	50.68	18.20
AS	2.52	2.41	2.66	49.49	29.34
θ	0.65	0.77	.51	1.02	.83
ϕ	1.40	1.22	1.62	1.48	2.24
#/samples		42	34	33	39
θ_M		-2.2	-2.9	-3.2	-2.4
ϕ_M		7.3	-12.7	-6.0	7.7

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$.

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-128.20	351.40	-452.60	12.65	1.64
ψ	-5.47	-12.69	9.28	-92.43	3.10
AS	5.24	-15.34	11.02	-82.86	2.90
θ	-1.28	-2.85	2.22	-0.27	0.05
ϕ	-3.85	8.62	-1.37	.56	.13

MANEUVER TYPE Normal Landing SECTOR 2

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	122	121	123	124
h	6.57	8.56	4.50	390.63	1687.60
ψ	7.92	8.51	7.30	28.28	252.81
AS	0.98	.99	.97	39.20	306.01
θ	0.82	.91	.73	1.47	1.84
ϕ	6.46	7.08	5.80	11.09	104.93
#/samples		24	23	18	29
θ_M		2.4	1.6	3.5	-4.7
ϕ_M		64.5	57.6	70.4	57.8

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-584.50	154.20	-88.40	256.60	-4.39
ψ	310.20	147.20	34.72	30.41	-2.45
AS	313.90	-56.10	10.35	47.17	-3.29
θ	-.66	2.84	-3.53	-0.05	0.01
ϕ	-59.34	142.6	-125.00	-15.52	1.85

MANEUVER TYPE Normal Landing SECTOR 3

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	122	121	123	124
h	32.64	28.67	37.71	943.50	503.05
ψ	0.83	0.73	0.95	1606.30	465.32
AS	1.80	1.76	1.86	1365.00	387.16
θ	0.82	.61	1.10	13.03	1.09
ϕ	1.77	1.77	1.85	71.38	24.31
#/samples		23	18	26	37
θ_M		-1.91	4.44	5.26	-3.93
ϕ_M		8.07	17.90	28.6	7.7

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_0	B_1	B_2	B_3	B_4
h	-1445.00	22.47	-35.08	-169.40	108.50
ψ	2278.00	-6.14	3.62	260.60	-169.30
AS	1758.00	-28.66	-3.45	221.60	-144.40
θ	-13.81	0.85	-0.03	-1.31	1.25
ϕ	92.24	-35.60	32.43	12.44	-7.75

MANEUVER TYPE Normal Landing SECTOR 4

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	122	121	123	124
h	12.03	14.36	9.70	912.02	593.75
ψ	3.08	3.02	3.15	39.75	41.51
AS	3.21	3.74	2.67	28.68	24.70
θ	0.98	1.08	.89	1.60	2.00
ϕ	4.12	4.42	3.83	13.97	9.58
#/samples		40	40	42	47
θ_M		-6.69	-5.96	-10.6	-6.14
ϕ_M		32.5	32.3	37.5	27.0

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	2759.00	-639.10	38.91	181.90	-56.04
ψ	165.90	130.40	32.90	-2.53	7.11
AS	263.00	-40.75	31.79	1.17	-3.86
θ	12.59	-12.33	10.93	0.56	-0.32
ϕ	56.93	13.35	-9.62	1.99	-0.75

MANEUVER TYPE Normal Landing SECTOR 5

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	122	121	123	124
h	4.65	5.77	3.26	1032.30	1009.30
ψ	1.25	1.42	1.04	106.64	137.91
AS	2.85	3.47	2.09	309.46	411.38
θ		0.61		6.66	10.61
ϕ		1.72		17.95	24.68
#/samples		31	25	25	32
θ_M		6.6	4.9	-5.1	-6.6
ϕ_M		6.9	-8.4	-7.3	9.7

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-507.90	-319.10	314.40	-51.31	-4.59
ψ	520.70	0.51	5.04	10.34	-1.17
AS	-62.60	5.44	-26.42	30.57	-3.38
θ	0.84	1.53	5.99	-0.58	0.07
ϕ	-6.62	-22.68	10.85	1.87	-0.14

MANEUVER TYPE Normal Landing SECTOR 6

ERRORS

Dependent Variable	Excellent			Poor	
		122	121	123	124
h		8.59		854.85	18160.00
ψ		.76		216.71	12365.00
AS		1.44		2.10	249.33
θ		.22		0.53	27.51
ϕ		.13		0.12	2.88
#/samples		12	8	8	14
θ_M		5.5	3.6	3.8	8.9
ϕ_M		-0.5	-0.3	-0.3	-2.4

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-96.22	141.10	-19.52	1117.00	11920.00
ψ	266.40	-6.45	7.08	-782.4	-7990.00
AS	90.57	-11.15	7.98	-16.94	-159.00
θ	-1.84	0.66	-0.85	-0.37	-15.69
ϕ	0.04	-1.02	0.46	0.34	2.77

MANEUVER TYPE Barrel Roll SECTOR 1

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	101	107	114(L)	110(L)
h	0.20	.25	0.10	7126.60	62614.00
ψ					
AS	0.04	0.00	0.04	23.60	120.92
θ	0.02	0.00	0.02	2.95	7.78
ϕ	0.03	0.00	0.04	.75	4.24
#/samples		1	4	8	32
θ_M		-9.8	-9.5	-9.1	-9.5
ϕ_M		1.4	2.4	5.2	-10.3

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$.

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	184.40	-89.60	16.00	-3.53	-81.04
ψ					
AS	275.00	-2.09	4.39	3.48	-11.17
θ	5.28	-2.26	-.40	1.25	-0.06
ϕ	2.30	-2.32	1.60	3.12	0.74

MANEUVER TYPE Barrel Roll SECTOR 2

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	101	107	114(L)	110(L)
h	46.60	37.67	54.45	198.03	298.18
ψ	1.77	2.37	1.24	8.26	8.55
AS	2.62	2.31	2.89	34.87	17.56
θ	0.81	0.89	.74	2.67	6.76
ϕ	4.79	4.94	4.65	11.77	5.51
#/samples		29	33	32	26
θ_M		-13.1	-14.7	-18.7	-10.1
ϕ_M		-44.9	-37.7	-37.5	-41.8

FUNCTIONAL FORM: $DV = B_0 + B_1 t + B_2 t^2 + B_3 \theta_M + B_4 \phi_M$

REFERENCE FUNCTIONS

Variable	B_0	B_1	B_2	B_3	B_4
h	-1559.00	-1987.00	758.90	-21.96	-32.10
ψ	5.75	5.73	-35.52	-0.00	0.12
AS	60.18	87.77	-46.18	-6.51	-1.62
θ	-8.89	-41.90	56.58	0.09	0.07
ϕ	-12.73	-214.70	202.10	-2.66	0.05

MANEUVER TYPE Barrel Roll SECTOR 3

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	101	107	114(9)	
h	15.54	16.66	14.49	620.28	229.42
ψ	3.03	3.22	2.85	7.14	20.24
AS	2.08	2.25	1.92	70.79	13.95
θ	0.65	.71	.60	2.48	4.70
ϕ					
#/samples		14	15	15	17
θ_M		34.2	38.1	38.4	26.4
ϕ_M		83.9	85.0	81.3	84.5

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 \; B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_0	B_1	B_2	B_3	B_4
h	7751.00	22.03	-.07	-23.32	-95.66
ψ	117.40	0.10	0.00	-1.51	-1.03
AS	1579.00	-1.02	0.00	3.90	-17.28
θ	14.89	0.95	-0.00	0.94	-.51
ϕ					

MANEUVER TYPE Barrel Roll SECTOR 4

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	101	107	114(I)	110(L)
h	6.02	6.05	5.99	250.00	1644.00
ψ	3.32	3.36	3.27	80.01	
AS	0.69	.77	.61	1.33	
θ	0.23	0.17	0.29	7.28	
ϕ					
#/samples		9	9	11	12
θ_M		31.2	34.6	34.9	-24.5
ϕ_M		169.0	170.0	172.0	168.0

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-169.4	20.30	-.05	-31.13	0.34
ψ	848.5	2.18	-0.00	-13.08	-3.25
AS	801.50	-1.33	0.00	-2.87	-2.53
θ	-147.00	.01	0.00	0.83	0.95
ϕ					

MANEUVER TYPE Barrel Roll SECTOR 5

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	101	107(L)	114(L)	
h	3.40	3.79	3.01	251.04	
ψ	3.58	3.50	3.66	129.45	
AS	0.77	.88	0.65	7.90	
θ	0.85	0.44	1.26	15.19	
ϕ					
#/samples		9	9	7	
θ_M		-35.1	-35.2	-41.3	
ϕ_M		268.0	263.0	260.0	

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-3178.00	18.06	-0.05	18.64	10.96
ψ	-2158.00	2.83	-0.00	6.50	8.28
AS	-509.90	-1.01	0.05	-0.57	2.80
θ	73.60	0.87	0.00	-0.89	0.05
ϕ					

MANEUVER TYPE Barrel Roll SECTOR 6

ERRORS

Dependent Variable	Excellent			Poor	
	ϵ_C	101	107	114 (L)	
h	21.48	26.47	16.48	100.83	
ψ	3.07	3.32	2.83	192.91	
AS	1.10	1.51	.70	210.67	
θ	0.69	.46	.42	135.99	
ϕ					
#/samples		11	11	6	
θ_M		-36.5	-37.6	-44.2	
ϕ_M		349.0	353.0	348.0	

FUNCTIONAL FORM: $DV = B_0 + B_1\phi + B_2\phi^2 + B_3\theta_M + B_4\phi_M$

REFERENCE FUNCTIONS

Variable	B_o	B_1	B_2	B_3	B_4
h	-676.90	-5.60	-0.01	14.63	10.55
ψ	-311.70	-4.22	0.00	21.86	5.28
AS	-118.50	0.58	0.00	29.96	4.18
θ	-289.90	-4.36	0.00	18.90	4.53
ϕ					