

NASA CR 170398

(NASA-CR-170398) NON-INTRUSIVE PARAMETER
IDENTIFICATION PROCEDURE USER'S GUIDE Final
Report, Feb. - Jun. 1982 (Systems
Technology, Inc., Mountain View, Calif.)
138 p HC A07/MF A01

N83-23318

Unclas
11503

CSSL 01C G3/08

Non-intrusive Parameter Identification Procedure User's Guide

Gregory D. Hanson and Wayne F. Jewell

Contract NAS4-2941
April 1983



NASA CR-170398

Non-Intrusive Parameter Identification Procedure User's Guide

Gregory D. Hanson and Wayne F. Jewell
Systems Technology, Inc., Mountain View, California 94043

Prepared for
Ames Research Center
Dryden Flight Research Facility
under Contract NAS4-2941

1983

NASA

National Aeronautics and
Space Administration

Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523

Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

FOREWORD

This project was sponsored by the National Aeronautics and Space Administration (NASA) Dryden Flight Research Center (DFRC) at the Edwards Air Force Base, California, under Contract NAS2-2941. The NASA contract technical monitor was Ms. Mary F. Shafer. The Systems Technology, Inc., (STI) project engineer was Mr. Robert K. Heffley. The work was accomplished during the period from February 1982 to June 1982.

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION AND BACKGROUND	1
II	BACKGROUND AND THEORY OF OPERATION	5
	A. Description of NIPIP	5
	1. Basic Concepts	5
	2. Assumption of a General Pilot Control Strategy Form	10
	3. NIPIP Algorithm	14
	B. Additional Features of the NIPIP Technique	19
	1. Sliding Time Window	19
	2. Bias Degree of Freedom	20
	3. Multiple Control Structures	22
	4. Preprocessing of Input Data	22
	5. Frequency Response	22
III	NAS VARIABLE AND SUBROUTINE DEFINITIONS	25
	A. Control Flags	26
	B. Time History Variables	29
	C. NIPIP Variables	31
	D. Frequency Response Variables	34
	E. Warning Tolerances	37
	F. User Defined Subroutines	38
	G. Input Data File Format	40
IV	SOME ILLUSTRATIVE EXAMPLES	43
	A. A Single-Loop Example	44
	1. Statement of the Problem and Objectives	44
	2. Consideration of NIPIP Parameters	44
	3. Definition of the NAS Variables	48
	4. Description of the NAS Outputs	52
	5. Discussion of the Results	52

TABLE OF CONTENTS (Concluded)

Section	Page
B. A Multiloop Example	61
1. Statement of the Problem and Objectives	61
2. Consideration of NIPIP Parameters	63
3. Definition of the NAS Variables	68
4. Description of the NAS Outputs	75
5. Discussion of the Results	75
C. Closed-Loop Pilot-Vehicle Example	81
1. Statement of the Problem and Objectives	81
2. Consideration of NIPIP Parameters	83
3. Definition of the NAS Variables	84
4. Description of the NAS Outputs	86
5. Discussion of the Results	86
D. Deceleration to Hover	89
1. Statement of the Problem and Objectives	89
2. Consideration of NIPIP Parameters	93
3. Definition of the NAS Variables	93
4. Description of the NAS Outputs	97
5. Discussion of the Results	97
REFERENCES	103
APPENDICES	
A GENERIC F-8 DIGITAL SIMULATION	A-1
B OPERATING PROCEDURES	B-1
C SUMMARY OF NAS PROGRAM VARIABLES.....	C-1

LIST OF TABLES

Number		Page
1	Table of Z-Transforms	15
2	Default Listing of the MODE Subroutine	39
3	Default Listing of the PREPR Subroutine	42
4	Input File Used for Example A	49
5	NAS Output for Example A	53
6	Listing of Subroutine PREPR for Example B	69
7	Listing of Subroutine MODE for Example B	71
8	Input File Used for Example B	73
9	NAS Output for Example B	76
10	Exact and Estimated Pilot Gains Using Three Degrees of Freedom	80
11	Exact and Estimated Pilot Gains Using Five Degrees of Freedom	80
12	Input File Used for Example C	85
13	NAS Output for Example C	87
14	Partial List of the Input File Used for Example D ...	94
15	Listing of the PREPR Subroutine Used for Example D ..	96
16	NAS Output for Example D	98

LIST OF FIGURES

Number		Page
1	An Example of a Pilot-Vehicle System	10
2	Three Common Control Loop Structures	12
3	Three Common CTOL Piloting Techniques Appropriate for an ILS Approach	13
4	A Single-Input, Single-Output Example	45
5	Time History Used for Example A	46
6	Frequency Responses of Exact and Estimated Y_p^θ for Various Degrees of Freedom	60
7	Multi-Loop Control Task Example	62
8	Time Histories Used for Example B	64
9	Frequency Response of Exact and Estimated Y_p^h for Five Degrees of Freedom	82
10	Phase Plane Trajectories of Landing Flare	90
11	Block Diagram for Deceleration to Hover Example	92
12	Actual Data and NIPIP Results for Example D	101

LIST OF SYMBOLS

A	A measure of the effective size of the object being viewed
B	Bias degree of freedom
\underline{c}	Column-vector of the c_j
$\hat{\underline{c}}$	Estimated value of \underline{c}
c_j	Constant coefficients relating y and f_j .
\underline{F}	Row vector of the f_j
f_1, f_2, \dots	Selected variables from the state vector \underline{X} or explicit functions thereof
H	Matrix of adjoined measurements
h	Height
\dot{h}	Vertical velocity
\ddot{h}	Vertical acceleration
h_c	Desired altitude above the ground
\dot{h}_c	Desired rate of change of altitude
\dot{h}_e	Vertical velocity error
J	Least square error
K_h	Pilot gain on vertical velocity
K_p	Pilot gain
K_θ	Pilot gain on pitch attitude
k	The effective gain for the combined pilot-vehicle system
L_w	Scale length
R	True range
\dot{R}	Range rate
\ddot{R}	Range deceleration

ORIGINAL PAGE IS
OF POOR QUALITY

R_p	Perceived range
R^2	Correlation coefficient
T_I	Integration time constant
T_L	Lead time constant
T_R	Roll time constant
U_o	Forward velocity
V	Airspeed
V_{ref}	Reference airspeed
v	Error between actual and estimated system
w_g	Vertical gusts
\underline{X}	Independent variable vector, state vector
\underline{X}_e	State error vector
\underline{X}_{ref}	State reference vector
Y_c	Controlled element
Y_{cf}	Pilot crossfeed
Y_p	Pilot
$Y_p^{()}$	Pilot transfer function with respect to ()
$\hat{Y}_p^{()}$	Estimated pilot transfer function with respect to ()
y	Dependent variable of a given system
δ	Control
$\dot{\delta}$	Control rate
δ_e	Elevator control
δ_p	Perturbation control
δ_s	Stick deflection
δ_T	Throttle deflection
δ_{trim}	Control deflection for trim

ORIGINAL PAGE IS
OF POOR QUALITY

$\epsilon_{G/S}$	Flight director
ζ	Damping ratio
θ	Pitch attitude
θ_c	Pitch attitude command
θ_e	Pitch attitude error
θ_{TD}	Pitch attitude at touchdown
σ_w	RMS level
τ_e	Time delay
ϕ_c	Desired bank
ω	Frequency (rad/sec)
ω_1	Incremental frequenc

LIST OF SUB AND SUPERSCRIPTS

FD	Flight director
h	Height
\dot{h}	Vertical velocity
n	Denotes each set of y nd <u>F</u>
TD	Touchdown
trim	Trim
v	Airspeed
ϵ	Phase plane state
θ	Pitch
ϕ	Bank angle

LIST OF ABBREVIATIONS

CTOL	Conventional takeoff and landing
DH ^T H	Determinant of the H ^T H matrix
DOF	Degrees of freedom
ILS	Instrument landing system
NAS	Non-Intrusive Parameter Analysis Software
NIPIP	Non-Intrusive Parameter Identification Procedure

SECTION I

INTRODUCTION AND BACKGROUND

This user's guide, describing the Non-Intrusive Parameter Analysis Software (NAS), is intended as an introductory text. Its organization comprises an outline of topics for software operation by a new user. Some features are only partially presented at first; their complete explanation is postponed to the point where additional information will not be confusing or out of context. This guide is oriented toward a user who is familiar with FORTRAN. Familiarity with FORTRAN will be required to exercise some of the special features of this software; however, a knowledge of FORTRAN is not required for basic program operation. It is assumed that the user is familiar with the operating procedures for the computer system in which this software resides.

NAS incorporates many features over which the user has direct control and allows the user to adapt the software package to the identification problem at hand. The capabilities of this software include the ability to identify linear as well as nonlinear relations between input and output parameters; the only restriction is that the input/output relation be linear with respect to the unknown coefficients of the estimation equation. The output of the identification algorithm can be specified to be in either the time domain (i.e., the estimation equation coefficients) or in the frequency domain (i.e., a frequency response of the estimation equation). The frame length ("window") over which the identification procedure is to take place can be specified to be any portion of the input time history thereby allowing the freedom to start and stop the identification procedure within a time history. There also is an option which allows a "sliding window." Under this option the window is "slid" to obtain new data, while the data at the beginning of the window is "dropped" as the new data is added. This option gives essentially a moving average over the time history.

ORIGINAL FILES IS OF POOR QUALITY

The software package includes a user-defined subroutine which can manipulate the input data of the identification algorithm into a convenient form. Another user-defined subroutine specifies the form of the assumed solution. Using this subroutine it is possible to run many assumed solutions simultaneously, which is advantageous when the exact form of the solution is unknown.

ORGANIZATION

Section II describes the fundamental concepts of the Non-Intrusive Parameter Identification Procedure (NIPIP). It outlines what is required of the user to understand and implement this technique. Section IV also provides a discussion of the special options of the software that add scope and convenience to the user's ability to manipulate and interact with the program. As the user becomes more familiar with the concepts of the NAS software package, this section provides several additional features that are of immediate usefulness.

Sections III and IV comprise the majority of the information for the operation of the software package. Section III defines the program variables with which the user must be familiar to understand and operate the software. This section contains a description of the program's control flags and their use in routing the program during execution. It also contains a description of the input and output specifications and the internal warning messages which alert the user to possible error conditions and solutions of questionable validity. Section IV provides some illustrative examples in which problems are described from setting up the estimation equation through describing the output of the algorithm. Although this section does not examine every conceivable possibility, the user should be able to understand the operating procedure and run the software after completing these sections.

Appendix A describes a small perturbation simulation of the F-8 aircraft which was used to generate some of the examples used in Section IV. Appendix B contains detailed instructions for running the NAS program. Appendix C is a summary of the NAS program variables. Program

listings of NAS are contained on the microfiche taped to the inside back cover of this report.

SECTION II

BACKGROUND AND THEORY OF OPERATION

A. DESCRIPTION OF NIPIP

This section describes the basic concepts of the Non-Intrusive Parameter Identification Procedure. The general procedure presented here was demonstrated in Ref. 1 and was extended to real-time computation in Refs. 2 and 3. This method involves the assumption of a general parameter identification strategy (a mathematical structural model with undetermined coefficients) and determines the coefficients of the strategy by matching the model to the generated data using a running least-squares estimation technique (i.e., a multiple linear regression).

First, the general concepts of the least-squares approach are discussed. Next the method by which a general form of an assumed pilot control strategy is stated. Finally, a description of the NIPIP algorithm is presented. All of this is in preparation for the presentation of the program variables and input/output specifications which follow in the next section.

1. Basic Concepts

The numerical analysis concept central to the procedure presented here consists of applying a running least-squares estimation technique. A specific example of least-squares estimation occurs in curve-fitting problems where it is desired to obtain a functional form of some chosen order that best fits a given set of discrete measurements, Ref. 4. Thus the least-squares problem consists of correlating a dependent variable, y , of a given system with one or more elements of an independent variable vector, \underline{X} . That is, it is assumed that y and \underline{X} are related by an expression of the form

**ORIGINAL PAGE IS
OF POOR QUALITY**

$$y = f_1 c_1 + f_2 c_2 + \dots + f_j c_j \dots + f_m c_m \quad (1)$$

$$y = [f_1, f_2, f_3, f_4, \dots] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \end{bmatrix}$$

or $y \equiv \underline{F} \underline{c} \quad (2)$

where f_1, f_2, \dots are selected variables from the state vector \underline{X} or explicit functions thereof, and the c_j 's are constant coefficients relating y and f_j . In Eq. 2 \underline{F} is a row-vector of the f_j , and \underline{c} is a column-vector of the c_j . y and \underline{F} can be composed of a control or state, past or present discrete values, and linear or nonlinear functions thereof, e.g., " $y = \underline{F} \underline{c}$ " could include the following forms:

$$\delta_e = \theta c_1$$

$$\text{or } \theta = \dot{\theta} c_1 + c_2$$

$$\text{or } \delta = \dot{\delta} c_1 + \theta c_2 + \dot{\theta} c_3 + \dot{h} c_4 + c_5$$

$$\text{or } \epsilon^{1/2} = \eta c_1 + \eta^2 c_2 + \frac{\eta^{5/2}}{1+\eta} c_3 + c_4$$

If there exist sets of discrete measurements for y and \underline{F} , and the difference between the actual system output and the output of the assumed model given by Eq. 1 is given by v , then

$$y_n = \underline{F}_n \hat{\underline{c}} + v_n$$

ORIGINAL PAGE IS
OF POOR QUALITY

where \hat{c} is the estimated value of c given by Eq. 1, and the subscript n is used to denote each set of y and F . It is assumed that there exists a \hat{c} which will relate all values of y_n to the F_n .

To arrive at an estimate for \hat{c} , a minimum of m sets of measurements are required (where m is the number of degrees of freedom in Eq. 1). The solution for \hat{c} is found by adjoining these sets of discrete measurements as follows,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} \hat{c} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad (3)$$

where N is the number of measurements and $N > m$. Equation 3 can be written more concisely in a matrix notation as

$$\underline{y} = H \hat{c} + \underline{v} \quad (4)$$

where \underline{y} , H , and \underline{v} are defined by inspection. Note that H is an N by m matrix.

The least-squares estimate for the unknown coefficients, \hat{c} , is obtained by minimizing the sum of the squares of \underline{v} where

$$\underline{v} = \underline{y} - H\hat{c} \quad (5)$$

ORIGINAL PAGE IS
OF POOR QUALITY

The sum of squares of a vector is given by the vector inner product. Thus we wish to minimize the scalar J , where

$$J = (\underline{y} - H \hat{\underline{c}})^T (\underline{y} - H \hat{\underline{c}}) \quad (6)$$

Minimization of a scalar with respect to a vector is obtained when

$$\frac{\partial J}{\partial \hat{\underline{c}}} = \underline{0} \quad (7)$$

and the determinant of the Hessian matrix of the second partial derivatives of J is positive, semidefinite

$$\det \left[\frac{\partial^2 J}{\partial \hat{\underline{c}}^2} \right] > 0 \quad (8)$$

Differentiating J and setting the result equal to zero yields:

$$H^T H \hat{\underline{c}} = H^T \underline{y} \quad (9)$$

It is readily shown that the determinant of the Hessian matrix of J , with respect to $\hat{\underline{c}}$, is positive semidefinite, and thus Eq. 9 does, indeed, define a minimum. When $H^T H$ possesses an inverse (i.e., when it has a non-zero determinant) the least-squares estimate is

$$\hat{\underline{c}} = (H^T H)^{-1} H^T \underline{y} \quad (10)$$

**ORIGINAL PAGE IS
OF POOR QUALITY**

Implied in the preceding development is that all available measurements are utilized together at one time (i.e., in a so-called "batch processing" scheme).

Although the estimate of the unknown coefficients can be obtained using Eq. 10, it usually requires vast amounts of storage to save all of the accumulated data which is needed to form the H matrix. This storage requirement is sometimes prohibitive to the application of this scheme. However, it is easily shown that a summing of the measurements as they become available will yield the same $H^T H$ and $H^T \underline{y}$ matrices as those obtained by adjoining each set of measurements. Thus,

$$H^T H = \sum_{n=1}^N \begin{matrix} F_n^T \\ F_n \end{matrix} \quad (11)$$

and

$$H^T \underline{y} = \sum_{n=1}^N \begin{matrix} F_n^T \\ F_n \end{matrix} y_n \quad (12)$$

This gives the recursive least-squares estimate as

$$\hat{\underline{c}} = \left(\sum_{n=1}^N \begin{matrix} F_n^T \\ F_n \end{matrix} \right)^{-1} \sum_{n=1}^N \begin{matrix} F_n^T \\ F_n \end{matrix} y_n \quad (13)$$

where N is the number of data points up to the given point in time where the sum is truncated.

The following features of this solution can be noted. First, the main computational task consists of updating summations of products of the m by m matrix $\underline{F}^T \underline{F}$. The only storage requirement is that these summations alone be saved, not all of the accumulated data as with the batch processing scheme. Second, the most complex computational task is the inversion of the $\sum_{n=1}^N \begin{matrix} F_n^T \\ F_n \end{matrix}$ matrix; however, this is still not a serious complication.

2. Assumption of a General Pilot
Control Strategy Form

One use of this identification scheme is to identify the pilot's control strategy by matching an assumed strategy to pilot-generated data. Basic and applied research going back more than two decades (Refs. 5 through 10) has demonstrated that the human operator's control strategy can be fairly accurately described by linear differential equations. When the control task is time-varying, such as controlling an aircraft on final approach, then the coefficients of the differential equations will, in general, also be time varying. The major interests here, however, are the states that must be perceived and acted upon in order to perform the control task.

A general control task is depicted in Fig. 1.

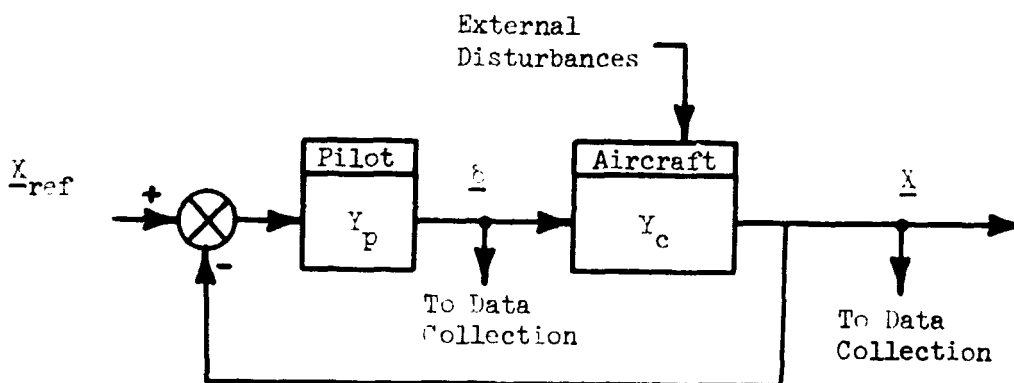


Figure 1. An Example of a Pilot-Vehicle System

This example represents the use of the controls, δ , and the aircraft states, X , for the identification of the pilot's actions. There are several concerns inherent in this kind of approach. The main concern is that since the raw data used in the recursive least-squares (i.e., curve fitting) procedure consists of the controls, δ , and the aircraft states, X , as shown in Fig. 1, there can be confusion as to whether the correlation

**ORIGINAL PAGE IS
OF POOR QUALITY**

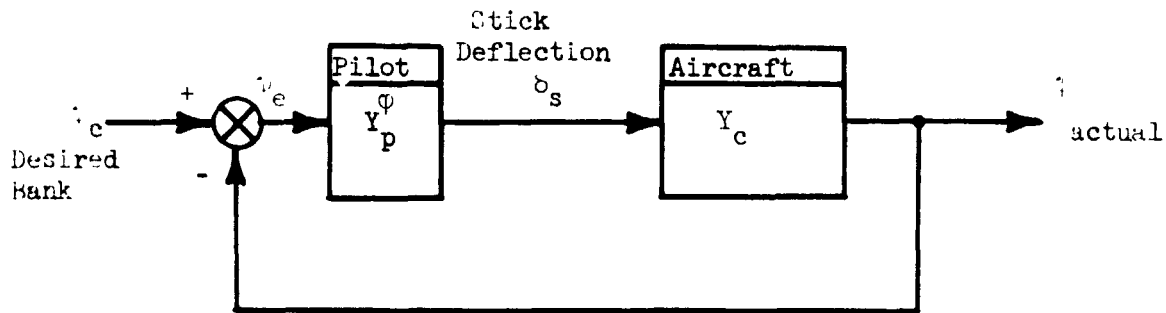
between $\underline{\delta}$ and \underline{X} reflects more accurately functions describing the pilot, Y_p , or those describing the aircraft controlled element, Y_c . However, since Eq. 13 guarantees a minimum of the scalar, J (Eq. 6), and this minimum is unique such that the functions describing the controlling element and those describing the inverse of the controlled element do not yield the same minimum for J , then Eq. 13 will yield the solution for which J is minimized. If the assumed model form more appropriately describes the controlling element, the guarantee of minimizing J assures that the correlation obtained using Eq. 13 will identify the controlling element. However, this guarantee of minimizing J does not guarantee that the solution will reflect either element if the assumed form of Eq. 1 does not adequately describe either element which is to be identified. It has been found that so long as the functional form of the assumed control law, i.e., $y = \underline{F} \underline{c}$, is appropriate for the piloting technique being used, this procedure does identify the control function of the pilot, Y_p . This fact was demonstrated in Refs. 2 and 3 and will be shown to be the case in a later section.

The advantage of this procedure is that it is mathematically simple, as stated earlier, but its success depends upon the judicious choice of an assumed pilot control strategy, Y_p . This, in turn, depends upon the experience of the experimenter, knowledge of the control task, and a thorough understanding of the dynamics of the controlled element, Y_c . It is essential that the analyst choose a likely candidate control strategy in order for the subsequently estimated parameters of the control strategy to be valid.

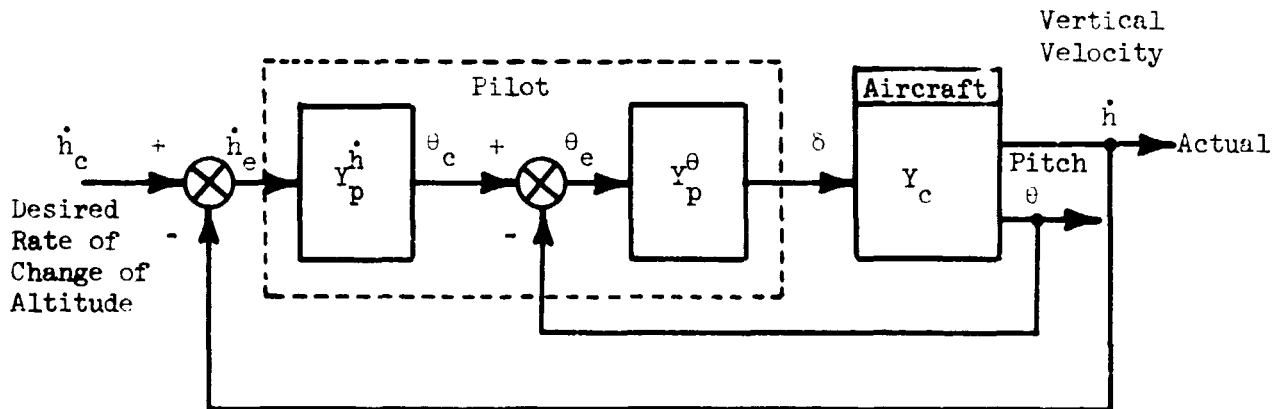
Some common examples of pilot control loop structures are shown in Figs. 2 and 3. These examples represent variations in the use of controls and variations in available displays or pilot-perceived aircraft states. The loop structures in Fig. 2 are examples of single input control-loop structures. Figure 3 gives examples of parallel loop structures (Fig. 3a), series loop structures (Fig. 3b), and control crossfeeds (Fig. 3c). The "Y" symbols within blocks in the loop structures represent describing functions (or control laws) which characterize the pilot's control strategy. It is important to recognize that each of these control

a. Bank Angle Regulation Task (Tracking Task)

ORIGINAL PAGE IS
OF POOR QUALITY



b. Altitude Rate of Change Task (Discrete Task)



c. Landing Maneuver (Discrete Task)

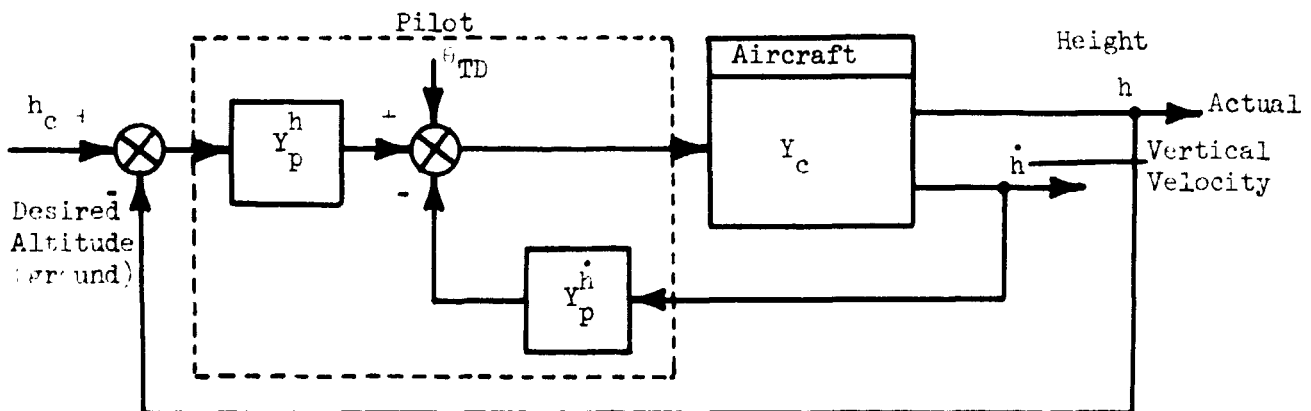
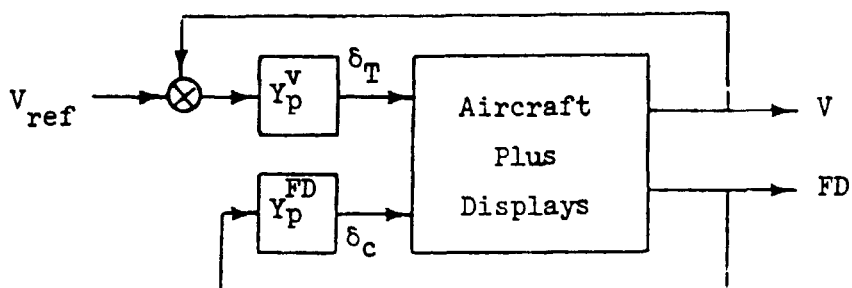
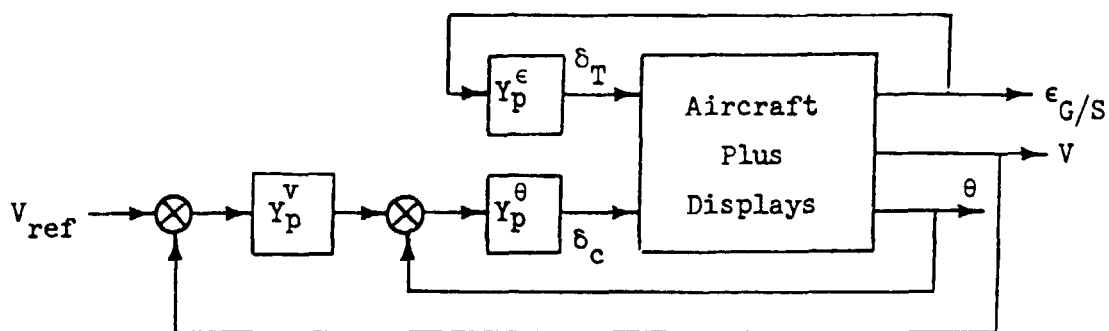


Figure . Three Common Control Loop Structures

a. Frontside Using Flight Director



b. Backside Using Raw Glide Slope



c. Frontside Using Raw Glide Slope

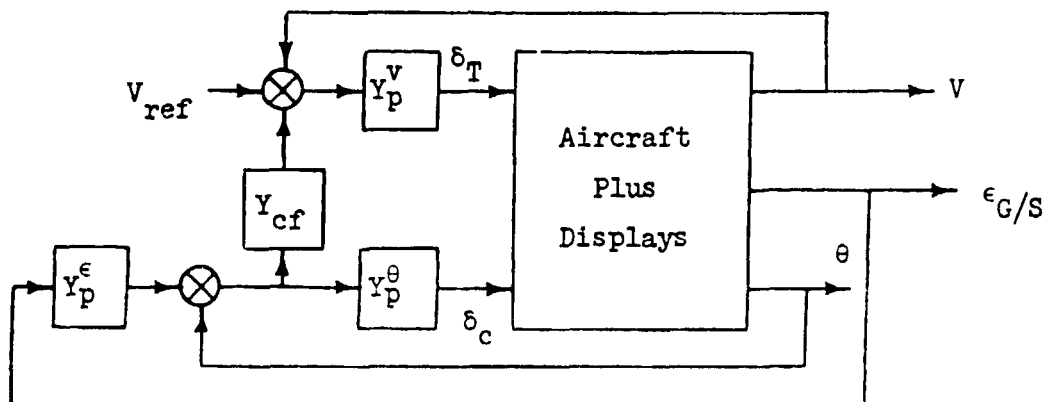


Figure 3. Three Common CTOL Piloting Techniques
Appropriate for an ILS Approach

ORIGINAL PAGE IS
OF POOR QUALITY

loop structures can be handled and identified by the procedure being considered.

The first step in applying the identification procedure is to formulate the pilot control strategy for the defined control task. There is an element of engineering judgment in the choice of the structure of a mathematical function which appropriately describes the general input/output relation (i.e., the form of Y_p) for the defined control task. Indeed, it is the analyst's exercise of a priori knowledge and experience which allows the judicious choice of the pilot control strategy. For example, a pure gain describing function is the most desirable so long as adequate pilot-vehicle performance can be obtained (Ref. 6). Various combinations of lead or lag or other equalization can also be adopted by the pilot depending upon the response of the vehicle and the overall level of performance desired.

The input/output characteristics which the pilot is likely to exhibit are readily expressed using Laplace transformations (transfer functions) of the differential equations describing the pilot control strategy; however, this transformation yields a continuous model of the pilot. Since the identification procedure involves a digital computer, it is necessary to formulate the relationship, Y_p , as a discrete relationship (i.e., a finite difference equation with undetermined coefficients). One way this is accomplished is by applying the z-transformation to the continuous system transfer function to obtain an equivalent discrete transfer function. Table 1 presents some common z-transforms of continuous systems, along with the number of undetermined coefficients. This table is by no means complete and additional z-transforms can be found in Refs. 11 and 12.

3. NIPIP Algorithm

Having discussed, in generalities, how to specify the form of the pilot control strategy, Y_p , one can form the difference equation (estimation equation) and apply the running least-squares error estimation

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 1
TABLE OF Z-TRANSFORMS

Entry No.	z-Transform	No. of Unknowns	Physical System
1	$b_1 z^{-1}$	1	K (pure gain)
2	$\frac{b_1 T z^{-1}}{1 - z^{-1}}$	1	$\frac{K}{s}$ (integrator)
3	$\frac{b_1 z^{-1}}{1 - a_1 z^{-1}}$	2	$\frac{Ka}{s+a}$ (1st-order lag)
4	$\frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$	4	$\frac{K\omega^2}{[\zeta; \omega]}$
5	$\frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} (z^{-p})$	4	$\frac{K(s+a)}{[\zeta; \omega]} e^{-ps}$
6	$\frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1}} (z^{-p})$	3	$\frac{K(s+a)}{(s+b)} e^{-ps}$
7	$\frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$	4	$\frac{K(s+a)}{s(s+b)}$
8	$\frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3}}$	6	$\frac{K(s+a)}{(s+b)[\zeta; \omega]}$
9	$\frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3}}$	7	$\frac{K[\zeta_n; \omega_n]}{[\zeta_1; \omega_1][\zeta_2; \omega_2]}$

**ORIGINAL PAGE IS
OF POOR QUALITY**

algorithm in order to solve for the unknown coefficients of the estimation equation. In the following discussion of the NIPIP algorithm, a control task with an assumed pilot control strategy is used as an example.

The control task is taken to be pitch attitude regulation (Fig. 2), with a lead/lag pilot control strategy (Entry 6 in Table 1). Thus,

$$Y_p(s) = \underbrace{K_p}_{\text{Pilot Gain}} \underbrace{\frac{(\tau_L + 1)}{(T_I s + 1)}}_{\text{Series Equilization}} \underbrace{\exp(-\tau_e s)}_{\text{Pure Time Delay}} = \frac{\delta(s)}{\theta_e(s)} \quad (14)$$

For this example, there is assumed to be no time delay, i.e., $\tau_e = 0$.

Once the control task and assumed control strategy have been specified, the general form of the estimation equation can be determined. Using the z-transform given in Table 1, the discrete relationship for the lead/lag strategy is given by

$$Y_p(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1}} \equiv \frac{\delta(z)}{\theta_e(z)} \quad (15)$$

The coefficients a_1 , b_0 , and b_1 involve combinations of the parameters in the pilot's strategy, and it is not, in general, easy to use them directly to interpret the attributes of the pilot control strategy described in Eq. 14. For this reason it is often necessary to use the frequency domain to interpret the coefficients of $Y_p(z)$. That is,

$$Y_p(s=j\omega) = Y_p(z=e^{j\omega T}) \quad (16)$$

where ω is frequency and T is the discrete sample period. Effective values of the Laplace transform frequency domain parameters K_p , T_L , T_I , and τ_e can, in turn, be approximated from the frequency response.

**ORIGINAL PAGE IS
OF POOR QUALITY**

The general procedure for identifying the coefficients of the estimation equation is started by expressing the z-plane transfer function for Y_p as the finite difference equation:

$$\delta = a_1 \delta z^{-1} + b_1 \theta_e z^{-1} + b_2 \theta_e z^{-2} \quad (17)$$

or

$$\delta_n = a_1 \delta_{n-1} + b_1 \theta_{e_{n-1}} + b_2 \theta_{e_{n-2}} \quad (18)$$

where the subscripts $n, n-1, n-2, \text{etc.}$, denote the values of δ and θ_e at times $t, t-T, t-2T, \text{etc.}$, respectively, and T is the sampling period. Expressing Eq. 18 in the form of Eq. 2 gives

$$y_n = \underline{F}_n \hat{\underline{c}} \quad (19)$$

where $y_n \equiv \delta_n$ and the \underline{F}_n vector is described by

$$\underline{F}_n \equiv [\delta_{n-1} \theta_{e_{n-1}} \theta_{e_{n-2}}] \quad (20)$$

and the $\hat{\underline{c}}$ vector is given by

$$\hat{\underline{c}} = \begin{pmatrix} a_1 \\ b_1 \\ b_2 \end{pmatrix} \quad (21)$$

**ORIGINAL PAGE IS
OF POOR QUALITY**

The values of the estimation equation coefficient, $\hat{\underline{c}}$, are determined using Eq. 13,

$$\hat{\underline{c}} = \left(\sum_{n=1}^N \underline{F}_{-n}^T \underline{F}_{-n} \right)^{-1} \sum_{n=1}^N \underline{F}_{-n} \delta_n \quad (22)$$

Equation 22 can be rewritten as

$$\hat{\underline{c}} = [H^T H]^{-1} H^T \underline{\delta} \quad (23)$$

where

$$H^T H = \sum_{n=1}^N \underline{F}_{-n}^T \underline{F}_{-n} = \sum_{n=1}^N \begin{bmatrix} \delta_{n-1} \\ \theta_{e_{n-1}} \\ \theta_{e_{n-2}} \end{bmatrix} \begin{bmatrix} \delta_{n-1} & \theta_{e_{n-1}} & \theta_{e_{n-2}} \end{bmatrix} \quad (24)$$

and

$$H^T \underline{\delta} = \sum_{n=1}^N \underline{F}_{-n}^T \delta_n = \sum_{n=1}^N \begin{bmatrix} \delta_{n-1} \\ \theta_{e_{n-1}} \\ \theta_{e_{n-2}} \end{bmatrix} (\delta_n) \quad (25)$$

In future references, the notation of Eq. 23 will be used. The solution for $\hat{\underline{c}}$ was shown earlier to minimize the squared error between the estimated control, $\hat{\delta} = \underline{F} \hat{\underline{c}}$, and the actual control, δ .

A measure of the goodness of the fit achieved by Eq. 13 can be computed as the integral square error between the actual and fitted data normalized with respect to the integral square of the actual data,

ORIGINAL PAGE IS
OF POOR QUALITY

$$R^2 = 1 - \frac{\sum_{n=1}^N [\delta_n - \hat{\delta}_n]^2}{\sum_{n=1}^N [\delta_n]^2} \quad (26)$$

where $\hat{\delta} = \underline{H}\hat{c}$ is the output of the model, N is the number of data points up to the given point in time, and R^2 is referred to as the correlation coefficient. Note that R^2 is referred to unity, thus when R^2 is 1.0, it means that the fitted model exactly matches the data. A low value of R^2 does not necessarily imply that the fitted model does not closely approximate the actual data, as the NIPIP technique will approximate a "best" fit of the data. For example, a low correlation coefficient will usually result when using data in which a large amount of noise is present. A low correlation coefficient should be cause for concern, and it should be determined whether the cause of this inability to fit the data is due to an inappropriate model or to problems within the input data.

B. ADDITIONAL FEATURES OF THE NIPIP TECHNIQUE

The previous sections described the basic function of the NIPIP estimation technique. This section presents some additional refinements or features of the identification procedure.

1. Sliding Time Window

The first of these refinements is the use of a "sliding window." The sliding window is effectively a moving average process which allows the identification of the time-varying behavior. Using the running solution for the estimation equation coefficients, the $H^T H$ and $H^T y$ matrices will become "stale" and mask the time-varying behavior of the pilot when the curve-fitting process is to take place over a long period of time. One method of eliminating this "staleness" of the $H^T H$ and $H^T y$ matrices is to reinitialize them periodically. However, periodic reinitialization of $H^T H$

ORIGINAL PAGE IS
OF POOR QUALITY

and $H^T \underline{y}$ was shown to be unacceptable in Ref. 3 because the solution for $\hat{\underline{c}}$ had to reconverge after each reinitialization; thus making it difficult to interpret the results. The problems of staleness and reconvergence in the solution can both be avoided by using the sliding window approach to updating the $H^T H$ and $H^T \underline{y}$ matrices.

The sliding window saves N values of \underline{y} and \underline{F} (the window is thus $N \cdot T$ seconds long). After N data points have been saved and the $H^T H$ and $H^T \underline{y}$ computed for these N data points, the window is "slid" to include the $N+1$ data point and the data point at the beginning of the window is dropped as follows:

$$[H^T H]_{n+1} = H_n^T H_n - \underline{F}_{-1}^T \underline{F}_{-1} + \underline{F}_{-n+1}^T \underline{F}_{-n+1} \quad (27)$$

$$[H^T \underline{y}]_{n+1} = H_n^T \underline{y}_n - \underline{F}_{-1}^T \underline{y}_{-1} + \underline{F}_{-n+1}^T \underline{y}_{-n+1} \quad (28)$$

where the \underline{F}_n vector is defined in Eq. 3. Future $H^T H$ and $H^T \underline{y}$ matrices are computed in a similar fashion and the process continues until the end of the time history.

2. Bias Degree of Freedom

Another feature is the capability of adding a bias degree of freedom (DOF) to the estimation equation. The bias DOF accounts for any uncertainties or biases in the input data, such as trimming about a non-zero reference operating point or unaccounted for command inputs to the overall system. An example of when a bias DOF may be needed in the estimation equation is given below.

From Fig. 1, the signals which are available are the controls, $\underline{\delta}$, and the states, \underline{X} . It is assumed

**ORIGINAL PAGE IS
OF POOR QUALITY**

$$\underline{\delta} = \delta_p - \delta_{trim} \quad (29)$$

and
$$\underline{X}_e = \underline{X}_{ref} - \underline{X} \quad (30)$$

Equations 29 and 30 are used to define perturbations with respect to a given operating point. That is, the controls are trimmed about some operating value, δ_{trim} , and there is some external command, \underline{X}_{ref} , which, in general, is not available for measurement. The pilot transfer function relates the perturbed values of the controls and states,

$$Y_p(z) = \frac{N(z)}{D(z)} = \frac{\underline{\delta}}{\underline{X}_e} = \frac{\delta_p - \delta_{trim}}{\underline{X}_{ref} - \underline{X}} \quad (31)$$

cross multiplying in Eq. 31,

$$(\delta_p - \delta_{trim}) D(z) = (\underline{X}_{ref} - \underline{X}) N(z) \quad (32)$$

Moving $\underline{\delta}_{trim}$ to the right-hand side and dividing both sides by $D(z)$ yields

$$\frac{\delta_p}{-p} = -\underline{X} \left(\frac{N(z)}{D(z)} \right) + \underbrace{\underline{X}_{ref} \left(\frac{N(z)}{D(z)} \right)}_{\text{Bias}} + \frac{\delta_{trim}}{-trim} \quad (33)$$

From Eq. 33 it is seen that the bias DOF is required when $\underline{\delta}_{trim}$ and/or \underline{X}_{ref} are non-zero, or their exact values are unknown. It is important to note that a single bias DOF can account for both of the effects of the unknowns \underline{X}_{ref} and $\underline{\delta}_{trim}$. If $\underline{\delta}_{trim}$ and \underline{X}_{ref} are known "exactly," a bias DOF is not required as the differences $\underline{\delta}_p - \underline{\delta}_{trim}$ and $\underline{X}_{ref} - \underline{X}$ can be determined prior to applying the identification procedure. However, for

**ORIGINAL PAGE IS
OF POOR QUALITY**

any uncertainty in the knowledge of \underline{X}_{ref} and $\underline{\delta}_{trim}$ it is usually a good idea to include the bias DOF in the estimation equation.

3. Multiple Control Structures

The AS software also has the ability to perform the least-squares estimation technique for several estimation equations simultaneously. The feature relies upon the user defined subroutine MODE which contains the NAS variable specifications for each estimation equation. This subroutine is located within a loop in the main program. The NAS variables for each estimation equation are indexed within the subroutine according to the loop variable IM. That is, when IM = 1 the control loop specification for the first estimation equation is used, when IM = 2 the control loop specification for the second estimation equation is used, and so on. An example of using this feature is presented in Section IV.

4. Preprocessing of Input Data

This feature relies upon the user-defined subroutine PREPR and is useful in manipulating the raw input data into the form required by the estimation equation. When multiple control loop structures are to be used and the raw input data is to be processed differently for each structure, the processing for each structure is again indexed using the loop variable, IM. Thus the same raw input data can be used for multiple control loop structures, where each estimation equation is made up of different combinations of the same raw input data. An example of using this feature is presented in Section IV.

5. Frequency Response

When using NIPIP to identify a continuous system (i.e., a system described by differential equations) by means of a discrete system (i.e., a

**ORIGINAL PAGE IS
OF POOR QUALITY**

z-transform of the continuous system), the identified coefficients, \hat{c} , taken by themselves offer only limited insight into the continuous system. Two reasons for this limited insight are that the coefficients change both with the data sample time and the number of degrees of freedom allowed for in the discrete representation of the continuous system. Thus it is not always possible to manipulate the identified discrete coefficients, \hat{c} , back into their respective continuous system counterparts. In fact it is hardly ever possible to perform this reverse manipulation, and it is likely to be an immense and time-consuming effort with very limited results.

This problem of interpreting the identified coefficients can be circumvented by interpreting the results in the frequency domain. Effective values of the corresponding Laplace transform frequency domain parameters can, in turn, be approximated from the frequency response. Besides being a convenient way to interpret the results, the frequency domain provides a means of comparing the NIPIP results with data collected by other techniques (i.e., describing function data). It also provides a means of computing combined system measurements (i.e., pilot-vehicle measurements).

Prior research using NIPIP to identify continuous systems has also shown that interpreting the results in the frequency domain reduces the sensitivity of the identified solution to changes in the sample time and the variations in the number of degrees of freedom allowed for in the discrete representation.

The stability of the frequency response (i.e., the repeatability of the frequency response computed at various intervals throughout an identification process) has been found to be closely related to the convergence of the identified solution (Ref. 13). That is, a frequency response which is essentially invariant in amplitude and phase throughout an identification process generally indicates a convergent and possibly correct solution. Frequency responses with large variations in amplitude or phase can be attributed to one or more causes. For example, the human operator may be a novice to the control task and may not be using a consistent control strategy. On the other hand, the human operator may be using a control strategy that is different from that assumed by the

**ORIGINAL PAGE IS
OF POOR QUALITY**

experimenter. In any case, it cannot be overemphasized that it is up to the experimenter, through knowledge of the control task and a thorough understanding of the dynamics of the controlled element, to evaluate the likelihood of the correctness of the identified solution regardless of the variations in the computed frequency responses.

SECTION III

NAS VARIABLE AND SUBROUTINE DEFINITIONS

To operate the NAS software efficiently, it is necessary for the user to be familiar with the program parameters which direct the flow of its operation (i.e., the control flags) and those used to define the problem at hand (i.e., program variables). This section defines these parameters and describes their use in the NAS routines. The control flags are defined first, followed by the variables specifying the time history data, the definition of the F-vector, the definition of the transfer function and frequency response, and, finally, the error constraint conditions. The number next to the name refers to the variable number in the namelist subroutine.

The NAS program parameters can be specified in two ways. The first method is through direct terminal input using a "namelist" subroutine which asks for the variable number and then the value. This method is called "namelist input," and the variable numbers required for this method are listed below. The second method is that of specifying the variables and their values in the user-defined subroutine, MODE. Both means of specifying the NAS program parameters will be demonstrated by examples in a later section.

Although there are a total of fifty-seven variables listed below, most are defaulted to values that need not be changed for the majority of the applications. The user will, in fact, quickly learn which variables are relevant to a particular problem. This will usually amount to only five to ten variables.

It is advisable to read and understand all of the NAS parameter descriptions presented below. Whenever possible, we refer to Example A in Section IV when specifying the numerical value of a particular parameter. After becoming familiar with the NAS parameters, the user should refer to the summary descriptions presented in Appendix C. This will speed the process of setting the values of the various parameters.

A. CONTROL FLAGS

IN (Var. No. 18)

Input control specifying the input device or logical unit number from which the time history data is to be read. In the Cyber system, this corresponds to the tape number in the program statement defining the filename under which the data is stored. (Default: 10)

IOUT (Var. No. 19)

Output control specifying the output device or logical unit number to which the algorithm's output is directed. In the Cyber, this corresponds to the tape number in the program statement defining the filename used for output. (Default: 6)

MOD (Var. No. 20)

Mode for program variable input.**

0

Direct user input of NAS program variables from input device (e.g., terminal, etc.).

1

NAS program variables are input from subroutine "MODE."

(Default: 0)

NPS (Var. No. 21)

Specifies the number of control loop structures to be run simultaneously. (Default: 1, Maximum number = 10 on Cyber, 2 on Apple)

**NOTE: In order to run more than one control loop structure simultaneously (i.e., NPS > 1), the program variables must be stored and input from subroutine "MODE;" hence MOD must be set equal to 1 for NPS > 1. However, a single control loop structure can be run either using MOD = 0 and NPS = 1, in which case the user inputs the program variables directly or using MOD = 1 and NPS = 1 and inputting the program variables using subroutine "MODE."

ORIGINAL TABLE IS
OF POOR QUALITY

Direct user input is only available when running the program using the software under interactive processing. When using batch processing, only subroutine MODE can be used.

INA (Var. No. 22)

NIPIP analysis verification

1

Verification of the F-vector. This is useful to check that the correct F-vector has been specified.

2

Verification of the frequency response transfer function. This is useful to check that the proper transfer function has been specified.

3

Verifies both the F-vector and the frequency response transfer function. This value verifies both, whereas, specifying INA = 1 or 2 verifies only the one specified.

4

Does no verification of the F-vector or of the transfer function. This can be useful when the same control loop structure is being used for successive time histories and there is no need to verify each time.

(Default: 1)

ISA (Var. No. 23)

System analysis (Frequency Response, Refer to Section II.B.5, pp. 22-24)

1

Does not compute a frequency. If ISA = 1, the user can omit input of frequency response variables and set INA = 1 or 4 depending upon desire for verification of F-vector.

ORIGINAL PAGE 13
OF POOR QUALITY

2

Frequency response is calculated for desired transfer function. Frequencies at which response is calculated are output, along with amplitude ratio and phase.

3

Frequency response is determined for desired transfer function; however, a search is made of the frequency response, and only those frequencies at which relative maxima occur in amplitude ratio are output. Subroutine "MODSH" is used for this search and its outputs are Mode No., frequency, amplitude ratio, phase, and the maximum slope in arriving at each relative maxima occurring within the specified frequency range. (Maximum of 5 relative maxima are allowed over the specified frequency range.)

4

Frequency response is determined as well as a search performed for relative maxima.

(Default: 1)

ISKIP (Var. No. 24)

Output printing option

0

Full print out (useful for debugging only)

1

Minimum print out

(Default: 1)

INFW (Var. No. 25)

Time window options (Refer to Section II.B.1, pp. 19-22)

0

Sliding time window

ORIGINAL PAGE IS
OF POOR QUALITY

1

Sliding time window inoperable.**

2

Finite-length time window

(Default: 1)

**NOTE: The same result will be obtained by specifying INFW = 0 and a time window length greater than the length of the time history used in the identification as that by specifying INFW = 1. This is useful when the amount of storage required for the length of the time window exceeds the storage allowed.

B. TIME HISTORY VARIABLES

These variables specify the times at which events (inputs, outputs, starting, stopping) or lengths of events (length of identification, length of sliding time window) occur within the identification process. The variables are usually specified in units of time, as data is usually sampled in time (e.g., samples per second, minute, hours, etc.); however, if the data is sampled in some other manner (e.g., in terms of distance, etc.) it is necessary only to substitute those units where the units of time appear in these definitions. In the following definitions it is assumed that the time history sample rate is in terms of samples/second. Thus when "seconds" appear in the following definitions we are referring to seconds of data on the input data file.

TST (Var. No. 1)

The number of seconds of data to be ignored before starting the identification procedure. (Default: 0 sec)

TEND (Var. No. 2)

The total length in seconds of the identification procedure, including the ignored time. (Default: 100 sec)

ORIGINAL PAGE IS
OF POOR QUALITY

TDT (Var. No. 3)

The sampling period in seconds used in the identification procedure, and it may be different from the time history sample period. However, it is required to be in even increments of the time history sample period. For example, if the time history sampling period is x seconds and a sample period equal to twice this is to be used in the identification procedure, TDT should be set equal to 2x seconds.

The sample rate used in the identification procedure should be based upon the expected bandwidth, ω_c , of each control loop, with high bandwidths requiring high sampling rates and low bandwidths allowing the use of low sampling rates. As a rule of thumb the minimum sampling rate should be:

$$TDT = \frac{\pi}{10\omega_c} \quad (34)$$

[Default: 0.1 (10 samples/sec)].

TOUT (Var. No. 4)

The time in seconds between outputs of the identification estimates and system analysis routines. (Default: 1 sec)

TC (Var. No. 5)

The time in seconds between computations of the identification estimates of the F-vector coefficients. This is usually equal to TOUT, as it is not usually necessary to compute the coefficients more (or less) often than they are output. It is important to note that the number of times these estimated coefficients are computed will not affect their values at future times or at the same points in time, i.e., with this variable equal to TOUT, it merely allows the user to monitor the estimation procedure as often as is desired. (Default: 5 sec)

TW (Var. No. 6)

The length in seconds of the sliding or finite time window. This needs to be specified only if INFW is equal to zero (0) or two (2). The length of the time window should be based upon the expected bandwidth, ω_c , of each control loop structure; with the higher the expected bandwidth the shorter the required time window and the lower the expected bandwidth the

ORIGINAL PAGE IS
OF POOR QUALITY

larger the required time window. As a rule of thumb an initial estimate for the length of the time window is

$$TW = \frac{10\pi}{\omega_c} \quad (35)$$

The approximate maximum length of the sliding time window is determined by

$$TW_{\max} = [(\text{amount of storage in VAR array})/(\text{NVAR})]*\text{TDT}$$

However, if specifying a time window length greater than that allowed, the program will reset TW to the maximum length allowed. Hence the maximum sliding time window is easily obtained by specifying a time length greater than the length of the time history. (Default: 100 sec) (Approx. max. for NVAR = 2 and TDT = 0.1 is 12.5 sec)

C. NIPIP VARIABLES

This group of variables defines the estimation equation describing the control loop structure to be identified. These variables are probably the most confusing to learn for the novice user. As a teaching aide, we will refer to Example A presented in Section IV.A (pp. 44-61) when specifying values for the NIPIP variables shown below; therefore, the reader should review these pages before proceeding.

X (Transparent to user, Refer to Section III.F, pp. 38-40)

This is an internal array of length m, where m is the number of variables stored in a single record on the data input file. This array accomplishes the data transfer from the data input file to the program.

For Example A in Section IV.A: X(1) = time, X(2) = θ , X(3) = δ_e , X(4) = w_g , and X(5) = h.

ORIGINAL F.
OF POOR QUALITY.

NC (Var. No. 26)

Specifies the number of estimation coefficients on the right-hand side of the estimation equation (i.e., the size of the F-vector), that is, NC is the number of degrees of freedom.

For Example A in Section IV.A: $NC = 5$; a_1 , a_2 , b_1 , b_2 , and B.

IBIAS (Var. No. 27)

Specifies a bias degree of freedom (DOF) to be added to the F-vector. This DOF is treated as a coefficient and, as such, must be included when specifying NC. (Refer to Section II.B.2, pp. 20-22)

0

Do not add bias DOF to F-vector.

1

Add bias DOF to F-vector.

For Example A in Section IV.A: $IBIAS = 1$.

NVAR (Var. No. 28)

Specifies the total number of independent parameters in the estimation equation. An independent parameter is defined as an element of the X array which is used in forming the estimation equation. Using an element of the X array more than once (i.e., using its current and/or past values) it is considered a single independent parameter.

For Example A in Section IV.A: $NVAR = 2$; δ_e and θ .

INDEX (Var. Nos. 29 to 33)

This is an array of length NVAR which specifies the elements of the X array to be used in forming the estimation equation. The elements of the INDEX array are specified using the subscript of the X array element corresponding to the parameters used in the estimation equation. The first element of this array specifies the X array element corresponding to the dependent variable of the estimation equation (i.e., left-hand side) as well as when this X array element is used as an independent variable in the estimation equation (i.e., when the F-vector contains past values of the dependent variable). The

**ORIGINAL PAPER
OF POOR QUALITY**

succeeding elements of this array specify the X array elements corresponding to the independent parameters of the estimation equation (i.e., right-hand side) (past values of the dependent or independent variables are not considered as new variables in the estimation equation). The indexes or subscripts of the INDEX array are given values which are used for future reference in specifying the F-vector. The first index is given a value of zero (0) and each succeeding index is given a value 1 greater than the previous index.

For Example A in Section IV.A: INDEX(1) = 3, X(3) = δ_e ;
INDEX(2) = 2, X(2) = θ

NVEC (Var. Nos. 34 to 43)

This integer array of length NC^* specifies how the independent parameters, as well as their past values, are entered into the F-vector. The arrangement of the parameters in the F-vector is determined by the order they are specified in the NVEC array. The elements of the NVEC array require an input of the form KI, where I points to the parameters in the INDEX array and K specifies the current or past values of the parameter. The codes for using I and K are shown below:

K	Past Value
0	Use current value
1	Use first past value
2	Use second past value
.	
.	
.	
K	Use Kth past value

I (Reference value)	Points to
0	Dependent parameter, i.e., INDEX(1)
1	First independent parameter, i.e., INDEX(2)
.	
.	
.	
I	Ith independent parameter, i.e., INDEX(I+1)

*NC-1 if a bias DOF is used.

**ORIGINAL PAGE IS
OF POOR QUALITY**

For example, NVEC(5) = 32 would be used to specify that the fifth element of the F-vector is the third (K=3) past value of the third parameter in the INDEX array (i.e., the second (I=2) independent parameter).

For Example A in Section IV.A: NVEC(1) = 10, $\delta_{e_{n-1}}$; NVEC(2) = 20, $\delta_{e_{n-2}}$; NVEC(3) = 11, θ_{n-1} ; NVEC(4) = 21, θ_{n-2} .

BV (Var. Nos. 13 to 17)

This array of length NVAR specifies the reference value associated with each parameter defined in the INDEX array. The first element of BV specifies the reference value for the first element defined in the INDEX array and so forth.

When using the user-defined subroutine MODE, BV is a two-dimensional array where the number of rows equals NVAR and the number of columns equals NPS. For example, BV(3,2) would be used to specify the reference value for the third element of the INDEX array defined in MODE for control loop structure number two.

For Example A in Section IV.A: BV(1) = -.10681, Trim δ_e (rad); BV(2) = 0.0889, Trim θ (rad).

D. FREQUENCY RESPONSE VARIABLES

This group of variables specifies the transfer function for which a frequency response is to be determined. The transfer function is made up of terms of the estimation equation and includes the dependent variable, i.e., left-hand side, as well as estimation coefficients, \hat{c} . To obtain a frequency response the user must set the ISA flag (see Subsection A above).

The range of the frequency response is based upon the bandwidth of the system to be identified. The frequency range should be specified such that it covers approximately a decade on a log plot of ω , with ω_c roughly centered within the frequency range. For our example, $\omega_c = 2.5$ rad/sec; hence a frequency range of $\omega = 0.5$ to 8 rad/sec should be sufficient.

ORIGINAL PAGE IS
OF POOR QUALITY

WMIN (Var. No. 7)

The minimum frequency in the frequency response. (Default: 0.1 rad/sec)

For Example A in Section IV.A: WMIN = 0.5.

WMAX (Var. No. 8)

The maximum frequency in the frequency response. (Default: 10. rad/sec)

For Example A in Section IV.A: WMAX = 8.0.

WINC (Var. No. 9)

The multiplying increment by which the frequency increases from w_{min} to w_{max} . Successive frequencies are determined by

$$w_n = w_{n-1} * w_{inc}$$

where $w_{min} < w_n < w_{max}$ (36)

(Default: 2.)

For Example A in Section IV.A: WINC = 2. which gives frequencies of 0.5, 1, 2, 4, and 8 rad/sec.

NOTE: The frequency response is calculated for a maximum of 25 frequencies between w_{min} and w_{max} ; hence if w_{inc} is too small and it requires more than 25 frequencies as defined by Eq. 36 to cover the frequency range only the first 25 frequencies will be used in calculating the response.

HFG (Var. No. 10)

A factor in the transfer function numerator which multiplies the high frequency gain by the value specified. Usually this factor is 1 or -1, which changes the phase such that it lies between 0 and -180 deg. The frequency response phase angle as output by the NAS program will lie between 90 and -270 deg. Thus, in those cases where the phase line crosses these boundaries, it may be difficult to interpret the phase with large discontinuities (e.g., going from +89 deg to -269 deg), and changing HFG from 1 to -1 or vice versa will change the phase by ± 180 deg, and it will then lie within the phase boundaries. (Default: 1.0)

ORIGINAL FILE
OF POOR QUALITY

For Example A in Section IV.A: HFG = 1.

NB (Var. No. 44)

Specifies the order of the numerator of the transfer function, its value is equal to the number of estimation coefficients of the c vector in the numerator.

For Example A in Section IV.A: NB = 2; b_1 and b_2 .

NA (Var. No. 45)

Specifies the order of the denominator of the transfer function; its value is equal to the number of elements of the estimation equation in the denominator.

For Example A in Section IV.A: NA = 3; 1, a_1 , a_2 .

KTAU (Var. No. 46)

Specifies a pure time delay in the transfer function. Its value is equal to the number of samples of delay. KTAU must be greater than or equal to zero, as there is no prediction of information prior to the current value. (Default: 0)

For Example A in Section IV.A: KTAU = 0.

NBI (Var. Nos. 47 to 51)

This array of length NB specifies which coefficients of the F-vector make up the numerator of the transfer function. The coefficients are usually specified in increasing powers of z (i.e., the lowest power of z in the numerator is the first element of NBI).

For Example A in Section IV.A: NBI(1) = 3, b_1 ;
NBI(2) = 4, b_2 .

NAI (Var. Nos. 52 to 56)

This array of length NA specifies which elements of the estimation equation make up the denominator of the transfer function. When specifying a transfer function denominator which contains the dependent variable of the estimation equation, i.e., the left-hand side, the first element of NAI must be equal to zero.

For Example A in Section IV.A: NAI(1) = 0, 1; NAI(2) = 1, a_1 ; NAI(3) = 2, a_2 .

ORIGINAL FORM
OF POOR QUALITY

IFOPT (Var. No. 57)

Specifies the form of output for the amplitude ratio of the transfer function.

1

Gives amplitude ratio squared.

2

Gives the linear amplitude ratio

3

Gives the amplitude ratio in dB.

(Default: 3)

For Example A in Section IV.A: IFOPT = 3.

E. WARNING TOLERANCES

This group of variables specifies the warning criteria which trigger warning messages within the NAS software. These warning messages are activated when the criteria are not satisfied. The warning criteria are specified as the maximum or minimum allowable ratios of certain variables within the software. Some of the reasons why these warning messages appear are also given in order to allow the user to determine if any corrective action should be taken.

TOL (Var. No. 11)

Specifies the maximum ratio of the largest to smallest elements along the diagonal of the $H^T H$ matrix. The purpose of this warning is to alert the user to the possibility of an ill-conditioned $H^T H$ matrix which may affect the solution for the estimation coefficients. An "ill-conditioned" $H^T H$ matrix usually results from large differences in the numerical values of the variables used to form the F-vector, e.g., if one variable is in units of degrees and another is in units of radians. This problem can usually be avoided by scaling the variables used to form the F-vector or using a consistent set of units. (Default: 1000)

**ORIGINAL PAGE IS
OF POOR QUALITY**

RLD (Var. No. 12)

Specifies the minimum ratio of the determinant of the $H^T H$ matrix to the smallest element along the diagonal of the $H^T H$ matrix. The purpose of this warning is to alert the user that the F-vector may contain redundant DOF, which can cause a fragile solution for the estimation coefficients. Redundant DOF result in linear dependence within the $H^T H$ matrix, causing the determinant to approach zero. This warning is used in conjunction with the correlation coefficient, R^2 , to assess the fragility of the estimated solution. The number of DOF are assumed appropriate when R^2 does not increase with additional DOF, and/or this warning does not appear. (Default: 0.0001)

F. USER DEFINED SUBROUTINES

This section defines the two user-defined subroutines which may be tailored to suit any particular estimation scheme.

MODE (involved by control flag, MOD; Refer to Section III.A, pp. 26-29)

Specifies the NAS variables when several control loop structures are being implemented simultaneously. It is necessary to specify only those NAS variables which are not defaulted or change from one control loop structure to another. To access this subroutine the control flag, MOD (variable No. 20), must be greater than zero. The default listing of MODE is given in Table 2. For additional examples of using subroutine MODE see Section IV.

PREPR

Performs any processing of input data prior to use in the NIPIP algorithm. This subroutine transfers the data from the data input array, DI, to the internal array, X, which is used when specifying the NIPIP variables. (Refer to Section III.C, pp. 31-34.) The default and most common use of subroutine PREPR is to simply transfer the data input file into the X

TABLE 2

DEFAULT LISTING OF THE MODE SUBROUTINE

```
C      5.5 SUBROUTINE MODE
C
C      THIS IS A USER DEFINED SUBROUTINE FOR SPECIFYING THE
C      NAS VARIABLES WHEN SEVERAL CONTROL LOOP STRUCTURES ARE TO BE
C      IMPLEMENTED SIMULTANEOUSLY
C
C      TO ACCESS THIS SUBROUTINE CONTROL FLAG MOD MUST BE GREATER
C      THAN ZERO
C
C      THIS IS THE DEFAULT LISTING
C
C      SYSTEMS TECHNOLOGY, INC.    MAY 1982    6DH
C
C      SUBROUTINE MODE(IM)
C
C      =====
C
C      IM...SPECIFIES WHICH CONTROL LOOP STRUCTURE IS BEING USED
C
C      NAS VARIABLES ARE SPECIFIED USING THE NAMES IN THE COMMON BLOCKS
C
C      =====
C
C      DIMENSION NVEC(10),INDEX(5),NBI(5),NAI(5),BV(5,2)
C
C      COMMON /TIME/ TST,TEND,TDI,TOUT,TC,TW,
1          WMIN,WMAX,WINC,HFG,TOL,RLD,BV
C
C      COMMON /BAS/  IN,IOUT,MOD,NPS,INA,ISA,ISKIP,INFW,
2          NC,IBIAS,NVAR,INDEX,NVEC,
3          NB,NA,KTAU,NBI,NAI,IFOPT
C
C
C      THE DEFAULT CASE IS FOR CONTROL FLAG MOD=0 ...
C      THEREFORE THIS SUBROUTINE IS NOT REQUIRED TO SPECIFY THE CONTROL
C      LOOP STURCTURES
C
C      RETURN
C      END
```

A>

ORIGINAL PROFILE OF POOR QUALITY

array. The default listing of PREPR is given in Table 3. Additional uses of subroutine PREPR are for smoothing, reconstructing, scaling, etc., of the raw input data prior to use in the NIPIP algorithm.

G. INPUT DATA FILE FORMAT

The input data files used by the NAS software must be given in the format shown in the previous examples. This format is required to allow the program to verify the F-vector and transfer function. The format of the input data is specified below.

INPUT DATA FILE FORMAT

```

4      Line 1: M; the number of input variables per line
N      -
R      FT      } Lines 2 through (M+1): The variable names and units; the first
R      DD      } four characters are used to verify the F-vector and transfer
R      FPS2    } function
NU      NU      }
LANDING PROFILE Line M+2: File identifying comments
+.1000E+01 +.2800E+04 +.8372E+00 +.1495E+02 } The first two entries of column one are used to
+.2000E+01 +.2760E+04 +.8533E+00 +.1479E+02 } compute the sample rate of the data stored on
+.3000E+01 +.2720E+04 +.8694E+00 +.1463E+02 } the input file
+.4000E+01 +.2690E+04 +.1191E+01 +.1312E+02 }
+.5000E+01 +.2650E+04 +.1206E+01 +.1300E+02 }
+.6000E+01 +.2600E+04 +.1191E+01 +.1297E+02 }
+.7000E+01 +.2570E+04 +.9016E+00 +.1418E+02 }
+.8000E+01 +.2530E+04 +.8694E+00 +.1428E+02 }
+.9000E+01 +.2480E+04 +.1288E+01 +.1244E+02 }
+.1000E+02 +.2450E+04 +.9016E+00 +.1395E+02 }
+.1100E+02 +.2410E+04 +.9982E+00 +.1342E+02 }
+.1200E+02 +.2360E+04 +.1256E+01 +.1234E+02 }
+.1300E+02 +.2320E+04 +.1127E+01 +.1272E+02 }
+.1400E+02 +.2290E+04 +.1352E+01 +.1192E+02 }
      :
      :
-1.0   0.0   0.0   0.0  ----- End-of-file marker

```

**ORIGINAL PAGE IS
OF POOR QUALITY**

As can be seen, the last line contains a negative number in the first column. This is a signal to the program that it has reached the end of the input data file and will keep it from attempting to read past the end of the file. Thus the first column of the input data file cannot contain any negative numbers prior to the last line, or the program will terminate reading data at the point it encounters the negative number in the first column.

Also it should be noted that the first two entries of the first column determine the sample rate of the data stored on the input file, and this sample rate is utilized when computing the number of samples to be used or ignored when specifying the time history variables. For example, the sampling rate in this case is

$$2 - 1 = 1 \text{ sample/unit}$$

where a "unit" is generally one second. Since specifying TST, TEND, TDT, TOUT, TC, and TW in the number of units (e.g., seconds) determine the number of units to be used or ignored. If an incremental counter or clock signal is available, placing it in the first column is recommended, since the first column cannot contain negative numbers prior to the last line. However, it is not necessary that this first column be the incremental counter or clock, only that the user recognize that the first two entries of column one determine the sample rate, as used by the program, that the time history variables should be specified based upon this sample rate, and that this column cannot contain negative numbers prior to the last line.

TABLE 3

DEFAULT LISTING OF THE PREPR SUBROUTINE

```
C      5.6 SUBROUTINE PREPR
C
C      THIS IS A USER DEFINED SUBROUTINE FOR PREPROCESSING THE INPUT
C      DATA TO NIPIP
C
C      THIS IS THE DEFAULT LISTING FOR TRANSFERING DATA FROM THE DI
C      ARRAY TO THE X ARRAY
C
C      SYSTEMS TECHNOLOGY. INC.    MAY 1982    GDM
C
C      SUBROUTINE PREPR(DI,X,IM,M)
C
C      =====
C
C      DI...INPUT DATA ARRAY FROM DATA FILE, TAPE, OR FLIGHT TEST
C      X ...OUTPUT DATA ARRAY AFTER PROCESSING
C      IM...SPECIFIES WHICH CONTROL LOOP IS BEING PROCESSED
C      M ...NUMBER OF VARIABLE S PER RECORD ON THE INPUT DATA FILE
C
C      =====
C
C
C      DIMENSION DI(1),X(1)
C
C
C      TRANSFER DATA FROM DI TO X ARRAY
C
C      DO 1 I=1,M
C      X(I)=DI(I)
C 1  CONTINUE
C
C
C      RETURN
C      END
A.
```

**ORIGINAL PAGE IS
OF POOR QUALITY**

**SECTION IV
SOME ILLUSTRATIVE EXAMPLES**

The four examples presented below are designed to illustrate how to use the NAS software package as well as to demonstrate its capabilities. Each example is presented as follows:

1. Statement of the problem and objectives
2. Consideration of NIPIP parameters
 - a. F-vector
 - b. Time window and frame time
 - c. Frequency response
3. Definition of the NAS variables
4. Description of the NAS outputs
5. Discussion of the results

The first example is a single-input, single-output compensatory tracking task. The second example is an extension of the first, an outer loop has been added to the control task, thereby providing a more complex example. The third example identifies the pilot's control strategy during a flare to landing. The fourth example identifies a helicopter pilot's control strategy while performing a deceleration to hover. This last example demonstrates how NIPIP can be used for solving nonlinear control problems.

A. A SINGLE-LOOP EXAMPLE

1. Statement of the Problem and Objectives

The functional block diagram shown in Fig. 4 summarizes a single-input, single-output, compensatory tracking control task. The "pilot," depicted as Y_p^θ , is regulating pitch attitude, θ , using the elevator, δ_e . The aircraft is flying through atmospheric turbulence, w_g , and the compensatory control task is to maintain the commanded value of pitch attitude, θ_c .

The pilot, aircraft, and vertical gusts were all simulated on a digital computer. Details of the aircraft and gust simulation are presented in Appendix A. The form and numerical values of the simulated pilot control strategy, Y_p^θ , were chosen to be representative of human pilot dynamics measured by other techniques (Ref. 6), and also to give good closed-loop characteristics. The Laplace transform of Y_p^θ is given below,

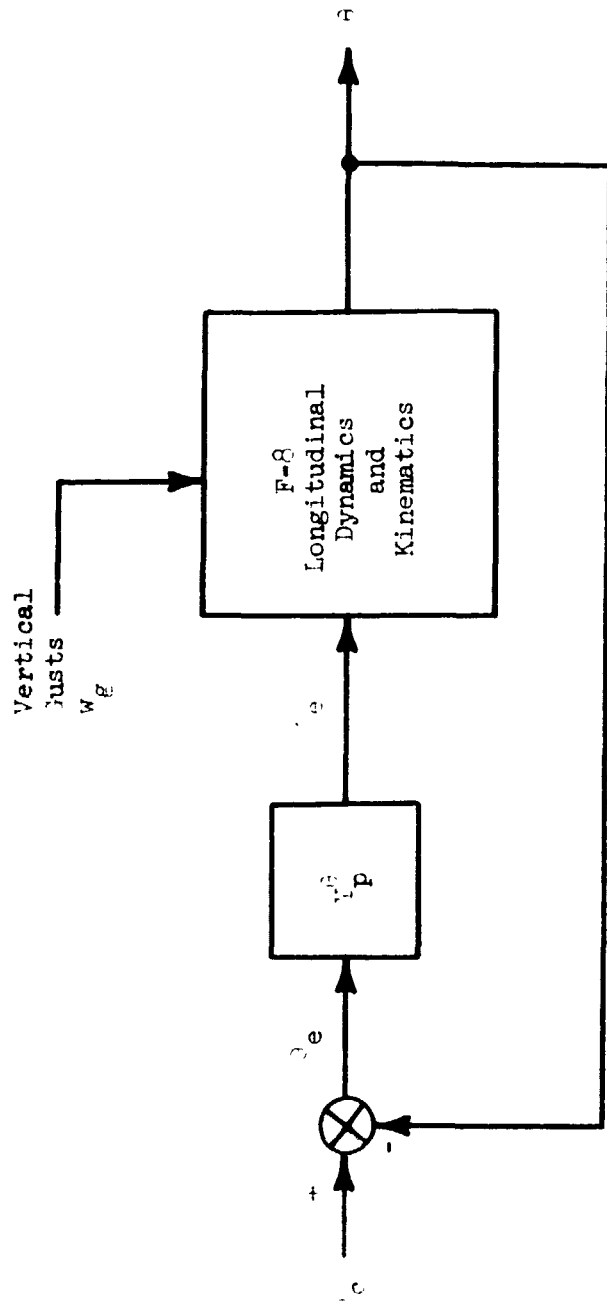
$$Y_p^\theta(s) = \frac{\delta_e}{\theta} = -4.87 \frac{s + 3.0}{s^2 + 2(0.3)(6.0)s + (6.0)^2} \left(\frac{\text{rad } \delta_e}{\text{rad } \theta} \right) \quad (37)$$

The $Y_p^\theta(s)$ shown in Eq. 35 yields a bandwidth of 2.5 rad/sec and a phase margin of 35 deg.

Figure 5 contains time histories of the closed-loop (i.e., aircraft and pilot) response to vertical gusts. The objective of NIPIP is to identify the parameters of Eq. 35 using the time histories of θ and δ_e . For this example θ_c is constant throughout the time history (i.e., θ_c is equal to the trim values of θ).

2. Consideration of NIPIP Parameters

a. Specifying the F-Vector. The first thing to decide is how many degrees of freedom are required to describe the system to be identified.



Notes: 1. Bandwidth of θ -loop is 2.5 rad/sec with $Y_p^\theta(s) = \frac{-4.87(3.0)}{[0.3; 6.0]}$, $\frac{\text{rad } \delta_e}{\text{rad } \theta_e}$

2. θ_c is a constant 5.09 deg

Figure 4. A Single-Input, Single-Output Example

ORIGINAL PAGE IS
OF POOR QUALITY

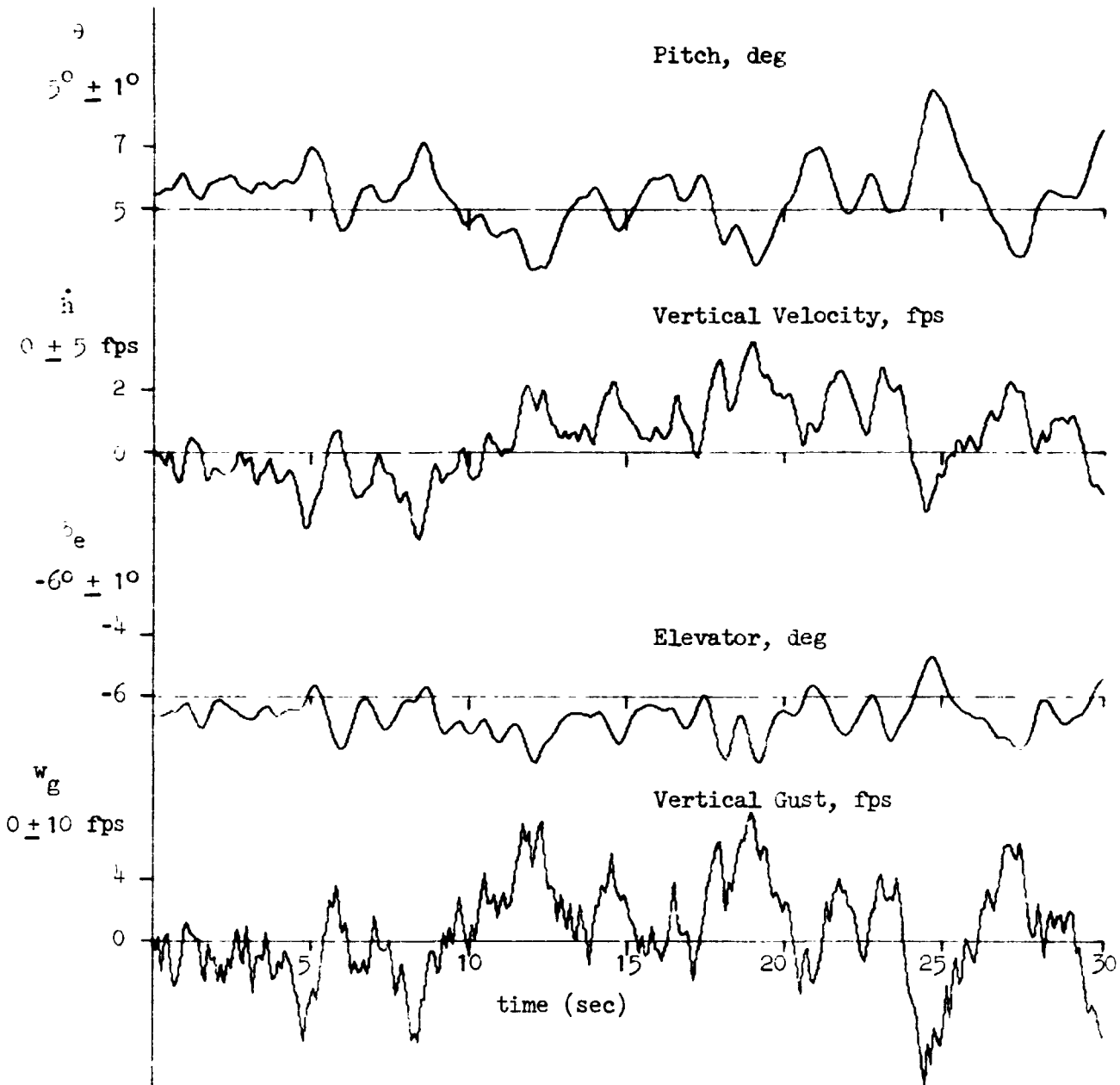


Figure 5. Time History Used for Example A.
(Closed-Loop Response to Vertical Gust.)

**ORIGINAL PAGE IS
OF POOR QUALITY**

For this example we know there are four parameters describing Y_p^θ : two in the numerator and two in the denominator, as shown in Eq. 37. Since NIPIP operates on discrete points in time, we express the unknown parameters of Y_p^θ using z-transforms (see Entry 5 in Table 1). Thus,

$$Y_p^\theta(z) = \frac{\delta_e}{\theta_e} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (38)$$

where $\theta_e = \theta_c - \theta$, as shown in Fig. 4. Using the notation derived in Section II, Eq. 38 can be used to write a difference equation of the form,

$$\begin{aligned} \delta_{e_n} &= a_1 \delta_{e_{n-1}} + a_2 \delta_{e_{n-2}} + b_1 \theta_{e_{n-1}} \\ &+ b_2 \theta_{e_{n-2}} + B \end{aligned} \quad (39)$$

$$= [a_1 \ a_2 \ b_1 \ b_2 \ B] \begin{bmatrix} \delta_{e_{n-1}} \\ \delta_{e_{n-2}} \\ \theta_{e_{n-1}} \\ \theta_{e_{n-2}} \\ 1 \end{bmatrix}$$

$$\equiv \underline{F}_n \hat{c}$$

The "B" term in Eq. 39 represents a bias DOF and is included for the reasons presented in Section II.

In general one will not know how many degrees of freedom to include in the F-vector. The effects of over and under specifying the number of degrees of freedom are presented later in this example. This topic is also addressed in Ref. 3.

**ORIGINAL PAGE IS
OF POOR QUALITY**

b. Frame Time and Time Window. The frame time (sample period) and time window can be calculated based upon the bandwidth of the system to be identified, ω_c , as discussed in Section II. Using Eqs. 34 and 35, with $\omega_c = 2.4$.

$$\text{TDT} = \frac{\pi}{10\omega_c} = 0.13 \text{ sec} \quad (\text{use } 0.10 \text{ sec})$$

$$\text{TW} = \frac{10\pi}{\omega_c} = 12.6 \text{ sec} \quad (\text{use } 11.6 \text{ sec}^*)$$

In general one will not know the exact bandwidth of the system to be identified. However estimates of ω_c can be computed based upon a closed-loop analysis of the task.

c. Frequency Response. The range of the frequency response is also based upon the bandwidth of the system to be identified. The frequency range should be specified such that it covers approximately a decade on a log plot of ω , with ω_c roughly centered within the frequency range. For $\omega_c = 2.5$ rad/sec, hence a frequency range of $\omega = 0.5$ to 8.0 rad/sec should be sufficient.

3. Definition of the NAS Variables

We are now ready to define the NAS input variables presented in Section III. Table 4 contains a partial list of the input file to be used. The time histories in this file were used to construct the plots shown in Fig. 4. The following subsections define only those variables

* $T_W = 11.6$ sec turned out to be a convenient number for the NAS software, due to the dimension of the VAR storage array.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 4

INPUT FILE USED FOR EXAMPLE A

(File name is EG1)

5
 TIME SEC _____ t
 THEY RAD _____ θ
 ELEV RAD _____ δ_e
 Wg FPS _____ w_g
 HDOT FPS _____ \dot{h}

EXAMPLE 1

+ .000000E+00	+ .8890780E-01	- .1068150E+00	+ .0000000E+00	+ .0000000E+00
+ .1000000E+00	+ .8899810E-01	- .1068150E+00	- .5907230E+00	- .8804700E-01
+ .2000000E+00	+ .8905940E-01	- .1067750E+00	+ .1955400E+00	- .9445950E-01
+ .3000000E+00	+ .8931070E-01	- .1067200E+00	- .1904910E+01	- .2672200E+00
+ .4000000E+00	+ .8956330E-01	- .1065800E+00	+ .1047440E+00	- .3803560E+00
+ .5000000E+00	+ .8942340E-01	- .1063890E+00	+ .3889030E+00	+ .1141360E-01
+ .6000000E+00	+ .8946960E-01	- .1063660E+00	- .1520230E+01	- .1445580E+00
+ .7000000E+00	+ .8998520E-01	- .1064000E+00	- .2790100E+01	- .6756900E+00
+ .8000000E+00	+ .9064760E-01	- .1062470E+00	- .2503280E+01	- .9568670E+00
+ .9000000E+00	+ .9105760E-01	- .1058820E+00	- .1503820E+01	- .8341710E+00
+ .1000000E+01	+ .9086740E-01	- .1055110E+00	+ .6070840E+00	- .3153000E+00
+ .1100000E+01	+ .9013000E-01	- .1054630E+00	+ .1131780E+01	+ .2874910E+00
+ .1200000E+01	+ .8933970E-01	- .1059240E+00	+ .5167260E+00	+ .4615900E+00
+ .1300000E+01	+ .8875460E-01	- .1067160E+00	+ .2644180E+00	+ .3705290E+00
+ .1400000E+01	+ .8840140E-01	- .1075080E+00	+ .1443370E+00	+ .2430000E+00
+ .1500000E+01	+ .8828550E-01	- .1080340E+00	+ .1737390E-01	+ .8440780E-01
+ .1600000E+01	+ .8865190E-01	- .1081560E+00	- .1962770E+01	- .3492200E+00
+ .1700000E+01	+ .8948780E-01	- .1077570E+00	- .2470100E+01	- .9154660E+00
+ .1800000E+01	+ .9003320E-01	- .1068590E+00	- .2185740E+00	- .7933200E+00
+ .1900000E+01	+ .9022900E-01	- .1059060E+00	- .1230040E+01	- .4952280E+00
+ .2000000E+01	+ .9034260E-01	- .1052860E+00	- .9719870E+00	- .5089950E+00
+ .2100000E+01	+ .9050010E-01	- .1050940E+00	- .2495250E+01	- .5965460E+00
+ .2200000E+01	+ .9065720E-01	- .1052240E+00	- .1474700E+01	- .7607980E+00
+ .2300000E+01	+ .9080150E-01	- .1055210E+00	- .2605610E+01	- .7104950E+00
+ .2400000E+01	+ .9087300E-01	- .1058350E+00	- .1057210E+01	- .7187840E+00
+ .2500000E+01	+ .9079450E-01	- .1060850E+00	- .1684380E+01	- .5534400E+00
+ .2600000E+01	+ .9055360E-01	- .1062740E+00	- .7274750E-01	- .4138530E+00
+ .2700000E+01	+ .8995160E-01	-	-	-

A.

t	θ	δ_e	w_g	\dot{h}
(sec)	(rad)	(rad)	(fps)	(fps)

ORIGINAL PAGE IS
OF POOR QUALITY

which are not defaulted. The "direct input mode" was used to define these variables in the NAS program.

a. NIPIP Variables (Refer to Section III.C, pp. 31-34). From Table 4 we can infer the X-vector,

$$\underline{X} = \{t, \theta, \delta_e, w_g, \dot{h}\}$$

From Eq. 39 and X, (Refer to Section III.C., pp. 31-34)

NC	= 5	Number of degrees of freedom
IBIAS	= 1	Bias degrees of freedom
NVAR	= 2	Number of independent parameters in the estimate equation; δ_e and θ
INDEX	= 3,2	Location of independent parameters in <u>X</u> array; $X(3) = \delta_e$ and $X(2) = \theta$
NVEC	= 10, 20, 11, 21	How independent parameter is entered in the F-vector
BV	= -.10681, 0.08890	Reference (trim) value of the independent variable

b. Time History Variables (Refer to Section III.B, pp. 29-31)

TDT	= 0.10	Sampling period
TW	= 11.6	Seconds of sliding or finite time window
TEND	= 30.0	Total time
TOUT	= 5.0	Seconds between outputs of identification estimate
TC	= 5.0	Seconds between computations of identification estimate

ORIGINAL PAGE IS
OF POOR QUALITY

c. Frequency Response Variables (Refer to Section III.D, pp. 34-37)

WMIN	=	0.5	
WMAX	=	8.0	→ $\omega = 0.5, 1.0, 2.0, 4.0, 8.0$ rad/sec
WINC	=	2.0	
NB	=	2	Order of numerator, number of coefficients of c
NA	=	3	Order of denominator, number of elements of c
NBI	=	{3, 4}	Specifies which coefficient is in the numerator; b_1 and b_2
NAI	=	{0, 1, 2}	Specifies which coefficient is in the denominator; 1, a_1 , and a_2

d. Control Flags (Refer to Section III.A, pp. 26-29)

INA	=	3	F-vector, transfer function frequency response option
ISA	=	2	Frequency response option
INFW	=	0	Time window option

e. User-Defined Subroutines

The default subroutines for MODE and PREPR will be used for this example (see Section III.F, pp. 32-35). The namelist option will be used to set all of the necessary NAS variables.

4. Description of the NAS Outputs

Table 5 contains an annotated output of the NAS program. Items typed by the user are underlined. The outputs for both 5 and 7 DOF are included in the table.

Note that the solution for \hat{c} is convergent after only 50 data points (see output at $t = 5.4$ sec), and is invariant throughout the run. For the 7 DOF case, a "warning" message is output due to an "overspecified F-vector." This message resulted from the determinant of the $H^T H$ matrix being very small (i.e., 10^{-34}), which indicates that there are redundant degrees of freedom in the F-vector.

5. Discussion of the Results

Figure 6 contains frequency responses of the exact and estimated pilot describing function, $Y_p^\theta(j\omega)$. Note that the estimated and exact $Y_p^\theta(j\omega)$ are in perfect agreement for the 5 DOF case.

In addition to the 5 DOF case, Fig. 6 also presents the $\hat{Y}_p^\theta(j\omega)$ using 3, 4, and 7 DOF. The correlation coefficients, R^2 , and determinants of the $H^T H$ matrix, Δ , are also shown in the figure. Note that the 3 DOF case did not fit the true frequency response and that R^2 is 0.88. The 4 DOF case came closer to fitting the true frequency response, and R^2 increased to 0.98. The 7 DOF case produced the same result as the 5 DOF case, with $R^2 = 1.00$, but note that Δ is 0.1×10^{-34} ! This low a value of Δ indicates a linearly dependent $H^T H$ matrix which makes the solution for \hat{Y}_p^θ very "fragile." That is, any noise in the measurements will cause variations in \hat{Y}_p^θ .

The logical choice for \hat{Y}_p^θ (even if we did not know the true Y_p^θ) would be the 5 DOF case. This choice is based both on the fact that R^2 peaks at 5 DOF and the dramatic decrease in Δ in going from 5 to 7 DOF.

ORIGINAL PAGE IS
OF POOR QUALITY.

TABLE 5

NAS OUTPUT FOR EXAMPLE A

NAS ————— Execute the NAS program

INPUT FILE ? EG1 ————— Filename

1 TIME SEC
2 THET RAD
3 ELEV RAD
4 WG FFS
5 HDJT FFS

Note: User inputs
are underlined

FILE ID : EXAMPLE 1

NO. VAR. 5

DIRECT INPUT USING NAMELIST

VAR. NO. ? 26
NC = 0 CHANGE TO 5
NC = 5
VAR. NO. ? 27
IBIAS = 0 CHANGE TO 1
IBIAS = 1
VAR. NO. ? 28
NVAR = 0 CHANGE TO 2
NVAR = 2
VAR. NO. ? 29
INDEX1 = 0 CHANGE TO 3
INDEX1 = 3
VAR. NO. ? 30
INDEX2 = 0 CHANGE TO 2
INDEX2 = 2
VAR. NO. ? 34
NVEC1 = 0 CHANGE TO 10
NVEC1 = 10
VAR. NO. ? 35
NVECC = 0 CHANGE TO 20
NVECC = 20
VAR. NO. ? 36
NVECC3 = 0 CHANGE TO 11
NVECC3 = 11
VAR. NO. ? 37
NVECC4 = 0 CHANGE TO 21
NVECC4 = 21
VAR. NO. ? 13
BV1 = 0.000 CHANGE TO -.10681
BV1 = -.1068
VAR. NO. ? 14
BV2 = 0.000 CHANGE TO .0889
BV2 = .8890E-01
VAR. NO. ? 3
TDT = .1000E+00 CHANGE TO .1
TDT = .1000

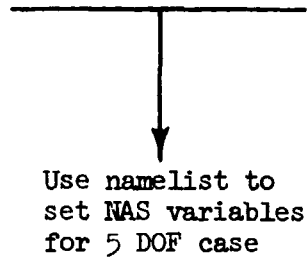


TABLE 5 (Continued)

VAR. NO. 7
 WMIN = .1000E+00 CHANGE TO .5
 WMIN = .5000
 VAR. NO. 8
 WMAX = 10.000 CHANGE TO 8.
 WMAX = 8.000
 VAR. NO. 9
 WINC = 2.000 CHANGE TO 2.
 WINC = 2.000
 VAR. NO. 44
 NB = 0 CHANGE TO 2
 NB = 2
 VAR. NO. 45
 NA = 0 CHANGE TO 3
 NA = 3
 VAR. NO. 47
 NB11 = 0 CHANGE TO 3
 NB11 = 3
 VAR. NO. 48
 NB12 = 0 CHANGE TO 4
 NB12 = 4
 VAR. NO. 52
 NA11 = 0 CHANGE TO 0
 NA11 = 0
 VAR. NO. 53
 NA12 = 0 CHANGE TO 1
 NA12 = 1
 VAR. NO. 54
 NA13 = 0 CHANGE TO 2
 NA13 = 2
 VAR. NO. 22
 INA = 1 CHANGE TO 3
 INA = 3
 VAR. NO. 23
 ISA = 1 CHANGE TO 2
 ISA = 2
 VAR. NO. 25
 INFW = 1 CHANGE TO 0
 INFW = 0
 VAR. NO. -1

ORIGINAL PAGE IS
 OF POOR QUALITY

{ A negative subscript
 signals the end of
 the namelist input*

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 1

0 ELEV(N-0) $y = \delta_{e_n}$

THE F-VECTOR (INDEPENDENT VARIABLE):

1 ELEV(N-1)
 2 ELEV(N-2)
 3 THET(N-1)
 4 THET(N-2)
 5 BIAS DOF

$F = [\delta_{e_{n-1}}, \delta_{e_{n-2}}, \theta_{n-1}, \theta_{n-2}, B]$

* A zero subscript causes all of the namelist variables to be output.

TABLE 5 (Continued)

ORIGINAL PAGE IS
OF POOR QUALITY

TRANSFER FUNCTION FOR CONTROL LOOP 1

$$\left. \begin{array}{l} \text{NUM : THET(N-1) THET(N-2) Z##- 0} \\ \text{DEN : 1.0 ELEV(N-1) ELEV(N-2)} \end{array} \right\} \frac{\delta_e}{\theta_e}(z) = \frac{\hat{c}_3 z^{-1} + \hat{c}_4 z^{-2}}{1 - \hat{c}_1 z^{-1} - \hat{c}_2 z^{-2}} \equiv \hat{Y}_p$$

VERIFY VARIABLES TO BE SAVED Note that NAS adjusted T_W to a maximum
ELEV WITH TRIM VALUE OF -.10681 of 11.6 sec.*
THET WITH TRIM VALUE OF .06890

TW,TDT,TC,TOUT = .116E+02 .100E+00 .500E+01 .500E+01

END VERIFICATION...

TYPE "0" TO CONTINUE; "-1" TO RETURN TO NAMELIST?

If F-vector and transfer functions are okay, type 0, otherwise type -1 and correct problem

0

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
5.4	5.3	.143E-05	.982E-18	.100E+01	50	1

CHAT= .14042E+01 -.69763E+00 .44813E+00 -.32906E+00 -.25216E-05 $\hat{c} = \{\hat{c}_i\}$

FREQ / AMP / PHASE

.50	1.00	2.00	4.00	8.00	ω_i (rad/sec)	$ \hat{Y}_p(j\omega) $
-7.67	-7.20	-5.49	-.20	.33		
4.87	9.18	14.46	5.00	-88.44	$\hat{Y}_p(j\omega)$	

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
10.4	10.3	.227E-05	.913E-15	.100E+01	100	1

CHAT= .14043E+01 -.69764E+00 .44812E+00 -.32905E+00 -.24969E-05

FREQ / AMP / PHASE

.50	1.00	2.00	4.00	8.00
-7.67	-7.20	-5.49	-.20	.33
4.87	9.18	14.46	5.00	-88.44

*The maximum allowable T_W is a function of the memory available on the computer used by NAS.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5 (Continued)

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
15.4	15.3	.356E-05	.797E-14	.100E+01	116	1

CHAT= .14043E+01 -.69768E+00 .44805E+00 -.32898E+00 -.25137E-05

FREQ / AMP / PHASE

.50	1.00	2.00	4.00	8.00
-7.67	-7.20	-5.49	-.20	.33
4.87	9.17	14.46	5.00	-88.45

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
20.4	20.3	.397E-05	.128E-13	.100E+01	116	1

CHAT= .14043E+01 -.69773E+00 .44801E+00 -.32893E+00 -.24797E-05

FREQ / AMP / PHASE

.50	1.00	2.00	4.00	8.00
-7.67	-7.20	-5.49	-.20	.33
4.86	9.17	14.45	5.00	-88.46

T	X(1)	Y2B	DHTH	R2	N	CONTRGL LOOP
25.4	25.3	.544E-05	.280E-13	.100E+01	116	1

CHAT= .14044E+01 -.69775E+00 .44800E+00 -.32893E+00 -.24976E-05

FREQ / AMP / PHASE

.50	1.00	2.00	4.00	8.00
-7.67	-7.20	-5.49	-.20	.33
4.86	9.17	14.45	5.00	-88.46

Last print out
for 5 DOF
case. Note that
solution was invariant
throughout run

TABLE 5 (Continued)

ORIGINAL PAGE IS
OF POOR QUALITY

INPUT FILE ? EG1

- 1 TIME SEC
- 2 THET RAD
- 3 ELEV RAD
- 4 WG FPS
- 5 HDOT FPS

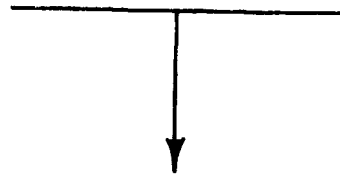
Program loops back
to this point. All
NAS variables are
unchanged from above

FILE ID : EXAMPLE 1

NO. VAR. 5

DIRECT INPUT USING NAMELIST

VAR. NO. ? 2
TEND = 6.000 CHANGE TO 12.
TEND = 12.00
VAR. NO. ? 26
NC = 5 CHANGE TO 7
NC = 7
VAR. NO. ? 38
NVECS = 0 CHANGE TO 30
NVECS = 30
VAR. NO. ? 39
NVEC6 = 0 CHANGE TO 31
NVEC6 = 31
VAR. NO. ? 44
NB = 2 CHANGE TO 3
NB = 3
VAR. NO. ? 45
NA = 3 CHANGE TO 4
NA = 4
VAR. NO. ? 49
NB13 = 0 CHANGE TO 5
NB13 = 5
VAR. NO. ? 55
NA14 = 0 CHANGE TO 6
NA14 = 6
VAR. NO. ? -1



Use namelist to
set NAS variables
for 7 DOF case

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 1

0 ELEV(N-0) $y = \delta_{e_n}$

THE F-VECTOR (INDEPENDENT VARIABLE):

- 1 ELEV(N-1)
- 2 ELEV(N-2)
- 3 THET(N-1)
- 4 THET(N-2)
- 5 ELEV(N-3)
- 6 THET(N-3)
- 7 BIAS DOF

$$\underline{F} = [\delta_{e_{n-1}}, \delta_{e_{n-2}}, \theta_{n-1}, \theta_{n-2}, \delta_{e_{n-3}}, \theta_{n-3}, B]$$

TABLE 5 (Continued)

ORIGINAL PAGE IS
OF POOR QUALITY

TRANSFER FUNCTION FOR CONTROL LOOP 1
 NUM : THET(N-1) THET(N-2) ELEV(N-3) Z##- 0 } Note incorrect
 DEN : 1.0 ELEV(N-1) ELEV(N-2) THET(N-3) } specification of
 transfer function

VERIFY VARIABLES TO BE SAVED
 ELEV WITH TRIM VALUE OF -.10681
 THET WITH TRIM VALUE OF .08890

TW,TDI,TC,TDUT = .116E+02 .100E+00 .500E+01 .500E+01

END VERIFICATION...
 TYPE '0' TO CONTINUE: '-1' TO RETURN TO NAMELIST? -1 Return to namelist
 to correct transfer
 function specification

DIRECT INPUT USING NAMELIST
 VAR. NO. ? 49
 NB13 = 5 CHANGE TO 6
 NB13 = 6
 VAR. NO. ? 55
 NA14 = 6 CHANGE TO 5
 NA14 = 5
 VAR. NO. ? -1

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 1

0 ELEV(N-0)

THE F-VECTOR (INDEPENDENT VARIABLE):

- 1 ELEV(N-1)
- 2 ELEV(N-2)
- 3 THET(N-1)
- 4 THET(N-2)
- 5 ELEV(N-3)
- 6 THET(N-3)
- 7 BIAS DOF

$$\underline{F} = [\delta_{e_{n-1}}, \delta_{e_{n-2}}, \theta_{n-1}, \theta_{n-2}, \delta_{e_{n-3}}, \theta_{n-3}, B]$$

TABLE 5 (Concluded)

ORIGINAL PAGE IS
OF POOR QUALITY

TRANSFER FUNCTION FOR CONTROL LOOP 1
 NUM : THET(N-1) THET(N-2) THET(N-3) Z##- 0
 DEN : .0 ELEV(N-1) ELEV(N-2) ELEV(N-3) } $\frac{\delta_e}{\theta_e}(z) = \frac{\hat{c}_3 z^{-1} + \hat{c}_4 z^{-2} + \hat{c}_6 z^{-3}}{1 - \hat{c}_1 z^{-1} - \hat{c}_2 z^{-2} - \hat{c}_5 z^{-3}} \equiv \hat{Y}_p$

VERIFY VARIABLES TO BE SAVED
 ELEV WITH TRIM VALUE OF -.10681
 THET WITH TRIM VALUE OF .08890

TM, TDT, TC, TOUT = .116E+02 .100E+00 .500E+01 .500E+01

END VERIFICATION...

TYPE "0" TO CONTINUE: "-1" TO RETURN TO NAMELIST" 0

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
5.5	5.4	.148E-05	.351E-34	.100E+01	50	1

CHAT= .96730E+00 -.84101E-01 .44802E+00 -.13305E+00 -.30475E+00
 -.14390E+00 -.35720E-05

*** WARNING *** POSSIBLY OVERSPECIFIED F-VECTOR } Warning message
 MIN. DIAG. ELEM. OF HTH= .5572E-04 DETERMINANT OF HTH= .3509E-34 } due to small value
 TRACE OF HTH= .752 E-22 RATIO LIMIT FOR DETERMINANT = .1000E-03 } of |Hth|

FREQ / AMP / PHASE

.50	1.00	2.00	4.00	8.00	} $\hat{Y}(j\omega)$ is same as 5 th DOF, even with redundant DOF
-7.67	-7.20	-5.49	-.20	.33	
4.87	9.18	14.46	5.00	-88.44	

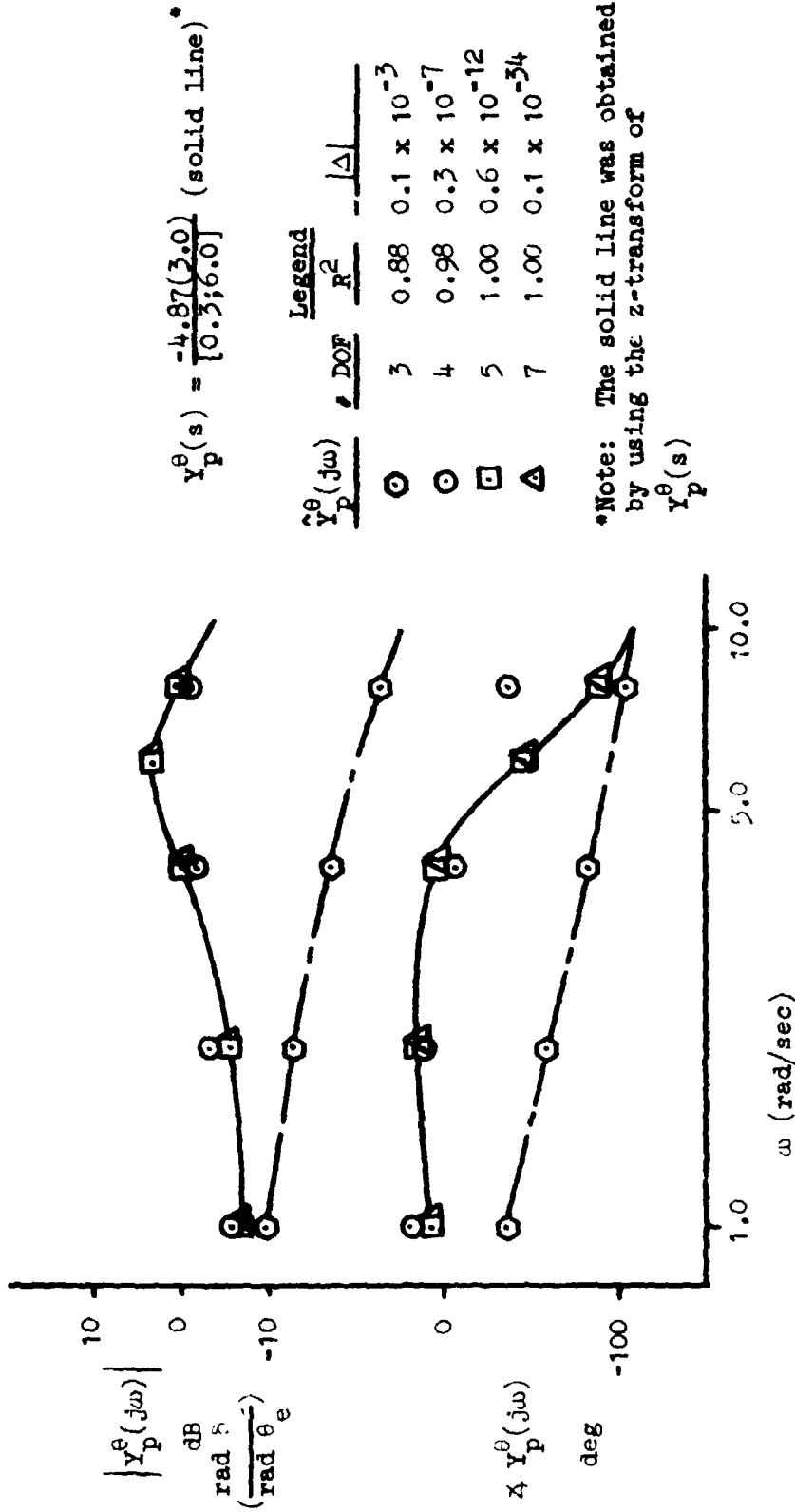


Figure 6. Frequency Responses of Exact and Estimated Y_p^θ for Various Degrees of Freedom

**ORIGINAL PAGE IS
OF POOR QUALITY**

Further discussion of the example presented above can be found in Ref. 3. In addition to the effects of the number of degrees of freedom, Ref. 3 also examines the effects of pilot remnant (which is, in effect, measurement noise), length of the sliding time window, and changes in the parameters of Y_p^θ .

B. A MULTILoop EXAMPLE

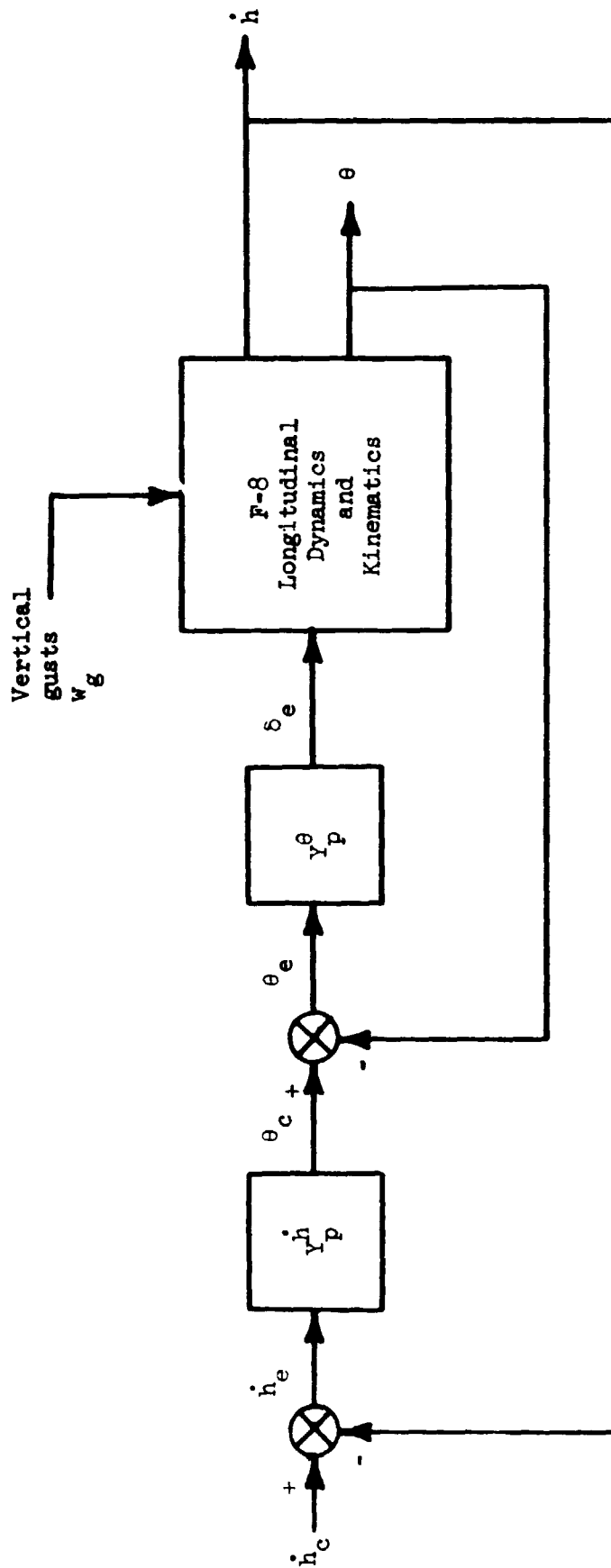
1. Statement of the Problem and Objectives

The next example demonstrates the use of NIPIP to identify multiloop pilot control strategies in the presence of discrete maneuvers. The multiloop control task is depicted in Fig. 7. The pilot's task is to maintain a commanded vertical velocity, \dot{h}_e , by feeding back \dot{h}_c to the commanded pitch attitude, θ_e . This is the "outer-loop" shown in Fig. 7. The pilot is also closing an "inner-loop" by feeding back θ_c to the elevator, δ_e . The inner- and outer-loop pilot control strategies are represented by Y_p^θ and Y_p^h , respectively. As in Example 1 above, the aircraft is flying through atmospheric turbulence, w_g . The pilot, aircraft, and turbulence were simulated on a digital computer. Details of the simulation are contained in Appendix A.

The simulated pilot control strategies, Y_p^θ and Y_p^h , were chosen to yield inner- and outer-loop bandwidths of 3.0 and 0.2 rad/sec, respectively. The Laplace transforms of Y_p^h and Y_p^θ are

$$Y_p^h(s) = \frac{\dot{\theta}_c}{\dot{h}_e} = \frac{K_h}{s} = \frac{0.0005}{s} \text{ (rad/ft)} \quad (40)$$

$$Y_p^\theta(s) = \frac{\theta_e}{\delta_e} = K_\theta = 1.675 \text{ (rad/rad)} \quad (41)$$



- Notes:
1. Bandwidth of θ -loop is 3.0 rad/sec with $Y_p^{\theta} = 1.675 \text{ rad/rad}$
 2. Bandwidth of \dot{h} -loop is 0.2 rad/sec with $Y_p^{\dot{h}} = \frac{0.0005 \text{ rad/ft}}{\text{s}}$
 3. $\dot{h}_c = \begin{cases} 0.0 & \text{for } 0 < t < 15 \text{ sec} \\ -87.5 \text{ ft/s} & \text{for } t \geq 15 \text{ sec} \end{cases}$

ORIGINAL PAGE 10
OF POOR QUALITY

Figure 7. Multi-Loop Control Task Example

**ORIGINAL PAGE IS
OF POOR QUALITY**

The closed-loop response (i.e., pilot plus aircraft) to vertical gusts and \dot{h}_c is shown in Fig. 8. The magnitude of w_g (not shown in the figure) is essentially the same as in Example 1 (see Fig. 5). From $t = 0$ to $t = 15$ sec the commanded vertical velocity is zero. At $t = 15$ sec, \dot{h}_c is instantaneously changed to -87.5 fps.

The objective of NIPIP is to identify Y_p^h and Y_p^θ using the time histories for \dot{h} , θ , and δ_e of Fig. 8. This example also shows how the user-defined subroutines PREPR and MODE are used in the NAS program.

2. Consideration of NIPIP Parameters

a. Specifying the F-Vector. From Fig. 7 the inner and outer pilot control strategies are given by

$$\delta_e = Y_p^\theta (\theta_c - \theta) \quad (42)$$

$$\theta_c = \frac{Y_p^h}{s} (\dot{h}_c - \dot{h}) \quad (43)$$

where Y_p^θ and Y_p^h are defined by Eqs. 40 and 41 above. Since NIPIP requires a time-domain representation of the control strategy equation, it is necessary to express Eqs. 42 and 43 as finite difference equations. This can be done by using the transfer function of the pilot, Y_p , and a table of z-transforms (e.g., Table 1). For the Y_p^θ and Y_p^h given in Eqs. 40 and 41,

$$Y_p^\theta = K_\theta \xrightarrow{\text{z-transform}} b_0 \quad (44)$$

continuous discrete

$$Y_p^h = \frac{K_h}{s} \xrightarrow{\text{z-transform}} \frac{b_1 T z^{-1}}{1 - z^{-1}} \quad (45)$$

continuous discrete

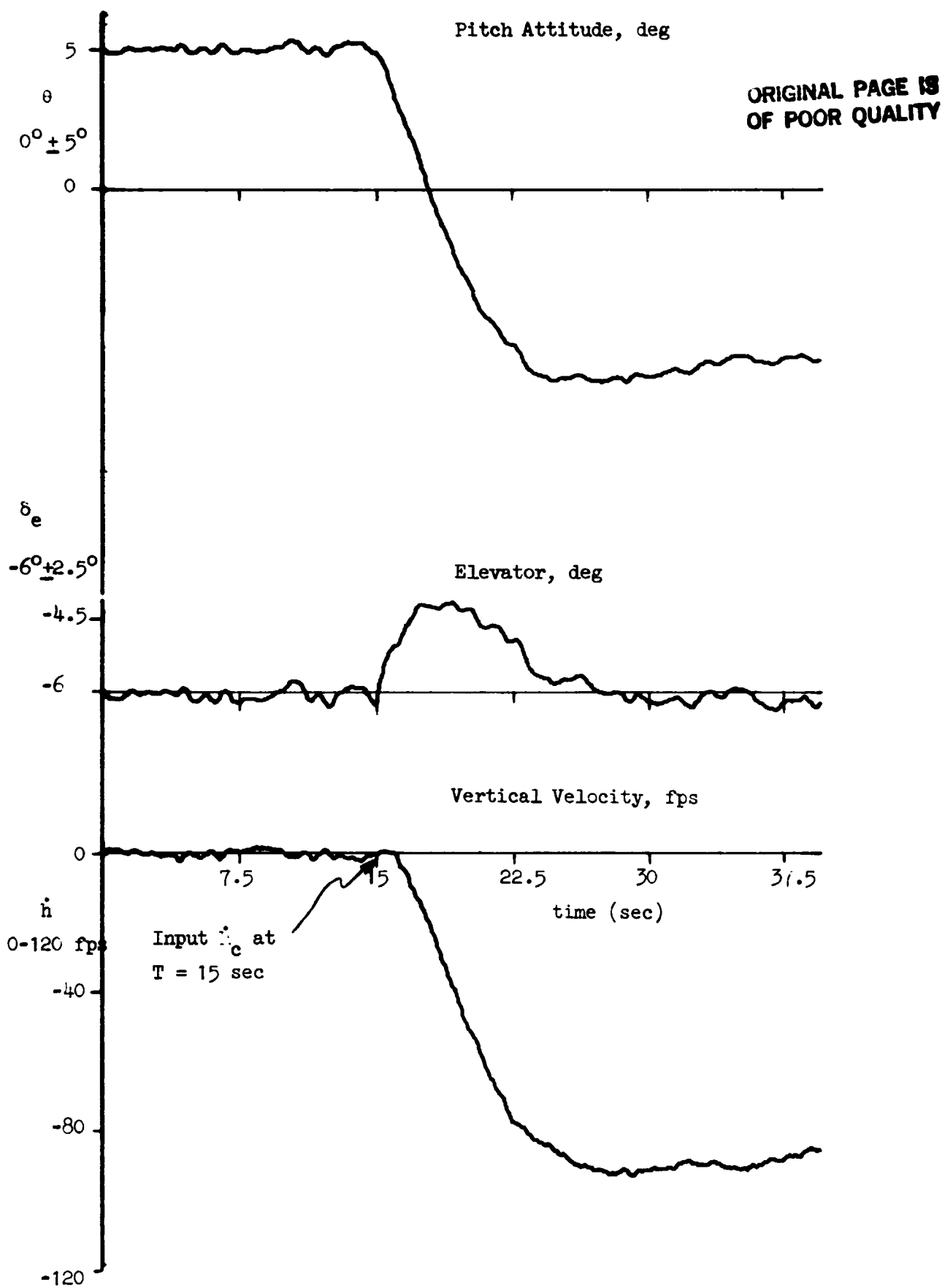


Figure 8. Time Histories Used for Example 2.
 (Closed-Loop Response to \dot{h} step Command of -87.5 fps)

**ORIGINAL PAGE IS
OF POOR QUALITY**

Substituting Eq. 44 into Eq. 42 and Eq. 45 into Eq. 53 the discrete domain representation of the control laws are given by

$$\delta_e = K_\theta (\theta_c - \theta) \quad (46)$$

$$\theta_c = \frac{K_h^* T z^{-1}}{1 - z^{-1}} (\dot{h}_c - \dot{h}) \quad (47)$$

where $b_0 = K_\theta$, $b_1 = K_h^*$, and T is the frame time.

In the case of our simplified simulation, θ_c can be obtained; however, in the realm of actual flight tests or pilot-in-the-loop simulations, θ_c is a command within the pilot's perception and is not obtainable. For this reason we will assume no direct θ_c information and relate θ_c to θ and \dot{h} .

θ_c can be eliminated by combining equations 46 and 47,

$$\delta_e = -K_\theta \theta + \frac{K_\theta K_h^* T z^{-1}}{1 - z^{-1}} (\dot{h}_c - \dot{h}) \quad (48)$$

multiplying both sides of Eq. 48 by $(1 - z^{-1})$ gives

$$(1 - z^{-1}) \delta_e = -K_\theta \theta (1 - z^{-1}) + K_\theta K_h^* T z^{-1} (\dot{h}_c - \dot{h}) \quad (49)$$

Using the notation derived in Section II, Eq. 49 is rewritten:

$$\begin{aligned} \delta_{e_n} - \delta_{e_{n-1}} &= -K_\theta (\theta_n - \theta_{n-1}) - K_\theta K_h^* T (\dot{h}_{n-1} - \dot{h}_c) + B \quad (50) \\ &= (\theta_n - \theta_{n-1}) \hat{c}_1 - (\dot{h}_{n-1} - \dot{h}_c) \hat{c}_2 + B \\ &\equiv \underline{F}_n \hat{c} \end{aligned}$$

**ORIGINAL PAGE IS
OF POOR QUALITY**

where the bias term, "B", has been added for the reasons presented in Section II.

From Eq. 50, it is possible to derive the information necessary to define the NIPIP inputs to the NAS program.

b. Reference Values. Using Eq. 50, the reference or bias values, BV (refer to Section III.C, pp. 31-34), can be thought of in a number of ways. The first way is to specify BV as the trim elevator, δ_e , and pitch attitude, θ , values and vertical velocity, \dot{h} , equal to \dot{h}_c . Using this approach the reference values would be subtracted in the NIPIP technique as follows:

$$(\delta_{e_n} - \delta_{e_{n-1}}) - \delta_{e_{trim}} = -K_\theta \{(\theta_n - \theta_{n-1}) - \theta_{trim}\} + K_\theta K_h^T (\dot{h}_{n-1} - \dot{h}_c) \quad (51)$$

Comparing Eq. 50 with Eq. 51, the bias DOF, B, would account for the trim values of δ_e and θ such that the B term of Eq. 50 would be equal to

$$B = \delta_{e_{trim}} + K_\theta \theta_{trim} \quad (52)$$

A second way is to note that any non-zero biases in δ_e or θ would be eliminated when using the differences shown in Eq. 50; thus BV could be specified as zero for δ_e and θ but still setting BV equal to \dot{h}_c for \dot{h} . This specification of BV would allow the bias DOF to be eliminated as Eq. 52 would be equal to zero.

A third way is to specify zero biases for all three variables δ_e , θ , and \dot{h} . In this case the B term of Eq. 50 would be equal to

$$B = K_\theta K_h^T \dot{h}_c \quad (53)$$

ORIGINAL PAGE IS
OF POOR QUALITY

In this example, the first approach for specifying BV will be used and the bias DOF retained in the estimation equation.

c. Frame Time and Time Window. The minimum frame time and time window are calculated based upon the inner and outer bandwidths of the system. Using Eq. 34 and 35

$$TDT_{IL} = \frac{\pi}{10(3)} = 0.104 \text{ sec}; \quad TDT_{OL} = \frac{\pi}{10(0.2)} = 1.57$$

$$TW_{IL} = \frac{10\pi}{3} = 10.5 \text{ sec}; \quad TW_{OL} = \frac{10\pi}{0.2} = 157 \text{ sec}$$

where the subscripts "IL" and "OL" refer to the inner and outer loops, respectively. To simultaneously identify both the inner- and outer-control loop elements, a frame time of 0.1 sec was used since it will identify both control loops. A time window equal to the length of the time history was chosen since it was found that a time window of 157 sec was not required for the identified solution to converge.

In general the frame time and time window lengths specified by Eqs. 34 and 35 are useful as starting points when making a first pass at identifying the control loops, with variations about these values permitted as deemed necessary.

d. Frequency Response. The frequency response of the inner loop, Y_p^θ , is determined by specifying the transfer function

$$Y_p^\theta = \frac{\delta_e}{\theta_e} = \hat{c}_1 = K_\theta \quad (54)$$

For $\omega_{c_{IL}} = 3.0$ rad/sec a frequency range of $\omega = 0.5$ to 8 rad/sec is specified. The outer-loop transfer function, Y_p^h , can be found by taking the ratio

$$\dot{Y}_p^h = \frac{\dot{\theta}_c}{h_e} = \frac{\theta_c}{\delta_e} \frac{\delta_e}{\dot{n}_e} = \frac{\hat{c}_2}{c_1} = K_h^* T \quad (35)$$

For $\omega_{c_{OL}} = 0.2$ rad/sec a frequency range of $\omega = 0.02$ to 0.8 rad/sec is specified.

e. User-Defined Subroutines. The subroutine PREPR, shown in Table 6, is used to compute the differences, $\theta_n - \theta_{n-1}$ and $\delta_{e_n} - \delta_{e_{n-1}}$. PREPR defines these differences as Elements 2 and 3, respectively, in the X array. \dot{h} is defined as Element 5 in the X array. (For a definition of the X array, refer to Section III.C, pp. 31-34.)

The forms of the control loop elements Y_p^θ and Y_p^h are specified using the subroutine MODE, shown in Table 7. The following section describes how subroutine MODE was defined.

3. Definition of the NAS Variables

In specifying the MODE subroutine, only those variables which are not defaulted or change from one control loop structure to another need to be defined. In the following subsections, the inner loop control element, Y_p^θ , is referred to as "control-loop 1" and the outer loop control element, Y_p^h is referred to as "control-loop 2."

A partial list of the input file to be used by NAS is shown in Table 8.

a. NIPIP Variables (Refer to Section III.C, pp. 31-34). From Table 8 and the description of subroutine PREPR we can infer the X-vector,

$$\underline{X} = \{t, \theta_n - \theta_{n-1}, \delta_{e_n} - \delta_{e_{n-1}}, w_g, \dot{h}\}$$

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 6

LISTING OF SUBROUTINE PREPR

```
C      5.7  SUBROUTINE PREPR
C
C      THIS IS AN EXAMPLE SUBROUTINE SHOWING HOW THE INPUT DATA
C      MAY BE PREPROCESSED PRIOR TO USE IN THE NIPIF TECHNIQUE
C
C      THIS SUBROUTINE CALCULATES THE DIFFERENCES BETWEEN THE CURRENT
C      AND FIRST PAST VALUES OF THE SECOND AND THIRD ELEMENTS OF THE
C      DI ARRAY AND OUTPUTS THEM AS THE SECOND AND THIRD ELEMENTS OF
C      THE X ARRAY, ALL OTHER ELEMENTS OF THE DI ARRAY ARE TRANSFERRED
C      TO THE X ARRAY WITH NO ADDITIONAL PROCESSING
C
C      SYSTEMS TECHNOLOGY, INC.    MAY 1982    GDH
C
C      =====
C
C      DI...INPUT DATA ARRAY FROM DATA FILE, TAPE, OR FLIGHT TEST
C      X...OUTPUT DATA ARRAY AFTER PROCESSING
C      IM...SPECIFIES WHICH CONTROL LOOP IS BEING PROCESSED
C      M...NUMBER OF VARIABLES PER RECORD ON THE INPUT DATA FILE
C
C      =====
C
C      SUBROUTINE PREPR(DI,X,IM,M)
C
C      DIMENSION DI(1),X(1)
C
C      GO TO 5 ON THE FIRST PASS TO PICK UP THE PAST VALUES OF THE
C      SECOND AND THIRD ELEMENTS OF THE DI ARRAY
C
C      USE IM=99 TO GO TO 5 ON THE FIRST PASS
C      99 IS USED SINCE IM WILL NOT BE EQUAL TO 99 DURING FUTURE
C      PROGRAM OPERATION.
C      FOR IM<=2 COMPUTE DIFFERENCES
C
C      IF (IM .EQ. 99)GO TO 5
C
C      TRANSFER DATA FROM DI TO X ARRAY
C
C      DO 1 I=1,M
C      X(I)=DI(I)
C 1  CONTINUE
```


ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 6 (Concluded)

```
C
C
C   COMPUTE DIFFERENCES
C
C   X(2)=X(2)-THL
C   X(3)=X(3)-DEL
C
C   GO TO RETURN IF IM<2 SINCE THE SAME DIFFERENCES ARE TO BE USED
C   IN CONTROL LOOP STRUCTURES 1 AND 2
C   PICK UP NEW PAST VALUES WHEN IM=2
C
C   IF(IM .LT. 2) GO TO 99
C
C   5   THL=DI(2)
C       DEL=DI(3)
C
C   99  RETURN
C       END
```

TABLE 7

LISTING OF SUBROUTINE MODE

```
C      5.6 SUBROUTINE MODE
C
C      THIS IS AN EXAMPLE OF A USER DEFINED SUBROUTINE FOR SPECIFYING
C      THE NWS VARIABLES WHEN SEVERAL CONTROL LOOP STRUCTURES ARE TO
C      BE IMPLEMENTED SIMULTANEOUSLY.
C
C      TO ACCESS THIS SUBROUTINE THE CONTROL FLAGS MOD AND NPS MUST BE
C      SET EQUAL TO "1" AND "2"
C      RESPECTIVELY, USING THE NAMELIST DIRECT INPUT.
C
C      THE CONTROL LOOP STRUCTURES FOR EXAMPLE 2 ARE SPECIFIED AS
C      FOLLOWS:
C      THE INNER LOOP IS SPECIFIED WHEN IM=1
C      THE OUTER LOOP IS SPECIFIED WHEN IM=2
C
C      ONLY THOSE VALUES WHICH ARE NOT DEFAULTED OR WHICH CHANGE FROM
C      CONTROL LOOP STRUCTURE 1 TO CONTROL LOOP STRUCTURE 2 ARE SPECIFIED.
C
C      NWS VARIABLES ARE SPECIFIED USING THE NAMES IN THE COMMON BLOCKS
C
C      SYSTEMS TECHNOLOGY, INC.      MAY 1982      GDH
C
C      =====
C      IM...SPECIFIES WHICH CONTROL LOOP STRUCTURE IS BEING USED
C      =====
C
C      SUBROUTINE MODE(IM)
C
C      DIMENSION NVEC(10),INDEX(5),NBI(5),NAI(5),BV(5,2)
C
C      COMMON /TIME/  TST,TEND,TDT,TOUT,TC,TW,
1          WMIN,WMAX,WINC,HFG,TOL,RLD,BV
C
C      COMMON /BAS/  IN,IOUT,MOD,NPS,INA,ISA,ISKIP,INFW,
1          NC,IBIAS,NVAR,INDEX,NVEC,
2          NB,NA,KTAU,NBI,NAI,IFCPT
C
C
C      GO TO (1,2),IM
C
C      NWS VARIABLES FOR CONTROL LOOP 1 (INNER LOOP)
C
```

TABLE 7 (Concluded)

1 CONTINUE
NC=3
IBIAS=1
NVAR=3
INDEX(1)=3
INDEX(2)=2
INDEX(3)=E
NVEC(1)=01
NVEC(2)=12
BV(1,1)=-.10681
BV(2,1)=.0889
BV(3,1)=-87.5
C
TST=15.
TDT=.1
TEND=30.
TOUT=3.
TC=3.
C
WMIN=.5
WMAX=8.
WINC=2.
C
NB=1
NA=1
NBI(1)=1
NAI(1)=0
C
MOD=1
NPS=2
INA=7
ISA=2
C
C
GO TO 99
C
C
NAS VARIABLES FOR CONTROL LOOP 2 (OUTER LOOP)
C
2 CONTINUE
BV(1,2)=-.10681
BV(2,2)=.0889
BV(3,2)=-87.5
WMIN=.02
WMAX=.8
WINC=1.5
C
NBI(1)=2
NAI(1)=1
C
C
99 RETURN
C
END
A>

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 8

INPUT FILE USED FOR EXAMPLE B
(File name is STEP)

5				
TIME SEC	_____			t
THEY RAD	_____	θ		
ELEV RAD	_____			δ_e
WG FPS	_____	w_g		
HGT FPS	_____	g		\dot{h}
DISCRETE STEP -87.5 FPS				
+.000000E+00	+.8890780E-01	-.1068150E+00	+.0000000E+00	+.0000000E+00
+.1000000E+00	+.8870710E-01	-.1068150E+00	+.2006810E-03	+.1945720E+00
+.2000000E+00	+.8818220E-01	-.1071350E+00	+.7158820E-03	+.5958630E+00
+.3000000E+00	+.8755490E-01	-.1077650E+00	+.1313370E-02	+.8468250E+00
+.4000000E+00	+.8732380E-01	-.1089440E+00	+.1502140E-02	+.6301350E+00
+.5000000E+00	+.8753650E-01	-.1092790E+00	+.1257160E-02	+.2335550E+00
+.6000000E+00	+.8753170E-01	-.109030E+00	+.1251080E-02	+.4543760E+00
+.7000000E+00	+.8739910E-01	-.1088730E+00	+.1369970E-02	+.6995320E+00
+.8000000E+00	+.8742340E-01	-.1090360E+00	+.1301680E-02	+.6021840E+00
+.9000000E+00	+.874350E-01	-.1089450E+00	+.1259370E-02	+.6722370E+00
+.1000000E+01	+.8763130E-01	-.1038690E+00	+.1030010E-02	+.5538370E+00
	t	θ	δ_e	w_g
	(sec)	(rad)	(rad)	(fps)
			\dot{h}	(fps)

**ORIGINAL PAGE IS
OF POOR QUALITY**

From Eq. 50 and X for control loops 1 and 2,

NC	=	3	Number of degrees of freedom
IBIAS	=	1	Bias degrees of freedom
NVAR	=	3	Number of independent parameters in estimation equation
INDEX	=	3, 2, 5	NVAR specific elements in <u>X</u> used in estimation equation, $\Delta\theta$, $\Delta\delta_e$, h
NVEC	=	01,12	How independent parameters are entered in the F-vector
BV	=	-.10681, 0.0889, -87.5	Reference (trim) value of the parameters

b. Time History Variables (Refer to Section III.B, pp. 29-31).

For control loops 1 and 2,

TST	=	15.
TEND	=	30.
TOUT	=	3.0
TC	=	3.0

c. Frequency Response Variables (Refer to Section III.D, pp. 34-37).

Control-Loop 1

WMIN	=	0.5
WMAX	=	8.0
WINC	=	2.0

Control-Loop 2

WMIN	=	0.02
WMAX	=	0.8
WINC	=	1.5

**ORIGINAL PAGE IS
OF POOR QUALITY**

Control-Loop 1

NB = 1
NA = 1 See Eqs. 48
NBI = {1} and 52
NAI = {0}

Control-Loop 2

NB = 1
NA = 1 See Eq. 48
NBI = {2} and 53
NAI = {1}

d. Control Flags (refer to Section III.A, pp. 26-29).

For control loops 1 and 2,

MOD = 1 } These two are input
NPS = 2 } using direct mode
INA = 3
ISA = 2

e. Description of Subroutines PREPR and MODE. Tables 6 and 7 contain annotated listings of the user-defined subroutines PREPR and MODE.

4. Description of the NAS Outputs

Table 9 contains an annotated output of the NAS program. Items typed by the user are underlined.

5. Discussion of the Results

For this case, it was not necessary to compute the frequency response to obtain the gains \hat{c}_1 and K_h^* . They could have just as easily been determined directly from the \hat{c} 's presented in Table 9. That is, From Eq. 48 it is recognized that \hat{c}_1 and \hat{c}_2 should equal K_θ and $K_\theta K_h^* T$, respectively; thus

TABLE 9. NAS OUTPUT FOR EXAMPLE B

ORIGINAL PAGE IS
OF POOR QUALITY

nas

INPUT FILE STEP _____ Filename

1 TIME SEC
2 THET RAD
3 ELEV RAD
4 MG FPS
5 HDOT FPS

Note: User inputs
are underlined

FILE ID : DISCRETE STEP -87.5 NO. VAR. 5

DIRECT INPUT USING NAMELIST

VAR. NO. 20
MOD = 0 CHANGE TO 1 _____ Use Subroutine MODE
MOD = 1
VAR. NO. 21
NPS = 1 CHANGE TO 2 _____ Two control loop structures
NPS = 2
VAR. NO. -1

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 1 — Inner loop
0 ELEV(N-0) _____ $y = \delta e_n$

THE F-VECTOR (INDEPENDENT VARIABLE):

1 THET(N-0)
2 HDOT(N-1) _____ $\underline{F} = [\Delta\theta_n, \Delta\dot{h}_{n-1}, B]$
3 BIAS DOF

TRANSFER FUNCTION FOR CONTROL LOOP 1

NUM : THET(N-0) Z##- 0
DEN : 1.0 $\frac{\delta e}{\theta_e}(z) = \hat{c}_1 \equiv \hat{y}_p$

VERIFY VARIABLES TO BE SAVED

ELEV WITH TRIM VALUE OF -.1068!
THET WITH TRIM VALUE OF .08890
HDOT WITH TRIM VALUE OF -87.50000

TW.TDT.TC.TOUT = .150E+02 .100E+00 .300E+01 .300E+01

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 2 — Outer loop

0 ELEV(N-0)

THE F-VECTOR (INDEPENDENT VARIABLE):

1 THET(N-0)
2 HDOT(N-1)
3 BIAS DOF
Same as for
Control Loop 1

TRANSFER FUNCTION FOR CONTROL LOOP 2

NUM : HDOT(N-1) Z##- 0
DEN : -THET(N-0) $\frac{\theta_c}{\theta_e}(z) = \frac{\hat{c}_2 z^{-1}}{\hat{c}_1} \equiv \hat{y}_p^i$

TABLE 9 (Continued)

VERIFY VARIABLES TO BE SAVED
ELEV WITH TRIM VALUE OF -0.10681
THET WITH TRIM VALUE OF 0.08890
WDOT WITH TRIM VALUE OF -87.50000

TW.TDT.TC.TOUT = .150E+02 .100E+00 .300E+01 .300E+01

NOTE: Program will not return to Namelist when MOD > 0

T	X(1)	Y2B	DPTH	R2	N	CONTROL LOOP
18.4	18.3	.116E-01	.738E+00	.100E+01	30	1
CHAT= .16725E+01 .88847E-04 .25502E+00						

FREQ / AMP / PHASE

FREQ	AMP	PHASE	ω_i (rad/sec)	
.50	1.00	2.00	4.00	8.00
4.47	4.47	4.47	4.47	4.47
0.00	0.00	0.00	0.00	0.00

Control Loop 1 (inner loop)

18.4	18.3	.116E-01	.738E+00	.100E+01	30	2
CHAT= .16725E+01 .88847E-04 .25502E+00						

FREQ / AMP / PHASE

FREQ	AMP	PHASE	ω_i (rad/sec)						
.02	.03	.04	.07	.10	.15	.23	.34	.51	.77
-85.49	-85.49	-85.49	-85.49	-85.49	-85.49	-85.49	-85.49	-85.49	-85.49
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Control Loop 2 (outer loop)

T	X(1)	Y2B	DPTH	R2	N	CONTROL LOOP
21.4	21.3	.116E-01	.440E+02	.100E+01	60	1
CHAT= .16757E+01 .85176E-04 .25560E+00						

Repeats above in TOUT intervals

FREQ / AMP / PHASE

FREQ	AMP	PHASE		
.50	1.00	2.00	4.00	8.00
4.48	4.48	4.48	4.48	4.48
0.00	0.00	0.00	0.00	0.00

21.4	21.3	.116E-01	.440E+02	.100E+01	60	2
CHAT= .16757E+01 .85176E-04 .25560E+00						

FREQ / AMP / PHASE

FREQ	AMP	PHASE							
.02	.03	.04	.07	.10	.15	.23	.34	.51	.77
-85.88	-85.88	-85.88	-85.88	-85.88	-85.88	-85.88	-85.88	-95.88	-85.88
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 9 (Concluded)

{	T	X(1)	Y2B	DTH	R2	N	CONTROL LOOP			
	24.4	24.3	.114E-01	.360E+03	.100E+01	90	1			
	CHAT=		.16768E+01	.83948E-04	.25579E+00					
FREQ / AMF / PHASE										
	.50	1.00	2.00	4.00	8.00					
	4.49	4.49	4.49	4.49	4.49					
	0.00	0.00	0.00	0.00	0.00					
	24.4	24.3	.114E-01	.360E+03	.100E+01	90	2			
	CHAT=		.16768E+01	.83948E-04	.25579E+00					
FREQ / AMF / PHASE										
	.02	.03	.04	.07	.10	.15	.23	.34	.51	.77
	-86.01	-86.01	-86.01	-86.01	-86.01	-86.01	-86.01	-86.01	-86.01	-86.01
	.00	0.00	0.00	0.00	-0.00	0.00	0.00	-0.00	0.00	0.00
	T	X(1)	Y2B	DTH	R2	N	CONTROL LOOP			
	27.4	27.3	.114E-01	.114E+04	.100E+01	120	1			
	CHAT=		.16820E+01	.83587E-04	.25629E+00					
FREQ / AMF / PHASE										
	.50	1.00	2.00	4.00	8.00					
	4.52	4.52	4.52	4.52	4.52					
	0.00	0.00	0.00	0.00	0.00					
	27.4	27.3	.114E-01	.114E+04	.100E+01	120	2			
	CHAT=		.16820E+01	.83587E-04	.25629E+00					
FREQ / AMF / PHASE										
	.02	.03	.04	.07	.10	.15	.23	.34	.51	.77
	-86.07	-86.07	-86.07	-86.07	-86.07	-86.07	-86.07	-86.07	-86.07	-86.07
	.00	0.00	0.00	0.00	0.00	0.00	.00	-0.00	0.00	0.00

- NOTES: 1) That solution for both control loops was essentially invariant throughout run
- 2) Program will loop back to the same point as in Example A; however, in this case no further computations were made.

**ORIGINAL PAGE IS
OF POOR QUALITY**

these gains could be determined by direct inspection of the \hat{c} coefficients. However, usually it is not possible to directly relate the coefficients, \hat{c} , to the pilot control parameters. In these cases, a frequency response from which specific piloting parameters (e.g., gain and phase) can be deduced will aid in recognizing the identified pilot control elements.

Table 10 gives the exact and 3 DOF estimated pilot gains K_θ and K_h^* . These gains were obtained from the frequency response amplitude ratios and \hat{c} coefficients of control loops 1 and 2 listed in Table 9. The frequency response for Y_p^h (control loop 2) can only identify the gain K_h^*T as the knowledge that Y_p^h is an integrator was used in deriving the 3 DOF estimation equation. To account for the sample frame time, T , which appears in Eq. 50, 20 dB was added to the amplitude ratio of control loop 2 to obtain K_h^* . Note that the 3 DOF estimated and exact pilot gains are in perfect agreement. Also presented in Table 10 is the value of the correlation coefficient, R^2 , which is equal to 1.

The use of a frequency response to deduce the pilot control parameters θ is further shown using 5 DOF to determine the pilot control elements Y_p and Y_p^h . 5 DOF were obtained by not computing the differences $\delta_{e_n} - \delta_{e_{n-1}}$ and $\theta_n - \theta_{n-1}$ prior to inputting them into NIP, but rather $\delta_{e_{n-1}}$ and θ_{n-1} are added as additional DOF. The estimation equation became

$$\delta_{e_n} = \hat{c}_1 \delta_{e_{n-1}} + \hat{c}_2 \theta_n + \hat{c}_3 \theta_{n-1} + \hat{c}_4 \dot{h}_{n-1} + B \quad (56)$$

where \hat{c} should equal

$$\hat{c} = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ \hat{c}_4 \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ K_\theta \\ -K_\theta \\ K_\theta K_h^* T \\ B \end{bmatrix} \quad (57)$$

TABLE 10

EXACT AND ESTIMATED PILOT GAINS USING THREE DEGREES OF FREEDOM

	K_{θ} (rad/rad)		K_h^* (rad/ft)		R^2
	Exact	Estimated*	Exact	Estimated*	
Frequency Response Amplitude Ratio	1.675	1.675	0.0005	0.0005	1
\hat{c} Coefficients	1.675	1.675	0.0005	0.0005	1

*Using estimate at $t = 21.4$ sec in Table 9.

TABLE 11

EXACT AND ESTIMATED PILOT GAINS USING FIVE DEGREES OF FREEDOM

	K_{θ} (rad/rad)		K_h^* (rad/ft)	
	Exact	Estimated*	Exact	Estimated*
Frequency Response Amplitude Ratio	1.675	1.66	0.0005	0.00041
\hat{c} Coefficients	1.675	1.64	0.0005	0.00043

*Using estimate at $t = 21.4$ sec, NAs output not shown.

**ORIGINAL PAGE IS
OF POOR QUALITY**

Values of \hat{c} listed in Eq. 57 were determined by inspection of Eq. 50.

In this case the a priori knowledge of Y_p^θ and Y_p^h is not used explicitly in deriving the estimation equation. The assumed form of Eq. 56 has enough DOF to identify Y_p^θ and Y_p^h as entries 1, 2, 3, and 6 of Table 1.

Figure 9 presents the exact and 5 DOF estimated pilot describing functions of Y_p^h . As before the frequency response identified Y_p^θ as a pure gain of magnitude essentially equal to that for 3 DOF. The frequency response of Y_p^h on the other hand now shows that Y_p^h is in fact an integrator. Although the phase angle of Y_p^h at the lower frequencies does not exactly match that of an integrator, this is due to an insufficient time window length. (Refer to the Section II discussion on time window length, pp. 19-20.) The phase angle at the higher frequencies does indeed match that of an integrator. Further increases in the number of DOF do not change the identified solutions for Y_p^θ and Y_p^h , and the same effects of increasing the number of DOF as in Example 1 were noted.

The coefficients, \hat{c} , for 5 DOF case again can be directly related to the pilot gains K_θ and k_h using Eq. 57. Table 11 gives the gains K_θ and K_h for the 5 DOF case as determined from the \hat{c} vector.

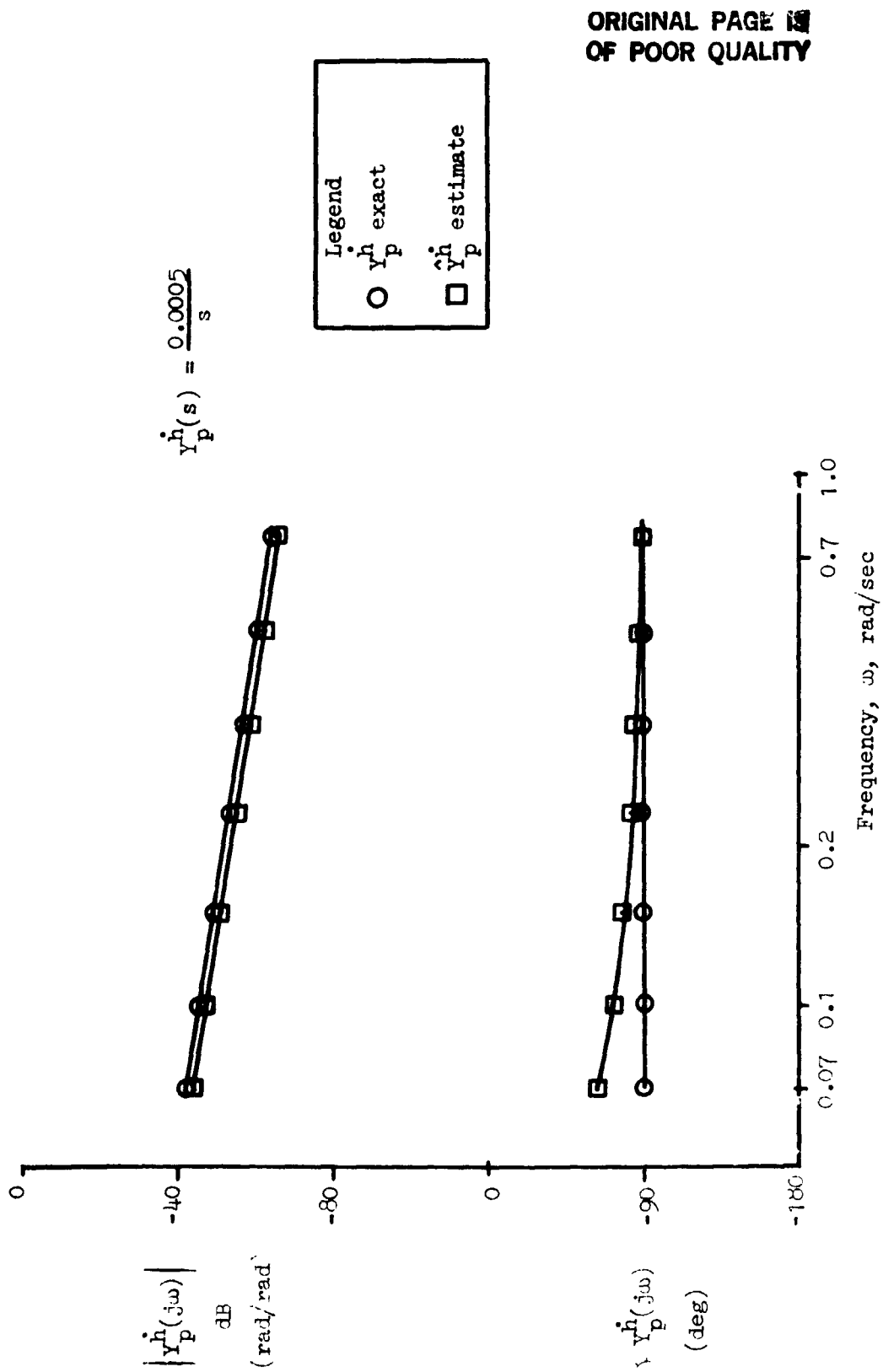
C. CLOSED-LOOP PILOT-VEHICLE EXAMPLE

1. Statement of the Problem and Objectives

The previous examples were concerned with identifying the pilot control elements and ignored the characteristics of the aircraft. This example combines the aircraft characteristics with those of the pilot, and the overall pilot-aircraft characteristics are identified for the landing maneuver (flare) of a DC-10 aircraft.

The pilot-aircraft characteristics for the landing flare are assumed to be described by a second-order model with a characteristic equation of the form

$$\ddot{h} + 2\zeta\omega\dot{h} + \omega^2h = 0 \quad (58)$$



ORIGINAL PAGE
OF POOR QUALITY

Figure 9. Frequency Response of Exact and Estimated \dot{Y}_p^h for five Degrees of Freedom

**ORIGINAL PAGE IS
OF POOR QUALITY**

where h is altitude and $(\dot{})$ refers to time differentiation. It is further assumed that this characteristic equation describing the flare corresponds to the response from an initial altitude, h , and sink rate, \dot{h} , with respect to the terminal conditions. Thus the flare is regarded as an unforced response from a set of initial conditions to a set of desired conditions at touchdown.

The pilot-aircraft natural frequency, ω , and damping, ζ , of Eq. 58 were identified from actual flight data of a DC-10 landing maneuver. A detailed discussion of the DC-10 flare and pilot-aircraft model is found in Ref. 14.

2. Consideration of NIPI Parameters

a. Specifying the F-Vector. Using Eq. 58, the estimated ω and ζ are determined by shifting the terms containing ω and ζ to the right-hand side. Thus the estimation equation is given by:

$$\begin{aligned}\ddot{h}_n &= -2\zeta\omega\dot{h}_n - \omega^2 h_n + B \\ &= \hat{c}_1 \dot{h}_n + \hat{c}_2 h_n + B \\ &\equiv \underline{\hat{F}}_n \hat{c} \end{aligned} \tag{59}$$

The "B" term in Eq. 59 represents a bias DOF and is included for the reasons presented in Section II.

From Eq. 59 it is seen that the pilot-aircraft parameters ω and ζ can be determined directly from the estimated coefficients, \hat{c} , that is,

$$\omega = \sqrt{-\hat{c}_2} \tag{60}$$

and

$$\zeta = \frac{-\hat{c}_1}{2\omega} \quad (61)$$

b. Frame Time and Time Window. The frame time and time window were determined by the sample period and length of the landing flare time history giving a sample period of 0.05 sec and a time window of 6.55 sec. This is one case where the rules of thumb are not followed when specifying the frame time and time window but using what is available.

c. Frequency Response. The pilot-aircraft characteristics are determined from Eqs. 60 and 61; thus a frequency response is not required and is therefore omitted.

d. User-Defined Subroutines. The default for subroutines MODE and PREPR are used, and the NAS variables are input using the "direct input mode."

3. Definition of the NAS Variables

Table 12 contains a partial listing of the input file to be used. The "direct input mode" was used to define those variables which are not defaulted.

a. NIPIP Variables. (Refer to Section III.C, pp. 31-34.) From Table 12 can be inferred as

$$\underline{X} = \{t, h, \dot{h}, \ddot{h}\}$$

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 12

INPUT FILE USED FOR EXAMPLE C
(File name is FLARE)

```

4
TIME SEC _____ t
H   FT _____ h
HDDT FPS _____ ḣ
HDDTFPS2 _____ ḧ
DC-10 FLARE
+.5200E+01 +.1005E+03 -.1453E+02 -.1737E+01
+.5250E+01 +.1003E+03 -.1408E+02 -.2059E+01
+.5300E+01 +.9872E+02 -.1553E+02 -.1673E+01
+.5350E+01 +.9761E+02 -.1553E+02 -.1995E+01
+.5400E+01 +.9682E+02 -.1566E+02 -.2156E+01
+.5450E+01 +.9619E+02 -.1564E+02 -.2091E+01
+.5500E+01 +.9476E+02 -.1641E+02 -.2349E+01
+.5550E+01 +.9413E+02 -.1635E+02 -.2156E+01
+.5600E+01 +.9366E+02 -.1612E+02 -.1963E+01
+.5650E+01 +.9287E+02 -.1626E+02 -.2670E+01
+.5700E+01 +.9160E+02 -.1685E+02 -.2542E+01
+.5750E+01 +.9049E+02 -.1724E+02 -.2059E+01
+.5800E+01 +.8954E+02 -.1746E+02 -.2284E+01
+.5850E+01 +.8954E+02 -.1673E+02 -.2509E+01
+.5900E+01 +.8812E+02 -.1746E+02 -.2509E+01
+.5950E+01 +.8670E+02 -.1814E+02 -.2316E+01
+.6000E+01 +.8654E+02 -.1753E+02 -.2316E+01
+.6050E+01 +.8622E+02 -.1709E+02 -.1963E+01
+.6100E+01 +.8543E+02 -.1715E+02 -.2091E+01
+.6150E+01 +.8464E+02 -.1719E+02 -.1769E+01
A>
      t      h      ḣ      ḧ
      (sec) (ft) (fps) (fps2)

```


ORIGINAL PAGE IS
OF POOR QUALITY

From Eq. 59 and X

NC = 3
IBIAS = 1
NVAR = 3
INDEX = 4, 3, 2
NVEC = 1,2

b. Time History Variables (Refer to Section III.B, pp. 29-31)

TDT = 0.05
TEND = 6.55
TOUT = 1.
TC = 1.

c. Frequency Response. This section is omitted, because it is unnecessary.

d. Control Flags. Use default settings.

4. Description of the NAS Outputs

Table 13 contains an annotated output of the NAS program. Items typed by the user are underlined.

5. Discussion of the Results

From Table 13 it is seen that the first three estimates of $\hat{c}(1)$ are positive in sign; thus computing ζ using Eq. 61 yields a negative (unstable) damping value. However, the last three estimates of $\hat{c}(1)$ are

**ORIGINAL PAGE IS
OF POOR QUALITY**

TABLE 13. NAS Output for Example C
NAS — execute the NAS program

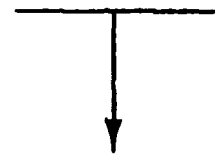
INPUT FILE " FLARE — Filename

- 1 TIME SEC
- 2 H FT
- 3 HDT FPS
- 4 HDT FPS2

FILE ID : DC-10 FLARE NO. VAR. 4

DIRECT INPUT USING NAMELIST

VAR. NO. 2
 TEND = 100.00 CHANGE TO 6.55
 TEND = 6.550
 VAR. NO. 3
 TDT = .1000E+00 CHANGE TO .05
 TDT = .5000E-01
 VAR. NO. 4
 TOUT = 5.000 CHANGE TO 1.
 TOUT = 1.0000
 VAR. NO. 5
 TC = 5.000 CHANGE TO 1.
 TC = 1.0000
 VAR. NO. 26
 NC = 0 CHANGE TO 3
 NC = 3
 VAR. NO. 27
 IBIAS = 0 CHANGE TO 1
 IBIAS = 1
 VAR. NO. 28
 NVAR = 0 CHANGE TO 3
 NVAR = 3
 VAR. NO. 29
 INDEX1 = 0 CHANGE TO 4
 INDEX1 = 4
 VAR. NO. 30
 INDEX2 = 0 CHANGE TO 3
 INDEX2 = 3
 VAR. NO. 31
 INDEX3 = 0 CHANGE TO 2
 INDEX3 = 2
 VAR. NO. 34
 NVEC1 = 0 CHANGE TO 01
 NVEC1 = 1
 VAR. NO. 35
 NVEC2 = 0 CHANGE TO 02
 NVEC2 = 2
 VAR. NO. -1



Use namelist to set
 NAS variables

Use negative subscript to
 exit Namelist

C-2

TABLE 13 (Concluded)

ORIGINAL PAGE IS
OF POOR QUALITY

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 1

0 HODT(N-0) $y = \ddot{h}_n$
THE F-VECTOR (INDEPENDENT VARIABLE):

1 HODT(N-0)
2 H (N-0)
3 BIAS DOF

} $F = [h_n, \dot{h}_n, B]$

VERIFY VARIABLES TO BE SAVED

HODT WITH TRIM VALUE OF 0.00000
HODT WITH TRIM VALUE OF 0.00000
H WITH TRIM VALUE OF 0.00000

TW, TDT, TC, TOUT = .655E+01 .500E-01 .100E+01 .100E+01

END VERIFICATION...

TYPE "0" TO CONTINUE; "-1" TO RETURN TO NAMELIST? 0

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
1.1	6.2	.465E+01	.249E+05	.986E+00	20	1

CHAT= .41154E+00 -.66509E-01 .10778E+02

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
2.1	7.2	.335E+01	.107E+07	.980E+00	40	1

CHAT= .38206E+00 -.67883E-01 .10402E+02

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
3.1	8.2	.229E+01	.150E+08	.969E+00	60	1

CHAT= .36911E+00 -.72305E-01 .10575E+02

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
4.1	9.2	.175E+01	.912E+08	.897E+00	80	1

CHAT= -.48387E-01 -.52732E 01 .18824E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
5.1	10.2	.162E+01	.347E+09	.890E+00	100	1

CHAT= -.68103E-01 -.52916E-01 .15655E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
6.1	11.2	.144E+01	.118E+10	.891E+00	120	1

CHAT= -.21262E+00 -.54116E-01 -.77539E+00 — Use this estimate of \hat{c} for computing ω and ξ

INPUT FILE ? — Program loops back to this point. No further runs made.

**ORIGINAL PAGE IS
OF POOR QUALITY**

negative in sign which yields a positive (stable) damping value. The estimate of $\hat{c}(2)$ on the other hand gives a real value for the natural frequency for all time.

Using $\hat{c}(1)$ and $\hat{c}(2)$ at $T = 6.1$, the estimated natural frequency and damping are 0.23 rad/sec and 0.46, respectively.

The correlation coefficient, R^2 , at $T = 6.1$ is equal to 0.86 which initially leads one to the conclusion that the estimated solution does not match the actual data. To determine the degree to which the estimation solution does or does not match the actual data, the initial condition response of Eq. 58 with ω and ζ equal to their estimated values was computed. The initial conditions from the actual time history of $h = 100.46$ ft and $\dot{h} = -14.53$ fps were used in computing the response.

Figure 10 presents the h to \dot{h} phase plane trajectories of the initial condition response of Eq. 58 with the estimated ω and ζ as well as the actual data. From Fig. 10 it appears that the estimated solution does fit the actual data with a reasonable degree of accuracy. Thus this low value of R^2 does not appear to be due to an improper modeling approach, but rather due to the noise which is present in the input data.

D. DECELERATION TO HOVER

1. Statement of the Problem and Objectives

The next example is based on the research presented in Ref. 15. The theoretical background and derivation of the control laws are omitted in the following text. The purpose here is to demonstrate how the NIPIP technique can be used to identify the parameters of nonlinear equations.

ORIGINAL PAGE 13
OF POOR QUALITY

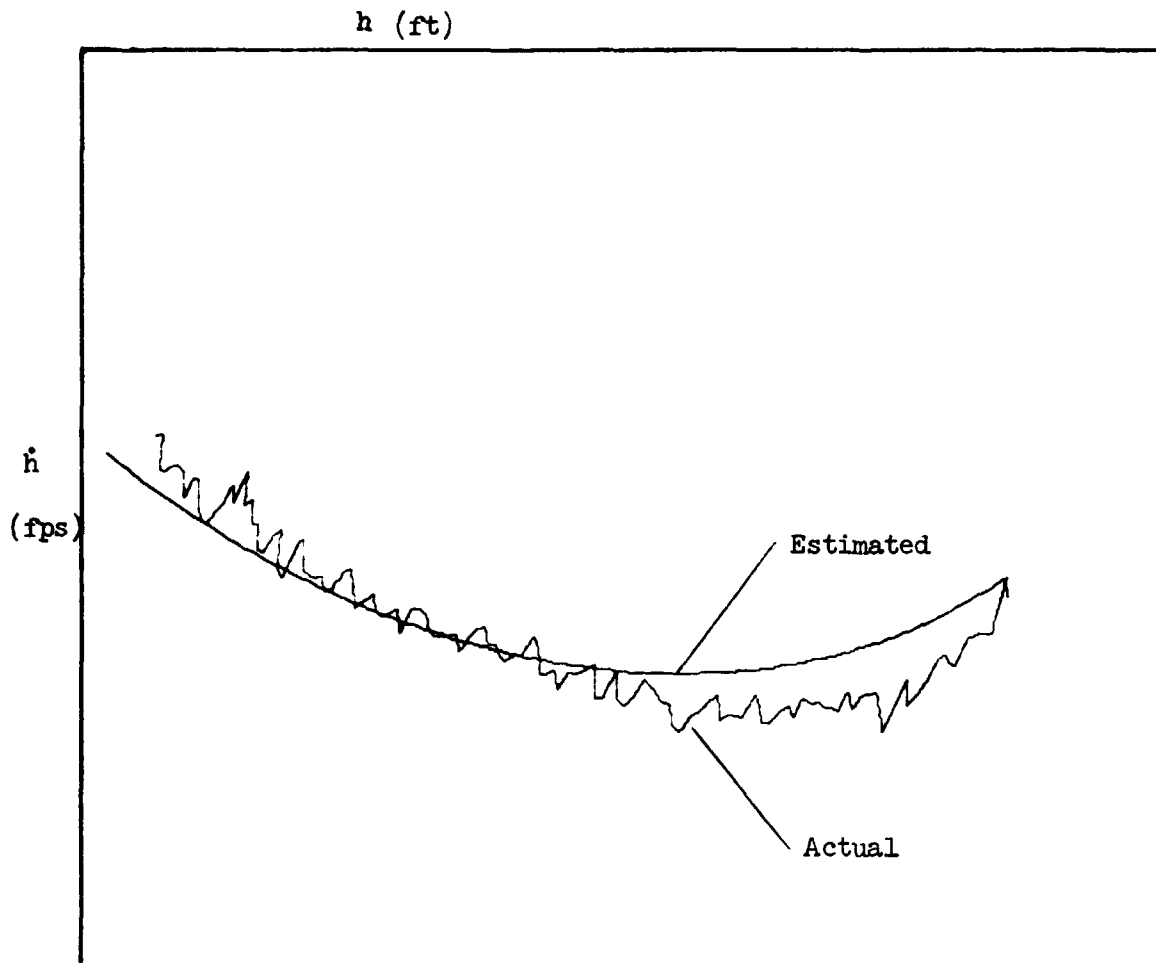


Figure 10. Phase Plane Trajectories of Landing Flare

**ORIGINAL PAGE IS
OF POOR QUALITY**

The block diagram shown in Fig. 11 summarizes the control task. Reference 15 derives the following relationship between deceleration, \ddot{R} , and range, R,

$$\ddot{R} = \frac{k^2 R}{(1 + R/A)^3} \quad (62)$$

The objective of NIPIP is to identify the parameters k and A based on a deceleration profile of \ddot{R} and R.

Equation 62 shows that k and A are nonlinearly related to \ddot{R} and R. Their relationship can be made linear by algebraic manipulation of Eq. 62,

$$(\ddot{R}/R)^{1/3} = c + dR \quad (63)$$

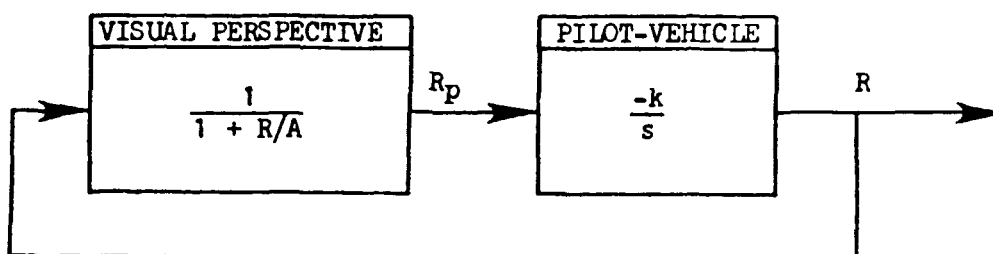
It is easily shown that c and d from Eq. 63 are related to k and A in Eq. 62 as follows:

$$k = (c)^{-3/2} \quad (64)$$

$$A = c/d \quad (65)$$

The method for using NIPIP is now clear: Identify c and d in Eq. 63 by using the pre-processor subroutine to calculate $(\ddot{R}/R)^{1/3}$, then calculate k and A using Eqs. 64 and 65. The details of how to set up the NIPIP input are presented below.

ORIGINAL PAGE IS
OF POOR QUALITY



where

- k = The effective gain for the combined pilot-vehicle system
- R = True range
- R_p = Perceived range
- A = A measure of the effective size of the object being viewed
- \dot{R} = Range rate
= $-kR/(1 + R/A)^3$
- \ddot{R} = Range deceleration
= $k^2R/(1 + R/A)^3$

Figure 11. Block Diagram for Deceleration to Hover Example

2. Consideration of NIPIP Parameters

a. Specifying the F-Vector. The F-vector is taken directly from Eq. 63,

$$y \equiv (\ddot{R}/R)^{1/3} = \{R, 1\} \begin{Bmatrix} d \\ c \end{Bmatrix} \quad (66)$$
$$\equiv \underline{F} \hat{c}$$

Thus only 2 DOF are required with one being a bias DOF (i.e., the c term is a bias DOF).

b. Frame Time and Time Window. For this example the data is in the form of ordered pairs of \ddot{R} and R. Time information was not recorded; thus the "frame time" is in terms of the incremental number shown in Table 14.. All data will be processed using the infinite time window option.

c. Frequency Response. No frequency response calculations are required, because k and A will be calculated from Eqs. 64 and 65.

3. Definition of the NAS Variables

a. NIPIP Variables (Refer to Section III.C, pp. 31-34) The X-vector can be inferred from the partial list of the input data file shown in Table 14,

$$\underline{X} = \{n, R, \ddot{R}, NU\}$$

ORIGINAL PAGE 19
OF POOR QUALITY

TABLE 14. PARTIAL LIST OF THE INPUT FILE
USED FOR EXAMPLE D

```

4
N - _____ n
R FT _____ R
RDD FPS2 _____ R̄
NU _____ not used
LANDING PROFILE
+.1000E+01 +.2800E+04 +.8372E+00 +.1495E+02
+.2000E+01 +.2760E+04 +.8533E+00 +.1479E+02
+.3000E+01 +.2720E+04 +.8694E+00 +.1463E+02
+.4000E+01 +.2690E+04 +.1191E+01 +.1312E+02
+.5000E+01 +.2650E+04 +.1208E+01 +.1300E+02
+.6000E+01 +.2600E+04 +.1191E+01 +.1297E+02
+.7000E+01 +.2570E+04 +.9016E+00 +.1418E+02
+.8000E+01 +.2530E+04 +.8694E+00 +.1428E+02
+.9000E+01 +.2480E+04 +.1288E+01 +.1244E+02
+.1000E+02 +.2450E+04 +.9016E+00 +.1395E+02
+.1100E+02 +.2410E+04 +.9982E+00 +.1342E+02
+.1200E+02 +.2360E+04 +.1256E+01 +.1234E+02
+.1300E+02 +.2320E+04 +.1127E+01 +.1272E+02
+.1400E+02 +.2290E+04 +.1352E+01 +.1192E+02
⋮
-1.0 0.0 0.0 0.0 _____ End-of-file marker

```

ORIGINAL PAGE IS
OF POOR QUALITY

The variable "n" is the incremental number used to number the data points. It is not time, but note that it is a monotonically increasing variable, as required by the NAS program. The variable "NU" is not used for this example.

From Eq. 63 and X,

NC = 2
IBIAS = 1
NVAR = 2
INDEX = 3, 2
NVEC = 01

b. Time History Variables. (Refer to Section III.B, pp. 29-31)

TDT = 1.0

With the time history variables set as shown above, NAS will process every data point, compute, and output \hat{c} every fifth data point.

c. Frequency Response Variables. (Refer to Section III.B, pp. 29-31) The frequency response option will not be used for this example.

d. Control Flags. (Refer to Section III.A, pp. 26-29) Default values for all of the control flags will be used.

e. User-Defined Subroutines. (Refer to Section III.F, pp. 38-40) The pre-processing subroutine, PREPR, will be used to calculate the dependent variable in Eq. 66 (i.e., the "y" term). A listing of the resulting PREPR subroutine is contained in Table 15.

LISTING OF THE PREPR SUBROUTINE USED FOR EXAMPLE D

```

C      5.6 SUBROUTINE PREPR
C
C      THIS IS A USER DEFINED SUBROUTINE FOR PREPROCESSING THE INPUT
C      DATA TO NIPJF
C
C      THIS IS THE DEFAULT LISTING FOR TRANSFERING DATA FROM THE DI
C      ARRAY TO THE X ARRAY
C
C      SYSTEMS TECHNOLOGY, INC.    MAY 1982    GDH
C
C      SUBROUTINE PREPR(DI,X,IM,M)
C
C      =====
C
C      DI...INPUT DATA ARRAY FROM DATA FILE, TAPE, OR FLIGHT TEST
C      X ...OUTPUT DATA ARRAY AFTER PROCESSING
C      IM...SPECIFIES WHICH CONTROL LOOP IS BEING PROCESSED
C      M ...NUMBER OF VARIABLE S PER RECORD ON THE INPUT DATA FILE
C
C      =====
C
C
C      DIMENSION DI(1),X(1)
C
C
C      TRANSFER DATA FROM DI TO X ARRAY
C
C      DO 1 I=1,M
C      X(I)=DI(I)
C 1  CONTINUE
C      R=X(2)
C      RDD=X(3)
C      X(3)=(R/RDD)**(1./3.)
C
C
C
C 99  RETURN
C      END
C

```

**ORIGINAL PAGE 19
OF POOR QUALITY**

4. Description of the NAS Outputs

Table 16 contains an annotated output of the NAS program. Items typed by the user are underlined. The values of k and A are calculated for a few values of the \hat{c} . Recall that k and A are related to $\hat{c}_1 = d$ and \hat{c}_2 via Eqs. 64 and 65.

5. Discussion of the Results

Plots of the actual data and the NIPIP curve fit are presented in Fig. 12. The curve was constructed by using Eq. 62 and the indicated values of k and A . Note that the fit is best from $R = 2800$ ft down to about $R = 300$ ft. This is because there are few data points for $R < 300$ ft. The fit in the $R < 300$ ft region could be improved by not including the data points for $R > 300$ ft or using the sliding time window approach (i.e., use a time window that would give a $\Delta R \approx 300$ to 500 ft).

TABLE 16. NAS OUTPUT FOR EXAMPLE D

A>B:EGD Execute the customized version of NAS for this example (i.e.,
EGD has the special version of the PREPR
subroutine in it).

INPUT FILE ? LAND

- 1 N -
- 2 R FT
- 3 RDD FPS2
- 4 NU

ORIGINAL PAGE IS
OF POOR QUALITY

FILE ID : LANDING PROFILE NO. VAR. 4

DIRECT INPUT USING NAMELIST

VAR. NO.? 26
 NC = 0 CHANGE TO 2
 NC = 2
 VAR. NO.? 27
 IBIAS = 0 CHANGE TO 1
 IBIAS = 1
 VAR. NO.? 28
 NVAR = 0 CHANGE TO 2
 NVAR = 2
 VAR. NO.? 29
 INDEX1 = 0 CHANGE TO 3
 INDEX1 = 3
 VAR. NO.? 30
 INDEX2 = 0 CHANGE TO 2
 INDEX2 = 2
 VAR. NO.? 34
 NVEC1 = 0 CHANGE TO 01
 NVEC1 = 1
 VAR. NO.? -1

Y (DEPENDENT VARIABLE), FOR CONTROL LOOP 1
 0 RDD (N=0) $y = (R/K)_n^{1/3}$

THE F-VECTOR (INDEPENDENT VARIABLE):
 1 R (N=0) $F = [R_n, B]$
 2 BIAS DOF

VERIFY VARIABLES TO BE SAVED
 RDD WITH TRIM VALUE OF 0.00000
 R WITH TRIM VALUE OF 0.00000

TW,TDI,TC,TOUT = .100E+03 .100E+01 .500E+01 .500E+01

END VERIFICATION...
 TYPE "0" TO CONTINUE; "-1" TO RETURN TO NAMELIST? 0

ORIGINAL PAGE #
OF POOR QUALITY

TABLE 16 (Continued)

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
7.0	7.0	.185E+03	.766E+05	.998E+00	5	1

CHAT= .24318E-02 .71437E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
12.0	12.0	.181E+03	.132E+07	.997E+00	10	1

CHAT= .24008E-02 .73195E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
17.0	17.0	.173E+03	.664E+07	.998E+00	15	1

CHAT= .29171E-02 .59754E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
22.0	22.0	.160E+03	.210E+08	.997E+00	20	1

CHAT= .42334E-02 .26491E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
27.0	27.0	.151E+03	.501E+08	.997E+00	25	1

CHAT= .39968E-02 .32303E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
32.0	32.0	.142E+03	.105E+09	.997E+00	30	1

CHAT= .40653E-02 .30704E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
37.0	37.0	.133E+03	.195E+09	.997E+00	35	1

CHAT= .40428E-02 .31226E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
42.0	42.0	.125E+03	.337E+09	.998E+00	40	1

CHAT= .40359E-02 .31376E+01

T	X(1)	Y2B	DHTH	R2	N	CONTROL LOOP
47.0	47.0	.117E+03	.541E+09	.998E+00	45	1

CHAT= .40502E-02 .31073E+01

d = 0.002432
c = 7.144
k = 0.0524
A = 293⁰.

d = 0.004050
c = 3.107
k = 0.1826
A = 767.

**ORIGINAL PAGE IS
OF POOR QUALITY**

TABLE 16 (Concluded)

T	X(1)	Y2B	DNTH	R2	N	CONTROL LOOP
52.0	52.0	.110E+03	.827E+09	.998E+00	50	1

CNAT= .40248E-02 .31613E+01

T	X(1)	Y2B	DNTH	R2	N	CONTROL LOOP
57.0	57.0	.103E+03	.122E+10	.998E+00	55	1

CNAT= .40442E-02 .31218E+01

T	X(1)	Y2B	DNTH	R2	N	CONTROL LOOP
62.0	62.0	.962E+02	.173E+10	.998E+00	60	1

CNAT= .40541E-02 .31024E+01

T	X(1)	Y2B	DNTH	R2	N	CONTROL LOOP
67.0	67.0	.899E+02	.240E+10	.998E+00	65	1

CNAT= .41209E-02 .29752E+01

INPUT FILE :

}

d = 0.004121
c = 2.975
k = 0.1949
A = 722.

ORIGINAL PAGE IS
OF POOR QUALITY

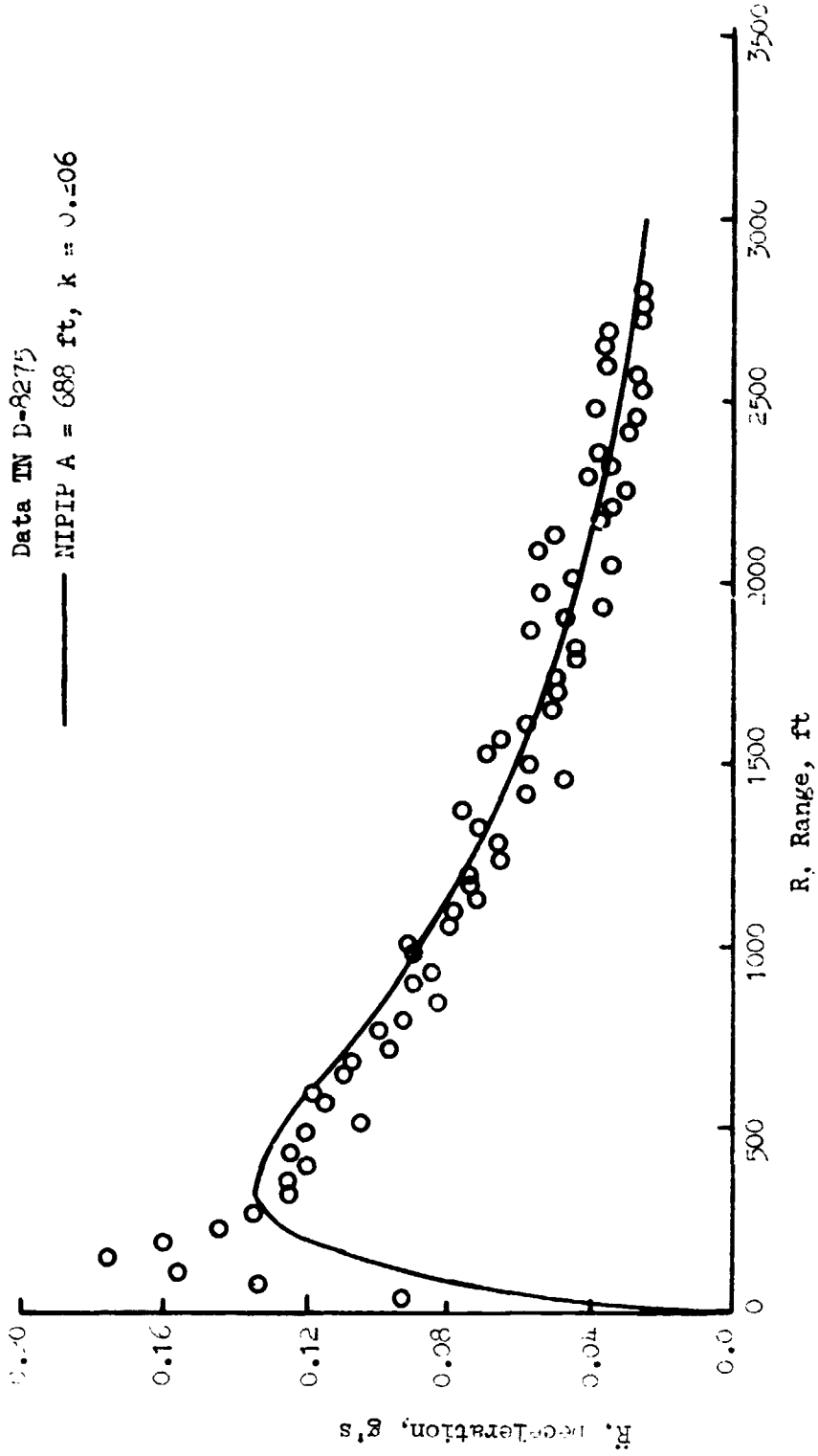


Figure 12. Actual Data and NIPIP Results for Example D

REFERENCES

1. Wingrove, R. C., Comparison of Methods for Identifying Pilot Describing Functions From Closed-Loop Operating Records, NASA TN D-6235, March 1971.
2. Heffley, R. K., and W. F. Jewell, Development of a CTOL Piloting Technique Measurement Scheme for a Real-Time Simulator Environment, NASA CR-152294, July 1979.
3. Jewell, W. F., and T. M. Schulman, A Pilot Control Strategy Identification Technique for Use in Multiloop Control Tasks, NASA CR-152374, August 1980.
4. Gelb, A., Applied Optimal Estimation, The M.I.T. Press, Cambridge, 1974, pp. 23-24.
5. McRuer, D. T., D. Graham, E. S. Krendel, and W. Reisener, Jr., Human Pilot Dynamics in Compensatory Systems--Theory, Models and Experiments with Controlled Element and Forcing Function Variations, AFFDL-TR-65-15, January 1965.
6. McRuer, D. T., and E. S. Krendel, Mathematical Models of Human Pilot Behavior, AGARDograph No. 188, January 1974.
7. Sheridan, Thomas B., and William R. Ferrell, Man-Machine Systems: Information, Control, and Decision Models of Human Performance, The M.I.T. Press, Cambridge, Mass., 1974.
8. McRuer, D. T., and Ezra Krendel, Dynamic Response of Human Operators, WADC-TR-56-524, October 1957.
9. Stapleford, Robert L., Samuel J. Craig, and John A. Tennant, Measurement of Pilot Describing Function in Single-Controller Multiloop Tasks, NASA CR-1238, January 1969.
10. Stapleford, R. L., D. T. McRuer, and R. E. Magdaleno, Pilot Describing Function Measurements in a Multiloop Task, NASA CR-542, August 1966.
11. Ragazzini, J. R., and G. F. Franklin, Sampled-Data Control Systems, McGraw-Hill, Inc., New York, 1958, pp. 314-320.
12. Jury, E. I., Sampled-Data Control Systems, Wiley, New York, 1958, pp. 56-63.

THIS PAGE IS BLANK AND NOT FILMED

13. Jewell, Wayne F., Data Collection Requirements, a Pilot Control Strategy Measurement Technique, and Verification Procedures for the Phase III HUD Program, Systems Technology, Inc., Technical Report No. 1153-1, February 1980.
14. Heffley, Robert K., Ted M. Schulman, Robert J. Randle, Jr., and Warren F. Clement, An Analysis of Airline Landing Flare Data Based on Flight and Training Simulator Measurements, Systems Technology, Inc., Technical Report No. 1172-1R, May 1982.
15. Heffley, Robert K. "A Model for Manual Decelerating Approaches to Hover," Proceedings of the Fifteenth Annual Conference on Manual Control, AFFDL-TR-79-3134, March 1979, pp. 545-554.

GENERIC F-8 DIGITAL SIMULATION

In Examples A and B of Section IV, the equations of motion for a generic F-8 aircraft with known "pilot" control laws were used to generate the time histories. The time histories were then used as the input to the NIPIP identification scheme. These equations of motion and known control laws were simulated on a digital computer. The equations of motion were the same for both of these examples; however, the "pilot" control laws were different in the two examples. The pilot control laws which were used are defined in Section IV.

The trim (i.e., straight and level) flight condition was specified at an altitude of 7000 ft, with an airspeed of 465 fps, giving a trim elevator control position of -6.1 deg and a trim pitch attitude of 5.1 deg. The airspeed was held constant at 465 fps throughout the time history by eliminating the axial acceleration in the equations of motion. The stability derivatives of this generic F-8 aircraft are listed in Table A-1. The bare airframe short period natural frequency and damping for this flight condition are 3.48 rad/sec and 0.6, respectively.

In this digital simulation the aircraft was assumed to be flying through atmospheric turbulence which was modeled by passing a uniform distribution through a first-order filter. That is,

$$\frac{w_g}{\eta} = \frac{\sigma_w \sqrt{L_w/U_0}}{s + 2U_0/L_w}$$

where w_g is the vertical turbulence and η is a random number with a uniform distribution. The scale length, L_w , was set at a constant 1000 ft, and the rms level, σ_w , was 3.0 fps.

ORIGINAL PAGE 19
OF POOR QUALITY

TABLE A-1
LONGITUDINAL STABILITY AND CONTROL
DERIVATIVES FOR THE F-8 AIRCRAFT

$X_u = 0$	$Z_u = -0.0578$ (rad/sec)	$M_u = 0.002$ (1/sec-ft)
$X_w = 0$	$Z_w = -0.996$ (rad/sec)	$M_w = -0.0204$ (1/sec-ft)
$X_q = 0$	$Z_q = 0$	$M_q = -3.187$ (1/sec)
$X_{\delta_e} = 0$	$Z_{\delta_e} = -63.325$ (fps ² /rad)	$M_{\delta_e} = -7.451$ (rad/sec ² /rad)

TRIM CONDITION:

$U_o = 465$ fps
$\gamma_o = 0.0$ deg
$\theta_o = 5.1$ deg
$\delta_{e_o} = -6.1$ deg

APPENDIX B

OPERATING PROCEDURES

This appendix presents a description of the disk and file operations necessary to run the NAS software. This description will be presented for two computer systems. The first is concerned with operations on an Apple II-Plus micro-computer and the second is for operation on the Dryden Cyber computer.

**A. OPERATION ON AN APPLE II-PLUS
MICRO-COMPUTER**

A prerequisite for using the NAS software on an Apple micro-computer is that the Apple must have 64K of random access memory (RAM) and must have the Microsoft Softcard CP/M operating system and Microsoft FORTRAN-80. This last requirement is solely due to the fact that this is the operating system and FORTRAN language used to write this particular version of the NAS software. However, Microsoft FORTRAN-80 is fairly standard FORTRAN, hence with minor modifications it may be possible to use this software on any computer with a standard FORTRAN language capability.

For additional information about the use of the CP/M text editor, FORTRAN-80 compiler and linker, or other operating system features, refer to Refs. B-1 through B-3.

1. The NAS Program Diskette

The NAS programs reside on one diskette. The files on the NAS diskette are:

a. FORTRAN sources files (.FOR files)

NSAP.FOR--Main line program

NIPIP.FOR--Subroutines

FUNC.FOR--Functions

MODE.FOR

PREPR.FOR

} Default versions of the user definable subroutines

b. Relocatable Object Files (.REL Files)

NSAP.REL

NIPIP.REL

FUNC.REL

MODE.REL

PREPR.REL

c. Executable Object Files (.COM Files)

NAS.COM

d. Submit Files (.SUB Files)

FL80.SUB

The FORTRAN source files (.FOR) consist of programs as typed in by the user. (Listings of these programs, in the form of microfiche, can be found in the envelope inside the back cover of this report.) The relocatable object files (.REL) are machine readable binary files produced by the FORTRAN-80 compiler from the FORTRAN source programs. The executable object file NAS.COM is the absolute version of the NAS program that is properly loaded and ready to be executed. This file is the only file which is necessary for program operation; however, the .REL and .FOR files are required to arrive at this executable version.

**ORIGINAL PAGE 19
OF POOR QUALITY**

FL80.SUB is a program which defines and executes macro instructions. FL80.SUB compiles and links the NAS software on a multi-drive system, i.e., performs the steps necessary to convert the source files into the executable version of the software.

2. Making a Backup Diskette

Always work with a backup copy of the NAS system master and store the original for safe keeping. Thus there will always be a default version of the software available. The procedure for making a backup is outlined in Ref. B-3. The following description relies on the fact that the backup copy is being used, as some files need to be erased when compiling and linking edited versions.

3. Sample Session for a Single-Drive System

a. Organization of Diskettes. Due to lack of disk space on a single diskette, the NAS programs and the FORTRAN-80 software will not fit on a single diskette. Thus it is necessary to use two diskettes. One contains a copy of the NAS programs listed above as well as the CP/M text editor. The other contains the FORTRAN-80 software. For convenience a third diskette contains the input data file.

<u>NAS Disk</u>	<u>FORTRAN-80 Disk</u>
.FOR Files	F80.COM
.REL Files	L80.COM
.COM Files	FORLIB.REL
.SUB Files	
ED.COM	

ORIGINAL PAGE IS
OF POOR QUALITY

b. Editing the Source Programs. The NAS program contains two user definable subroutines, MODE and PREPR, and at some point it may be necessary to change these programs from their default versions.

It should be noted; although it is possible to edit any of the source programs, it is not recommended or required to edit any source programs other than MODE or PREPR for program operation. The source files are edited using the CP/M editor, ED*, by typing

ED FN.FOR

where FN is the file name of the source file to be edited, i.e., MODE or PREPR.

After the source file, FN, has been edited, exit the editor by typing

E

c. Compiling and Linking Edited Source Programs. The next step is to compile and link the edited program. This procedure is explained and outlined in the FORTRAN-80 user's guide, Ref. B-1. The commands and the procedure are summarized below:

Place the FORTRAN-80 disk into the disk drive and type

F80

When the asterick prompt appears, place the copied NAS disk into the drive and type

= FN

*Refer to Ref. B-3 for text editor commands.

ORIGINAL PAGE IS
OF POOR QUALITY

where FN refers to the name of the source file to be compiled, i.e., PREPR or MODE. This creates the .REL object code with the same name. Then type control-c and enter*

ERA NAS.COM

Place the FORTRAN-80 disk into the drive, type control-c, and enter

L80

When the asterisk appears, insert the NAS disk and enter

NSAP,NIPIP,FUNC,MODE,PREPR,FILENAME/N

where filename specifies the name to be given to the executable object file. Note if future editing is to be done on this diskette and a filename other than the one previously used is specified, it is necessary to erase the earlier version of the executable object code as was done when NAS.COM was erased. This is due to disk space limitations. Thus substitute

ERA FILENAME.COM

for ERA NAS.COM in this procedure and specify the new file name in the line after the L80 command.

When finished, the linker will return an asterisk. Do not worry about the list of "undefined globals." Put the FORTRAN-80 disk back in the drive and enter

*First make sure this is a back-up copy of the master diskette, because this command will erase the default executable NAS program. If it is not a backup, first create a backup copy of the master diskette. See Ref. B-3 for details on how to do this.

FORLIB/S

This command searches the FORTRAN library and links the program with the required subroutines. When the asterisk prompt appears, insert the copied NAS disk into the drive and type

/E

This will save the machine-executable file Filename.COM on the disk.

- d. Run the Program. After the program has been compiled and linked, it is executed by typing

FILENAME

When the Apple responds

INPUT FILE?

and the red light goes out on the disk drive, remove the copied NAS disk and place the diskette containing the input data file into the disk drive. Then enter the filename of the input data file and the program execution continues as shown in the examples of Section IV.

This ends the sample session for a single-drive system. The next section presents a similar discussion for multi-drive systems.

4. Sample Session on a Multi-Drive System

- a. Organization of Diskettes. Due to disk space limitations, two diskettes are required for the FORTRAN-80 programs and the NAS programs. One contains a copy of the NAS programs. The other contains the FORTRAN-80 software, the CP/M text editor, and the file SUBMIT.COM. A third diskette contains the input data file.

ORIGINAL PAGE IS
OF POOR QUALITY

<u>NAS Disk</u>	<u>FORTRAN-80 Disk</u>
(Drive A:)	(Drive B:)
.FOR Files	F80.COM
.REL Files	L80.COM
.COM Files	FORLIB.REC
.SUB Files	ED.COM
	SUBMIT.COM

b. Editing the Source Programs. The NAS program contains two user definable subroutines, MODE and PREPR, and at some point it may be necessary to change these programs from their default versions. It should be noted, although it is possible to edit any of the source programs, it is not recommended or required to edit any source program other than MODE or PREPR for program operation. The source programs are edited using the CP/M text editor, ED, by typing

B:ED FN.FOR

where FN is the file name of the source file, i.e., MODE or PREPR. After the source file FN has been edited, to exit the editor type

E

c. Compiling and Linking Edited Source Programs. The next step is to compile and link the edited program. This procedure uses the macro command file FL80.SUB. The macro commands are summarized below:

**ORIGINAL PAGE IS
OF POOR QUALITY**

Place the copied NAS disk into Drive A:, the FORTRAN-80 disk into Drive B:, type control-c, and type*

ERA NAS.COM

B:SUBMIT FL80 FILENAME

where FILENAME is the name to be given to the machine-executable object file. Note, if changing the NAS programs from their default version, it is recommended that some name other than NAS be used for FILENAME in the above command. Thus NAS.COM will always be the default version of the software.

The Apple will respond

ERA *.BAK -- Erases all .BAK files created by the CP/M text editor.

B:F80 = MODE -- Compiles the MODE.FOR program.

F:F80 - PREPR -- Compiles the PREPR.FOR program

B: -- go to Drive B:

L80 A:NSAP,A:NIPIP,A:FUNC,A:MODE,A:PREPR, A:FILENAME/N/E

Creates the machine-executable object file. This file--called FILENAME.COM--is saved on Drive A:

A: -- Return to Drive A:

*First, is this a back-up copy of the master diskette, as this command will erase the executable NAS program. If not, first create a back-up copy of the master diskette, see Ref. B-3 for details.

**ORIGINAL PAGE IS
OF POOR QUALITY**

Note that if future editing is to be done on this diskette and a filename other than the one previously used is specified in the submit command, it is necessary to erase that version of the executable object code as was done when NAS.COM was erased. This is due to disk space limitations. Thus, substitute

ERA FILENAME.COM

for ERA NAS.COM in this procedure and specify the new filename in the submit command.

d. Run the Program. Turn on printer if hard copy is desired. After the program has been compiled and linked, it is executed by placing the diskette with the new executable object code into Drive B: and the input data disk into Drive A:, typing control-c, and entering

B:FILENAME

When the program responds

INPUT FILE?

Enter the input data file name and program execution continues as shown in the examples of Section IV.

B. OPERATION ON THE DRYDEN CYBER COMPUTER

The NAS software resides on the Cyber in two forms: one is an interactive version and the other a batch version. The only difference between these two versions is that, in the batch version, all NAS variables must be defined prior to execution using subroutine MODE. The control flag MOD is inoperable in the batch version, and the program always calls subroutine MODE; whereas, in the interactive version, the user has the option of using the namelist or subroutine MODE for defining the NAS variables.

Both of the Cyber versions (interactive and batch) are written in standard FORTRAN hence should run on any computer with FORTRAN capability with minor modifications. FORTRAN-V is the Cyber standard FORTRAN (see Ref. B-4); however, the NAS software does not rely on any special capabilities of FORTRAN-V, hence the NAS software can be run using FORTRAN IV.

1. The NAS Program Files

The two NAS versions reside in two separate files. The source files on the Cyber are:

NASIAF — Interactive FORTRAN version

NASSUB — Batch version

These files are the FORTRAN source files, and they contain the mainline program, all function subprograms, subroutines, and the default user-defined subroutines, MODE and PREPR. Thus the NAS program is self-contained in the above files for both applications.

2. Making a Backup Source File

Always work with a backup copy of the NAS system master and store the original for safe keeping. Thus there will always be a default version of the software available. The procedure for making a backup is outlined in Ref. B-5. The following description relies on the fact that the backup copy is being used, as some sections may need to be edited when adapting the program to the problem at hand.

ORIGINAL PAGE IS
OF POOR QUALITY

3. Sample Session Using the Interactive Version

a. Calling the Interactive Version. The following commands assume that you have a copy of the NASIAF version on your disk. If not, obtain a suitable copy. After logging onto the Cyber (see Ref. B-5). The interactive FORTRAN (IAF) version is run using the FORTRAN compiler.

Enter the FORTRAN compiler by typing

FTNTS.

IAF responds with the following:

OLD, NEW, OR LIB FILE:

Enter OLD, NASIAF

IAF responds:

READY.

The ready response indicates that IAF is ready to accept another command.

b. Editing the Source Programs. The NASIAF program contains two user-definable subroutines, MODE and PREPR, and at some point it may be necessary to change these programs from their default versions.

It should be noted, although it is possible to edit any of the source programs, it is not recommended or required to edit any source programs

other than MODE or PREPR for program operation. The source file is edited using XEDIT* by typing

XEDIT, NASIAF

where NASIAF is the file name of the interactive source file to be edited. The user-defined subroutines are at the end (bottom) of the source file. They can be found using the locate command.

After the source file has been edited, exit the editor by typing

Q, FILENAME, RL

where FILENAME is the filename to be given to the edited source file. RL will replace the file and make the file a local file.

c. Attaching the Data Input File. The next step is to attach the data input file. The attach procedure is explained and outlined in the NOS Reference Manual, Ref. B-5. The commands and the procedure are summarized below:

The default input file is DATA; thus input data files are required to be attached under the name DATA using

ATTACH, DATA = INFILE

Where INFILE is the file name of the input data file to be attached.

d. Run the Program. After the input data file has been attached, the program is executed by typing

RUN

*Refer to Ref. B-6 for text editor commands.

When the Cyber responds

INPUT FILE?

Type the name of the input file (i.e., INFILE) This is used only for future reference when reviewing the program output; the program will use the file attached as DATA. Program execution continues as shown in the examples of Section IV.

This ends the sample session for the interactive version. The next section presents a similar discussion for the batch version.

4. Sample Session Using the Batch Version

a. Calling the Batch Version. The following commands assume that you have a copy of the NASSUB version on your disk. If not, obtain a suitable copy. The batch version is run using an interactive active terminal and submitting the program as a batch job. Thus all NAS variables must be defined prior to program execution using subroutine MODE. After logging onto the Cyber, the batch version is obtained by typing

FTNTS

IAF responds

OLD, NEW, OR LIB FILE:

Enter OLD, NASSUB

IAF responds

READY.

**ORIGINAL PAGE IS
OF POOR QUALITY**

The ready response indicates that IAF is ready to accept another command.

b. Editing the Source Programs. Since the batch version requires the NAS variables to be stored in subroutine MODE, it is always necessary to edit this subroutine to define the appropriate NAS variables. It may also be necessary to change subroutine PREPR from its default version. It should be noted, although it is possible to edit any of the source programs, it is not recommended or required to edit any source programs other than MODE or PREPR for program operation. The source file is edited using XEDIT* by typing

XEDIT,NASSUB

where NASSUB is the file name of the interactive source file to be edited. The user-defined subroutines are at the end (bottom) of the source file. They can be found using the locate command.

After the source file has been edited, exit the editor by typing

Q, FILENAME, RL

where FILENAME is the file name to be given to the edited source file. RL will replace the file and make the file a local file.

*Refer to Ref. B-6 for text editor commands.

c. Batch Job Control Cards. The batch version, NASSUB, contains job control cards. These cards are summarized below:

```
/JOB
JOB NAME.                Name to be given to Job
USER, USERNAME, PASSWORD.
CHARGE, NO1, NO1, UTILITY.  Charge numbers
SETTL, 2000.              Sets time limits
FTN5(ET=F,PL,L=0).        Use the FORTRAN V compiler
ATTACH, DATA = INFILE.    Attaches input data file
                           INFILE under name DATA
LGO.                      Load and execute
/EOR

NASSUB PROGRAM

/EOF
```

These cards must be adapted to a valid user when running the program.

d. Running the Batch Program. After the proper job control cards have been specified, the batch version is run by typing

```
SUBMIT, FILENAME, B
```

where FILENAME is the name given to the program when exiting the editor, and B specifies the batch mode. The default specification for the program output device is the line printer. If any other device is to be specified, the control flag IOUT needs to be changed as well as specifying the suitable device in the program card in the mainline program

This ends the sample session for the batch version.

**ORIGINAL PAGE IS
OF POOR QUALITY**

REFERENCES

- B-1 FORTTRAN-80 Users Guide, Microsoft Consumer Products, Bellevue, Washington, 1980.
- B-2 FORTTRAN-80 Reference Manual, Microsoft Consumer Products, Bellevue, Washington, 1980.
- B-3 Microsoft, Softcard, Vol. 1, Microsoft Consumer Products, Bellevue, Washington, 1980.
- B-4 FORTTRAN Version 5 Reference Manual, CDC Operating Systems: NOS 1, NOS/BE 1, SCOPE 2, Control Data Corporation, CDC 60481300, July 20, 1979.
- B-5 NOS Version 1 Reference Manual, Volume 1 of 2. CDC Computer Systems: Cyber 170 Series; Cyber 70, Models 71, 72, 73, 74; 6000 Series, Control Data Corporation, CDC 60435400, June 17, 1975.
- B-6 XEDIT Version 3 Reference Manual, CDC Operating System: NOS 1, Control Data Corporation, CDC 60455730, June 2, 1978.

APPENDIX C

SUMMARY OF NAS PROGRAM VARIABLES

Table C-1 contains summary descriptions of all of the NAS parameters. More complete descriptions of the parameters can be found in Section III of the main body of this report.

The list contained in Table C-1 will be very useful to the experienced NAS user for a number of reasons. First, the table can be used as a check list for specifying the NAS parameter values. Second, the "Variable No." column in the list can be used when inputting the values using the name-list. Finally, the list significantly speeds up the time required to use the NAS program.

TABLE C-1

SUMMARY OF NAS PROGRAM VARIABLES

Variable Name	Variable No.	Default Value	Frequently Changed?	Brief Description	Page No.
IN	18	10	No	Logical unit for input device	26
IOUT	19	6	No	Logical unit for output device	26
MOD	20	0	No	"Director" or program control for inputs of NAS variables	26
NPS	21	1	No	Number of control loops	26
INA	22	1	Yes	Verify F-vector	27
ISA	23	1	Yes	Compute frequency response	27
ISKIP	24	1	No	Debug option	28
INFW	25	1	Yes	Sliding time window ON (0), OFF (1)	28
TST	1	0.0	Yes	Start time	29
TEND	2	100.0	Yes	Stop time	29
TDT	3	0.1	Yes	Frame time	30
TOUT	4	5.0	Yes	Output time	30
TC	5	5.0	Yes	Compute time	30
TW	6	--	Yes	Time window	30
NC	26	--	Yes	Number of DOF	32
IBIAS	27	--	Yes	Bias DOF, ON (1), OFF (0)	32
NVAR	28	--	Yes	Number of independent variables	32
INDEX ₁	29	--	Yes	Indirect address of variables used in the estimation equations. Length is NVAR	32
INDEX ₂	30	--	Yes		32
INDEX ₃	31	--	Yes		32
INDEX ₄	32	--	Yes		32
INDEX ₅	33	--	Yes		32
NVEC ₁	34	--	Yes	Specify format of the F-vector. Length is NC if IBIAS = 0, NC-1 if IBIAS = 1.	33
NVEC ₂	35	--	Yes		33
NVEC ₃	36	--	Yes		33
NVEC ₄	37	--	Yes		33
NVEC ₅	38	--	Yes		33
NVEC ₆	39	--	Yes		33
NVEC ₇	40	--	Yes		33
NVEC ₈	41	--	Yes		33
NVEC ₉	42	--	Yes		33
NVEC ₁₀	43	--	Yes		33
BV ₁	13	--	Yes	Reference values of the variables in the estimation equation. Length is NVAR	34
BV ₂	14	--	Yes		34
BV ₃	15	--	Yes		34
BV ₄	16	--	Yes		34
BV ₅	17	--	Yes		34

**ORIGINAL PAGE IS
OF POOR QUALITY**

TABLE C-1 (Concluded)

Variable Name	Variable No.	Default Value	Frequently Changed?	Brief Description	Page No.
WMIN	7	0.1	No		35
WMAX	8	10.0	No		35
WINC	9	2.0	No		35
HFG	10	1.0	No		35
NB	44	--	No		35
NA	45	--	No		35
KTAU	46	--	No		35
NBI ₁	47	--	No	Parameters used to compute frequency response	35
NBI ₂	48	--	No		35
NBI ₃	49	--	No		35
NBI ₄	50	--	No		35
NBI ₅	51	--	No		35
NAI ₁	52	--	No		35
NAI ₂	53	--	No		35
NAI ₃	54	--	No		35
NAI ₄	55	--	No		35
NAI ₅	56	--	No		35
IFOPT	57	--	No		37
TOL	11	1000.0	No	Warning tolerances	37
RLD	2	0.0001	No	Warning tolerances	38

**ORIGINAL PAGE IS
OF POOR QUALITY**

Microfiche supplement for NASA CR-170398

**NON-INTRUSIVE PARAMETER IDENTIFICATION
PROCEDURE USER'S GUIDE**

by Gregory D. Hanson and Wayne F. Jewell