AD-A118 117 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 1/3
DIGITAL FLIGHT CONTROL SYSTEM DESIGN USING SINGULAR PERTURBATIO--ETC(U)
DEC 81 J S SMYTH
UNCLASSIFIED AFIT/EE/GE/81D-55 NL

1 of 3
AD
A118117

DDC

0

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

DIGITAL FLIGHT CONTROL SYSTEM DESIGN

USING SINGULAR PERTURBATION METHODS

THESIS

AFIT/EE/GE/81D-55

Joseph S. Smyth
2nd Lt USAF

Approved for public release· distribution unlimited

AFIT/EE/GE/81D-55

DIGITAL FLIGHT CONTROL SYSTEM DESIGN
USING SINGULAR PERTURBATION METHODS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Joseph S. Smyth, B.S.E.E.
2nd Lt USAF

Graduate Engineering Electrical

December 1981

Approved for public release; distribution unlimited.

PREFACE

This report is the result of an effort to develop a longitudinal tracker digital flight control system using singular perturbation methods developed by Professor Brian Porter of the University of Salford. Also, a joint effort between Capt. Douglas Porter, Lt. Randall Paschall and myself produced the user-interactive computer program MULTI from a core of algorithms provided by Professor Porter.

The thesis topic parallelled my graduate courses in flight control and digital control. The insights gained concerning engineering trade offs and judgement proved quite valuable. I would like to express my gratitude to Dr. John D'Azzo for his insights and help on technical matters. I also would like to thank Dr. C. H. Houpis for his comments, Capt. J. Silverthorn for his help, and Elisha Rachovitsky for his guidance on behalf of the Flight Dynamics Lab.

Finally, my deep-felt gratitude to Sharie Taylor whose typing and sympathy are an invaluable part of this project.

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A, A_{11}, A_{12}, A_{21}, A_{22}	Continuous Plant Matrix and Partitions
ACG_x	Acceleration of Center of Gravity Along x-Body Axis (ft/sec^2)
CG_z	Acceleration of Center of Gravity Along z-Body Axis (ft/sec^2)
B, B_2	Input Matrix and Partition
C, C_1, C_2,	Output Matrix and Partitions
CDI	Drag Inlet Coefficient
D	Feed-Forward Matrix
$\mathbf{e}$	Error Vector
f	Sampling Frequency
F	Single Engine Thrust (lb)
F_{total}	Total Engine Thrust (lb)
F, F_1, F_2,	Augmentea Output Matrix and Partitions
g	Acceleration of Gravity (ft/sec^2)
h	Height (ft)
kT	Sampling Time Interval

K_o	Proportional Error Controller Matrix
K_1	Integral Error Controller Matrix
ℓ	Number of Outputs; Number of Inputs
M	Mach
	Measurement Matrix
n	Number of States
q	Pitch Rate (rad/sec)
Tsamp	Sampling Time (sec)
u	Forward Velocity Perturbation (ft/sec)
	Input Vector
v	Command Vector
VEL	Total Airspeed Perturbation (ft/sec)
w	Downward Velocity Perturbation (ft/sec)
	Augmented Output Vector
x	Down Range Position Perturbation (ft)
$\underline{x}$	State Vector
y	Output Vector

z	Integral Error Vector
$z_1, z,_2, z,_3$	Closed Loop Asymptotic Roots
α	Angle of Attack (rad)
	Proportion of Direct and Integral Feedback
γ	Flight Path Angle (rad)
$\Gamma(\lambda), \tilde{\Gamma}(\lambda), \hat{\Gamma}(\lambda)$	Asymptotic Transfer Function Matrices
δ_c	Canard Deflection (deg)
δ_{LF}	Lift Flap Deflection (deg)
ϵ	Sigma Matrix Multiplier
λ	Eigen Value
σ_i	Sigma Matrix Elements
Σ	Sigma Output Weighting Matrix
θ_j	Jet Flap Deflection (deg)
θ	Pitch Angle (rad)

ABSTRACT

In this report a single longitudinal tracker is developed for the FPCC aircraft for three different flight conditions. The method used is the singular perturbation method applied to fast-sampling, output-feedback digital control. Each flight condition has three command modes: positive pitch pointing, vertical translation and straight climb. A sensitivity study is performed to validate the design and illustrate design parameter influences on system response.

A computer-aided-design program, MULTI, is developed to assist in the iterative design process. The program is fully interactive, user-oriented, and provides error protection. The program allows complete design and simulation of three types of control law designs: known-regular plants, known-irregular plants, and unknown plants. The report contains a brief but complete summary of each of these control law design methods.

A user's manual and a programmer's manual are provided for further development of the program.

CHAPTER I

INTRODUCTION

Digital flight control systems (DFCS) for aircraft can provide more flexibility and ability to perform widely varying tasks than analog flight control systems (AFCS). As an example, the space shuttle currently uses a digital fly-by-wire (FBW) system to control its motion over a large range of speeds and altitudes.

Flight propulsion control coupling (FPCC) is one of a series of test studies conducted by the Flight Dynamics Lab to investigate the possibilities of integrating one or more subsystems on an aircraft to provide greater performance. These subsystems include guidance and navigation, propulsion, flight control, fire control and others. By integrating these subsystems together, pilot workload is reduced, mission performance is optimized and survivability is enhanced. The FPCC "paper" aircraft used for this thesis is a control-configured vehicle (CCV) and can provide direct lift and sideforce as well as the normal modes of control. This is accomplished by the use of jet flaps and horizontal and vertical canards. The aircraft is also statically unstable to provide greater pitch response.

The final key to the solution of combining a DFCS with a CCV aircraft is a method to integrate the controls to achieve the performance desired. The method studied and presented in this thesis is based on the latest work by Professor Brian Porter of the University of Salford, England. He is currently under contract to the Flight Dynamics Lab and his latest work concerns the application of singular perturbation methods to multivariable digital controls (Refs. 1, 2 and 3). By using output feedback, fast sampling, and a proportional-plus-integral controller network, fast tracking coupled with good disturbance rejection is achieved. Futhermore, the controller is robust enough to handle plant, input, and output parameter changes (such as would occur at various flight conditions) and still return good performance.

STATEMENT OF PROBLEM

The thesis is broken into two separate and distinct tasks:

(1) Develop an interactive, user-oriented computer program based on the algorithms furnished by Professor Porter (Ref. 4). It includes the three design methods outlined in Chapter II and contains a flexible plotting package to provide a quick graphical interpretation of the results. Finally, the program is flexible enough to allow a meaningful iterative design process and the ability to test a single control law design over a wide range of flight conditions.

(2) Using the computer program, named MULTI, design a digital flight control system for the longitudinal axis to track three different command vectors: positive pitch pointing, vertical translation, and straight climb. The design is based upon the transonic flight condition, Mach 0.9, h = 30,000 ft. A robustness check is made by applying the newly developed control law to two other flight conditions, subsonic, Mach 0.6, 0 ft., and supersonic, Mach 2.3, 40,000 ft.

APPROACH

A top down approach is employed and modularity is stressed in the development of MULTI. The list of program requirements and programming techniques are given in Chapter III. The core of algorithms supplied by

Professor Porter are adapted for use on the CDC 6600 computer at AFIT and provide an iterative design tool.

The equations of motion for the FPCC aircraft are derived by the use of the FPCCSIM program provided to the Flight Dynamics Lab by Honeywell (Refs. 5 and 6). Three flight conditions are examined, subsonic, transonic and supersonic, and a reduced 4-state, 3 input, 3 output model is derived for each.

The program MULTI is used to develop a longitudinal tracker control law for the transonic flight condition. The control law is then applied to the subsonic and supersonic flight conditions to test the robustness of the design. Finally, a sensitivity study is performed by changing the individual design parameters, both to validate the original design and to provide guidance to future designers as to the influence of the individual design parameters.

REVIEW OF LITERATURE

The purpose of this thesis was to obtain a working knowledge of Porter's latest work. The review of literature consisted of the series of papers (previously cited) dealing with singular perturbation methods in multi-variable output feedback control. Other literature included topics such as transmission zeros (Ref. 7) and the course material presented in EE 7.08 - Linear Multivariable Feedback Systems.

CHAPTER II.

DESIGN METHODOLOGY USING SINGULAR PETURBATION METHODS

INTRODUCTION

The theory presented here is an overview of Dr. Brian Porter's papers (Refs. 1, 2, 3) presented during a series of lectures at AFIT in May, 1981.

Dr. Porter's latest work concentrates on the application of singular-perturbation methods for multi-variable control. Output feedback is used to regulate the output and track the desired input command vector. The state equations are:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \qquad (1)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \qquad (2)$$

where:

$\mathbf{x}$ = state vector

$\mathbf{u}$ = input vector

$\mathbf{y}$ = output vector

The structure of the controller is a proportional-plus-integral controller. Figure 1 shows its structure in the overall system block diagram.

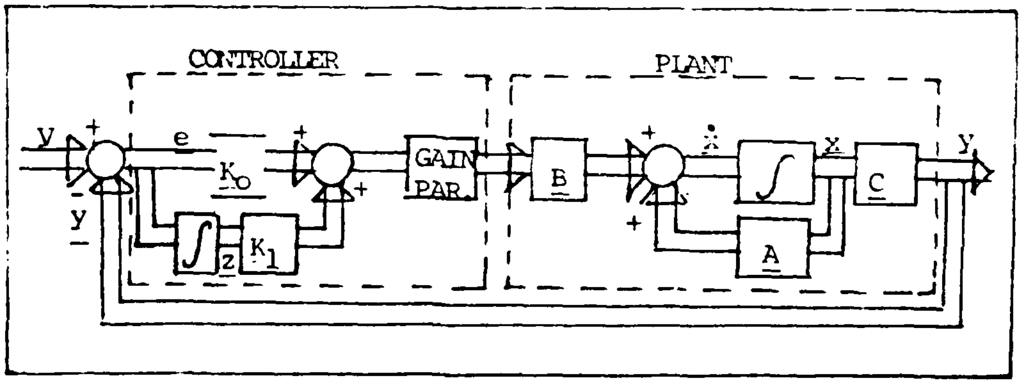


Fig. 1. Block diagram structure of plant and controller

The controller may either be analog or digital. This thesis concentrates on the digital case. Samplers are assumed to be placed in the feedback loop and at the command input vector, x. A sample-and-hold device is assumed to be placed after the gain parameter to maintain piece-wise constant inputs to the plant. The design method of this thesis does not require a z-domain analysis.

The controller matrices, K_o and K_1 are computed by using the first Markov parameter of the continuous plant as their basis. The concept of the first Markov parameter is important and is developed in each of the following sections. It is unique for each of the 3 types of plants studied : Known Plants (regular or irregular) and Unknown Plants.

KNOWN PLANTS

The category of plants consists of those that have a known detailed state space description. Equations (1) and (2) are constrained to the following format:

$$\begin{bmatrix} \dot{\chi}_1(t) \\ \dot{\chi}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \chi_1(t) \\ \chi_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} u(t) \tag{3}$$

$$y(t) = [C_1, C_2] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{4}$$

where

n = number of states

ℓ = number of inputs = number of outputs

x_1 = state vector partition; length (n-ℓ)

x_2 = state vector partition; length (ℓ)

A_{11} = A matrix partition; size $(n-\ell)$ X $(n-\ell)$

A_{12} = A matrix partition, size $(n-\ell)$ X ℓ

A_{21} = A matrix partition, size ℓ X $(n-\ell)$

A_{22} = A matrix partition, size ℓ X ℓ

B_2 = non-zero B matrix partition; full rank; size ℓ X ℓ

C_1 = C matrix partition; size ℓ X $(n-\ell)$

C_2 = C matrix partition; size ℓ X ℓ

If the state equation is not in the form of Eq. (3), this form can be obtained by a transformation. However, it is not necessary to perform this transformation to synthesize the control law. The control law is:

$$u(kT) = f\left\{K_o e(kT) + K_1 z(kT)\right\} \qquad (5)$$

where

f = sampling frequency

e = error vector = $v - y$

z = integral error vector

K_o = proportional error controller matrix

K_1 = integral error controller matrix

kT = sampling time interval

v = constant command vector

The closed-loop system augmented with integrators, in order to obtain the control law of Eq. (5), tracks a constant command input, that is,

$$\lim_{t \to \infty} e(t) = 0$$

However, this is possible only if the matrix pair (A,B) is controllable and the system augmented with the integrators is controllable. The latter condition is satisfied if

$$\text{rank}\begin{bmatrix} A - \lambda I & B \\ -C & 0 \end{bmatrix} = n + \wp$$

Note, to satisfy the controllability requirement, the B_2 matrix must have full rank. This requirement can be satisfied two ways. One way is to combine the servo dynamics into the equations. This increases the number of states and complicates the design procedure. The second way is to combine two inputs that are functionally similar. This was done in this thesis by combining the jet flaps and maneuver flaps, both of which provided large lift with minimal pitching moment. This approach was taken since the aircraft studied has a rank-deficient B_2 matrix. This is discussed in Chapter IV.

For most aircraft, the first partitioned set of states, x_1, usually consists of kinematic equations that have no control inputs, such as $\dot{\theta} = q$, $\dot{\phi} = p$ and $\dot{\psi} = r$. The second partition, x_2, consists of states which have control inputs through the B_2 matrix. Note that no provision for a feed forward (D) matrix is included. The presence of a D matrix (caused by choosing accelerations as outputs) can be circumvented by including the servo outputs (surface deflections) as states.

Once the plant is in the required format, its first Markov parameter is determined. If C_2 is full rank, the plant is described as "Regular" and the matrix $C_2 = B_2$ has full rank and is the first Markov parameter. Note that $CB = C_2B_2$.

If C_2 is rank deficient, the plant is described as "Irregular" and C_2 must be augmented to make it full rank. Design methodologies for "Regular" and "Irregular" plants are described below.

(A) REGULAR PLANTS

If the plant is regular, then the first Markov parameter, C_2B_2, is invertible. This is the basis for computing the controller matrices:

$$K_o = \alpha(C_2B_2)^{-1}\Sigma \qquad (6)$$

$$K_1 = (C_2B_2)^{-1}\Sigma \qquad (7)$$

where

K_o = proportional error control matrix

K_1 = integral error controller matrix

α = proportion of direct and integral error control

C_2B_2 = first Markov parameter

Σ = output weighting matrix (selected as a diagonal matrix to reduce intercoupling)

For regular plants the error is the difference between the command input vector v(t) and the output vector y(t), that is,

$$e(t) = v(t) - y(t) \qquad (8)$$

Since the complete system is augmented with a vector integrator, with a constant command vector the steady-state value of the error vector is zero.

$$e_{ss}(t) = 0 \qquad (9)$$

Porter demonstrates that the closed loop transfer-function matrix, $G(\lambda)$, assumes the asymptotic form, $\Gamma(\lambda)$, as the perturbation parameter, T_{samp}, approaches zero (or $f \rightarrow \infty$). $\Gamma(\lambda)$ may be partitioned into two components given by:

$$\Gamma(\lambda) = \widetilde{\Gamma}(\lambda) + \widehat{\Gamma}(\lambda) \qquad (10)$$

where

$$\tilde{\Gamma}(\lambda) = [K_o{}^{-1}K_1, 0]\ [\lambda I_N - I_N - TA_o]^{-1} \begin{bmatrix} 0 \\ \\ TA_{12}C_2{}^{-1} \end{bmatrix} \tag{11}$$

$$\hat{\Gamma}(\lambda) = C_2[\lambda I_L - I_L + B_2K_oC_2]^{-1}B_2K_o \tag{12}$$

and

$$A_o = \begin{bmatrix} -K_o{}^{-1}K_1 & 0 \\ \\ A_{12}C_2{}^{-1}K_o{}^{-1}K_1 & A_{11} - A_{12}C_2{}^{-1}C_1 \end{bmatrix} \tag{13}$$

Since A_o is block structured, the roots of Eq. (11) can be determined. They are finite and are identified with the "slow modes". They are divided into two subsets

$$Z_1 = \left\{\lambda \in C : \left|\lambda I_L - I_L + TK_o{}^{-1}K_1\right| = 0\right\} \tag{14}$$

$$Z_2 = \left\{\lambda \in C : \left|\lambda I_{N-L} - I_{N-L} - TA_{11} + TA_{12}C_2{}^{-1}C_1\right| = 0\right\} \tag{15}$$

The structure of Eq. (11) shows that the Z_1 roots (associated with the upper left partition of A_o) become unobservable. Also, the Z_2 roots (associated with the lower right partition of A_o) become uncontrollable.

The final set of roots, Z_3, are identified with the "fast mode" and are determined as:

$$Z_3 = \left\{\lambda \in C : \left|\lambda I_L - I_L + C_2B_2K_o\right| = 0\right\} \tag{16}$$

As the sampling frequency (f) increases, these roots provide the dominant transient output for the system. Evaluation of Eq. (11) yields the result that $\tilde{\Gamma}(\lambda) = 0$. This leaves the asymptotic transfer function as

$$\Gamma(\lambda) = \hat{\Gamma}(\lambda) = (\lambda I_L - I_L + C_2B_2K_o)^{-1}C_2B_2K_o \qquad (17)$$

This transfer function is the result in the limit as the sampling time goes to zero. The main thrust of this development shows that as the sampling time is decreased, only the fast modes are present in the output. The matrix K_o is chosen such that

$$C_2B_2K_o = \Sigma = \text{diag}\ \{\sigma_1, \sigma_2 \ldots \sigma_L\} \qquad (18)$$

Then the asymptotic transfer function matrix becomes

$$\Gamma(\lambda) = (\lambda I_L - I_L + \Sigma)^{-1} \qquad (19)$$

$$= \text{diag}\ \left\{\frac{\sigma_1}{\lambda - 1 + \sigma_1}, \frac{\sigma_2}{\lambda - 1 + \sigma_2} \ldots \frac{\sigma_L}{\lambda - 1 + \sigma_L}\right\}$$

Therefore decoupling is accomplished.

Equation (19) can be simplified even further by specifying $\Sigma = I_L$. Equation (19) now becomes

$$\Gamma(\lambda) = \text{diag}\left\{\frac{1}{\lambda_1}, \frac{1}{\lambda_2} \ldots \frac{1}{\lambda_L}\right\} \quad (20)$$

Equation (18) dictates the form of the K_o matrix:

$$K_o = (C_2B_2)^{-1}\sum \qquad (21)$$

For the purposes of compatibility with the computer program, MULTI, and the other design methods, K_o and K_1 and given in Eqs. (6) and (7) and are repeated below:

$$K_o = \alpha(C_2B_2)^{-1}\sum \qquad (6)$$

$$K_1 = (C_2B_2)^{-1}\sum \qquad (7)$$

LIMITATIONS

Transmission zeros become a problem if they are located at the origin or in the right-half plane (Ref. 7). When the state and output equations have the form of Eqs. (3) and (4), Porter shows that the transmission zeros are:

$$Z_T = \left\{\lambda \in \mathcal{C} : \left|\lambda I_{N-L} - A_{11} + A_{12}C_2^{-1}C_1\right| = 0\right\} \qquad (22)$$

It is noted that this expression contains terms which are similar to those found in Eq. (15). This indicates that the second set of finite roots, Z_2, approach the transmission zeros as the sampling frequency approaches infinity. If these transmission zeros are at the origin or in the right-half plane, the Z_2 roots may become unstable as they approach the transmission zeros. If this occurs, it may be necessary to define different output variables to achieve a more desirable transmission zero location.

(B) IRREGULAR PLANTS

Irregular plants are those whose first Markov parameter (C_2B_2) does not have full rank. Since B_2 is constrained to have full rank by Eq. (3), C_2 must be augmented to make up its deficiency. This is accomplished with a transducer or measurement matrix M. The feedback vector is now defined as W.

$$w(t) = [F_i, F_2] \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \qquad (23)$$

$$\begin{aligned} w(t) &= [C_1 + MA_{11}, C_2 + MA_{12}] \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \\ &= y(t) + M\,[A_{11} \;\; A_{12}] \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \end{aligned} \qquad (24)$$

$$w-y = M\,[A_{11} \;\; A_{12}] \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \qquad (25)$$

With constant command inputs the steady state values of the states are constant. Therefore

$$\dot{X}_1 = 0 = [A_{11}\; A_{12}] \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \qquad (26)$$

It is evident from Eqs. (24) through (26) that $W(t)_{ss} \rightarrow y(t)_{ss}$.

A modified block diagram for the irregular plant design is shown in Fig. 2.

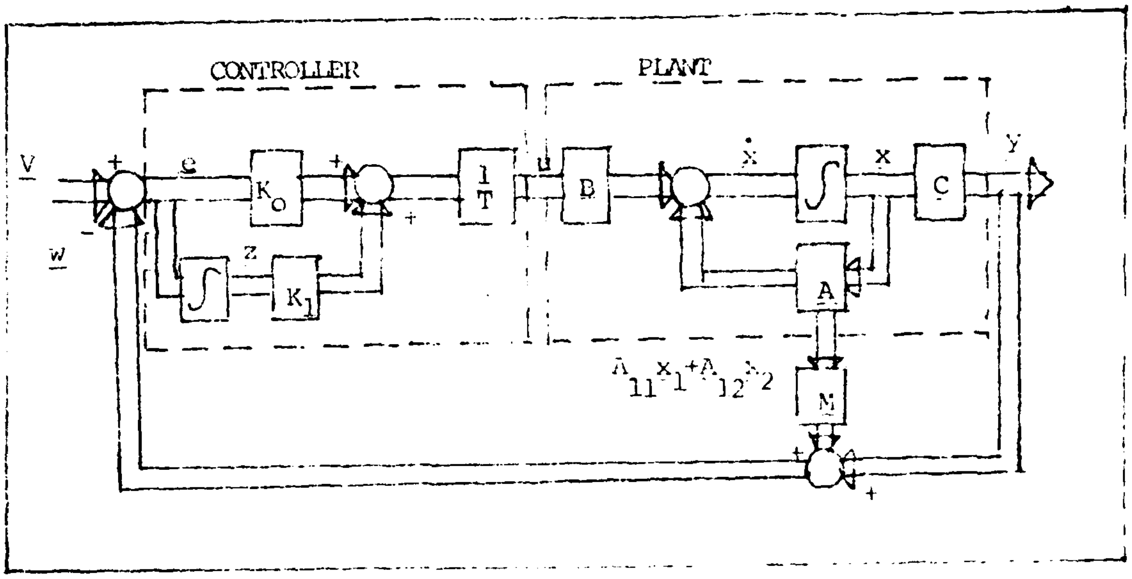


Fig. 2 Block diagram - Irregular Plant and Controller

The only difference between Figs. 1 and 2 is the extra measurements of the states in the feed-back loop. The new error vector is:

$$e = v - w \qquad (27)$$

The new first Markov parameter is now F_2B_2. This requires F_2 to have full rank in order for the first Markov parameter to be invertible.

$$F_2 = [C_2 + MA_{12}] \qquad (28)$$

where

F_2 is $\ell \times \ell$
C_2 is $\ell \times \ell$
M is $\ell \times (n-\ell)$
A_{12} is $(n-\ell) \times \ell$

Some insight is needed in order to pick a measurement matrix. This is addressed later in this section.

Once M is chosen properly and F_2 is calculated, the controller matrices are:

$$K_o = \alpha(F_2B_2)^{-1}\Sigma \qquad (29)$$

$$K_1 = (F_2B_2)^{-1}\Sigma \qquad (30)$$

Note the similarities between Eqs. 29 and 30, and Eqs. 6 and 7, respectively. The control law, Eq. 5, is unchanged.

As with the regular plant design, using $f = \frac{1}{T}$ as the perturbation parameter, the asymptotic transfer function as $f \to \infty$ assumes the form:

$$\Gamma(\lambda) = \tilde{\Gamma}(\lambda) + \hat{\Gamma}(\lambda) \qquad (31)$$

where

$$\tilde{\Gamma}(\lambda) = [C_2F_2^{-1}K_o^{-1}K_1, C_1 - C_2F_2^{-1}F_1] \quad [\lambda I_N - I_N - TA_o]^{-1} \begin{bmatrix} 0 \\ TA_{12}F_2^{-1} \end{bmatrix} \qquad (32)$$

$$\hat{\Gamma}(\lambda) = [C_2]\ [\lambda I_L - I_L + B_2K_oF_2]^{-1}BK_o \qquad (33)$$

and

$$A_o = \begin{bmatrix} -K_o^{-1}K_1 & 0 \\ A_{12}F_2^{-1}K_o^{-1}K_1 & A_{11} - A_{12}F_2^{-1}F_1 \end{bmatrix} \qquad (34)$$

Note the similarities between Eqs. 32 through 34 and Eqs. 11 through 13. If $F_2 = C_2$ in Eqs. 32 through 34, the result is Eqs. 11 through 13, respectively.

As in the regular plant design, the A_o matrix is block structured and the roots can be easily determined. Equation 32 yields two sets of finite, closed-loop eigen values associated with the "slow modes".

$$Z_1 = \left\{ \lambda \in \mathcal{C} : \left| \lambda I_L - I_L + TK_o^{-1}K_1 \right| = 0 \right\} \tag{35}$$

$$Z_2 = \left\{ \lambda \in \mathcal{C} : \left| \lambda I_{N-L} - I_{N-L} - TA_{11} + TA_{12}F_2^{-1}F_1 \right| = 0 \right\} \tag{36}$$

Analogous to the regular plant, modes associated with the Z_1 roots become uncontrollable as $f \rightarrow \infty$ and effectively dissappear as sampling time T is decreased. On the other hand, the mode associated with the Z_2 roots remain observable due to the extra measurements generated by the M matrix. This is quite different from the regular plant design where the modes associated with the Z_2 roots became unobservable as T is decreased.

The third set of roots are associated with the "fast modes" derived from Eq. 33.

$$Z_3 = \left\{ \lambda \in \mathcal{C} : \left| \lambda I_L - I_L + F_2B_2K_o \right| = 0 \right\} \tag{37}$$

Again, the Z_3 roots are the desired outputs since tracking becomes "tighter" and interaction minimal as $f \rightarrow \infty$.

The asymptotic transfer function matrix for the irregular plant contains both slow and fast modes, rather than just the fast modes for the regular plant seen in Eq. 17.

$$\Gamma(\lambda) = \underbrace{[C_1 - C_2F_2^{-1}F_1]\,[\lambda I_{N-L} - I_{N-L} - TA_{11} + TA_{12}F_2^{-1}F_1]^{-1}\,[TA_{12}F_2^{-1}]}_{\text{(SLOW MODES)}}$$

$$+$$

$$\underbrace{[C_2F_2^{-1}]\,[\lambda I_L - I_L + F_2B_2K_0]^{-1}\,[F_2B_2K_0]}_{\text{(FAST MODES)}} \tag{38}$$

It can be shown that this is a diagonal matrix and therefore interaction is reduced as $f \rightarrow \infty$.

Equation 37 dictates the form of the controller matrices by choosing K_o such that

$$F_2B_2K_o = (C_2 + MA_{12})B_2K_o = \Sigma = \text{diag}\,\{\sigma_1, \sigma_2 \ldots \sigma_l\} \tag{39}$$

This leads to the expression for K_o:

$$K_o = (F_2B_2)^{-1}\Sigma \tag{40}$$

For the purpose of compatibility of the computer program, MULTI, and the other design methods, the matrices K_o and K_1 are re-defined as

$$K_o = \alpha(F_2B_2)^{-1}\Sigma \tag{29}$$

$$K_1 = (F_2B_2)^{-1}\Sigma \tag{30}$$

MEASUREMENT MATRIX

The transmission zeros of the augmented system are:

$$Z_T = \{\lambda \in C : |\lambda I_{N-L} - A_{11} + A_{12}F_2^{-1}F_1| = 0\} \tag{41}$$

Note that this equation contains similar terms as Eq. 36, the expression for the second set of finite, slow modes, Z_2. Since

$$F_2^{-1}F_1 = [C_2 + MA_{12}]^{-1}\,[C_1 + MA_{11}] \tag{42}$$

It is evident that any choice of M influences the transmission zero location for the plant. A proper choice of M is a rather easy choice if M is an L x 1 vector, in other words, if the number of states is one more than the number of outputs. This involves solving Eq. 36 for one root. However, this becomes rather involved when Eq. 36 contains two or more roots. Ridgely (Ref. 8) has a possible solution to this problem by suggesting the use of eigenstructure assignment. Although his work is for an analog system, the method is adapted in this thesis for digital system designs.

The structure roots are determined by

$$\left| \lambda I_{N-L} - I_{N-L} - TA_{11} + TA_{12}F_2^{-1}F_1 \right| = 0 \qquad (43)$$

This is similar in structure to the state feedback eigenstructure approach where

$$\left| \lambda I_{N-L} - A - B\underset{\sim}{K} \right| = 0 \qquad (44)$$

where

$$A = I_{N-L} + TA_{11} \qquad (45)$$

$$B = -TA_{12} \qquad (46)$$

$$\underset{\sim}{K} = F_2^{-1}F_1 \qquad (47)$$

Equations 45 and 46 can be calculated for a specific sampling time, T. They are entered into the interactive program, CESA (Ref. 9), as the ASD and BSD matrices, respectively. The CESA program requests the number of eigen values to be assigned. The program then generates a null space for each eigenvalue and the user picks one or more linearly independant eigenvectors from those null spaces. The program then generates the control law matrix K . Once K is known, the measurement matrix can be formed as follows:

$$K = [C_2 + MA_{12}]^{-1} \; [C_1 + MA_{11}] \qquad (48)$$

$$M[A_{12}K - A_{11}] = [C_1 - C_2K] \qquad (49)$$

$$M = [C_1 - C_2K] \; [A_{12}K - A_{11}]^{-1} \qquad (50)$$

There is a solution to Eq. 50 if the inverse of the matrix $[A_{12}K - A_{11}]$ exists.

The matrix manipulations shown above are included in option 18 of MULTI, the computer package developed as part of this thesis. Chapter III and Appendices B and C contain a complete description of the program. Option 18 returns an M matrix once a K matrix from CESA is entered.

The one draw back to this method is that the K matrix, and consequently the M matrix, is not a unique solution to the eigenvalue assignment problem. K is assumed to be an independant variable by the program CESA. However, the interpretation of K for the output feedback methods of this thesis is that it is a function of A_{11} and A_{12} which are also present in the ASD and BSD matrices.

Porter used measurement matrices that include 0.25 as one of the non zero elements. This may be a trial and error figure that returned suitable results. (Refs. 3 and 10). In both reference papers the rank deficiency of the C_2 matrix is eliminated by clever placement of non-zero elements in the M matrix.

A final word of caution when choosing a measurement matrix is to observe the vector space of A_{12} *and* C_2. *Since the* F_2 *matrix is a linear* combination of A_{12} and C_2 (via the M matrix), full rank may not be possible if A_{12} and C_2 share one or more row vectors. If A_{12} and C_2 share one or more row vectors, the dimension of the vector space from which F_2 is chosen may be less than ℓ . This can occur if one of the output equations is identical to one of the X_1 equations and can lead to a non-invertible first Markov parameter.

UNKNOWN PLANTS

Porter's paper (Ref. 1) addresses the need for obtaining an accurate state-space representation of a dynamic plan and then synthesizing a digital controller for it. Porter's contention is that such a detailed investigation of the plant's dynamic qualities is unnecessary and that a digital controller can be synthesized using the plant's steady-state transfer function matrix, G(0). The matrix G(0) is easily obtained by offline tests and serves as the first Markov parameter for the plant.

The control law is stated as:

$$u = T_{samp} \left\{ K_o e(kT) + K_1 z(kT) \right\} \qquad (51)$$

where

u is the input vector

T_{samp} is the sampling time

K_o is the proportional error controller matrix

K_1 is the integral error controller matrix

e is the error vector, $v - y$

z is the integral error vector

The only difference between Eq. 51 and the control law equation for known plants, Eq. 5, is the scalar multiplier T_{samp} (instead of f). This is the gain parameter shown in Fig. 1. The controller matrices are given by

$$K_o = \sigma [G(0)]^{-1} \Sigma \qquad (52)$$

$$K_1 = [G(0)]^{-1} \Sigma \qquad (53)$$

where

σ is the proportion of direct to integral control

G(0) is the steady-state tranfer function matrix

Σ is the output weighting matrix

Equations 51 through 53, used with the controller structure in Fig. 1, lead two sets of closed loop characteristic roots, Z_1 and Z_2.

$$Z_1 = \left\{ \lambda \in C : \left| \lambda I_N - I_N - TA + O(T^2) \right| = 0 \right\} \qquad (54)$$

$$Z_2 = \{\lambda \in C : \left| \lambda I_L - I_L + T^2 \Sigma + O(T^3) \right| = 0\} \qquad (55)$$

where

Z_1 = set of roots due to plant states

Z_2 = set of roots due to integrator states

N = number of plant states

L = number of inputs = number of outputs

T = sampling time

A = open-loop plant matrix

$O(T^2)$ = analytic expression whose value is on the order of T^2

$O(T^3)$ = analytic expression whose value is on the order of T^3

For example, a plant whose open-loop matrix contains only the diagonal elements $\{-1,-1,-1\}$, the Z_1 roots would be:

$$Z_1 = \{1 - T + O(T^2),\ 1 - 2T + O(T^2),\ 1 - 3T + O(T^2)\} \qquad (56)$$

If a Σ matrix is chosen that contains only the diagonal elements $\{120,60\}$, the Z_2 roots would be:

$$Z_2 = \{1 - 120T^2 + O(T^3),\ 1 - 60T^2 + O(T^3)\} \qquad (57)$$

It is readily apparent that sampling time has a dominant effect on the closed-loop roots. As T approaches zero, all roots converge inside the unit circle to the Z = 1 point.

While it is possible to obtain an analytic solution for the closed-loop roots, the premise of this design method is that it is better to observe the output time responses of the system and to adjust the sampling time to obtain suitable responses. This, in effect, bypasses the conventional approach of assigning closed-loop eigen values and lets the designer deal strictly with the system output in the time domain.

LIMITATIONS AND PARAMETER CHOICE

The unknown plant design method has a few limitations of particular interest to the flight control engineer. The main limitation is that the open-loop system must be stable, i.e. all roots of the discrete-time representation of the system must be inside the unit circle. This is evident from the following example. In an unstable plant with a single root at $s = +3$ is put into the form of Eq. 53, the Z_1 root is:

$$Z_1 = \left\{ 1 + 3T - 0(T^2) \right\} \qquad (58)$$

No value of T can bring this root inside the unit circle. The limitation of requiring open-loop stability precludes this design method's use for aircraft with relaxed static stability in the pitch axis or spiral divergence in the yaw axis.

The other limitation is that this design can not be used if the plant has zero-valued transmission zeros, since this reduces the rank of G(0). This leads to a non-invertible first Markov parameter which is the basis of the controller matrices, K_o and K_1. This limitation occurs primarily when the outputs are angular rates such as P, Q, and R. This can be avoided by reforming the output matrix C so that just the angles are the observed output.

While this method is identified as useful for an unknown plant, it can also be used when the plant model is known. The plant equations are set up in the normal matrix equation format using the A, B, C, D matrices. Servos can be included in the equations.

Parameters affecting the response are sampling time T, the proportion of direct and integral feedback α, and the output weighting matrixΣ . There is an additional scalar parameter, ϵ that multiplies the sigma matrix. Since the design is an iterative process, the most logical route is to set nominal values for the parameters and to adjust accordingly. Sampling time can be nominally set at $T = 0.01$. Alpha can be initially 1 for equal direct and integral feedback. Setting $\Sigma = I_L$ and $\epsilon = \frac{1}{T}$ provides an iterative baseline from which adjustments can be made. If one output is too slow, an increase in its corresponding element in the Σ matrix will correct this. If the input energies and outputs are all too small, an increase in the sigma matrix multiplier corrects this. Decreasing sampling time causes the outputs to track the command vector faster. Finally, adjustment of α affects the amount of undershoot or overshoot of the outputs.

CONCLUSION

Three distinct design methods are presented: known plants - regular, known plants - irregular, and unknown plants. Each has its own benefits and limitations. Figure 3 represents a designer with a flow chart that outlines the design process.

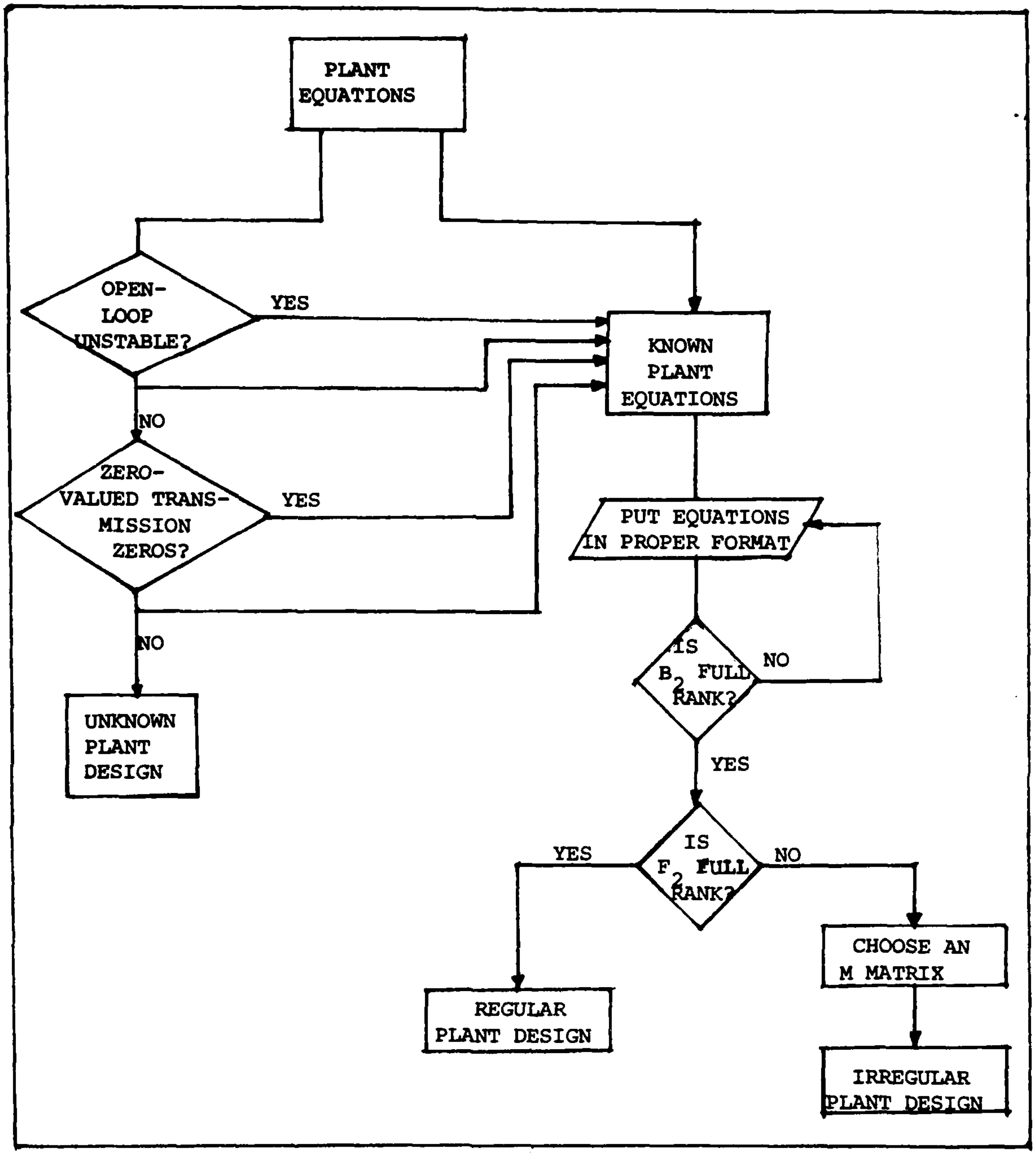


Figure 3. Design Methodology Flowchart

CHAPTER III

"MULTI" COMPUTER PROGRAM STRUCTURE AND DEVELOPMENT

INTRODUCTION

The computer program "MULTI" is an interactive, user oriented computer program that develops digital controllers for error actuated trackers using singular perturbation methods. It is designed to fulfill the following objectives.

1. The computer program must be interactive, user oriented, and have the ability to recover from user input errors.

2. The program must be flexible enough to allow for a meaningful interactive design process.

3. It must be able to test various plant parameter variations (such as different flight conditions) with a single control law in order to demonstrate the robustness of a design.

4. It must contain a flexible plotting package in order to graphically interpret design parameter changes.

5. The program must have the ability to write information to and read information from local data files. This provides fast design iterations and removes the laborious necessity of entering plant data every time the program is accessed.

6. It should incorporate all three design methodologies : known - regular plants, known - irregular plants, and unknown plants. It should also direct the user to an alternate design procedure if the first design has limitations.

7. It should incorporate such features as inclusion of servo and

sensor dynamics in the simulation.

8. It should also feature a provision to include limits for the input values (such as rudder and aileron limits) in the simulation.

9. The program must be written in a top-down approach so that future improvements can be made with minimal problems.

The program "MULTI" is based on a core of algorithms written by A. Hemani, entitled "Package 200" (Ref. 4). An analysis of this program reveals that it does not satisfy all of the previously cited objectives. Capt. Porter (Ref. 11) undertook the major effort of developing and refining the program MULTI by modifying the algorithms in Hemani's program in order to satisfy some of the desirec objectives. The author of this thesis made a large contribution to that effort. MULTI is fully interactive, compatible with operation on the CDC 6600's INTERCOM, and is easily accessed. Lt. Paschall provided the matrix inversion subroutine, Capt. Porter wrote the plotting package, and the author provided the routine to solve the differential equations with the control surface constraints.

The inclusion of these subroutines into "Package 200" required too much core memory for use on INTERCOM. It also lacked the known-regular and known-irregular design packages, and did not have a provision to save information on data files. The program also lacked the inherent flexibility required in a user-oriented, iterative design package.

The problem of flexibility is solved by the use of "options" and is similar to the approach used in the interactive program "TOTAL" (Ref. 13). Plant, sensor and servo information is entered in options 1-9.

Design parameter information is entered and calculation of the control law matrices is performed in options 11-19. Simulation parameters are entered and the simulation is performed in options 21-29. Plotting options, both CALCOMP and terminal plots, are included in options 31-39. Options 0,10,20 and 30 provide the lists of input, design, simulation and plotting options, respectively. Options 101 to 139 provide a print out of the data values entered in options 1 to 39, respectively.

OVERLAY STRUCTURE

One main problem in this stage of the development of MULTI is the resultant size of the program. It can not be compiled on INTERCOM, which has a 65,000 limit. In order to adapt the program to fit the computer limits there are two software structures to chose from: (1) segmented load using loader commands to direct program execution or (2) th^e^ use of overlays. The use of overlays is chosen as the approach for this thesis in order to reduce the amount of storage required and to make efficient use of the available field length. The program is divided into overlays. Each overlay is an executable program and the overlays are a collection of programs combined into an overlay structure. Since the program is divided into options and a top-down approach is used, the overlay structure is easily implemented.

Overlays are classified as main overlays, primary overlays and secondary overlays. There is a single main (or zero) overlay that is the executive program and directs the loading of other overlays. The main overlay is loaded first and remains in memory at all times. A primary overlay can be loaded following the main overlay, and a secondary overlay can be loaded immediately following the primary overlay. This allows the program to load into memory only the primary overlay that is required, and a secondary overlay if necessary. Since primary overlays can be as large or small as necessary, a large amount of code can be partitioned among the primary overlays in order to drastically reduce the size of the resident memory core. The program "MULTI" uses 12 overlays. These are discussed in detail in Appendix B. A simplified block diagram is shown in Fig. 4. Additional information concerning overlays can be found in the CDC FORTRAN 5 manual (Ref. 14).

DATA STORAGE AND ERROR PROTECTION

Data is passed between overlays by the use of labeled common blocks. If an overlay uses any variable or array that is used in the main overlay or any other primary overlay, it must be included in a labeled common block in the specific overlays. Otherwise, any arrays that are used or initialized are destroyed when another overlay is loaded. There are 24 labeled common blocks in "MULTI".

A useful feature of "MULTI" is the ability to open data files and write information to them. Similarly, the program can also open specified data files to read information from them. Overlay (14,0) is called when the program is halted. This automatically writes the plant, servo and sensor state equation information to a file called "MEMO". It also writes the design parameters and control law matrices to a file called MEM1O". These local files may be cataloged in the normal fashion to save their contents.

In order to read plant, servo and sensor state equation information, the user calls Option #9 and types in the name of the local file to be accessed. If the user wishes to read the G(0) matrix

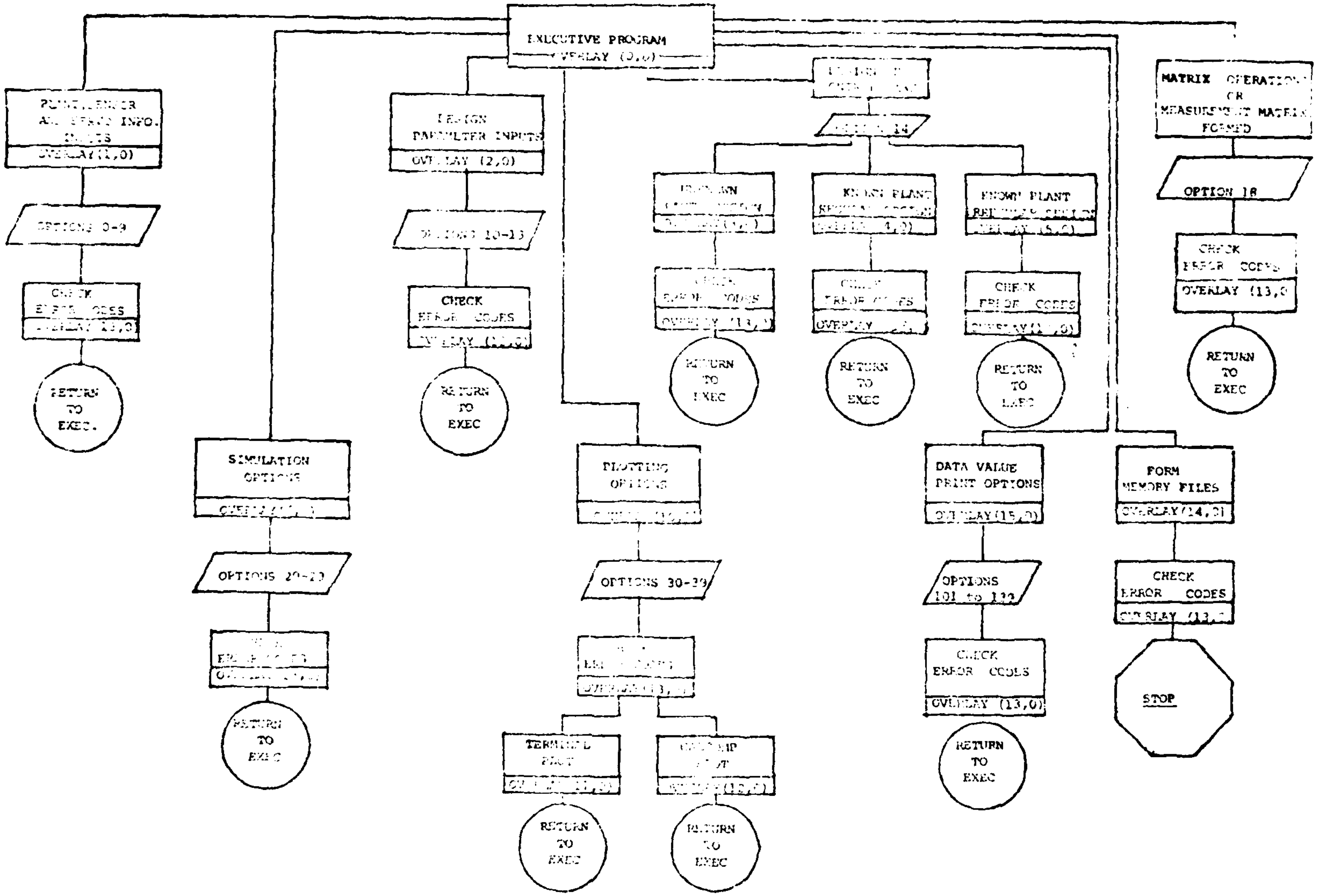


Figure 4. Block Diagram-MULTI Program Flowchart

information (used for unknown plant design), the user calls Option #8. In order to read in the design parameters and control law matrices, the user calls Option #19. This gives the designer the ability to enter different flight condition for an aircraft and to test them against a single control law design.

Error protection is provided on 2 levels. The first level is the ability of the designer to correct a mistake when entering matrices without having to enter all the information once again. The second level of error protection informs the user if he has bypassed an entry before progressing further. For example, if all the design parameters have not been entered (Options 11-13), the control laws are not calculated. In this case the program instructs the user as to which parameter is missing.

As a final safeguard, Options 101 through 139 print out the current values of data corresponding to Options 1 through 39, respectively. This permits a double-check of all parameters.

With the modifications noted in this Chapter all nine objectives listed previously are met.

CHAPTER IV

FPCC AIRCRAFT MODEL DEVELOPMENT

INTRODUCTION

The aircraft model used in this thesis is a "paper" aircraft. It is the result of a joint effort between the Flight Dynamics Lab and three companies, Lockheed, Honeywell and Pratt & Whitney, to explore the effects of coupling the propulsion and flight control systems of an aircraft. The resulting study (Ref. 15) and a follow-up study (Ref. 16) demonstrated the possiblities of flight propulsion control coupling (FPCC) and its performance advantage.

The study set up a methodology to choose between a number of new technologies in order to satisfy the following guidelines:

Greatly expanded flight envelope

Increased agility

Improved stability and control

Improved handling qualities

Reduced size/weight/cost

Extended opterational lifetime

The technologies chosen to be incorporated into the aircraft are presented in Table I along with the relative merits of a canard configuration or an aft-tail configuration for the aircraft.

The canard configuration is chosen, primarily because the aft-tail configuration proves to be a limitation when using jet flaps/vectored thrust.

TABLE I ADVANCED AIRCRAFT TECHNOLOGY SUITABILITY

ADVANCED TECHNOLOGY		CANARD CONFIGURATION		AFT-TAIL CONFIGURATION	
		CANDIDATE	RATIONALE	CANDIDATE	RATIONALE
Fly-By-Wire (Digital)		Yes	RFQ Reqmt, Adv Control Mechanization	Yes	RFP Reqmt Adv Control Mechanization
Relaxed Inherent Static Stability		Yes	Improved Controllability	Yes	Trim Drag Reduction
Maneuver Flaps		Yes	Reduced Drag, Delayed Buffet Onset	Yes	Reduced Drag, Delayed Buffet Onset
Jet Flap/ Vectored Thrust		Yes	Reduced Drag, Delayed Buffet Onset	No	Wing-Tail Interference, Control Limitation
Direct Side Force/ Lift Control		Yes	Improved Aiming Accuracy, Ride Quality	Yes	Improved Aiming Accuracy, Ride Quality
Thrust/ Drag Control	Thrust Reversers	Tradeoff	Integration/ Redundancy With Jet Flap	Tradeoff	Integration/ Redundancy With Jet Flap
	Modulating Dive Brakes	Yes	Permits Higher Engine RPM Hold	Yes	Permits Higher Engine RPM Hold
CCV ● Flutter Suppression ● Maneuver Load Control ● Gust Load Alleviation		Yes	Reduced Weight/ Drag Potential	Yes	Reduced Weight/Drag Potential
Composite Primary and Secondary Structure		Yes	Reduced Weight	Yes	Reduced Weight

The bulk of the analysis of the use of jet flaps was done by Honeywell with engine, inlet and aerodynamic models provided by Lockheed and Pratt & Whitney.

GENERAL DESCRIPTION

The aircraft studied is shown in Figure 5. It is a single seat, high-performance, supersonic aircraft designed for air combat with an air-to-surface secondary role. It is powered by two Pratt & Whitney F-100 engines which have modulating jet flap nozzles along the trailing edges of the wings. A horizontal canard provides pitch control and counter balancing when the jet flaps are deployed.

The aircraft is statically unstable at M=0.9 and L=30,000 ft.

The flight control functions and their respective limits are indicated in Table II. In addition to the conventional pitch, roll and yaw controls, vertical chin canards are available for direct side force. The function and deflection limits for the surfaces are also shown in the table. The control deflection sign convention is chosen so that a positive deflection tends to produce a positive lift, side force, or positive rolling movement.

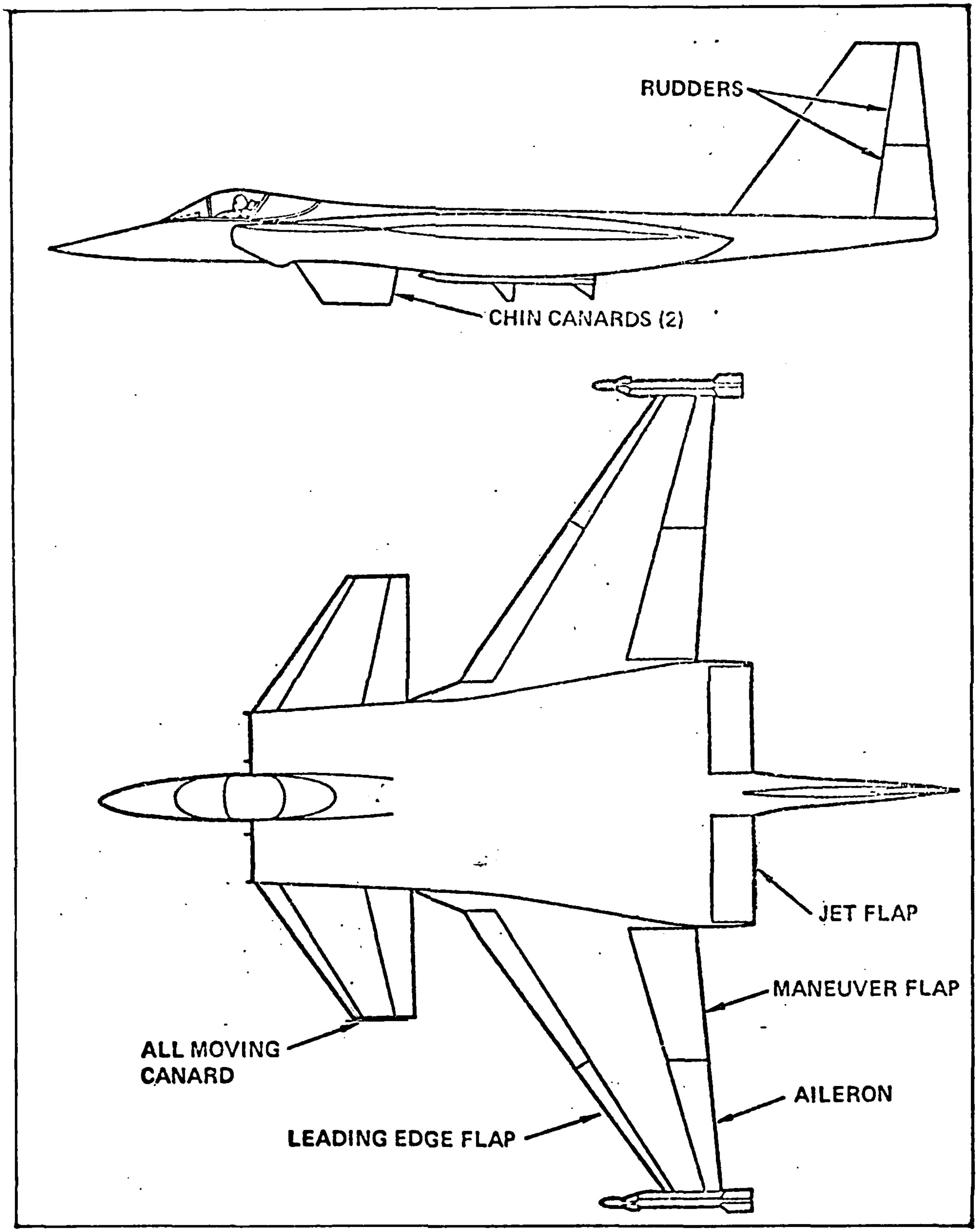


Figure 5. FPCC Aircraft - Top and Side Views

TABLE II

CONTROL SURFACE FUNCTIONS AND LIMITS

CONTROL	CONTROL	DEFLECTION LIMITS	
FUCTION	SURFACE	M<1.0	1.0<M
LIFT	MANEUVER FLAP	±15°	-
	L.E. FLAP	0° TO 20°	-
	JET FLAP	0° TO 90°	LIMITED
	CANARD	(SEE BELOW)	
SIDE FORCE	CHIN CANARDS	±35°	±15°
	LOWER RUDDER	(SEE BELOW)	
PITCH	CANARD	±20°	+5° TO -10°
	CANARD FLAP	±30°	±15°
YAW	UPPER RUDDER	±30°	LOCKED
	LOWER RUDDER	±30°	±15°
ROLL	AILERON (PER SIDE)	±20°	-

The horizontal canard is a variable camber surface having geared leading edge and trailing edge flaps. For the purposes of this study, they are assumed to be fixed. The two vertical chin canards are slab surfaces which are deflected in parallel. The ailerons and rudder are conventional surfaces.

FPCCSIM PROGRAM

Appendix A contains the longitudinal equations of motion for the aircraft. The scope of this study did not include the lateral dynamics. As can be seen from the equations in the appendix, there are several non-linear equations dealing with the supercirculation effect and its corresponding lift and drag. Obviously, these equations do not lend themselves to the usual methods of determining the equations of motion. In order to analyze the aircraft, the computer program FPCCSIM (Ref. 5) is used. FPCCSIM is a full nonlinear 6 DOF simulation. It transfers the block diagram into state equations for each of 10 flight conditions listed in Table III. This study uses the longitudinal data from flight conditions 7, 8, and 10. These correspond to a low altitude subsonic condition, high altitude transonic condition and a high altitude supersonic condition, respectively. The engine model used by the program had been transformed to a state equation format by Pratt & Whitney.

The FPCCSIM program is large; the source listing fills eight boxes of punched cards. Much of the program deals with the engine simulation. The program was modified by John E. Houtz of the Flight Dynamics Lab so that it would provide just the state equations and eigen values of the aircraft itself without the flight-control system (Ref. 6). This was necessary because the original program would provide a simulation complete with the engine dynamics and the flight control system inputs which would be useless for the purpose of this thesis. The revised program is much shorter and less time-consuming. The matrices for the three flight conditions are generated using a data deck similar to the one listed in Ref. 4. Different flight conditions are analyzed by switching a single parameter change card (PCC) in the deck.

The output of the program is in state equation format.

TABLE III. Flight Conditions and Trim Values

Flight Condition Number	Mach Number	Altitude (ft)	Dynamic Pressure (psf)	Engine Power Setting	Trim Values			
					Angle of Attack (deg)	Horizontal Canard Position (deg)	Engine* Thrust F1=F2 (lb)	Normal Load Factor (g's)
1	0.6	0	533	Inter-mediate	5.73	-6.76	11,270	5.51
2	0.6	0	533	Max A/B	8.60	-9.90	24,170	8.47
3	0.9	20,000	551	Max A/B	6.49	-5.88	16,930	8.04
4	0.9	30,000	356	Inter-mediate	4.39	-4.07	5,517	3.46
5	0.9	30,000	356	Max A/B	6.80	-6.14	12,040	5.46
6	2.3	40,000	1450	Max A/B	2.21	-1.00	19,970	3.47
7	0.6	0	533	--	1.07	-1.31	2,593	1.0
8	0.9	20,000	551	--	.83	- .79	2,609	1.0
9	0.9	30,000	356	--	1.29	-1 22	2,860	1.0
10	2.3	40,000	1450	--	.64	- .29	16,300	1.0

* Net thrust from each engine.

$$x = Ax + Bu \qquad (59)$$

$$y = Cx + Du \qquad (60)$$

For flight condition 7, Mach 0.6, zero ft., the state equations are:

$$\begin{bmatrix} \dot{q} \\ \dot{w} \\ \dot{u} \\ \dot{\theta} \\ \dot{h} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -0.463 & 0.0639 & -0.0011 & 0 & 0.374\text{x}10^{-6} & 0.471\text{x}10^{-6} \\ 674 & -2.82 & -0.0568 & -0.577 & -0.489\text{x}10^{-3} & -0.461\text{x}10^{-3} \\ -12.1 & 0.105 & -0.0279 & -32.1 & 0.674\text{x}10^{-4} & -0.734\text{x}10^{-4} \\ 1.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.00 & 0.018 & 669 & 0 & 0 \\ 0 & 0.018 & 1.00 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ w \\ u \\ \theta \\ h \\ x \end{bmatrix}$$

$$+ \begin{bmatrix} 0.640 & -0.0966 & -0.043 & -0.224\text{x}10^{-5} & -0.224\text{x}10^{-5} & 0 & 0 \\ -1.43 & -0.473 & -0.616 & -0.345\text{x}10^{-4} & -0.345\text{x}10^{-4} & 0 & 0 \\ 0.518 & 0.0198 & 0.923\text{x}10^{-3} & 0.109\text{x}10^{-2} & 0.109\text{x}10^{-2} & -387 & -387 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_c \\ \delta_{MF} \\ \theta_j \\ F_1 \\ F_1 \\ CDI_1 \\ CDI_2 \end{bmatrix} \qquad (61)$$

$$\begin{bmatrix} \alpha \\ VEL \\ A_{CG_X} \\ A_{CG_Z} \end{bmatrix} = \begin{bmatrix} 0 & 0.149\text{x}10^{-2} & -0.268\text{x}10^{-4} & 0 & 0 & 0 \\ 0 & 0.0179 & 1.0 & 0 & 0 & 0 \\ -0.0713 & 0.105 & -0.297\text{x}10^{-1} & 0 & 0.674\text{x}10^{-4} & -0{,}734\text{x}10^{-4} \\ 4.08 & -2.82 & -0.568\text{x}10^{-1} & 0 & -0.489\text{x}1o^{-3} & -0.461\text{x}10^{-3} \end{bmatrix} \begin{bmatrix} q \\ w \\ u \\ \theta \\ h \\ x \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.518 & 0.0198 & 0.923\text{x}10^{-3} & 0.109\text{x}10^{-2} & 0.109\text{x}10^{-2} & -387 & -387 \\ 1.43 & -4.73 & -0.616 & -0.345\text{x}10^{-4} & 0.345\text{x}10^{-4} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta c \\ \delta_{MF} \\ \theta_j \\ F_1 \\ F_2 \\ CDI_1 \\ CDI_2 \end{bmatrix} \qquad (62)$$

where

δ_C = canard deflection; degrees

δ_{MF} = maneuver flap deflection; degrees

θ_j = jet flap deflection; degrees

$F_{1,2}$ = net thrust, engine 1, 2; lb.

$CDI_{1,2}$ = inlet drag coefficient; engine 1, 2

α = angle of attack; radians

VEL = total airspeed; ft./sec.

ACG_x = accel of C.G along x-body axis; ft$_1$/sec$_2$

ACG_z = accel of C.G along z-body axis; ft/sec$_2$

q = body axis pitch rate; rad/sec

w = z-body axis velocity; ft/sec

u = x-body axis velocity; ft/sec

θ = pitch angle; radians

h = altitude; ft.

x = down range position referenced to initial body axis; ft.

All variables are perturbation quantities from the equilibrium point.

As discussed in Chapter II, the design methodology for known plants for synthesizing the flight control system (Ref. 2) has no provision for a feed forward matrix, D. This can be dealt with by either transforming the equations into a new state space that includes actuator dynamics, or by reconfiguring the output matrix to include terms that do not contain feed-forward terms. The latter route is chosen after the former method required a large measurement matrix that was 3 X 6. There is little guidance in choosing the elements of a large measurement matrix. The two engine thrusts are combined into one input, F total. The CDI terms are dropped on the assumption that the inlet drag does not change significantly at each flight condition. The jet flaps and maneuver flaps are geared together so that full deflection occurs simultaneously -15° for maneuver flaps and 75° for the jet flaps. This is done because the B_2 matrix, with four inputs, does not have full rank. These two surfaces provided the same input, large lift with small negative pitching moment, and were therefore redundant. The output matrix consists of flight path angle, pitch angle and forward velocity. The resulting simplified 6 state, 3 input, 3 output euations for flight condition #7 are:

$$\begin{bmatrix} \dot{x} \\ \dot{h} \\ \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0.018 & 0 \\ 0 & 0 & 669 & 0.018 & -1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ -0.734\text{x}10^{-4} & 0.674\text{x}10^{-4} & -32.1 & -0.0297 & 0.105 & -12.1 \\ -0.461\text{x}10^{-4} & -0.489\text{x}10^{-3} & -0.557 & -0.0568 & -2.82 & 674 \\ 0.471\text{x}10^{-6} & 0.374\text{x}10^{-6} & 0 & -0.11\text{x}10^{-2} & 0.0639 & -0.463 \end{bmatrix} \begin{bmatrix} x \\ h \\ \theta \\ u \\ w \\ q \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.518 & 0.4223\text{x}10^{-3} & 0.218\text{x}10^{-2} \\ -1.43 & -1.404 & -0.69\text{x}10^{-4} \\ 0.64 & -0.0591 & -0.448\text{x}10^{-5} \end{bmatrix} \begin{bmatrix} \delta_c \\ \delta_{LF} \\ F_{TOTAL} \end{bmatrix} \tag{63}$$

$$\begin{bmatrix} \theta \\ \gamma \\ \mathrm{VEL} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0.268\mathrm{x}10^{-4} & -0.149\mathrm{x}10^{-2} & 0 \\ 0 & 0 & 0 & 1.0 & 0.0179 & 0 \end{bmatrix} \begin{bmatrix} x \\ h \\ \theta \\ u \\ w \\ q \end{bmatrix}$$

where

(64)

γ = flight path angle; radians

δ_{LF} = lift surface deflection, geared combination of jet flap and maneuver flap; degrees

F total = total net thrust; lb.

The same approach is used for flight condition #9, high altitude transonic cruise, Mach. 0.9, 30,000 ft. The equations are:

$$\begin{bmatrix} \dot{x} \\ \dot{h} \\ \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0.0219 & 0 \\ 0 & 0 & 895 & 0.0219 & -1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.61\mathrm{x}10^{-4} & -0.165\mathrm{x}10^{-3} & -32.1 & -0.0822 & 0.0472 & -19.7 \\ -0.581\mathrm{x}10^{-3} & -0.665\mathrm{x}10^{-3} & -0.705 & -0.0558 & -1.68 & 898 \\ 0.52\mathrm{x}10^{-6} & -0.939\mathrm{x}10^{-5} & 0 & -0.317\mathrm{x}10^{-2} & 0.0303 & -0.253 \end{bmatrix} \begin{bmatrix} x \\ h \\ \theta \\ u \\ w \\ q \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.359 & -0.0479 & 0.218\mathrm{x}10^{-2} \\ -0.634 & -2.007 & -0.744\mathrm{x}10^{-5} \\ 0.527 & -0.065 & -0.484\mathrm{x}10^{-5} \end{bmatrix} \begin{bmatrix} \delta_C \\ \delta_{LF} \\ F_{TOTAL} \end{bmatrix}$$

(65)

$$\begin{bmatrix} \theta \\ \gamma \\ VEL \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0.24\times10^{-4} & -0.112\times10^{-2} & 0 \\ 0 & 0 & 0 & 1.0 & 0.0219 & 0 \end{bmatrix} \begin{bmatrix} x \\ h \\ \theta \\ u \\ w \\ q \end{bmatrix} \qquad (66)$$

The equations for flight condition #10, high altitude supersonic cruise, Mach 2.3, 40,000 ft., are:

$$\begin{bmatrix} \dot{x} \\ \dot{h} \\ \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0.0111 & 0 \\ 0 & 0 & 2220 & 0.0111 & -1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ -0.842\times10^{-3} & -0.632\times10^{-3} & -32.1 & -0.0287 & -0.0113 & -24.7 \\ -0.764\times10^{-3} & -0.77\times10^{-3} & -0.356 & -0.441\times10^{-2} & -1.41 & 2230 \\ 0 & 0 & 0 & -0.205\times10^{-4} & 0.0123 & -1.67 \end{bmatrix} \begin{bmatrix} x \\ h \\ \theta \\ u \\ w \\ q \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.96 & -0.0853 & 0.218\times10^{-2} \\ -9.35 & -4.841 & 0 \\ 1.05 & -0.174 & 0 \end{bmatrix} \begin{bmatrix} \delta_C \\ \delta_{LF} \\ F_{TOTAL} \end{bmatrix} \qquad (67)$$

$$\begin{bmatrix} \theta \\ \gamma \\ \mathrm{VEL} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0.497\times10^{-5} & -0.449\times10^{-3} & 0 \\ 0 & 0 & 0 & 1.0 & 0.0111 & 0 \end{bmatrix} \begin{bmatrix} x \\ h \\ \theta \\ u \\ w \\ q \end{bmatrix} \qquad (68)$$

CHAPTER V

CONTROL LAW FOR FPCC A/C

INTRODUCTION

The longitudinal tracker developed for the FPCC aircraft is based on a 4-state model. This model reduction results from an inability to choose a measurment matrix for the 6-state models developed in Chapter IV. The original design is accomplished using flight condition #9 - Mach 0.9, 30,000 ft. (which is considered to be) the most critical of the three flight conditions under study. Thus, by applying the control law for this transonic flight condition to the subsonic (flight condition #7) and the supersonic (flight condition #10) conditions, a rigorous test of Porter's claim of robustness can be made. His claim that fast sampling, error-actuated control is robust enough to handle wide parameter changes in the plant is supported by the results obtained in this chapter. In hindsight, however, it appears that the most critical flight condition may have been the subsonic condition, based on the velocity tracking results.

Three command vectors are chosen: positive pitch pointing, vertical translation, and straight climb. The positive pitch pointing command vector is

$$v^T = [0.1,0,0]^T \qquad (69)$$

This is a 0.1 radian positive pitch angle with no change in flight path angle and no change in velocty.

The vertical translation command vector consisted of:

$$v^T = [0,0.1,0]^T \qquad (70)$$

This represents a 0.1 radian positive flight path angle with no change in pitch angle nor velocity.

The straight climb command vector consisted of:

$$v^T = [0.1,0.1,0]^T \qquad (71)$$

This represents a combination of a 0.1 radian pitch angle coupled with a 0.1 radian flight path angle. This effectively regulates angle of attack to zero. Velocity change is regulated to zero.

TRANSONIC FLIGHT CONDITION

The four state plant and output equations for the transonic flight condition, Mach 0.9, 30,000 ft. is as follows:

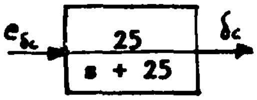
$$\begin{bmatrix} \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1.0 \\ -32.1 & -0.0822 & 0.0472 & -19.7 \\ -0.705 & -0.0558 & -1.68 & 898 \\ 0 & -0.317\text{x}10^{-2} & 0.0303 & -0.253 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0.359 & -0.0479 & 0.218\text{x}10^{-2} \\ -0.634 & -2.007 & -0.744\text{x}10^{-4} \\ 0.527 & -0.065 & -0.484\text{x}10^{-5} \end{bmatrix} \begin{bmatrix} \delta_C \\ \delta_{LF} \\ F_{TOTAL} \end{bmatrix} \qquad (72)$$

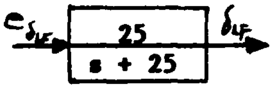
$$\begin{bmatrix} \theta \\ \gamma \\ VEL \end{bmatrix} = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 1.0 & 0.245\text{x}10^{-4} & -0.112\text{x}10^{-2} & 0 \\ 0 & 1.0 & 0.0219 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix} \qquad (73)$$

The servos are chosen as first-order lags, based on the design specifications from Lockheed for the FPCC aircraft. The engine was modeled as a first-order lag based on the Pratt & Whitney test data (Ref. 16). They are as follows:

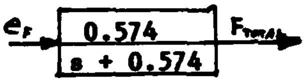
CANARD



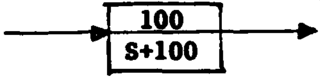
LIFT FLAP



ENGINE



All sensors are modeled as first-order lags as:



The design parameters are chosen as:

SAMPLING TIME = .01 sec

ALPHA = 2.5

SIGMA MATRIX = [1,.5,.03]
DIAGONAL ELEMENTS

EPSILON = 0.1

MEASUREMENT = $[.2,0,0]^T$
MATRIX

These design parameters are the result of a lengthy, iterative design study. The rationale behind the choice of these parameters is included in Chapter VI. These parameters result in an excellent combination of input and output responses for all three command vectors with respect to input and output figures of merit. Figures of merit for the outputs include overshoot, rise time and settling time. The figures of merit for the inputs were the maximum deflections and number of oscillations (# of peaks).

The controller matrices formed from the design parameters are:

$$K_o = \begin{bmatrix} 2.281 & 6.604 & 0.554\text{X}10^{-5} \\ -0.706 & 53.45 & -0.218\text{X}10^{-3} \\ -391.2 & 1208 & 3.433 \end{bmatrix} \qquad (74)$$

$$K_1 = \begin{bmatrix} 0.9125 & 2.642 & 0.221\text{X}10^{-5} \\ -0.2825 & 21.38 & -0.843\text{X}10^{-4} \\ -156.5 & 483 & 1.373 \end{bmatrix} \qquad (75)$$

The input and output responses are shown in Fig. 6a to Fig. 6e for the positive pitch pointing command. Theta (pitch angle) and gamma (flight path angle) are combined on a single graph since these two outputs are of prime importance. Theta reaches a steady state value of 0.1 radian while gamma approaches zero. Velocity perturbation has a minimum of -2.5 ft/sec and a maximum of 1.3 ft/sec. This perturbation is from a nominal velocity of 830 ft/sec, and is hardly noticeable. The steady-state values for the inputs are:

CANARD: -14°
LIFT FLAP: -70°

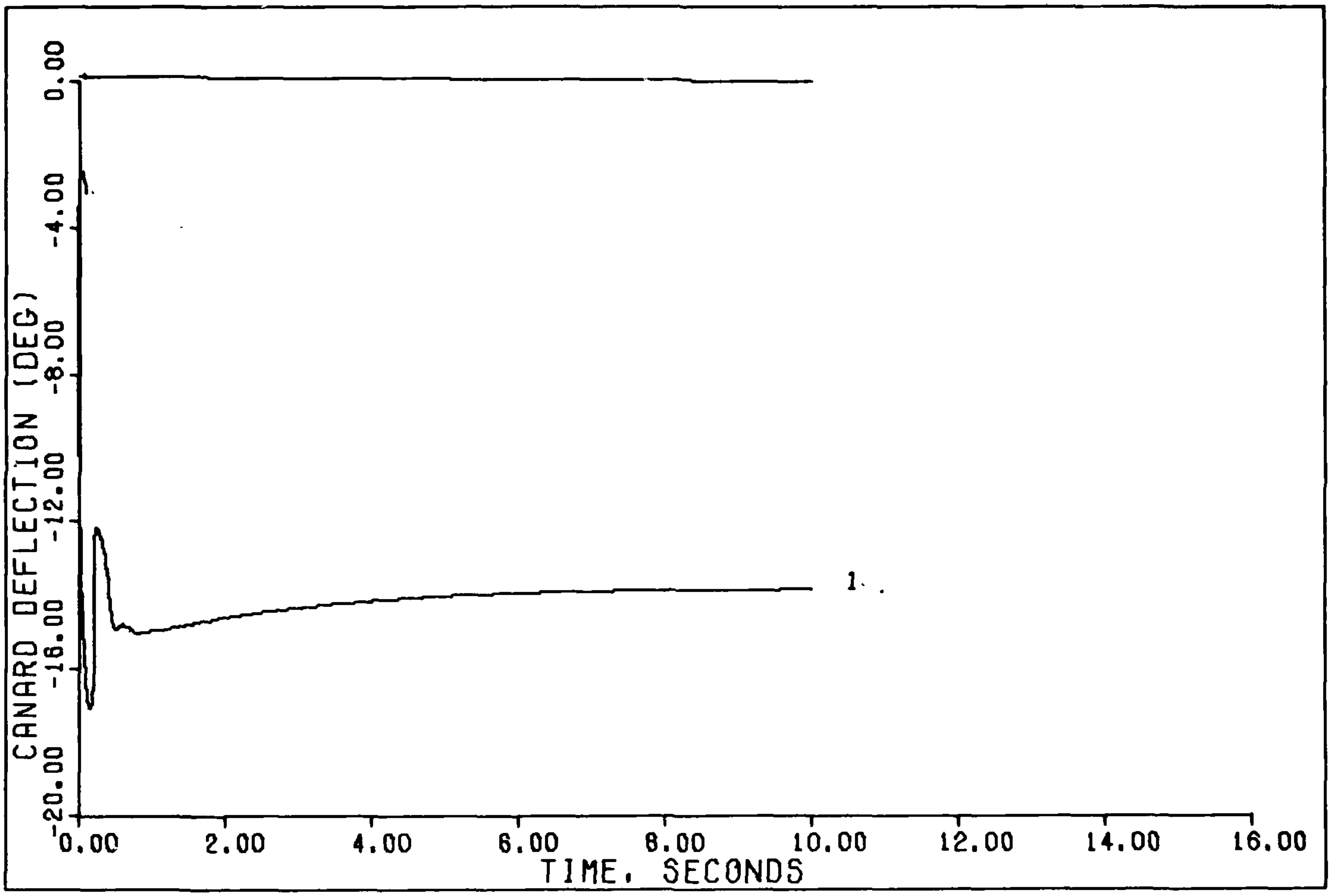


Figure 6a. Canard Deflection, Positive Pitch Pointing Command (Transonic Flight Condition)

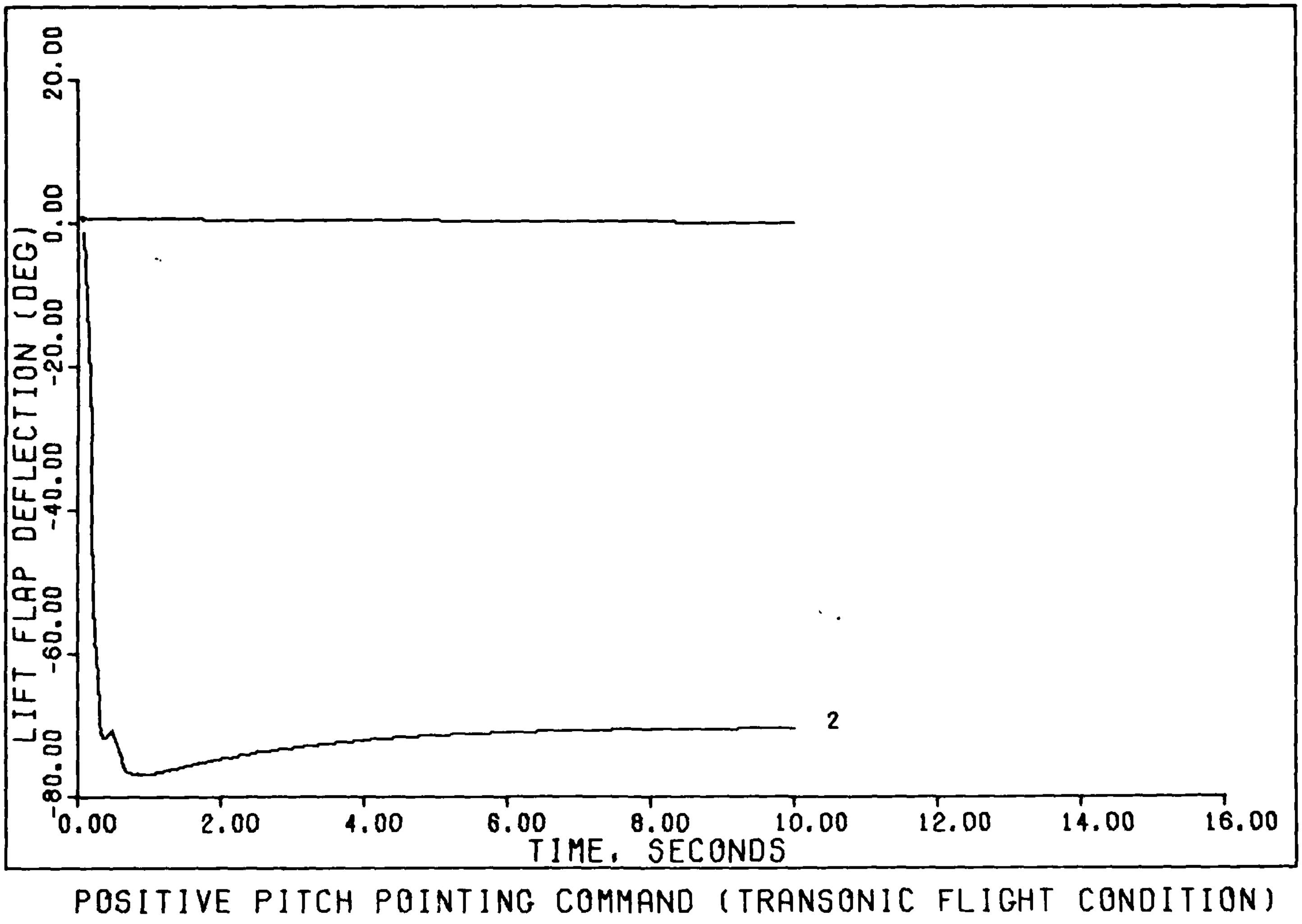


Figure 6b. Lift Flap Deflection

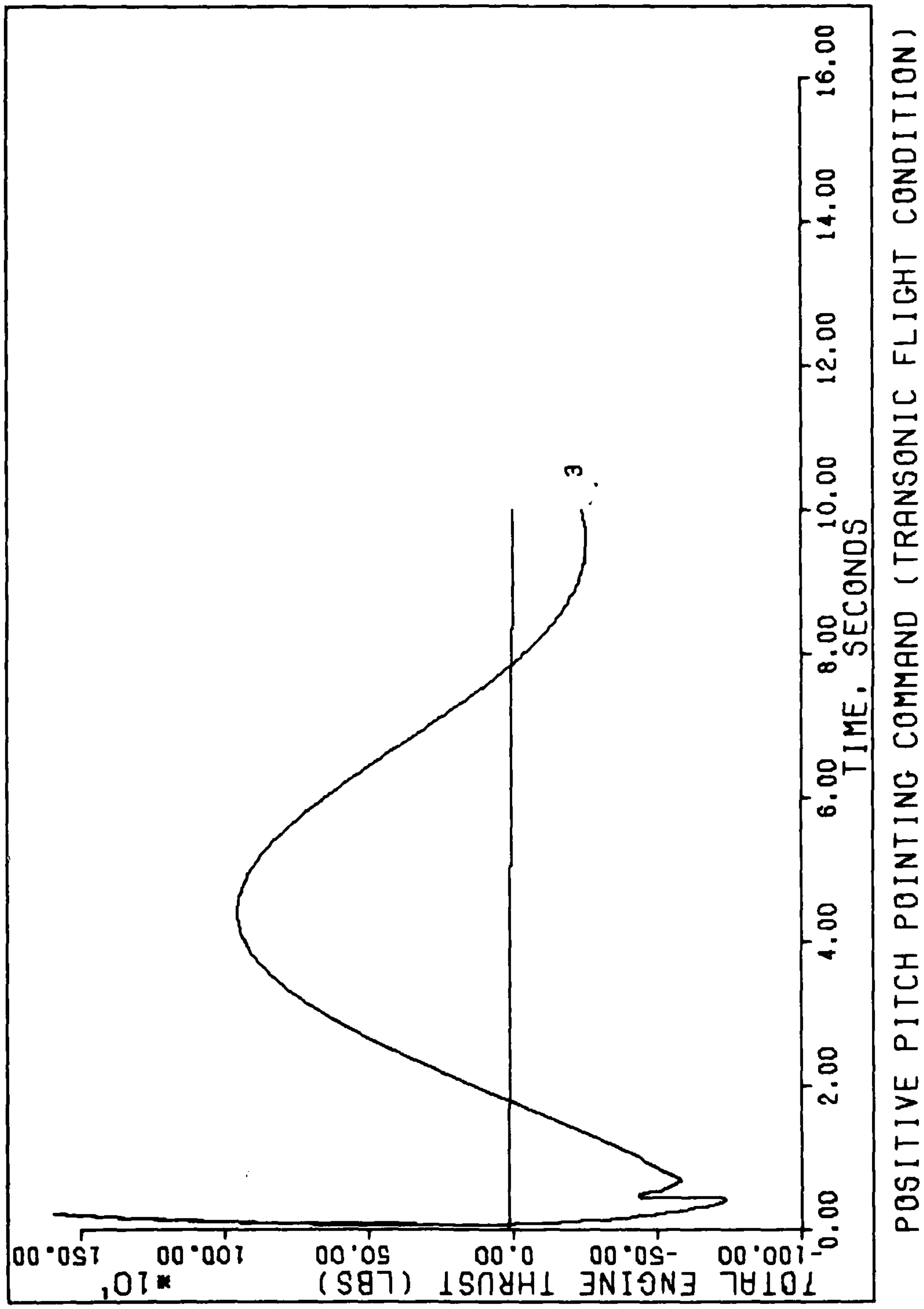


Figure 6c. Total Engine Thrust

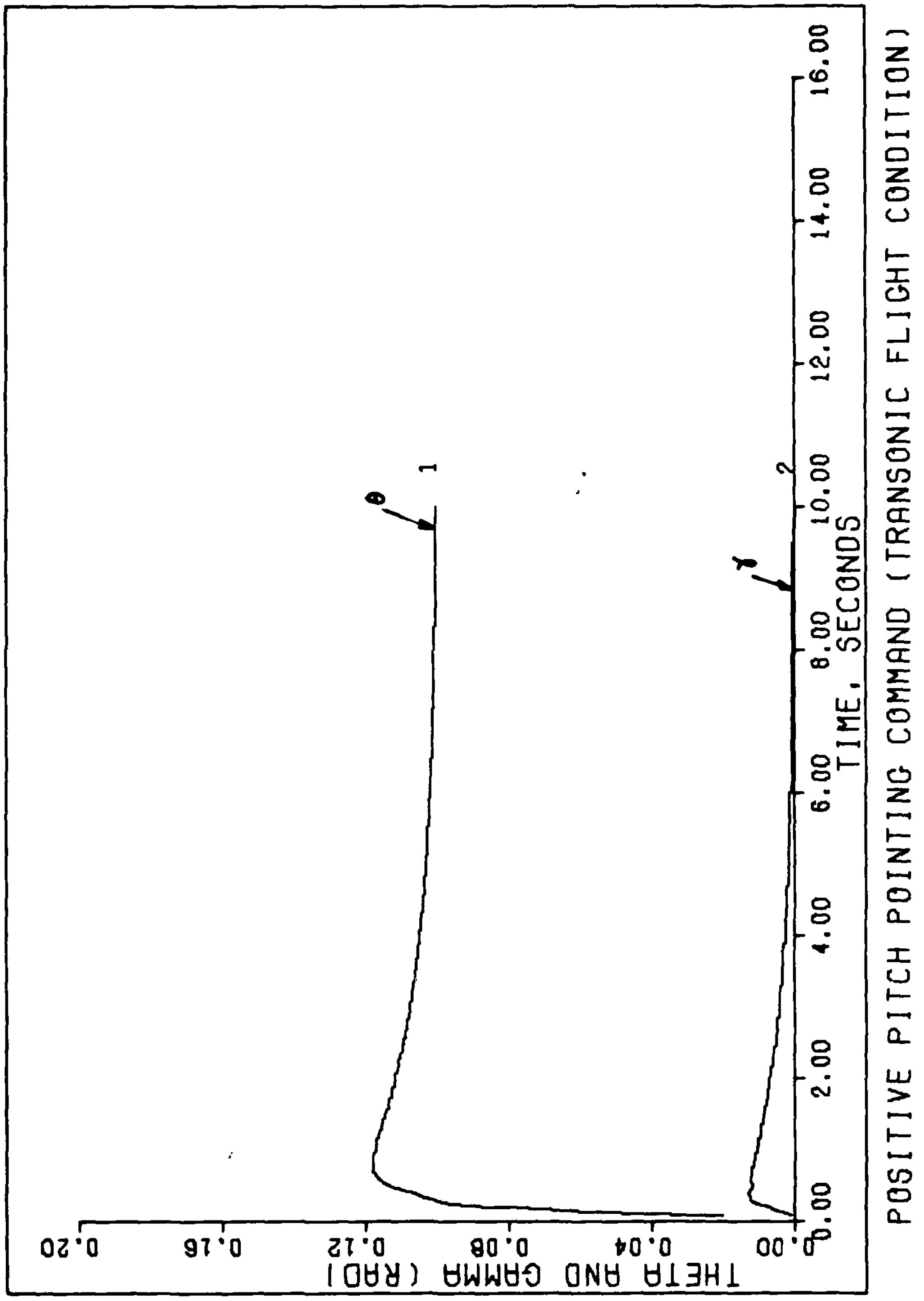
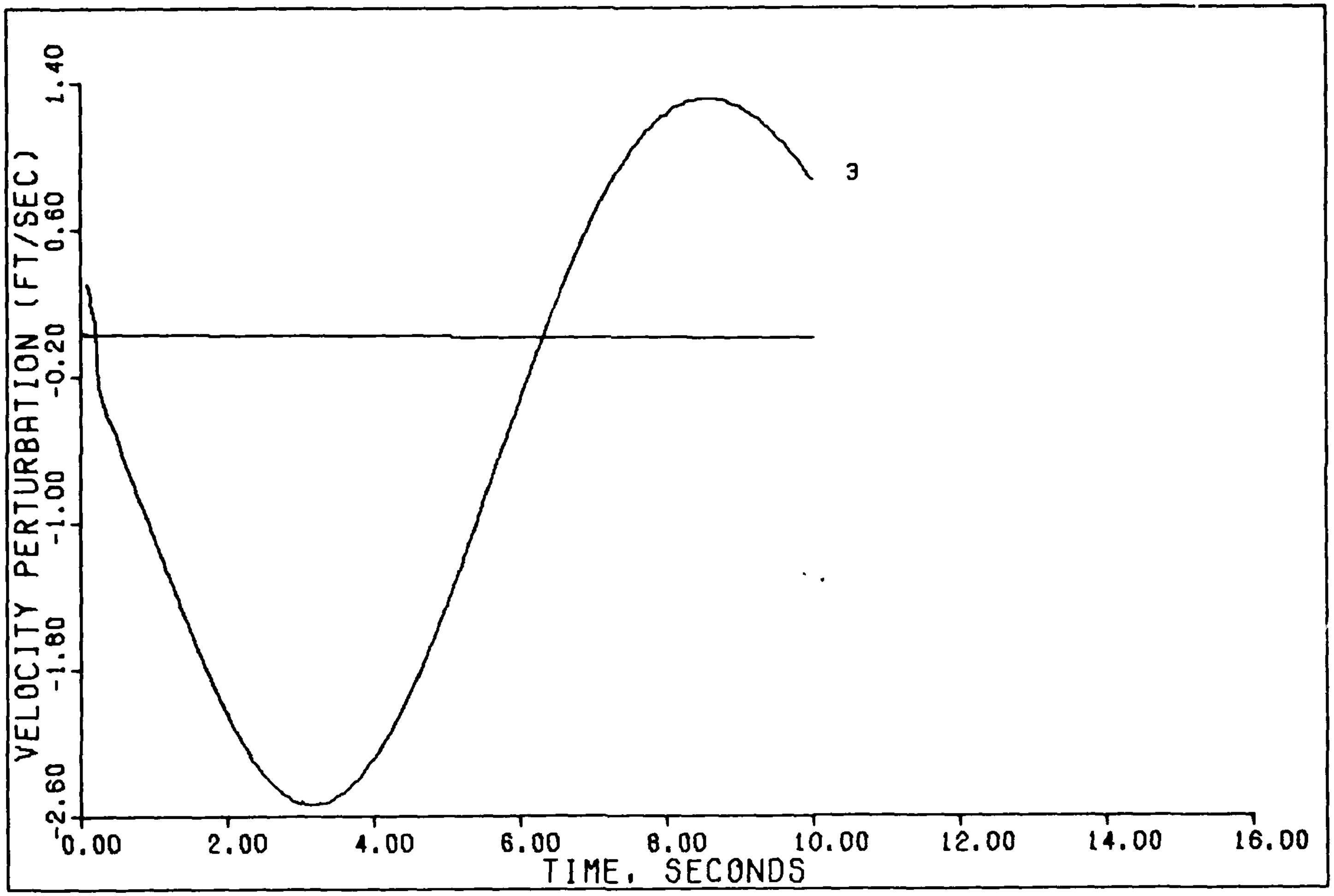


Figure 6d. Pitch and Flight Path Angles



POSITIVE PITCH POINTING COMMAND (TRANSONIC FLIGHT CONDITION)

Figure 6e. Velocity Perturbation

The engine is still oscillatory at T=10 sec.

As seen in the lift flap response, a negative steady state angle for the lift flap is needed to achieve positive pitch pointing. This is needed to counteract the additional lift generated by maintaining an abnormal angle of attack. The resultant positive pitch moment is countered by downloading the canard. This is a feasible method to achieve positive pointing except that the jet flap (geared together with the maneuver flap to form a lift flap) can not have a negative deflection. If the point-of-view is taken that this aircraft is still in the design stage, these results would dictate the need for a jet flap capable of positive and negative deflections.

The vertical translation command vector results in the inputs and outputs shown in Fig. 7a through Fig. 7e. In this mode, gamma is commanded to 0.1 radian with theta regulated to zero. Velocity perturbation has a minimum of -2.4 ft/sec and a maximum of 1.4 ft/sec. The steady state values of the inputs are:

CANARD: 14°

LIFT FLAP: 70°

The engine is still oscillatory at T=10 secs.

The straight climb vector results in the input an outputs shown in Fig. 8a through Fig. 8e. Both theta and gamma track to 0.1 radian velocity perturbation has a minimum of -4.1 ft/sec and a maximum of 2 ft/sec. The steady values for the input are:

CANARD: 0°

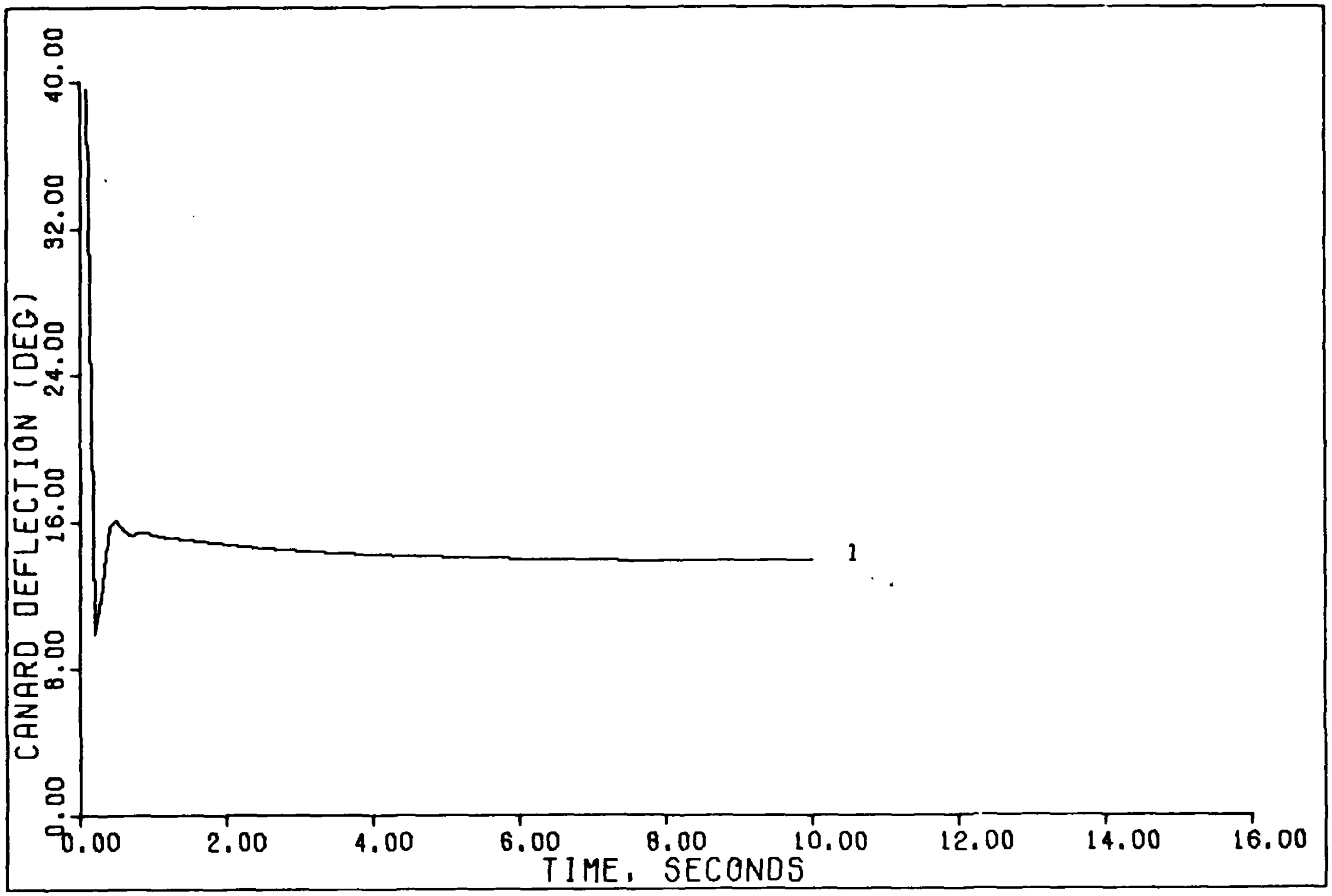
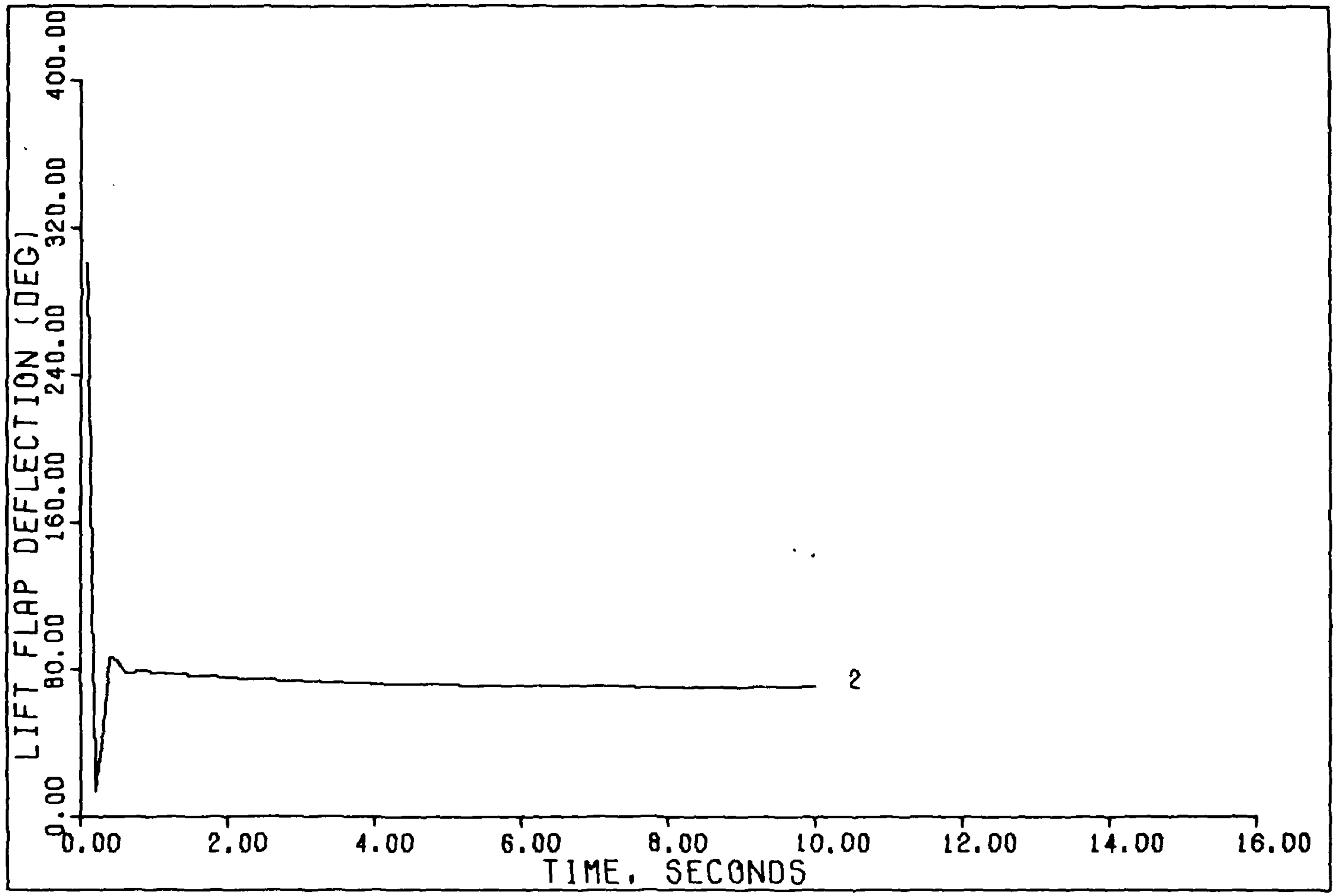
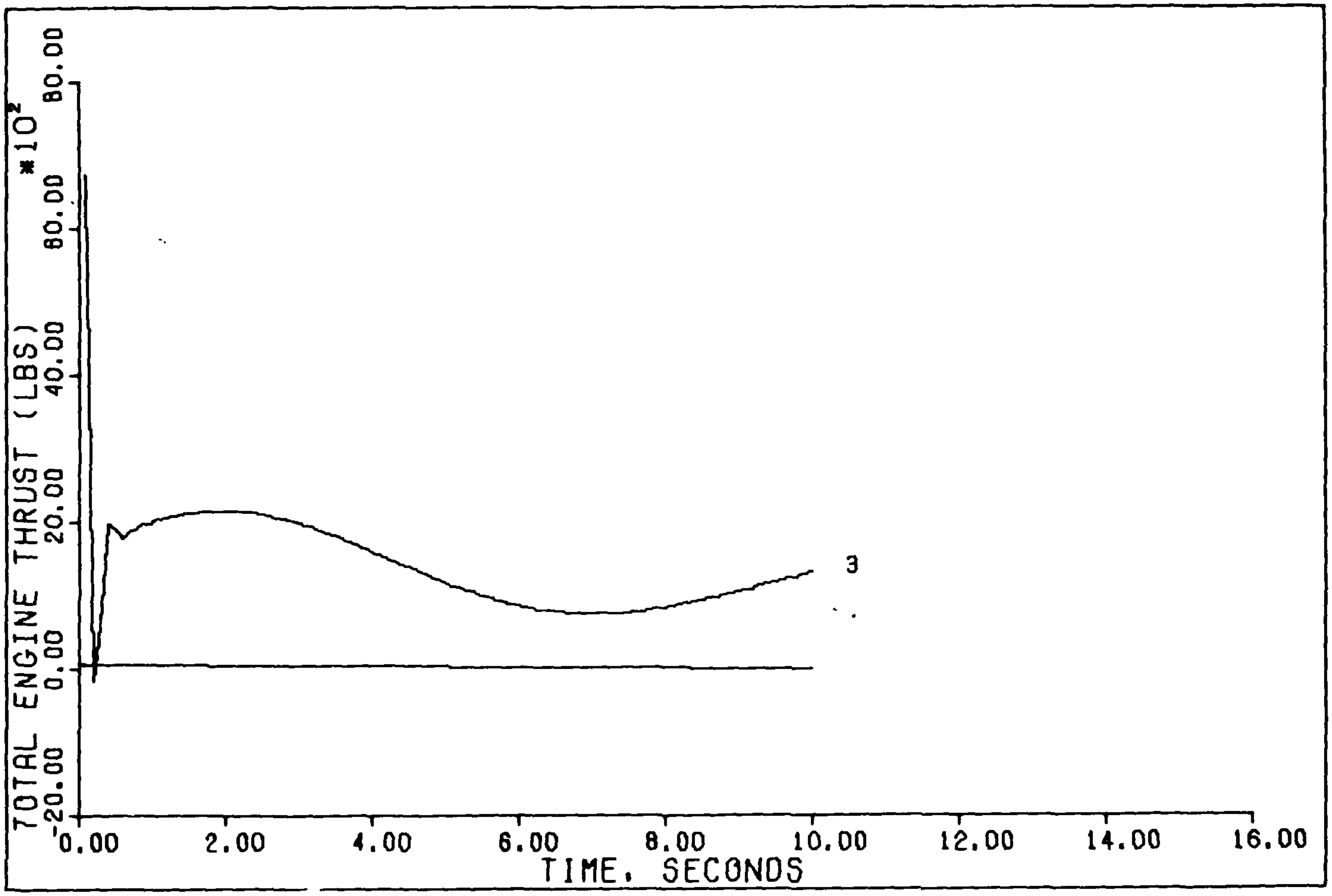


Figure 7a. Canard Deflection, Vertical Translation Command (Transonic Flight Condition)



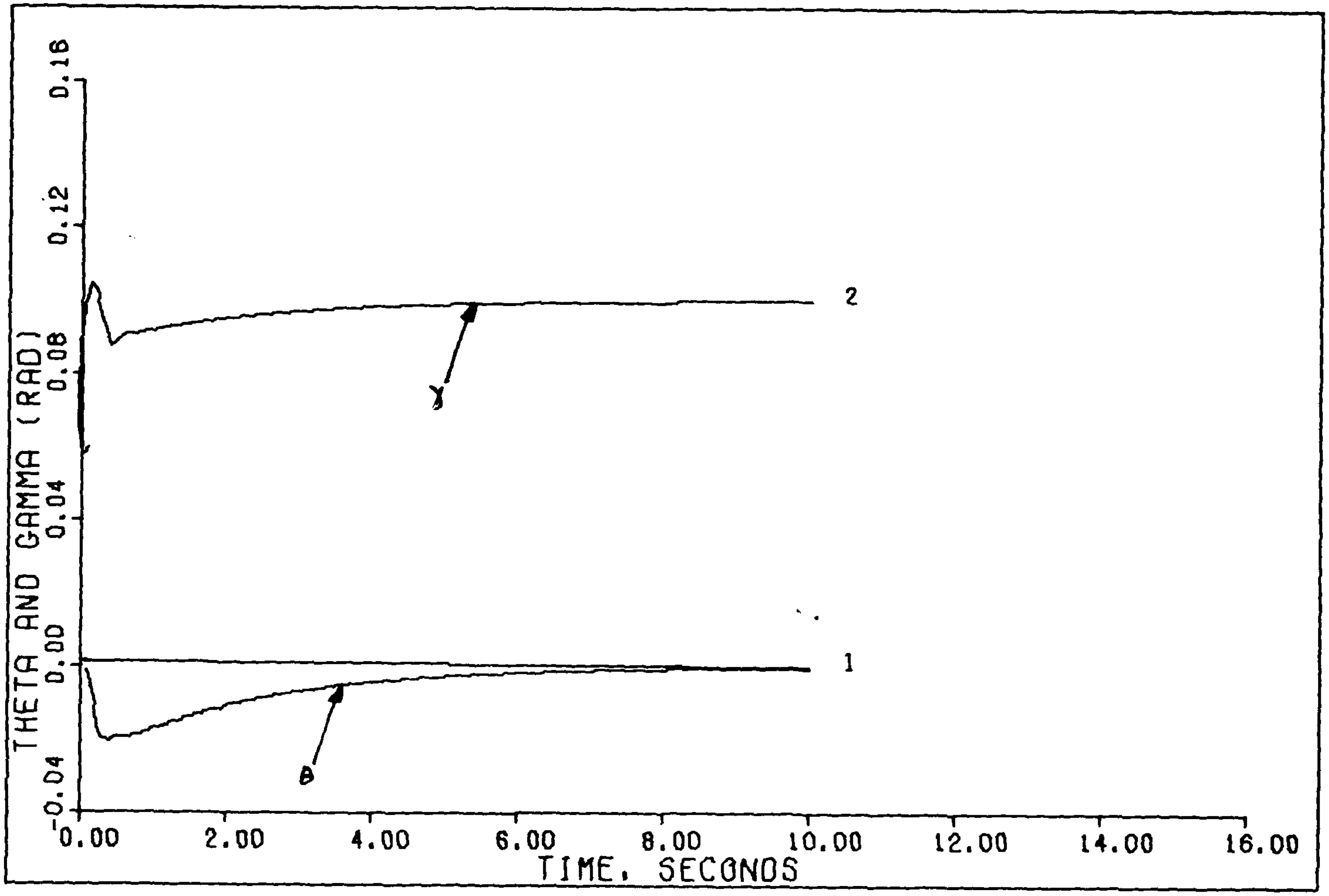
VERTICAL TRANSLATION COMMAND (TRANSONIC FLIGHT CONDITION)

Figure 7b. Lift Flap Deflection



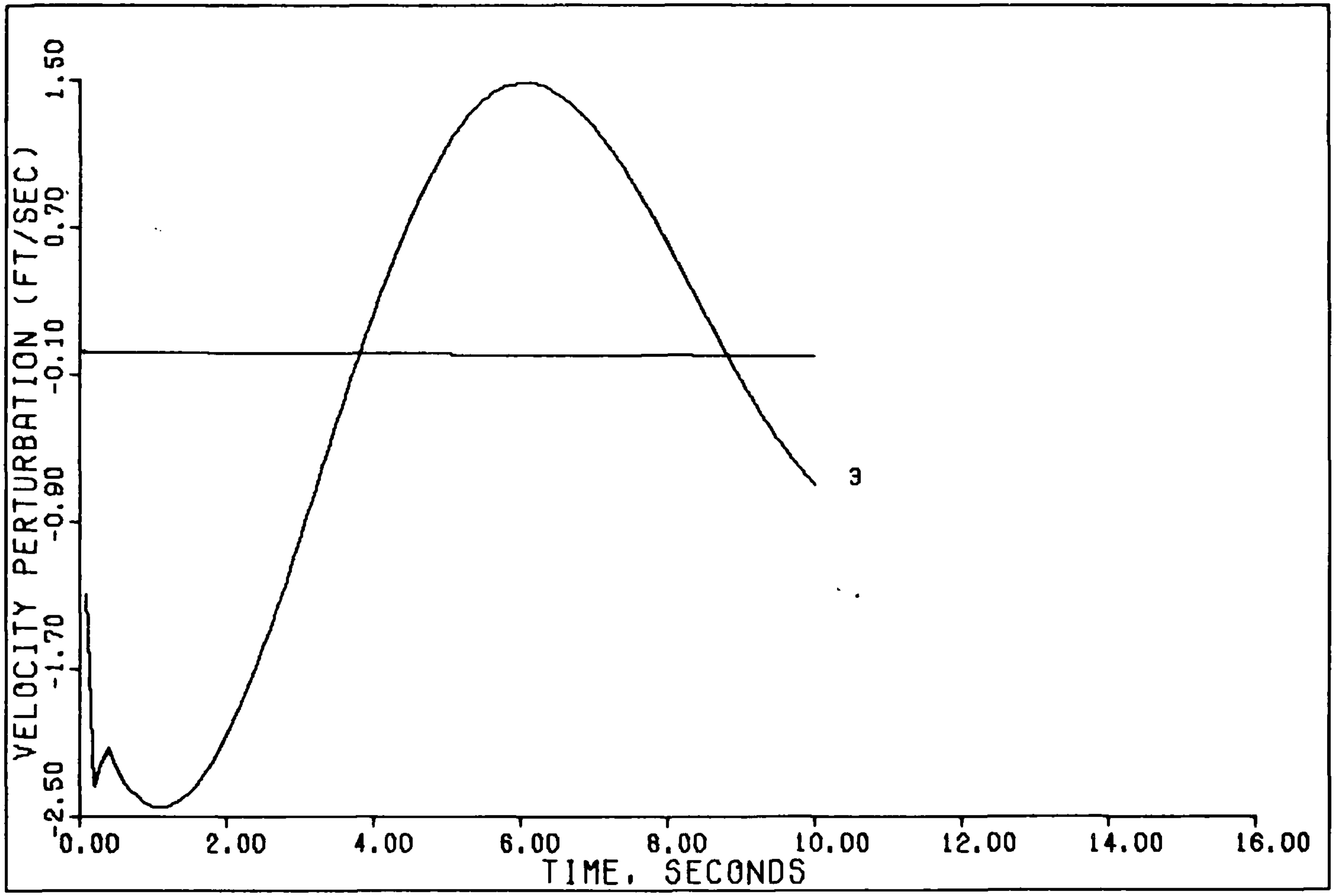
VERTICAL TRANSLATION COMMAND (TRANSONIC FLIGHT CONDITION)

Figure 7c. Total Engine Thrust



VERTICAL TRANSLATION COMMAND (TRANSONIC FLIGHT CONDITION)

Figure 7d. Pitch and Flight Path Angles



VERTICAL TRANSLATION COMMAND (TRANSONIC FLIGHT CONDITION)

Figure 7e. Velocity Perturbation

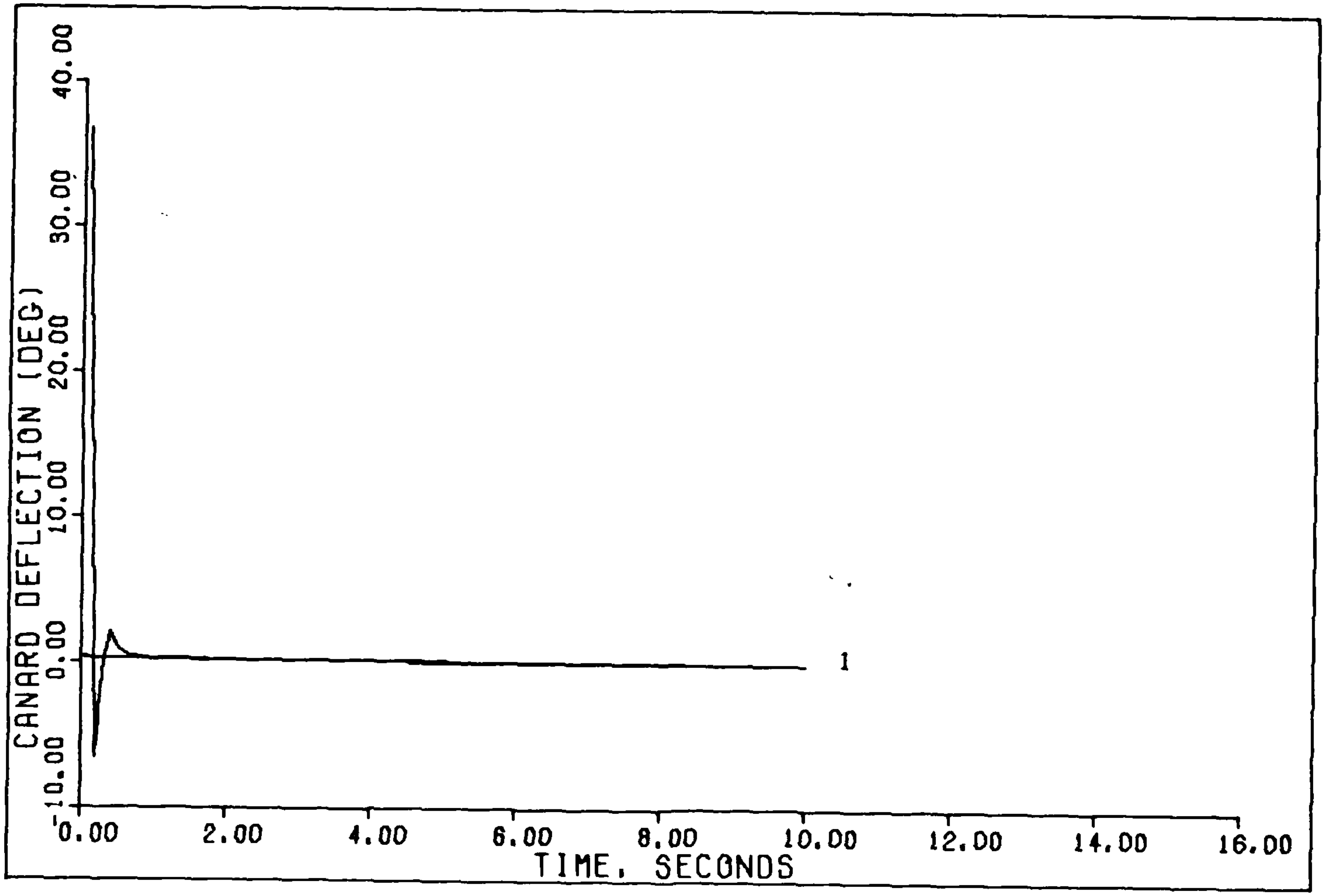


Figure 8a. Canard Deflection, Straight Climb Command (Transonic Flight Condition)

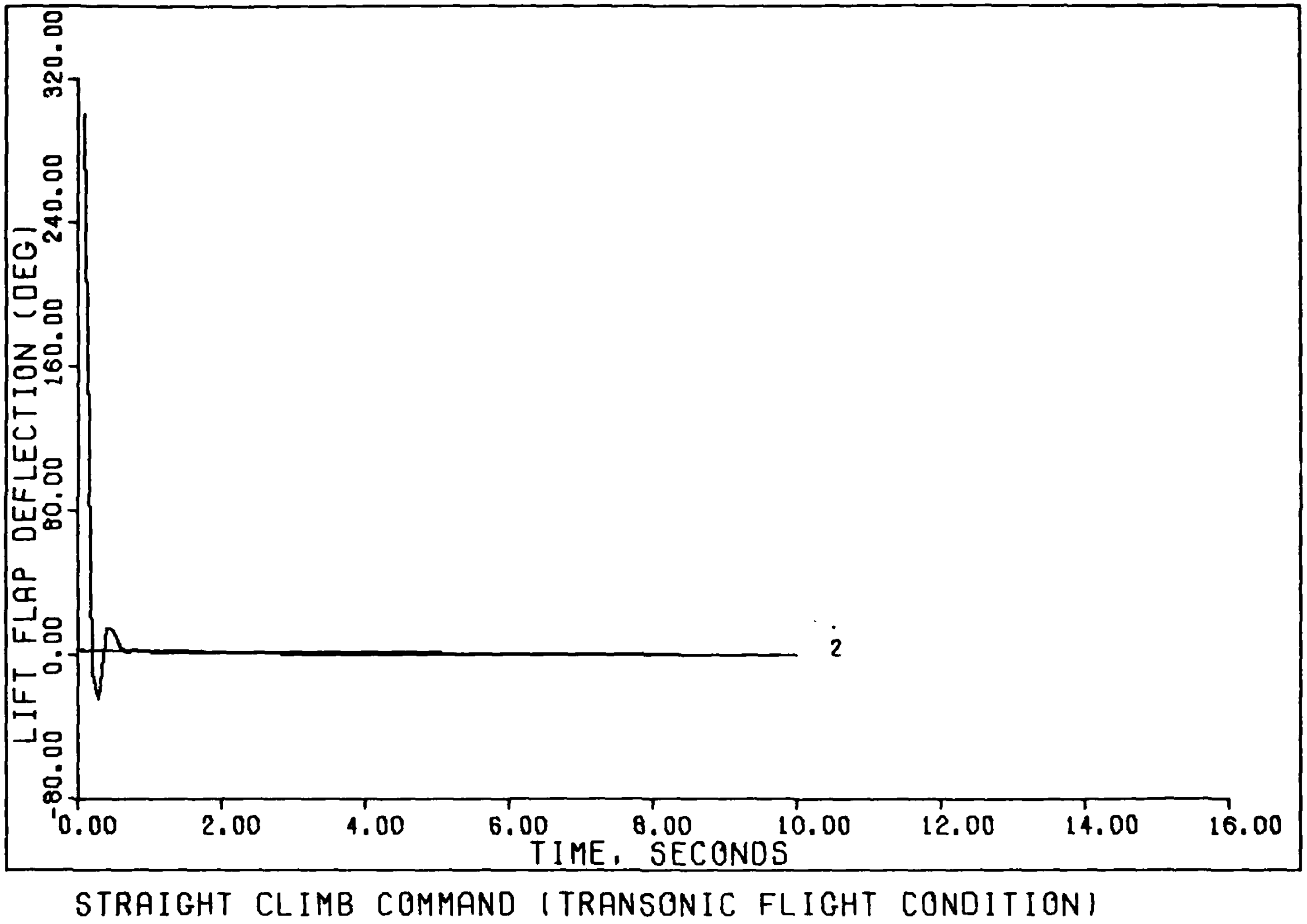
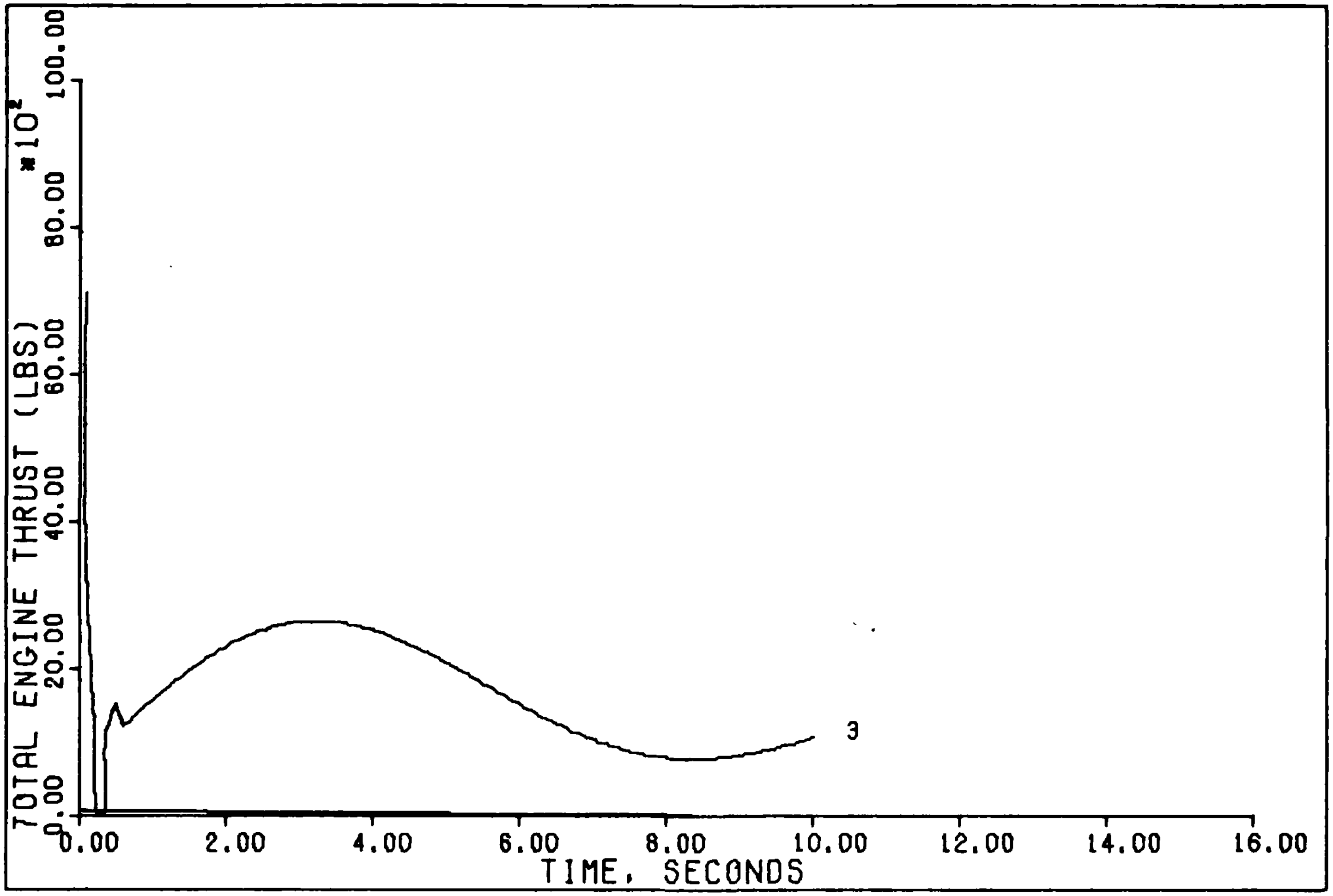


Figure 8b. Lift Flap Deflection



STRAIGHT CLIMB COMMAND (TRANSONIC FLIGHT CONDITION)

Figure 8c. Total Engine Trust

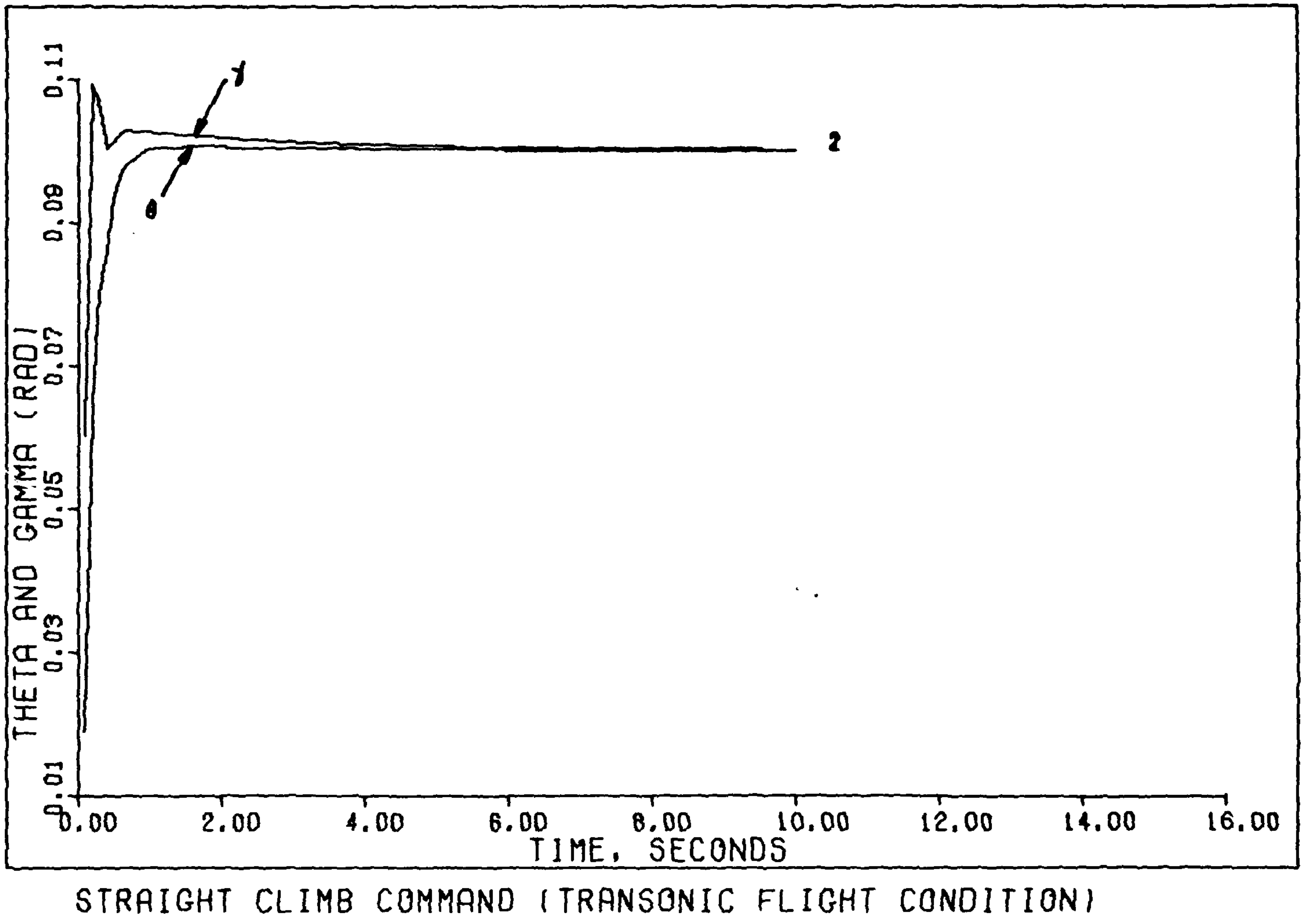


Figure 8d. Pitch and Flight Path Angles

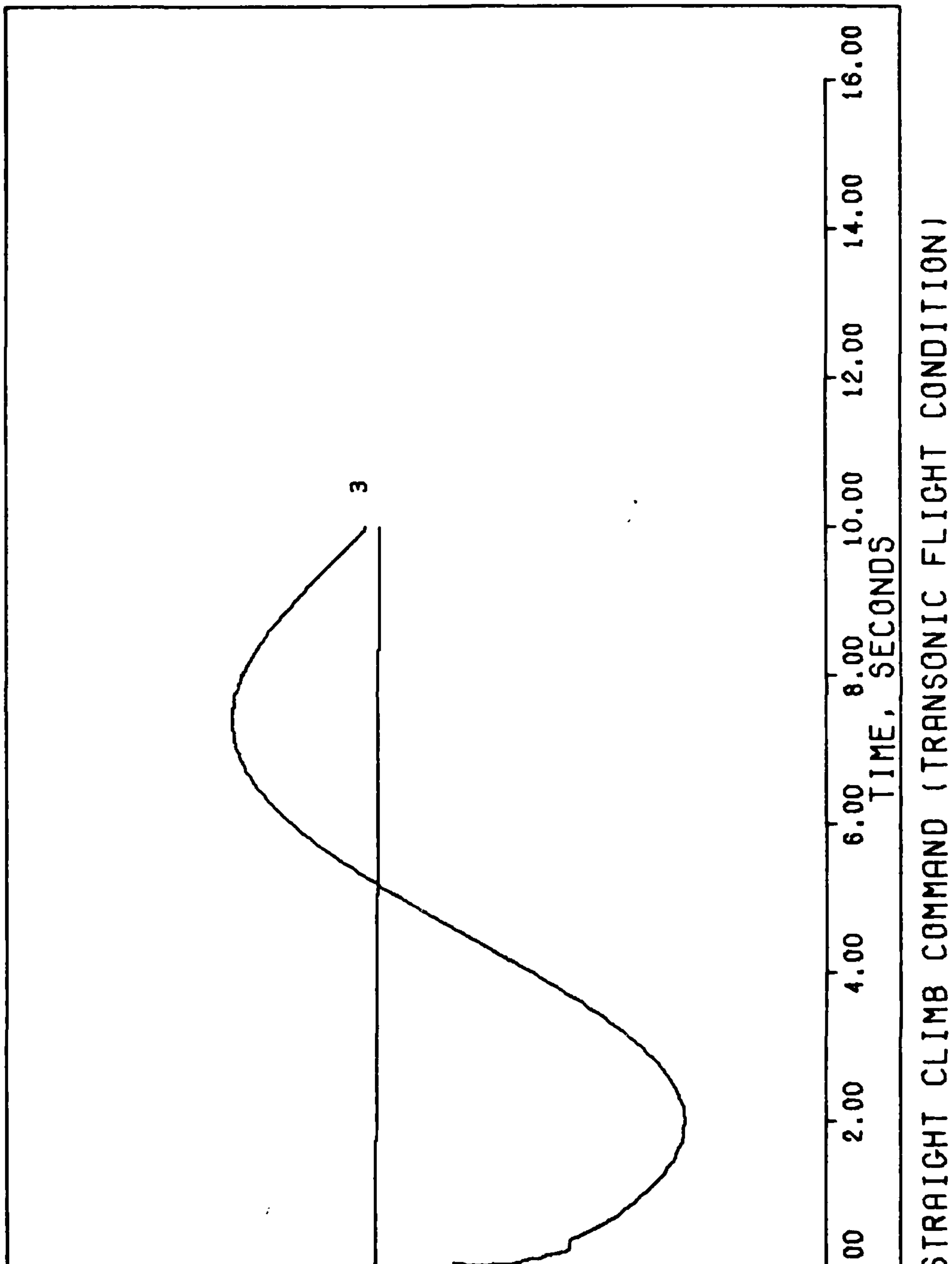


Figure 8e. Velocity Perturbation

LIFT FLAP: 0^{o}
ENGINE: 1300 lb.

All pertinent figures of merit for the three command vectors are presented in Table IV for this flight condition.

SUBSONIC FLIGHT CONDITION

The four state plant and output equations for flight condition #7 - Mach 0.6, zero ft. are as follows:

$$\begin{bmatrix} \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1.0 \\ -32.1 & -0.0297 & 0.105 & -12.1 \\ -0.557 & -0.0568 & -2.82 & 674 \\ 0 & -0.111\text{x}10^{-2} & 0.0639 & -0.463 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0.518 & 4.223\text{x}10^{-3} & 0.218\text{x}10^{-2} \\ -1.43 & -1.404 & -0.69\text{x}10^{-4} \\ 0.64 & -0.0591 & -0.448\text{x}10^{-5} \end{bmatrix} \begin{bmatrix} \delta_{c} \\ \delta_{LF} \\ F_{TOTAL} \end{bmatrix} \qquad (76)$$

$$\begin{bmatrix} \theta \\ \gamma \\ VEL \end{bmatrix} = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 1.0 & 0.268\text{x}10^{-4} & -0.149\text{x}10^{-2} & 0 \\ 0 & 1.0 & 0.0179 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix} \qquad (77)$$

TABLE IV FPCC CONTROL LAW-THREE FLIGHT CONDITIONS

COMMAND VECTOR	INPUT OR OUTPUT	FIGURES OF MERIT	TRANSONIC M=0.9,h=30k	SUBSONIC M=0.6,h=Ø	SUPERSONIC M=2.3,h=40k
POSITIVE PITCH POINTING	THETA	OVERSHOOT	17%	8%	16%
		SETTLING TIME	5.9 sec	5 sec	5.6 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	INTERACTION	12%	17%	8%
		# OF PEAKS	1 pk	1 pk	1 pk
	VELOCITY	MAX VALUES	-2.5 to 1.3	-13 to 7.8	-54 to 37
	CANARD	MAX VALUES	-3° to -16°	-7° to -17°	-2° to -15°
		# OF PEAKS	3 pks	2 pks	4 pks
	LIFT FLAP	MAX VALUES	-12° to -75°	-7° to -117°	-9° to -50°
		# OF PEAKS	1 pk	1 pk	1 pk
	F_{TOTAL}	MAX VALUES	1400 to -750	-1800 to 6900	-600 to 34000
		# OF PEAKS	3 pks	2 pks	2 pks
VERTICAL TRANSLATION	GAMMA	OVERSHOOT	3%	Ø	12%
		SETTLING TIME	3.9 sec	5.8 sec	2.6 sec
		# OF PEAKS	2 pks	2 pks	2 pks
	THETA	INTERACTION	-20%	+22% to -14%	-39%
		# OF PEAKS	1 pk	4 pks	3 pks
	VELOCITY	MAX VALUES	-2.4 to 1.4	9.2 to -5.5	51 to -36
	CANARD	MAX VALUES	39° to 10°	21° to 6°	64° to -11°
		# OF PEAKS	2 pks	5 pks	2 pks
	LIFT FLAP	MAX VALUES	300° to 15°	300° to 31°	251° to -42°
		# OF PEAKS	3 pks	3 pks	4 pks
	F_{TOTAL}	MAX VALUES	6700 to -1400	4300 to -1070	-31800 to 124
		# OF PEAKS	3 pks	3 pks	2 pks
STRAIGHT CLIMB	THETA	OVERSHOOT	Ø	1%	Ø
		SETTLING TIME	0.7 sec	1 sec	0.7 sec
		# OF PEAKS	1 pk	3 pks	1 pk
	GAMMA	OVERSHOOT	9%	10%	17%
		SETTLING TIME	1.7 sec	0.8 sec	0.5 sec
		# OF PEAKS	2 pks	3 pks	3 pks
	VELOCITY	MAX VALUES	-4.1 to 2	-4.4 to 2.5	18 to -17
	CANARD	MAX VALUES	36° to -9°	14° to -10°	49° to -13°
		# OF PEAKS	2 pks	2 pks	2 pks
	LIFT FLAP	MAX VALUES	298° to -25°	302° to -38°	239° to -68°
		# OF PEAKS	4 pks	2 pks	2 pks
	F_{TOTAL}	MAX VALUES	7000 to -570	10040 to -2900	-7000 to 7600
		# OF PEAKS	4 pks	4 pks	2 pk

The servos and actuators are the same as stated previously.

In order to check the robustness of the transonic design, the same controller matrices, K_0 and K_1, and the same measurement matrix, M, are used for the subsonic flight condition.

The positive pitch pointing command vector produces the input and output results shown in Figs. 9a through Fig. 9e. The theta overshoot is less than the transonic condition, but the gamma interaction is slightly higher. The velocity perturbation is larger with a -13 ft/sec minimum and a 7.8 ft/sec maximum. This is the perturbation from a nominal velocity of 670 ft/sec and may or may not be readily apparent to the pilot. The steady state values for the inputs are:

CANARD: -17°
LIFT FLAP: -117°

The engine is still oscillating at T=10 sec. The steady state value for the lift flap exceeds the physical maximum of 75° which indicates that the command vector is too large or that the lift flap does not have the "authority" to perform the needed task. This should be taken into account if the FPCC aircraft evolves any further than its current status as a "paper" aircraft.

The inputs and outputs for the vertical translation command vector are shown in Fig. 10a through Fig. 10e. Both the gamma overshoot and the theta interactior are less than the for transonic condition but the velocity perturbation is greater. The maximum perturbation is 9.2 ft/sec and the minimum is -5.5 ft/sec. Although this is less than a 2% deviation from the nominal value of 670 ft/sec, it may be noticeable.

The steady state values for the inputs are:

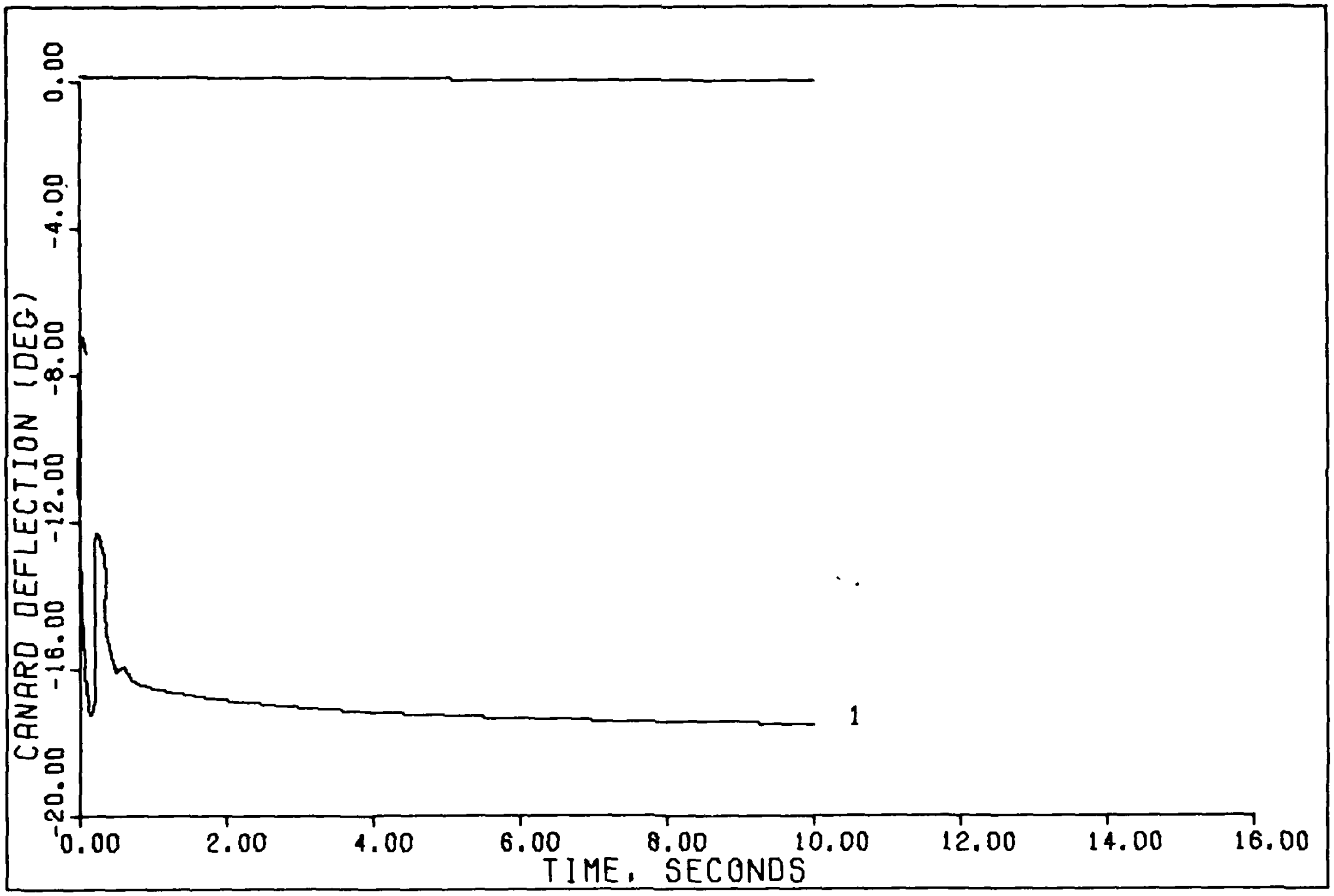


Figure 9a. Canard Deflection, Positive Pitch Pointing Command (Subsonic Flight Condition)

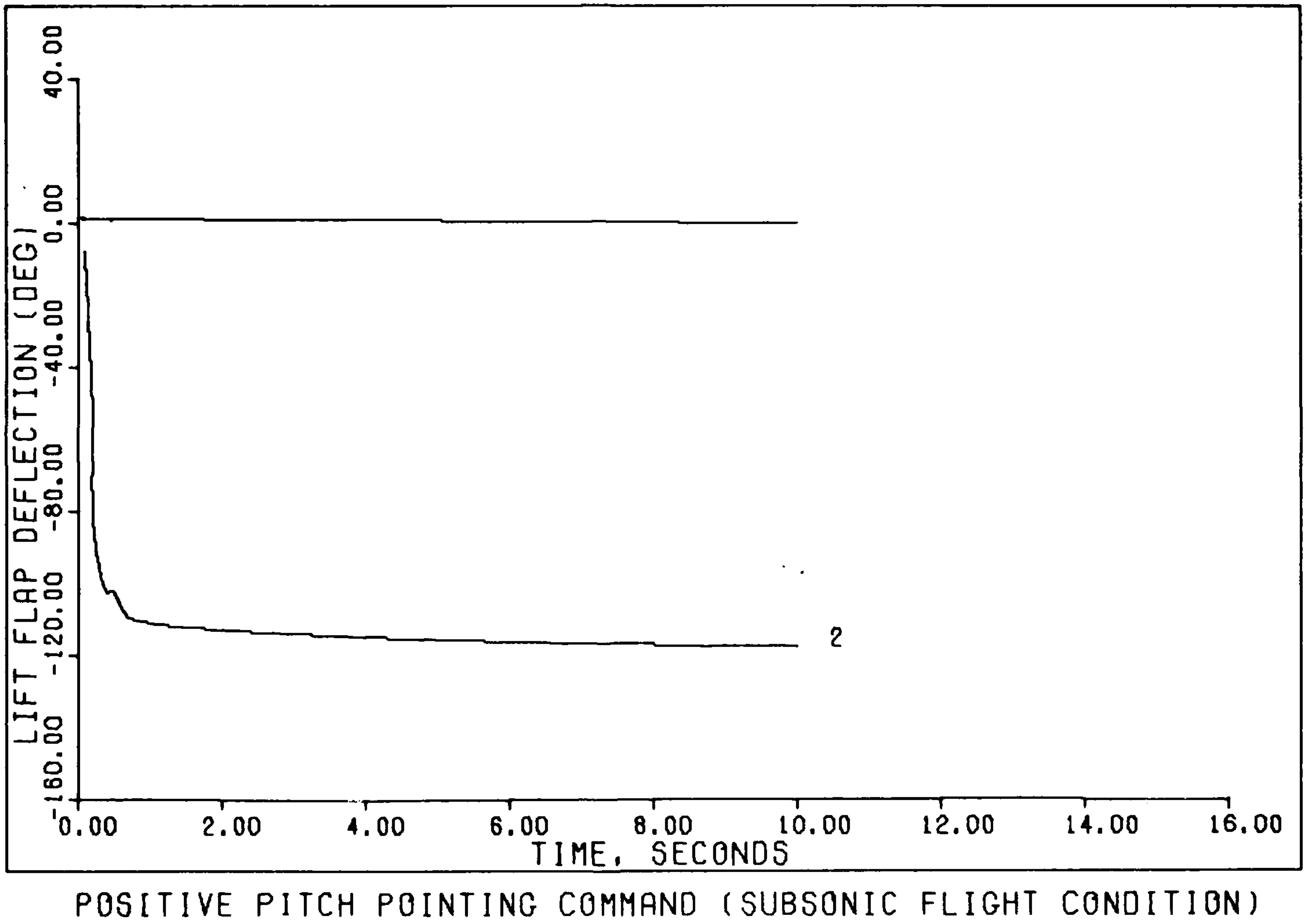


Figure 9b. Lift Flap Deflection

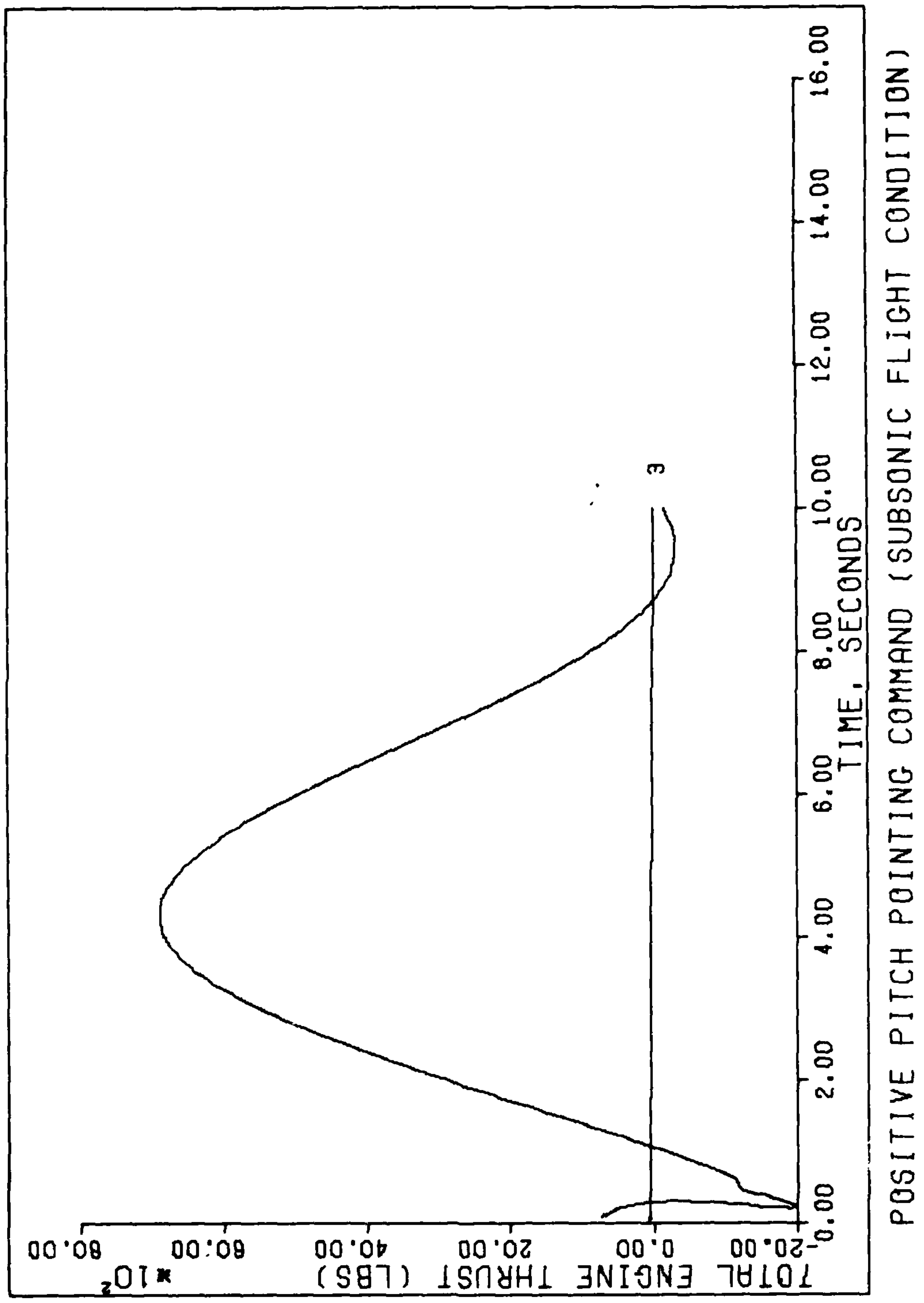


Figure 9c. Total Engine Thrust

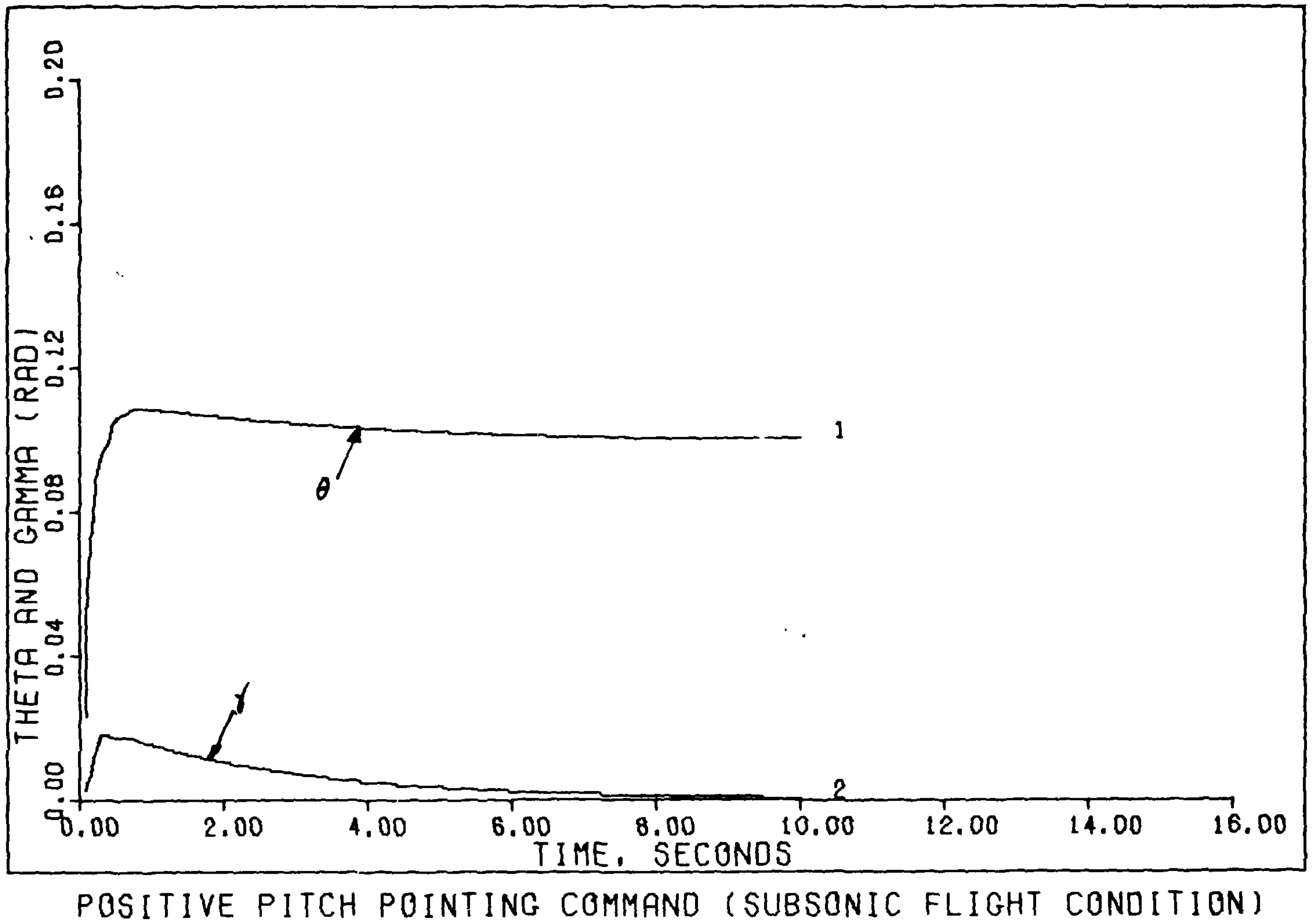
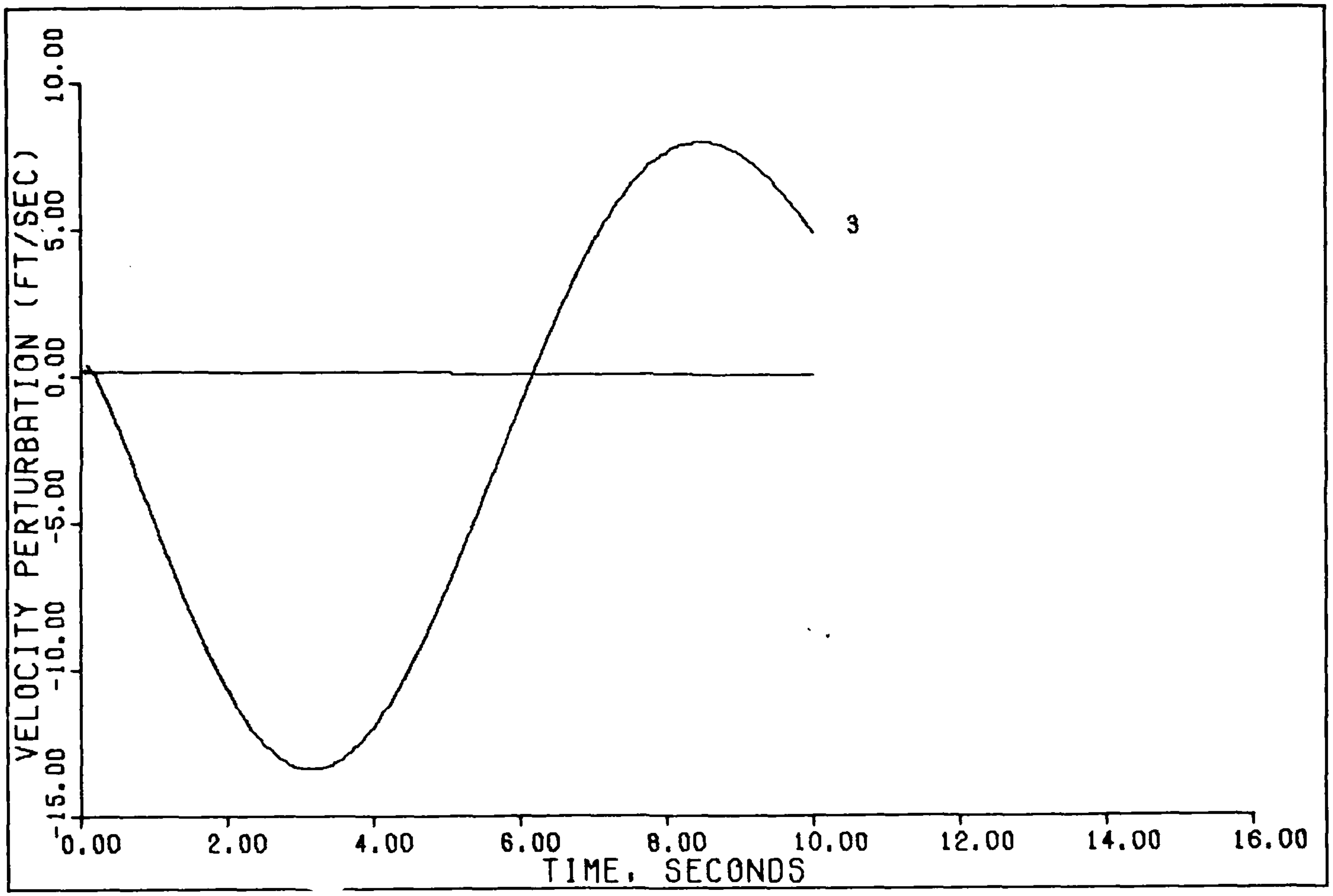


Figure 9d. Pitch and Flight Path Angles



POSITIVE PITCH POINTING COMMAND (SUBSONIC FLIGHT CONDITION)

Figure 9e. Velocity Perturbation

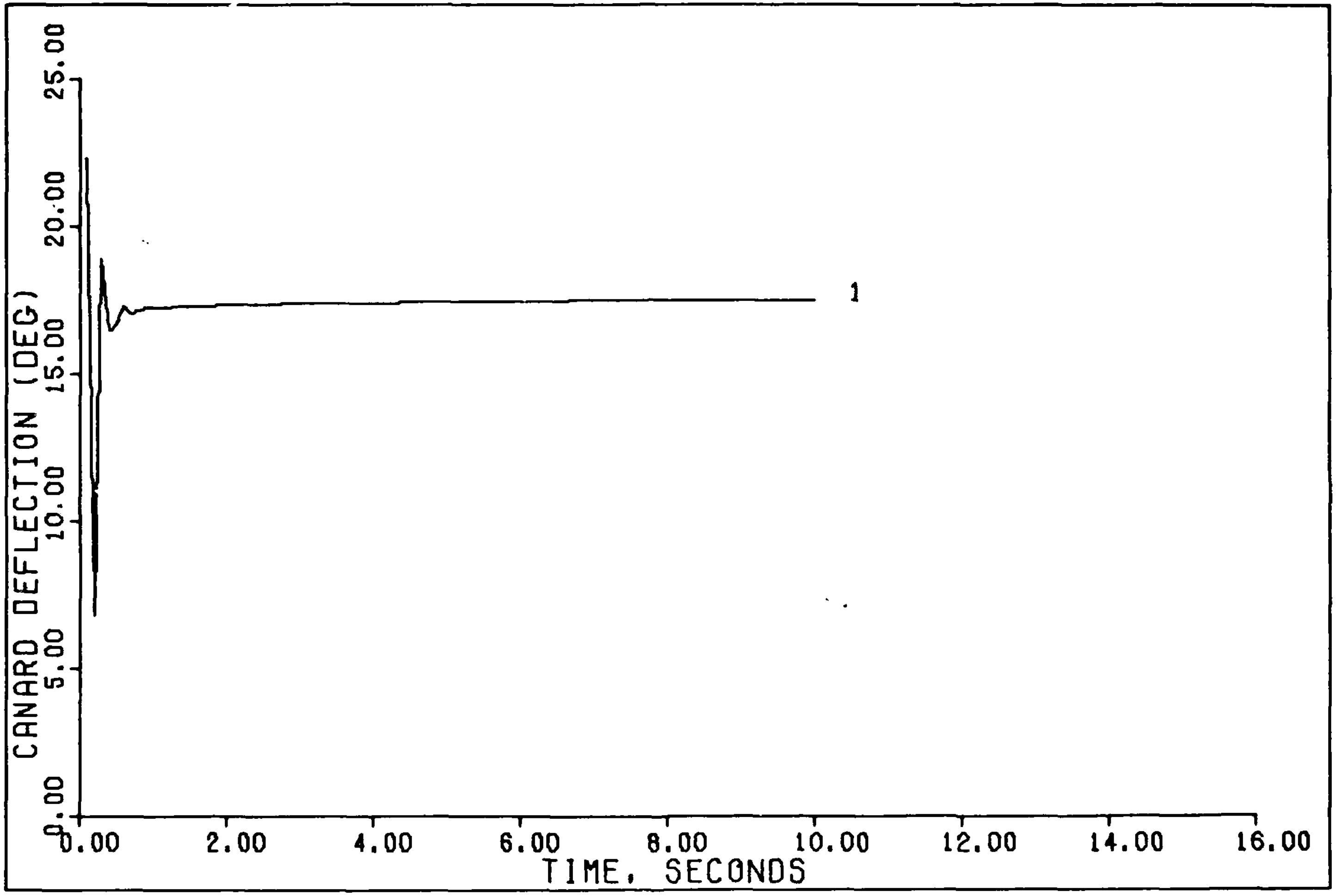


Figure 10a. Canard Deflection, Vertical Translation Command (Subsonic Flight Condition)

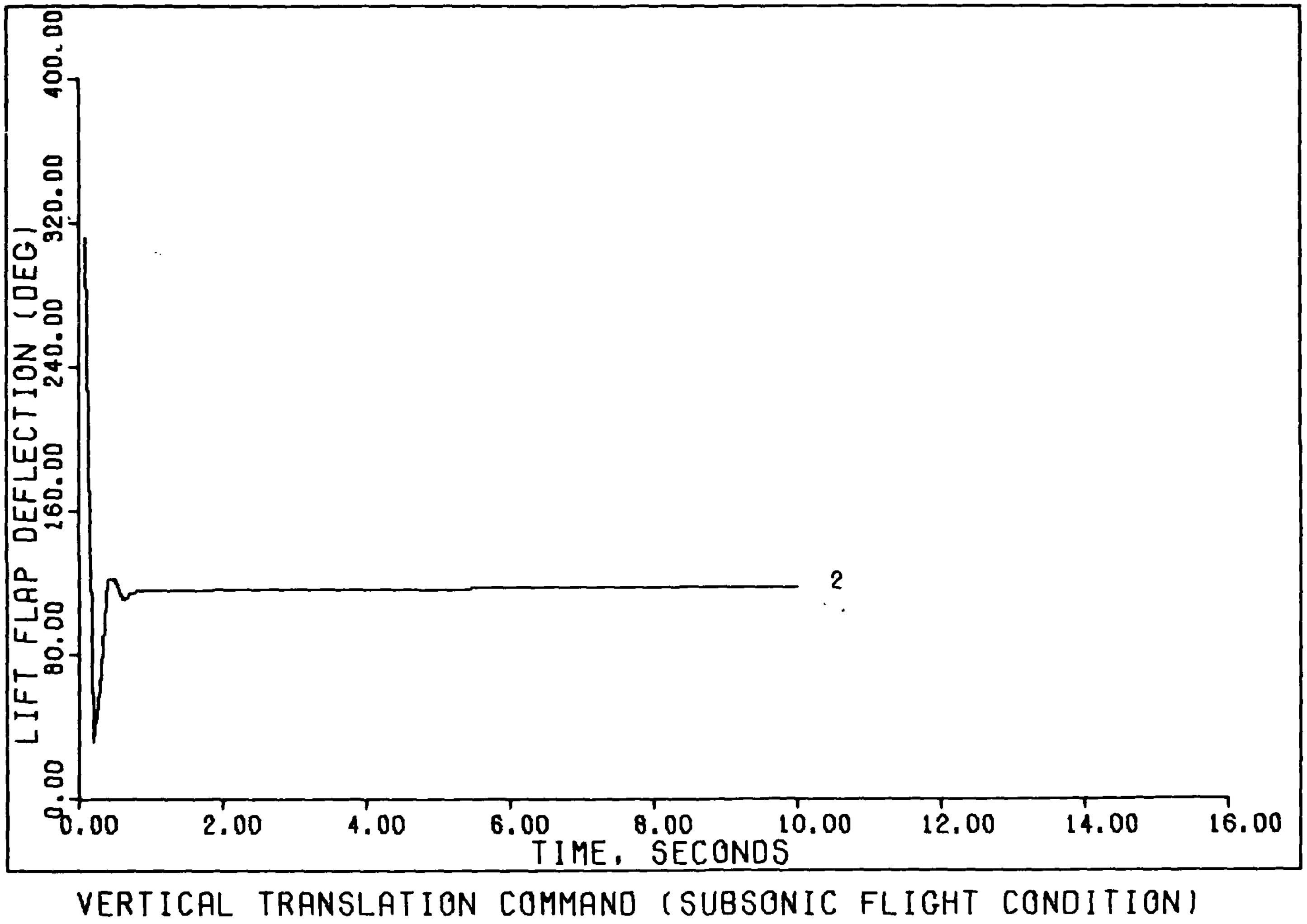


Figure 10b. Lift Flap Deflection

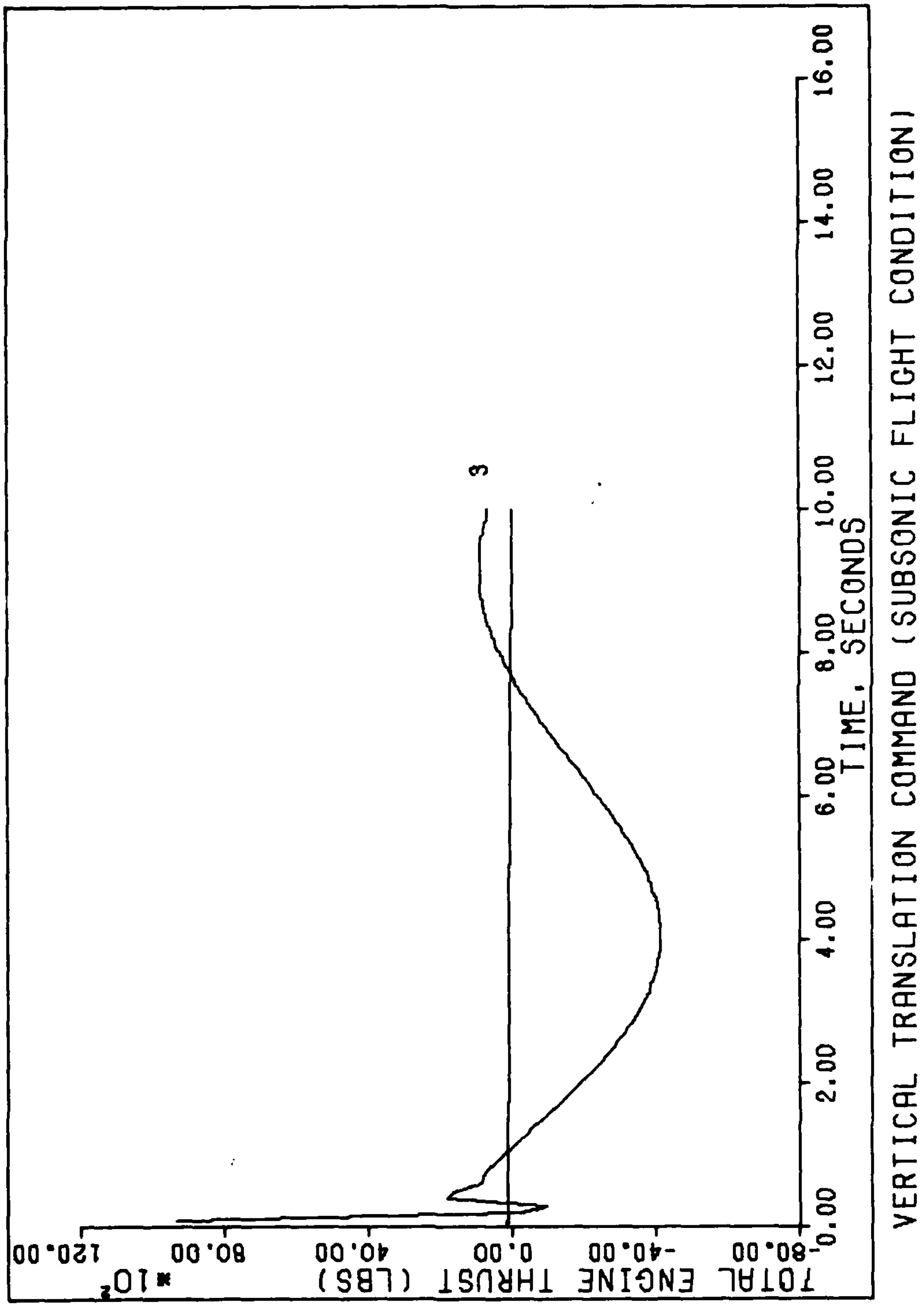


Figure 10c. Total Engine Thrust

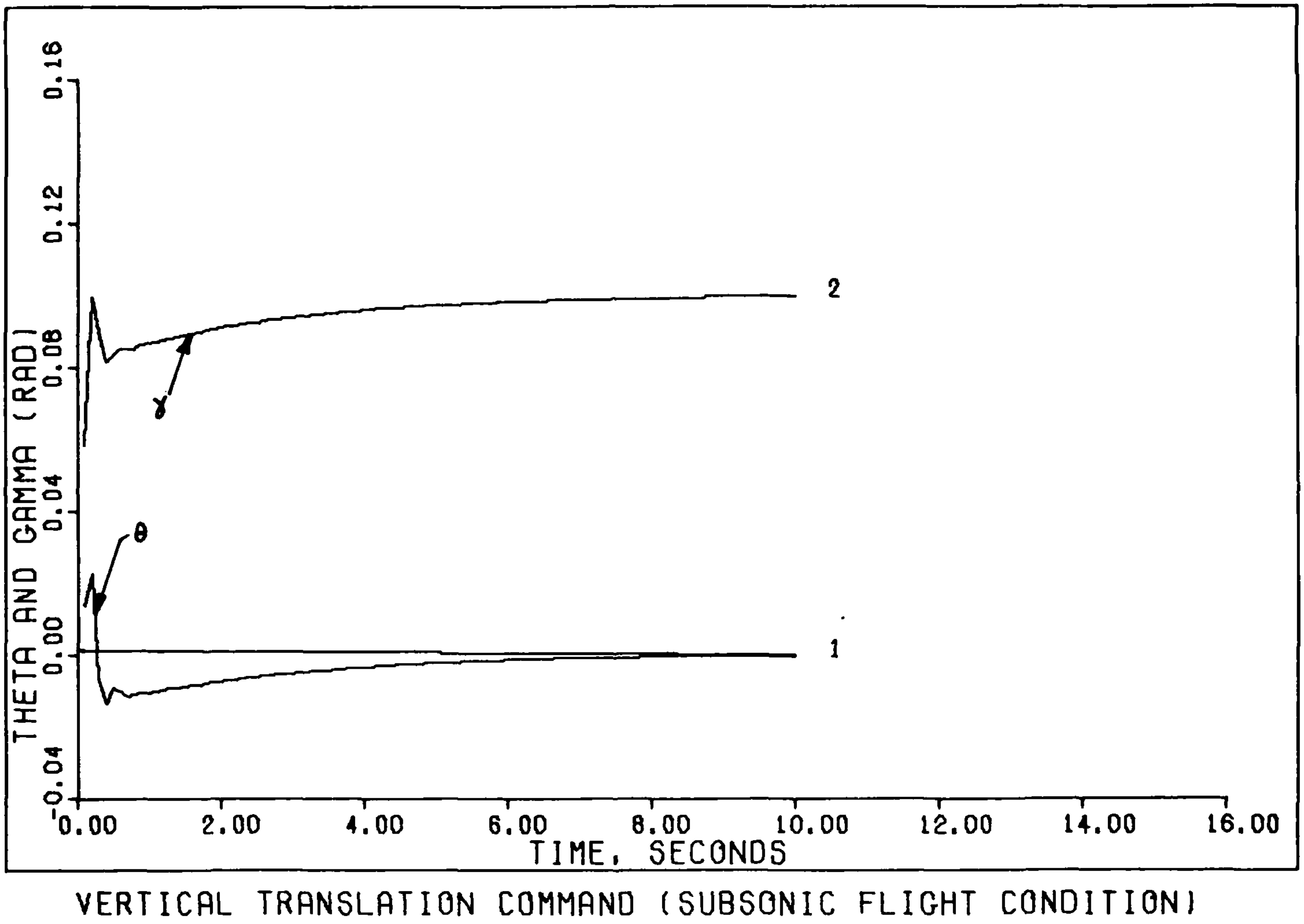
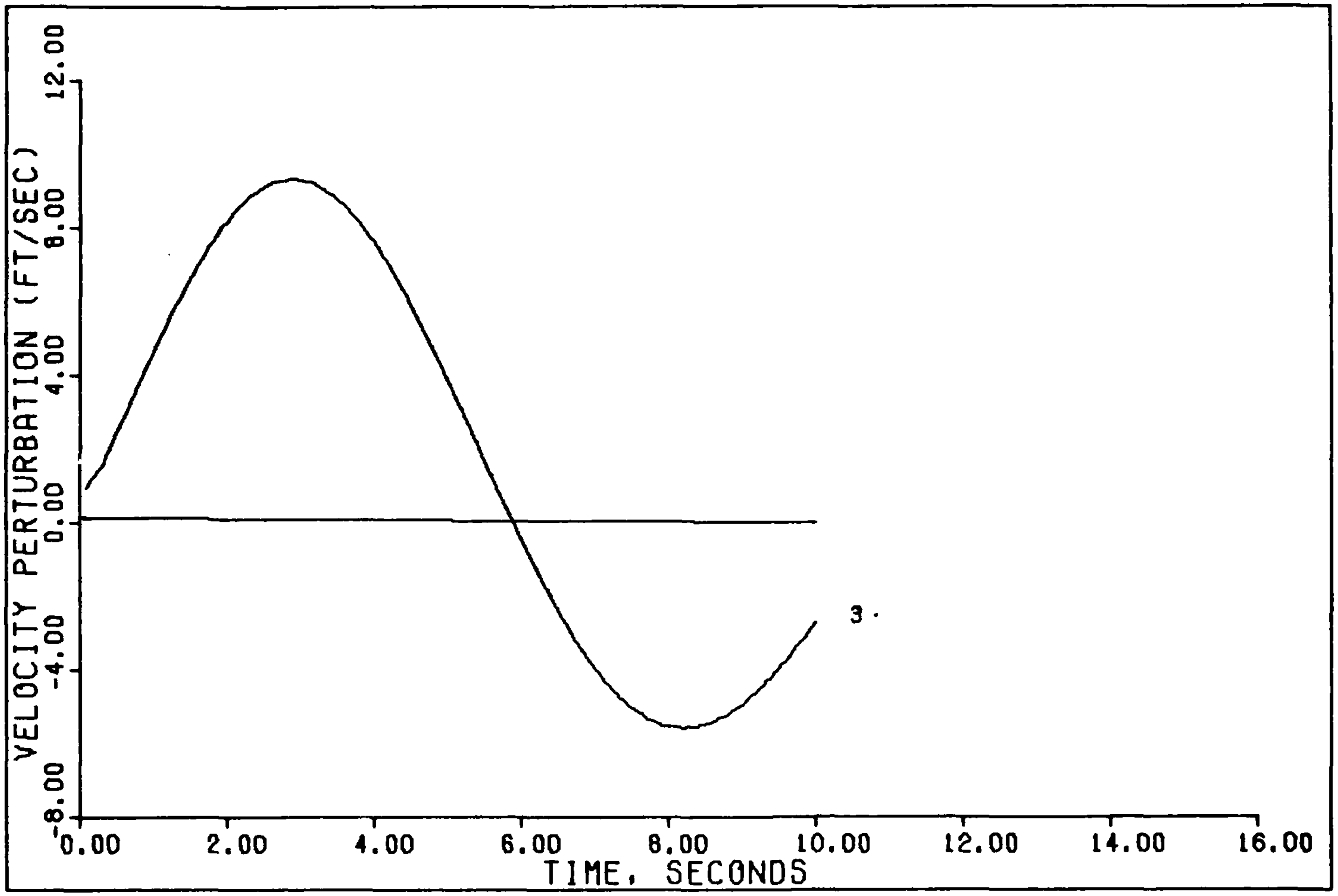


Figure 10d. Pitch and Flight Path Angles



VERTICAL TRANSLATION COMMAND (SUBSONIC FLIGHT CONDITION)

Figure 10e. Velocity Perturbation

CANARD: 17°

LIFT FLAP: 117°

The engine is still oscillatory at T=10 seconds. The steady state value for the lift flap again indicates that either the command vector is too large or the lift flap lacks the "authority" to perform a vertical translation with a 0.1 radian flight path angle.

The straight climb command vector result are shown in Figs. 11a through 11e. Theta and gamma track a 0.1 radian command with slightly more oscillatory behavior than the transonic condition results of Figs. 8a through Figs. 8e. The velocity perturbation is also quite similar to the transonic results; the maximum perturbation is 2.5 ft/sec, the minimum is -4.4 ft/sec. These would hardly be noticed by the pilot.

The steady state values for the inputs are:

CANARD: 0°

LIFT FLAP: 0°

The engine is still oscillatory at T=10 sec.

The pertinent figures of merit for all three command vectors are presented in Table IV under the subsonic flight condition heading.

SUPERSONIC FLIGHT CONDITION

The four state plant and output equations for the supersonic flight condition #10 - Mach. 2.3, 40,000 ft - are as follows:

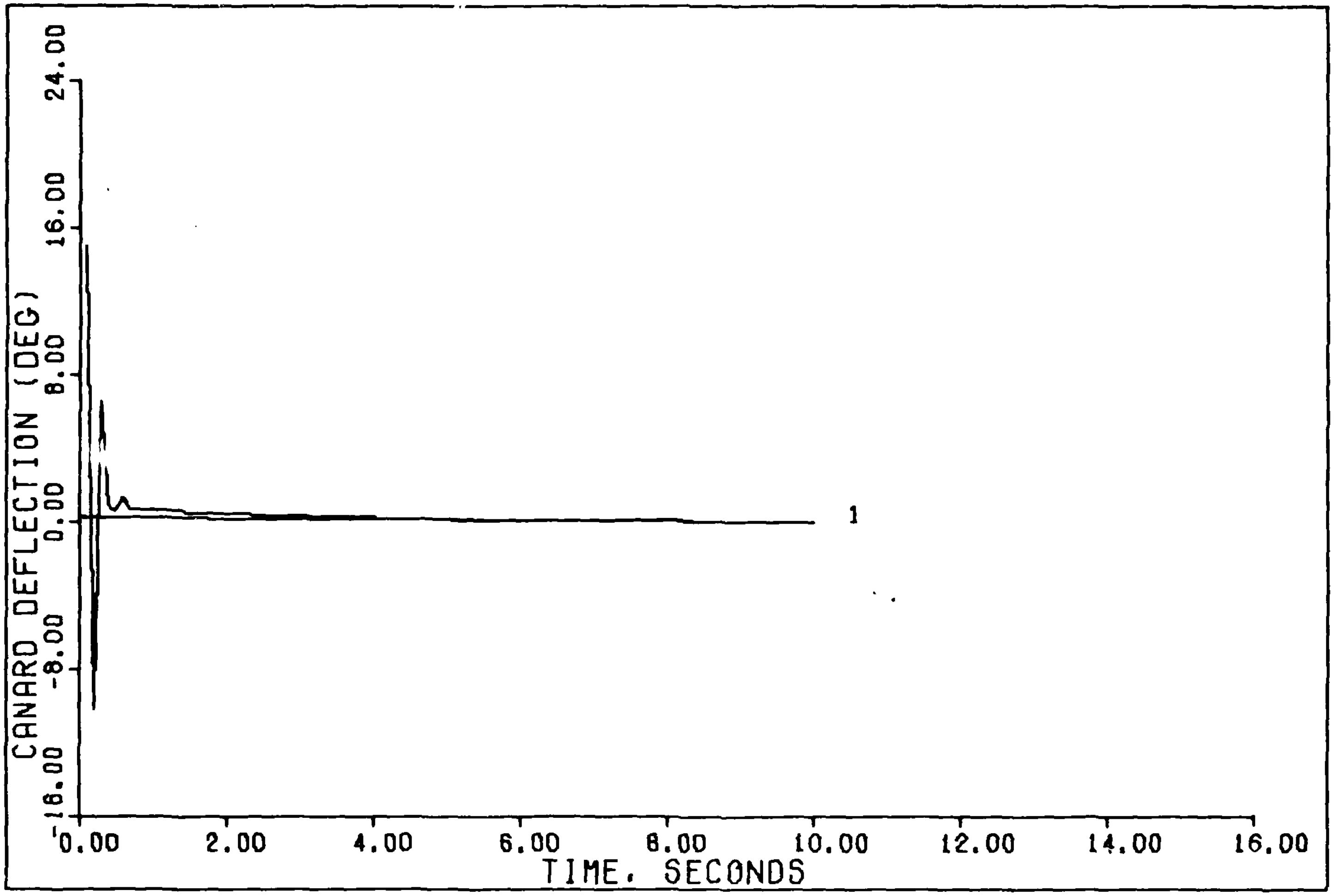


Figure 11a. Canard Deflection, Straight Climb Command (Subsonic Flight Condition)

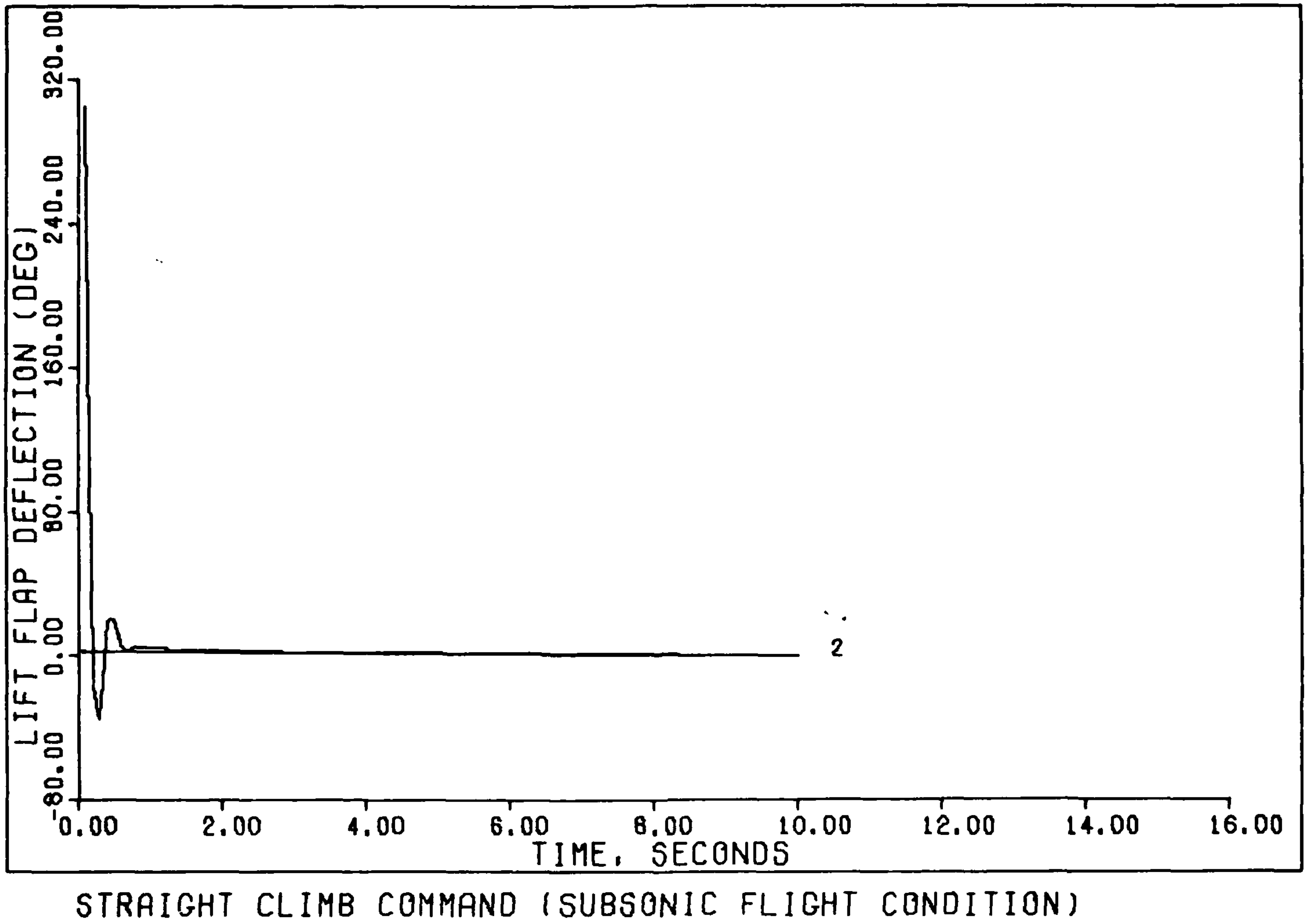


Figure 11b. Lift Flap Deflection

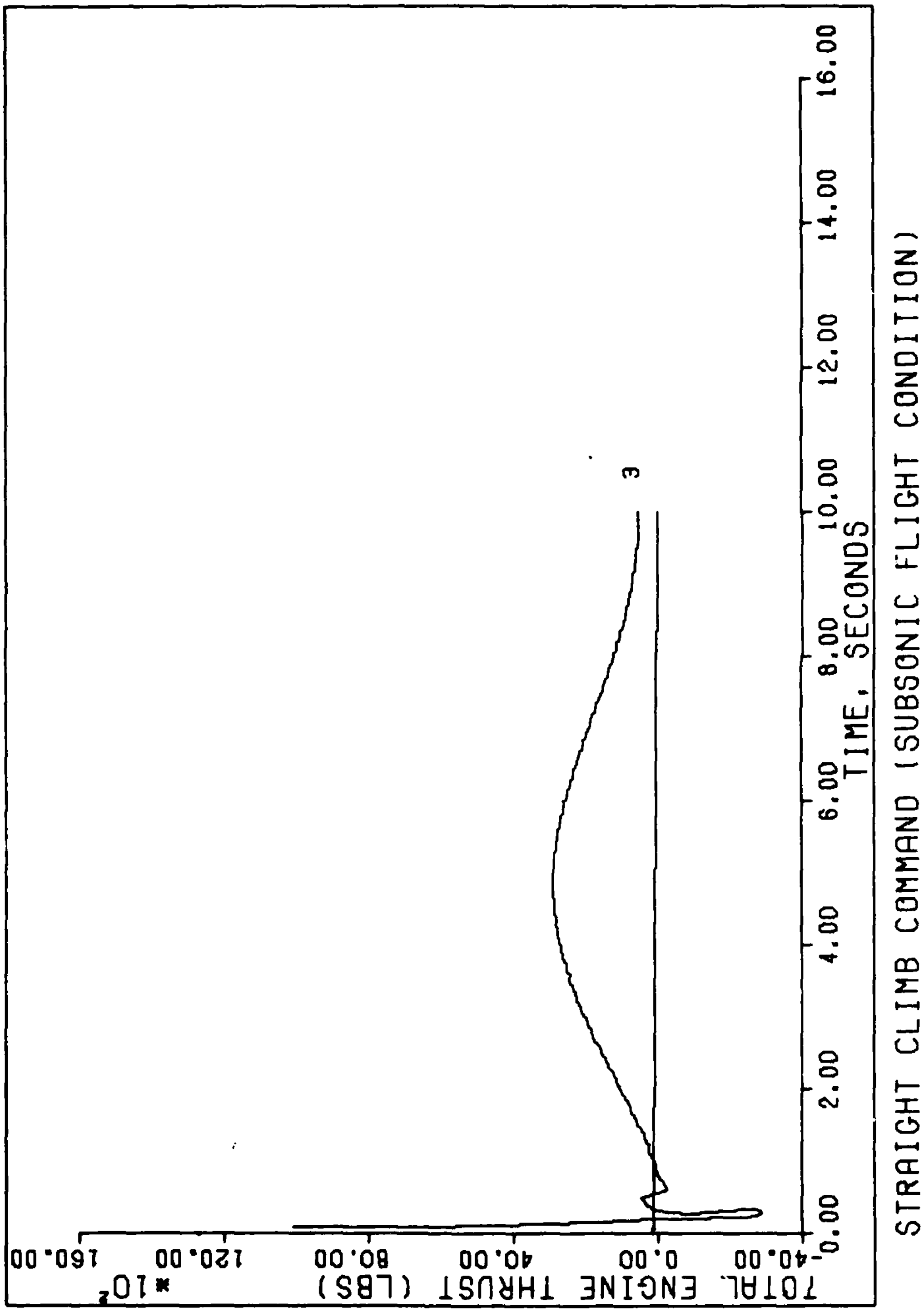


Figure 11c. Total Engine Thrust

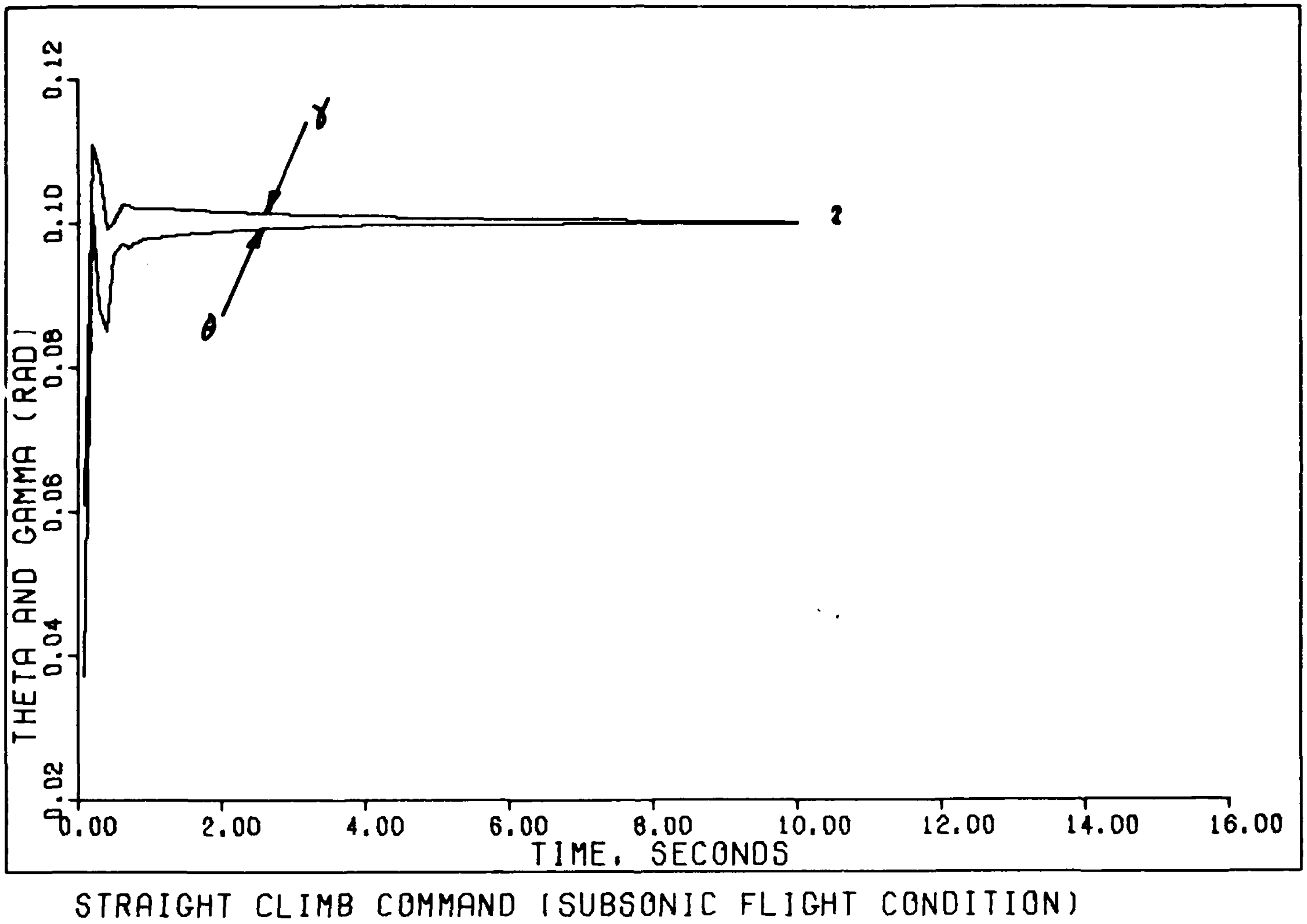


Figure .1d. Pitch and Flight Path Angles

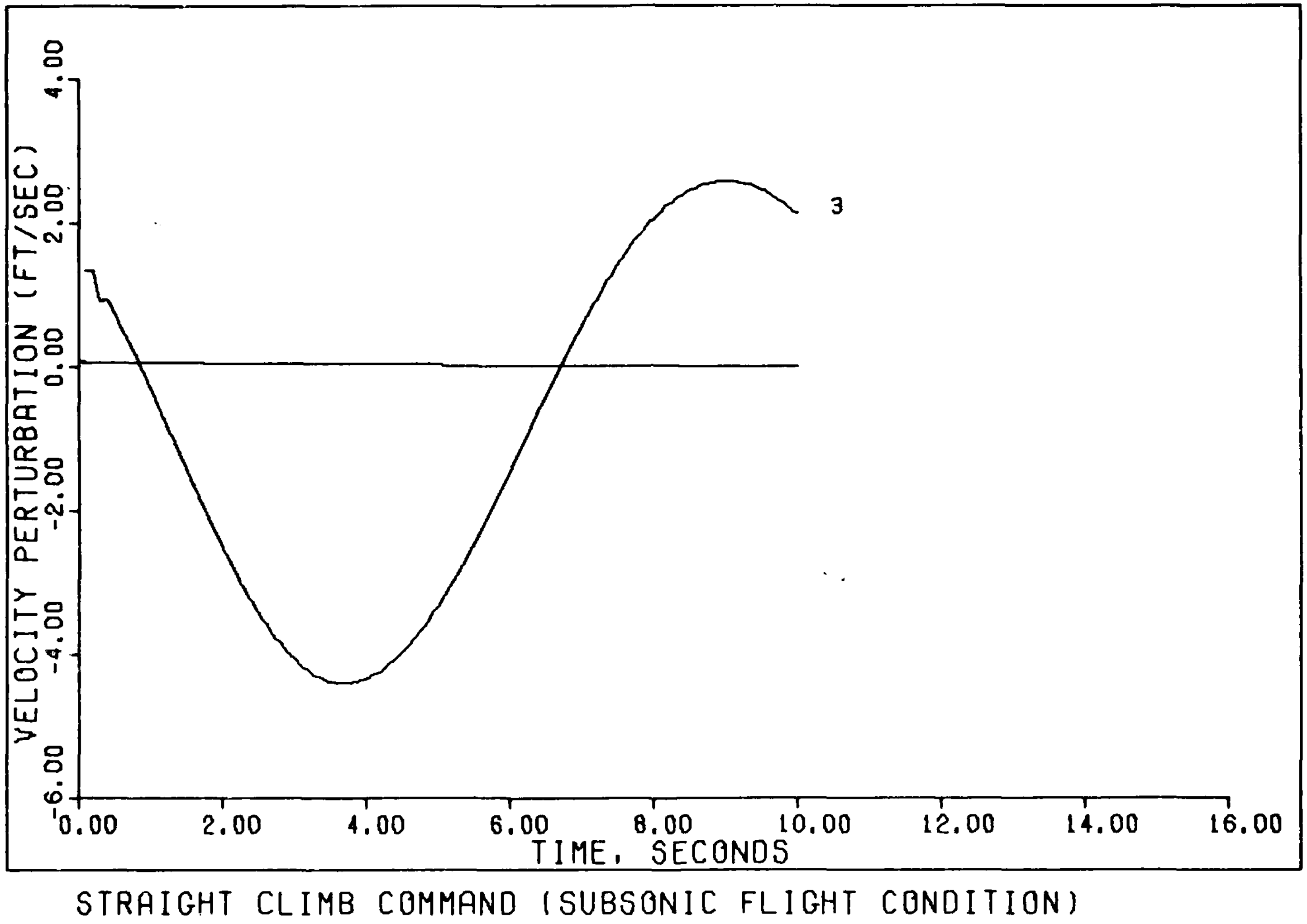


Figure 11e. Velocity Perturbation

AD-A118 117   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/6 1/3
DIGITAL FLIGHT CONTROL SYSTEM DESIGN USING SINGULAR PERTURBATIO--ETC(U)
DEC 81 J S SMYTH
UNCLASSIFIED   AFIT/EE/GE/81D-55   NL

2 of 3
AD
A118 117

$$\begin{bmatrix} \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1.0 \\ -32.1 & -0.0287 & -0.0113 & 24.7 \\ -0.356 & -0.441\text{x}10^{-2} & -1.41 & 2230 \\ 0 & -0.205\text{x}10^{-4} & 0.0123 & -1.67 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 2.96 & -0.0853 & 0.218\text{x}10^{-2} \\ -9.35 & -4.841 & 0 \\ 1.05 & -0.174 & 0 \end{bmatrix} \begin{bmatrix} \delta_c \\ \delta_{LF} \\ F_{TOTAL} \end{bmatrix} \qquad (78)$$

$$\begin{bmatrix} \theta \\ \gamma \\ VEL \end{bmatrix} \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 1.0 & 0.497\text{x}10^{-5} & -0.449\text{x}10^{-3} & 0 \\ 0 & 1.0 & 0.0111 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix} \qquad (79)$$

The servos and sensors are the same as stated previously.

This flight condition is the final check for robustness. The controller matrices, K_o and K_1, and the measurement matrix are the same ones developed for the transonic flight condition.

The positive pitch pointing results are shown in Figs. 12a through 12e. The theta overshoot and gamma interaction are reduced from the values for the transonic flight condition. Velocity perturbation has a minimum of -54 ft/sec and a maximum of 37 ft/sec from the nominal velocity of 2230 ft/sec. This corresponds to less than a 2 1/2%

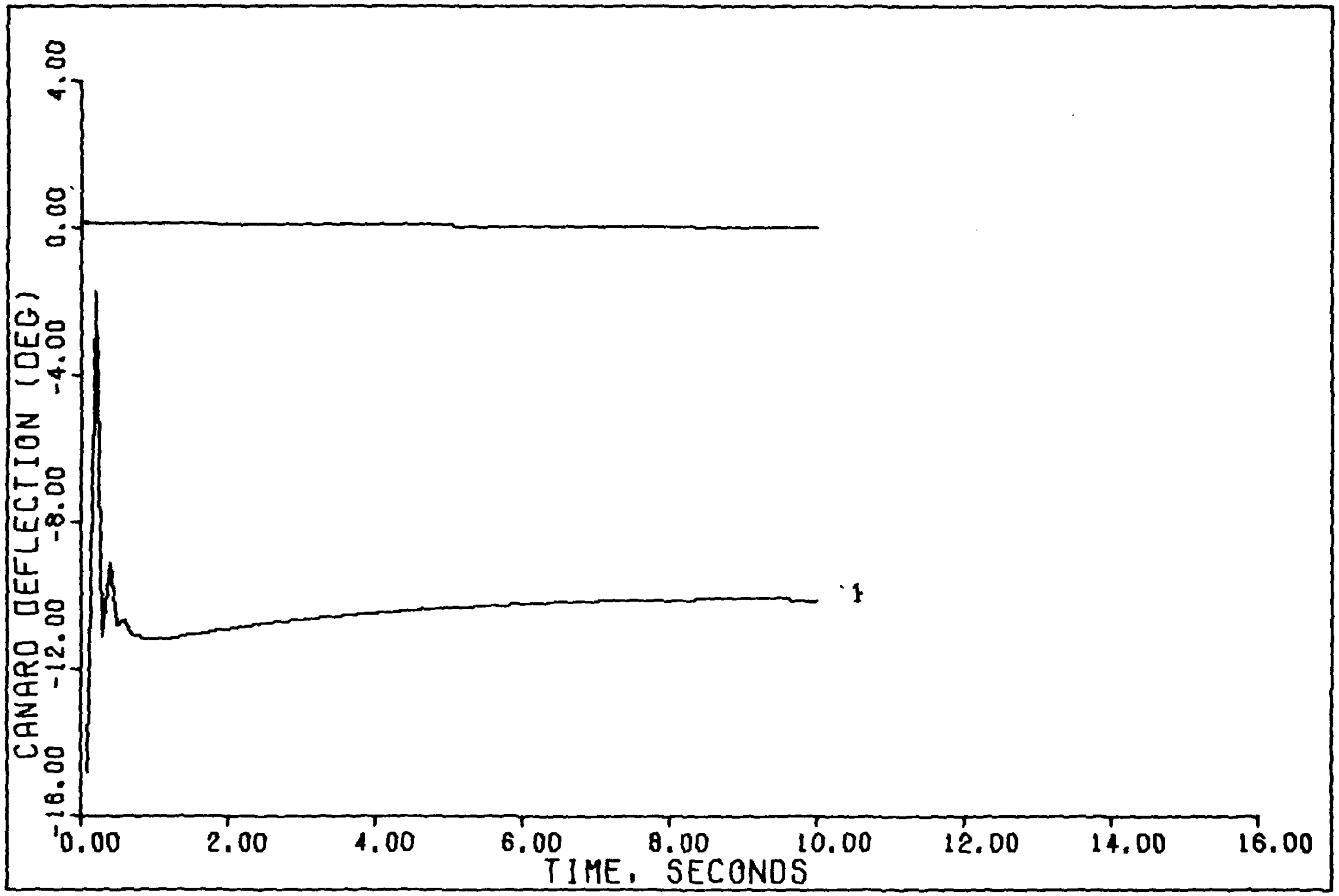


Figure 12a. Canard Deflection, Positive Pitch Pointing Command (Supersonic Flight Condition)

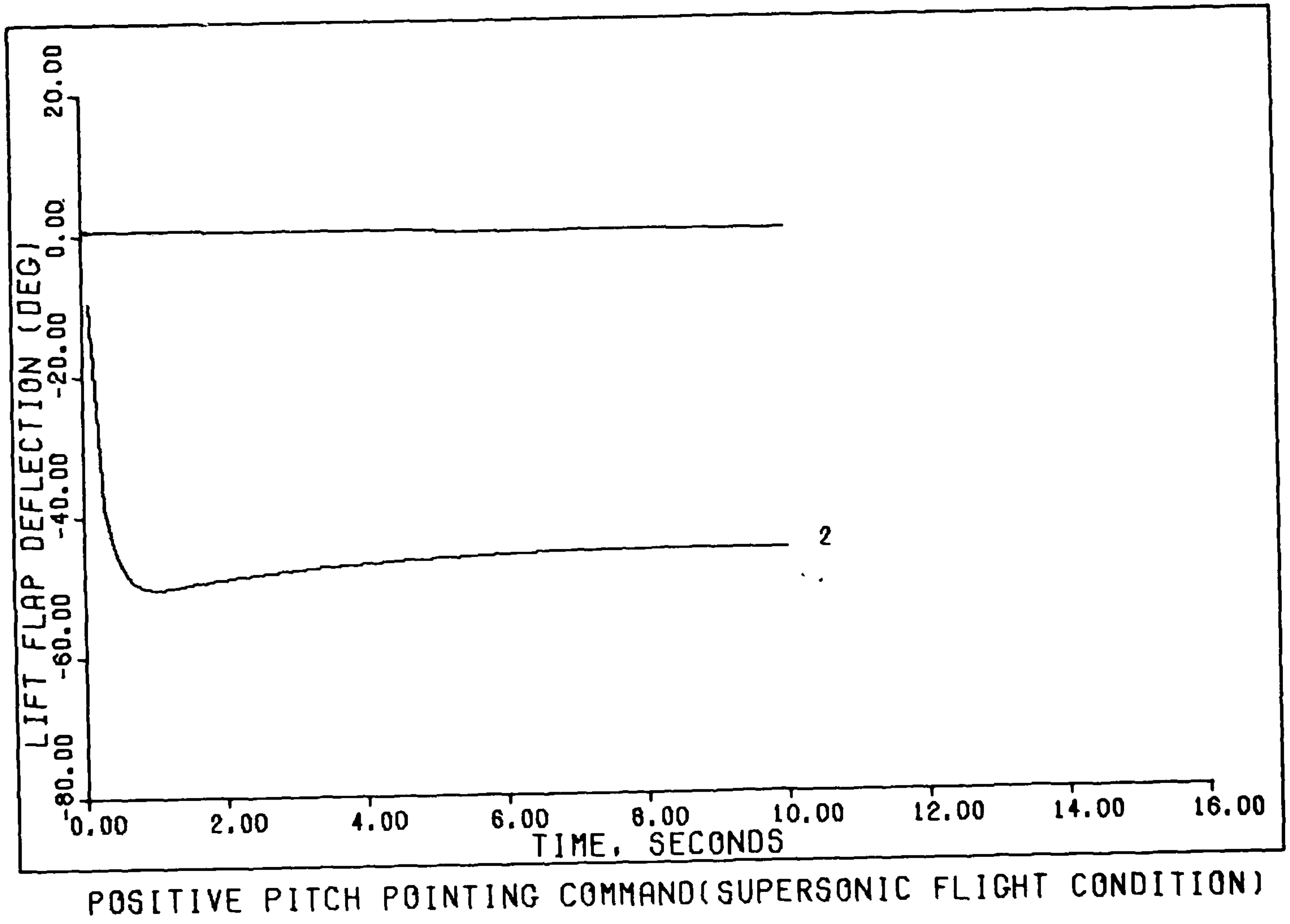


Figure 12b. Lift Flap Deflection

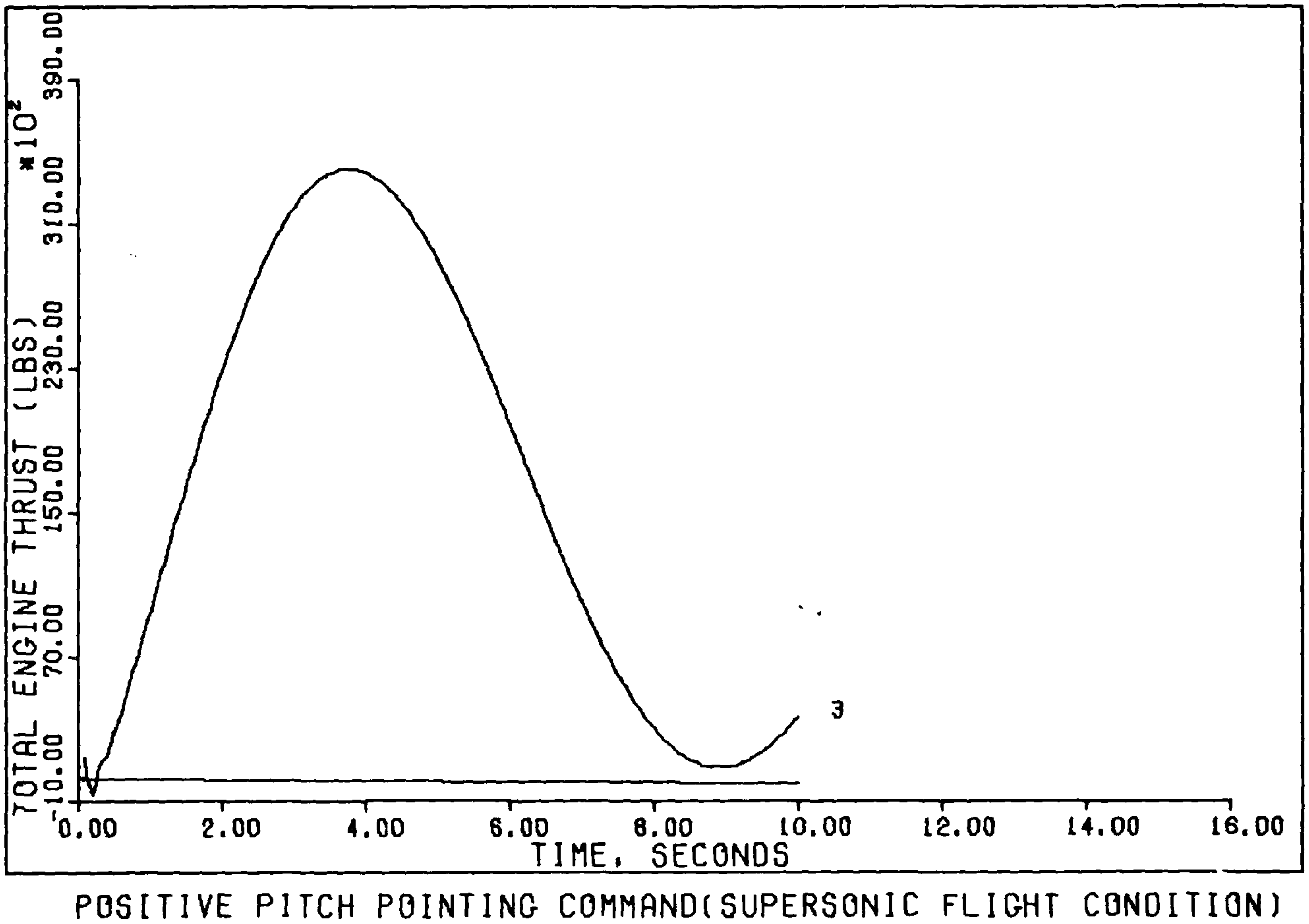
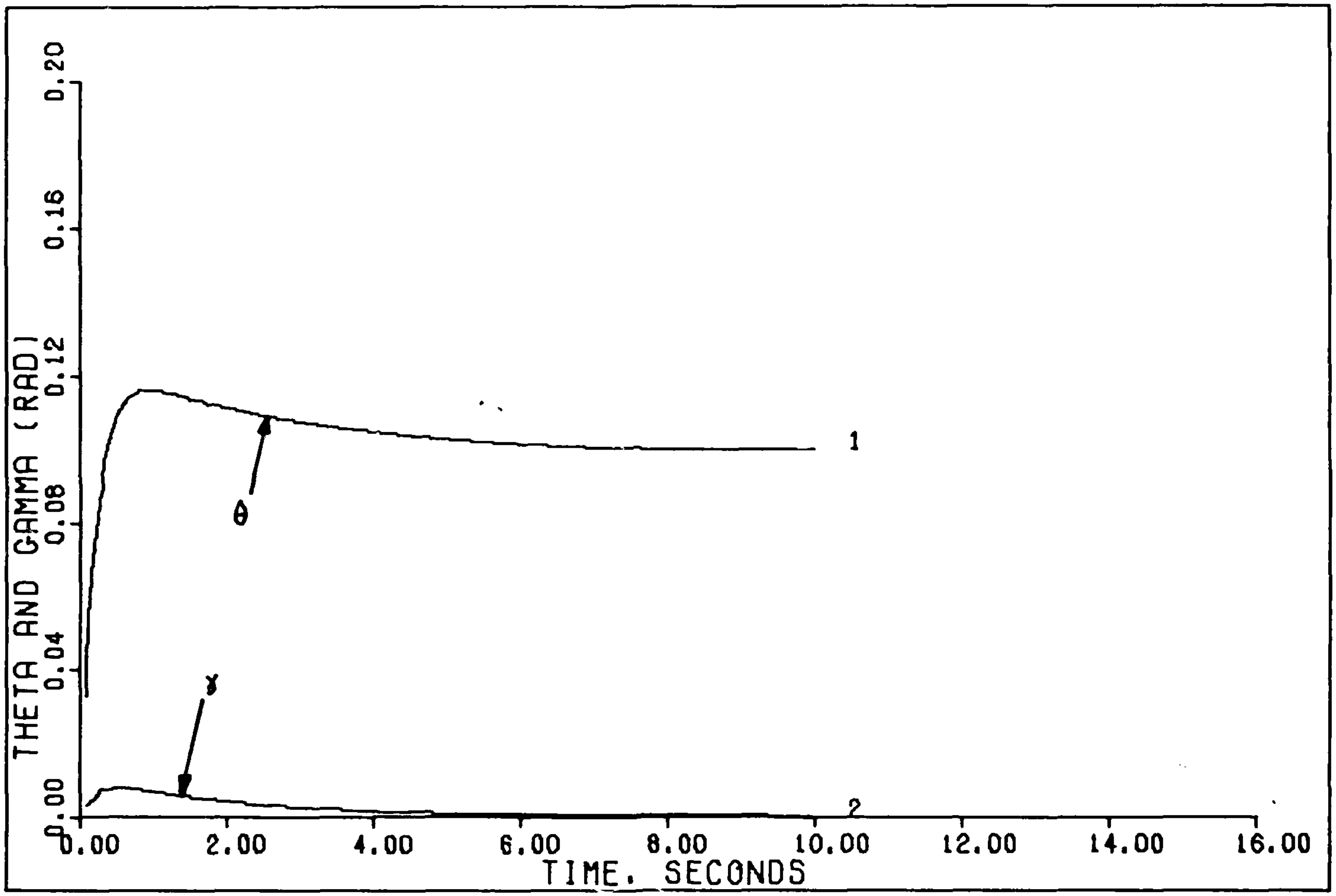


Figure 12c. Total Engine Thrust



POSITIVE PITCH POINTING COMMAND(SUPERSONIC FLIGHT CONDITION)

Figure 12d. Pitch and Flight Path Angles

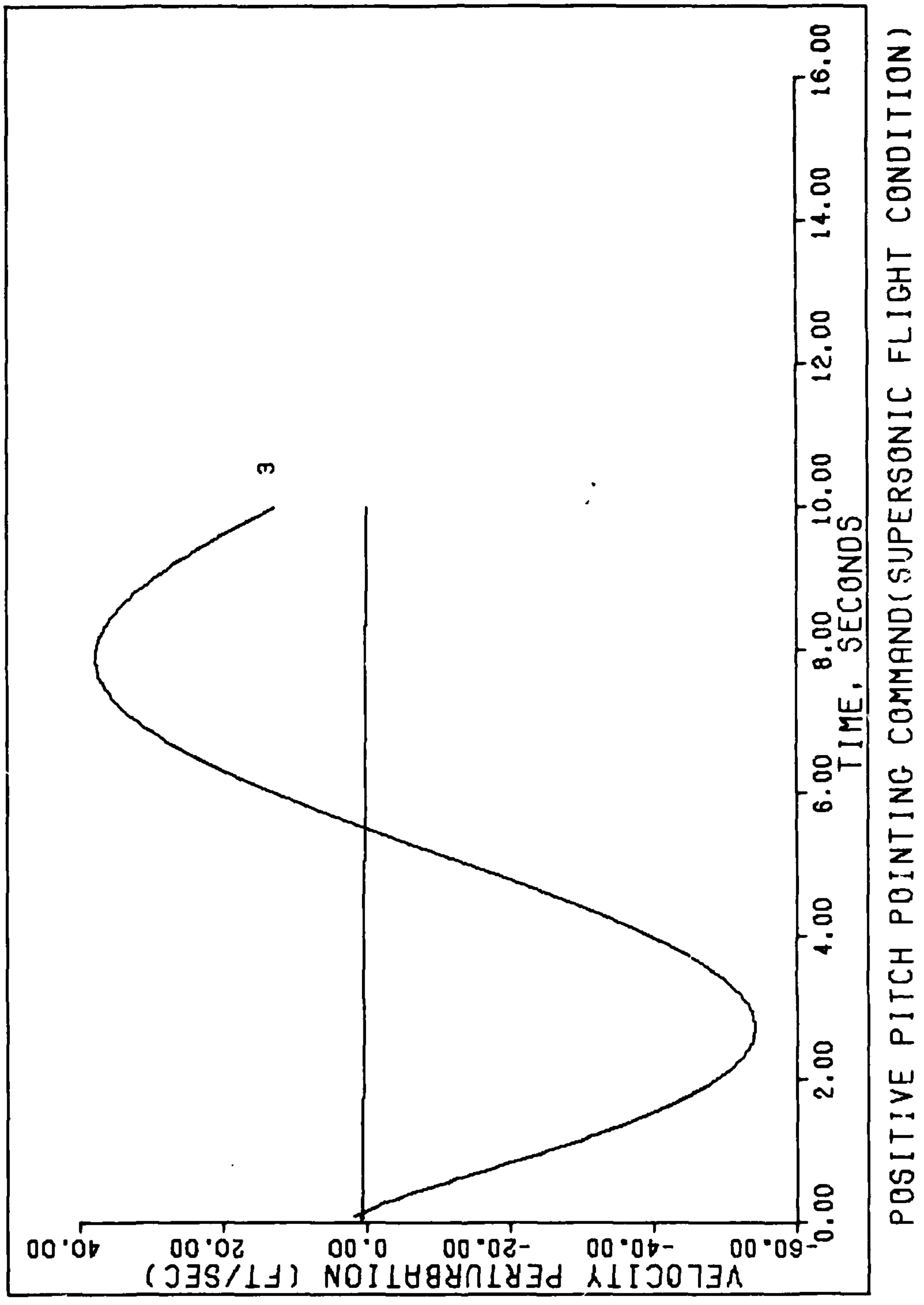


Figure 12e. Velocity Perturbation

variation in speed which may be detected by the pilot.

Steady state input values are:

CANARD: -10°
LIFT FLAP: -45°

The engine is still oscillatory at T=10 sec. The maximum variations from the nominal thrust is +34,000 and -600 lb. Since the maximum augmented thrust from both engines is 31,500 lb., it is apparent that this command vector is too large for the aircraft to maintain nominal velocity.

The vertical translation command vector results are shown in Figs. 13a through 13e. The gamma overshoot is 12% but the theta interaction peaks at -0.039, a 39% negative interaction. This is the only glaring problem in the robustness test. Velocity perturbation has a maximum of 51 ft/sec and a minimum of -36 ft/sec from the nominal velocity of 2230 ft/sec. This is slightly more than a 2% variation in velocity which may be noticeable.

The input steady state values are:

CANARD: 10°
LIFT FLAP: 45°

The engine is still oscillatory at T=10 sec. The total thrust varies from -31,800 lb to +124 lb from the nominal thrust. This again is clearly outside the normal operation of the engine. The command vector is too large for the aircraft at this supersonic condition.

The straight climb vector results are shown in Figs. 14a through 14e. The overshoot is 17%, larger than the transonic case. Theta still exhibits no overshoot. Velocity perturbation is quite small; maximum perturbation is 18ft/sec, minimum is -17 ft/sec. This is less than a 1%

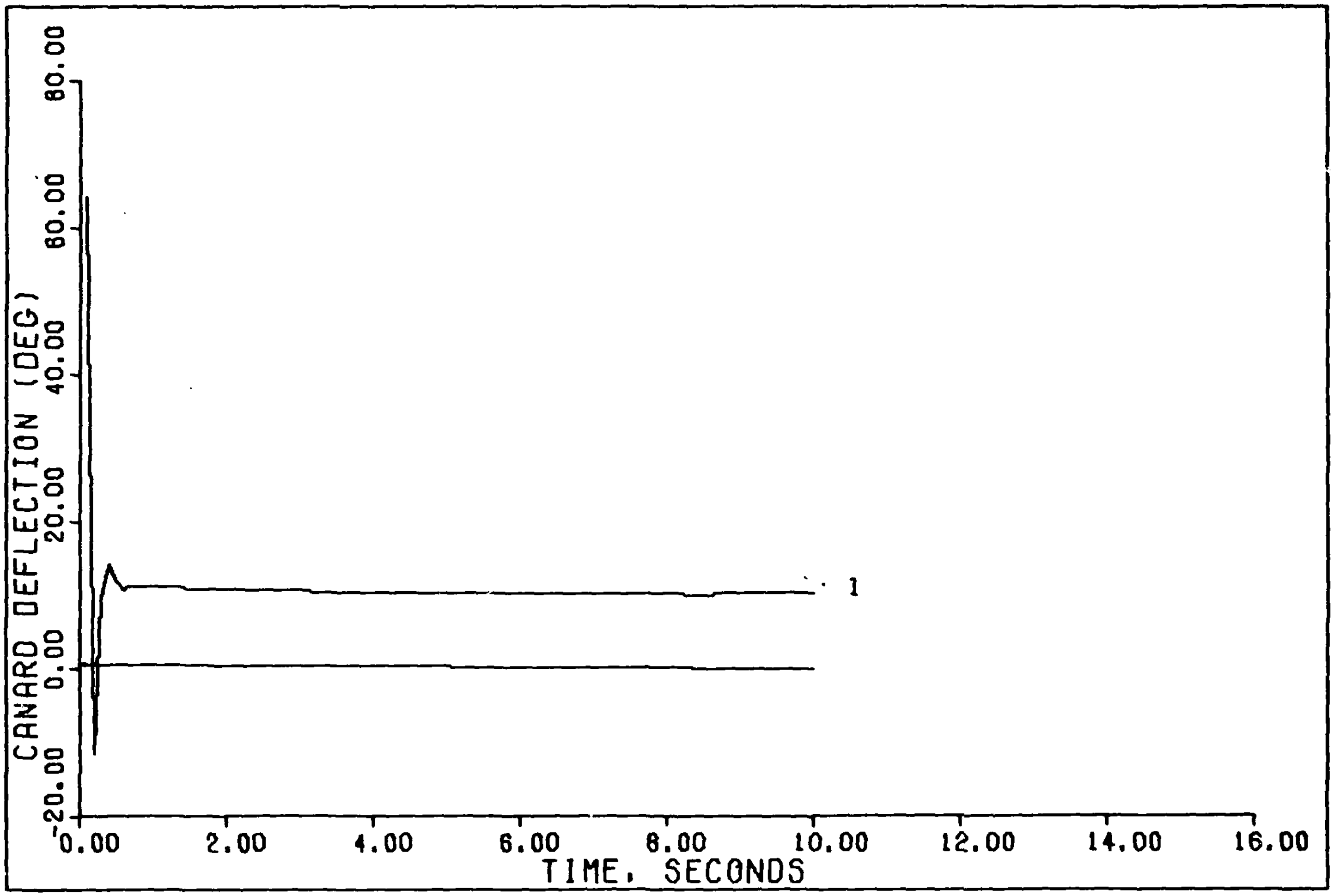


Figure 13a. Canard Deflection, Vertical Translation Command (Supersonic Flight Condition)

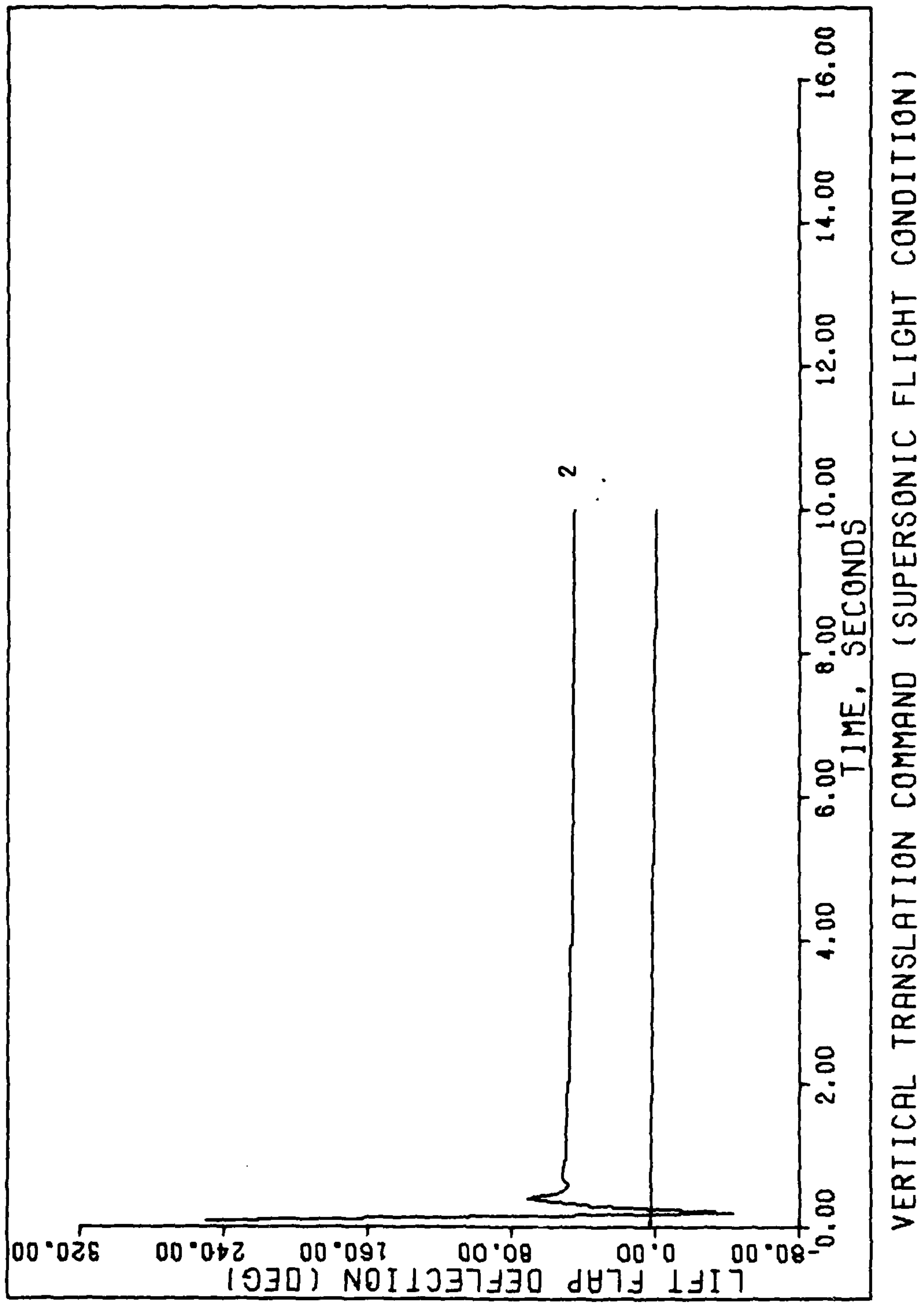


Figure 13b. Lift Flap Deflection

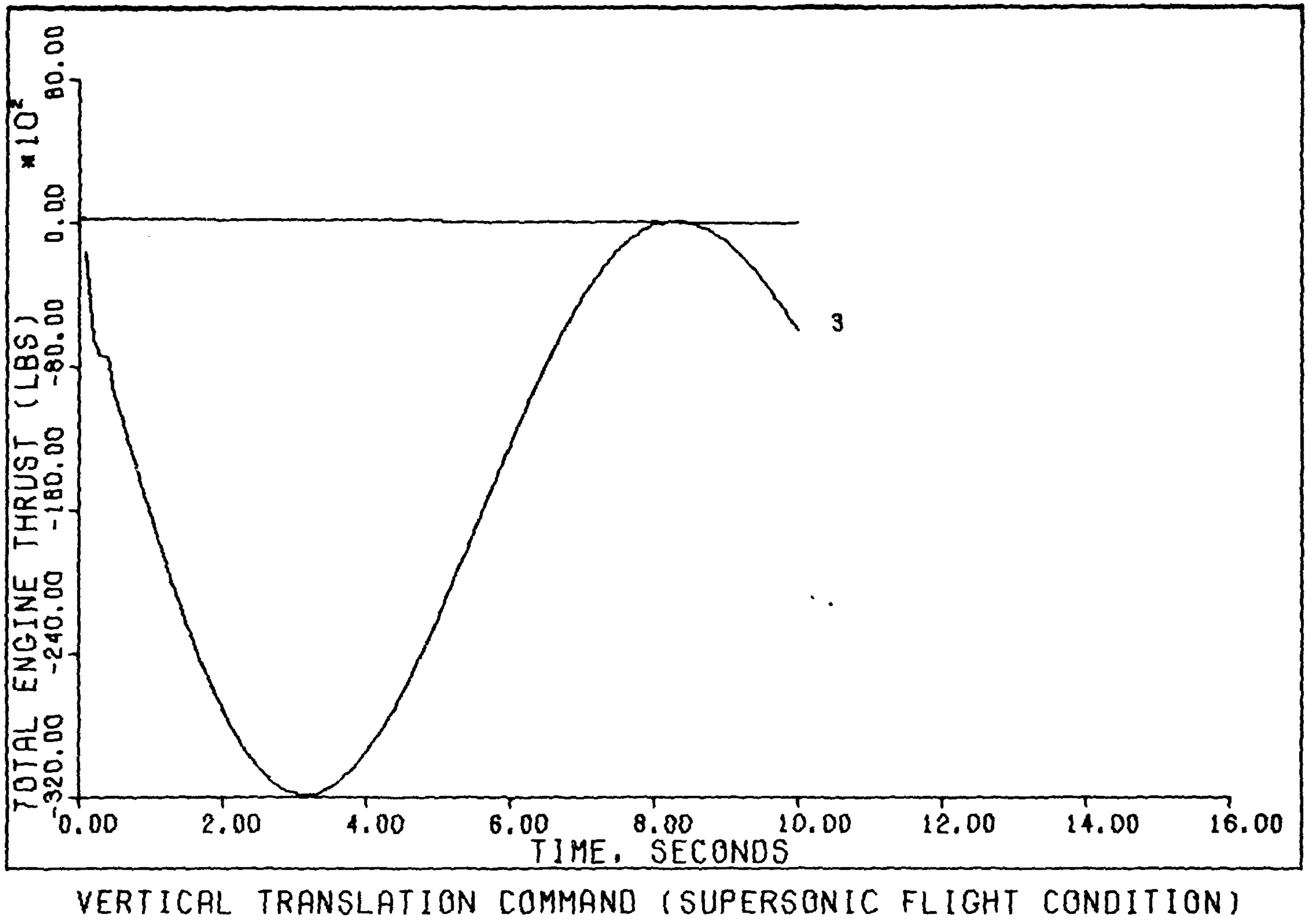


Figure 13c. Total Engine Thrust

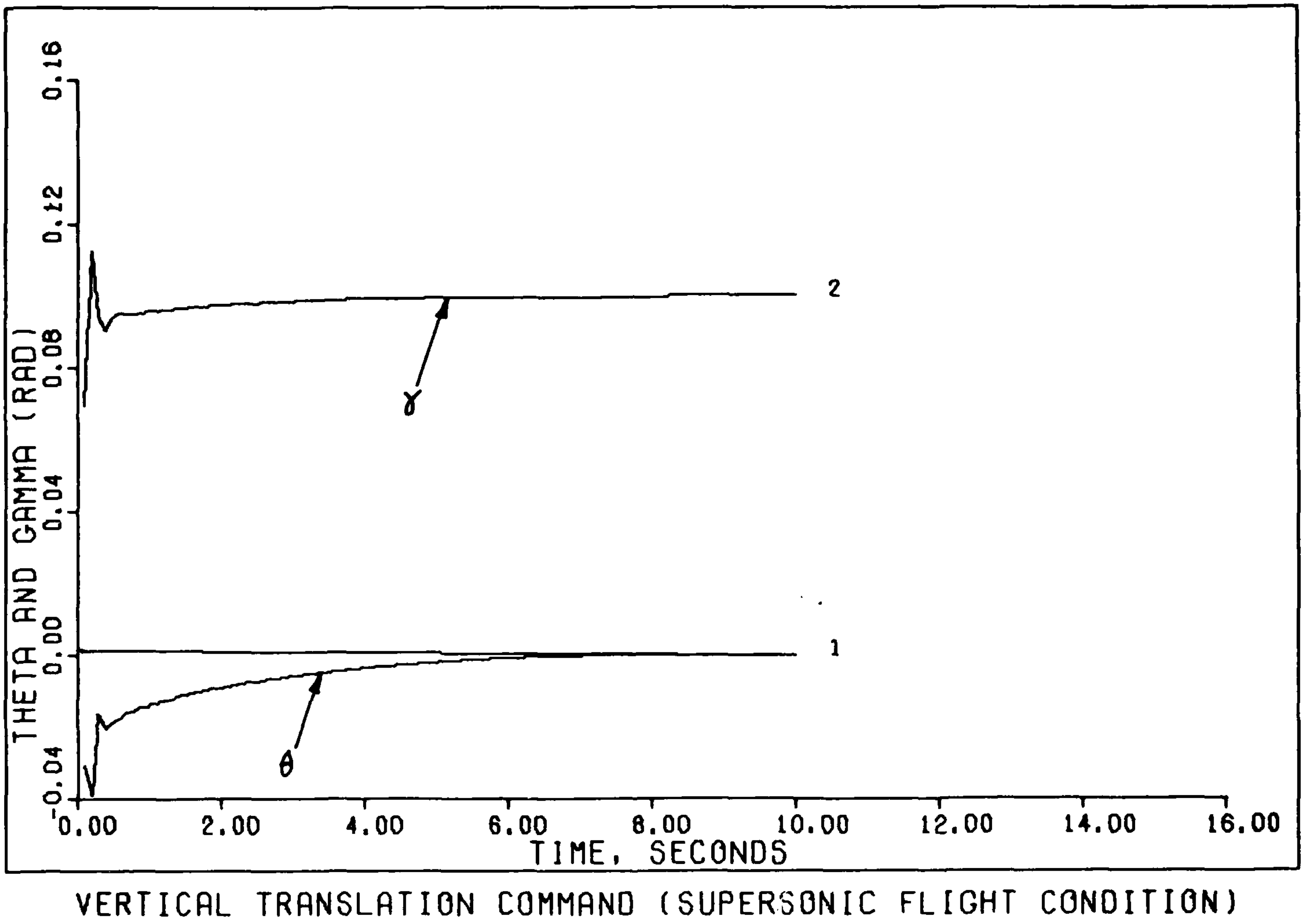


Figure 13d. Pitch and Flight Path Angles

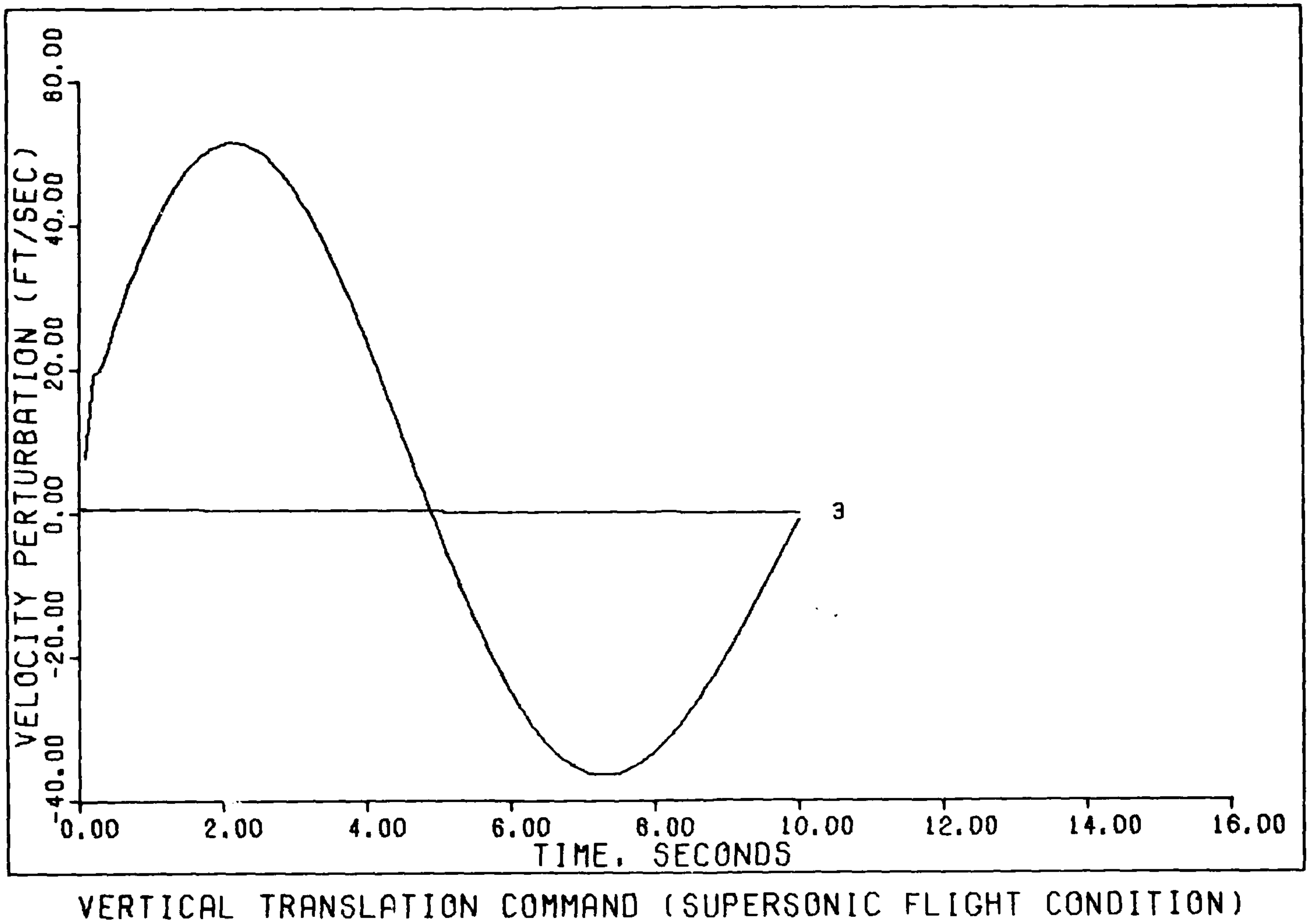


Figure 13e. Velocity Perturbation

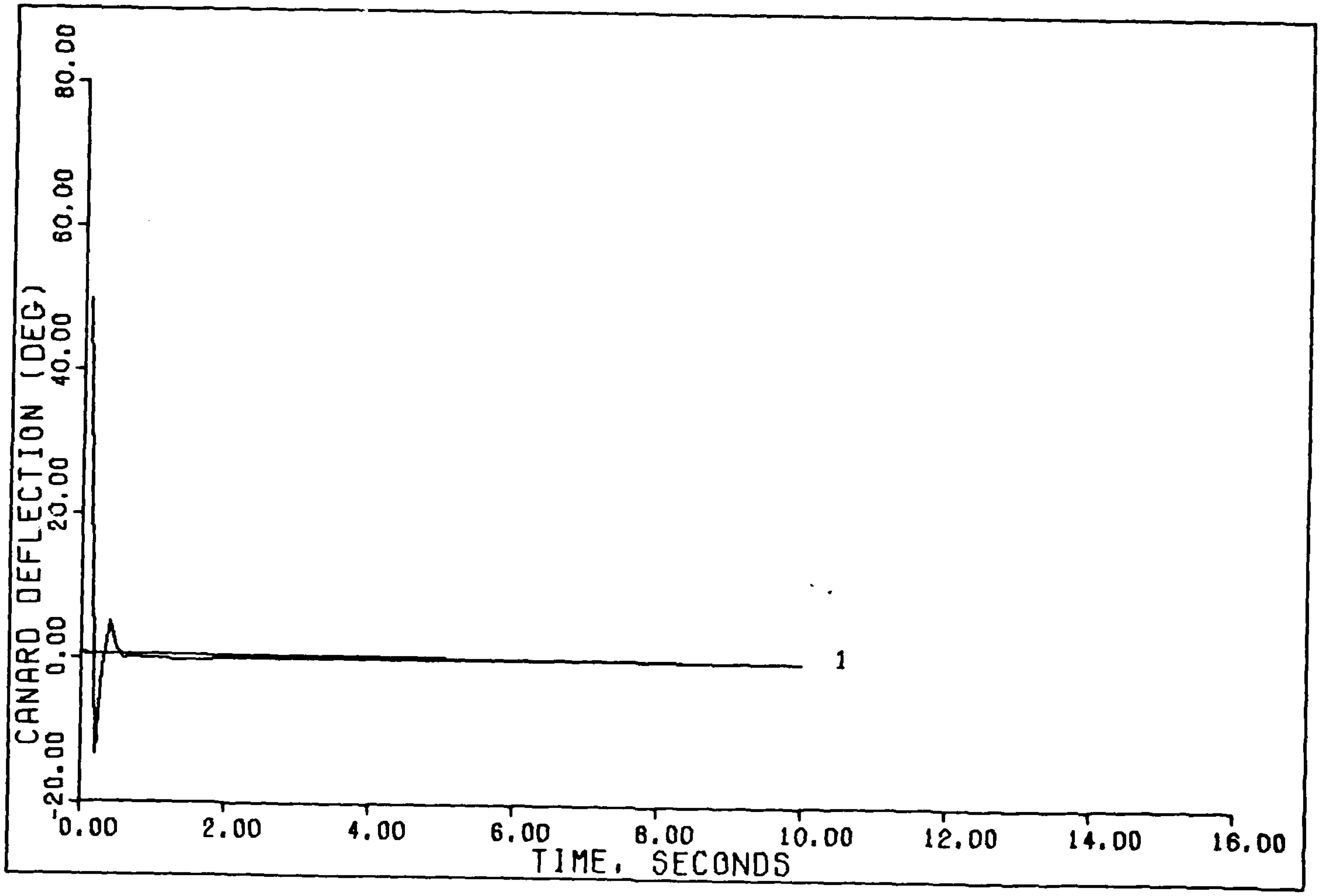


Figure 14a. Canard Deflection, Straight Climb Command (Supersonic Flight Condition)

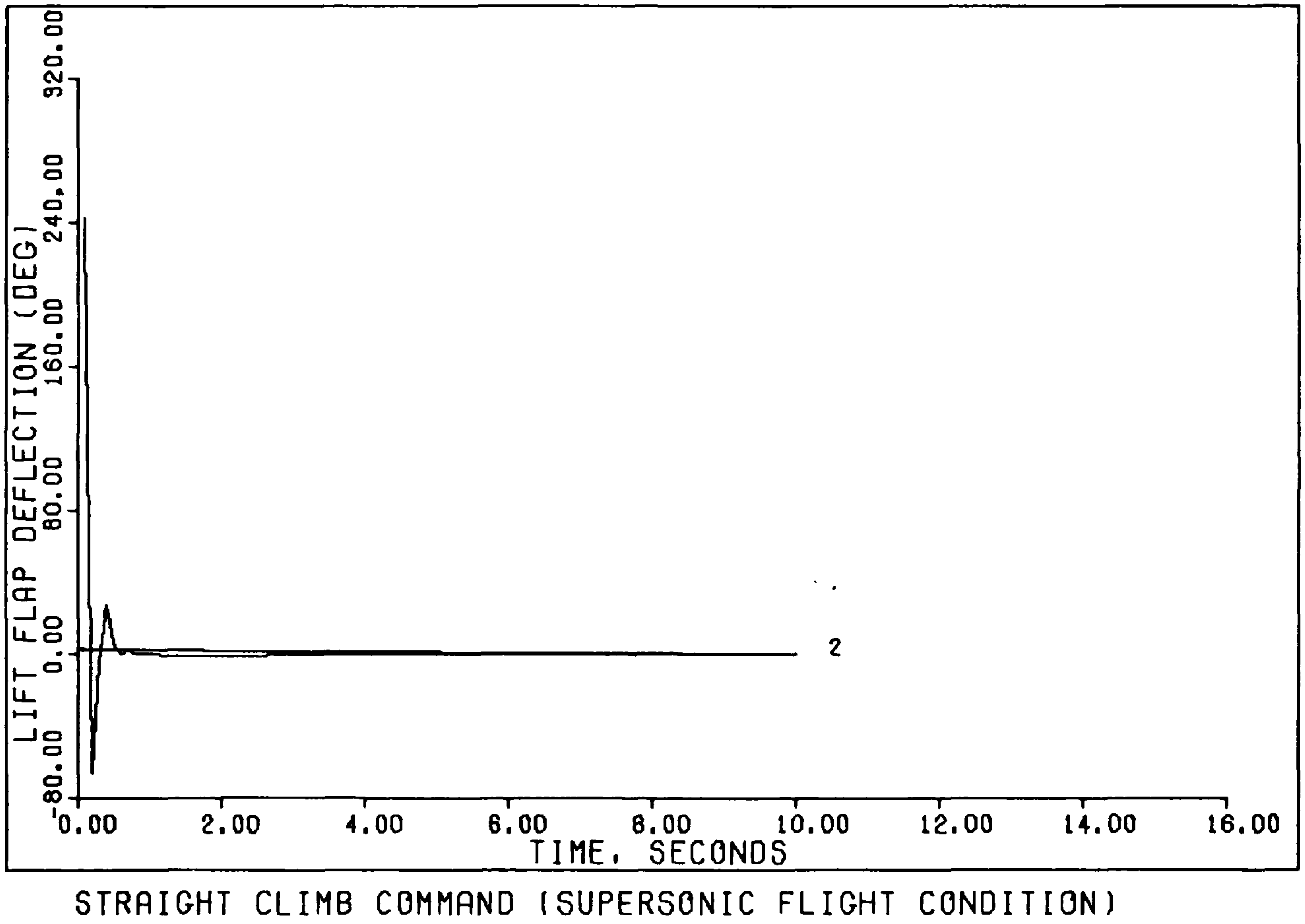
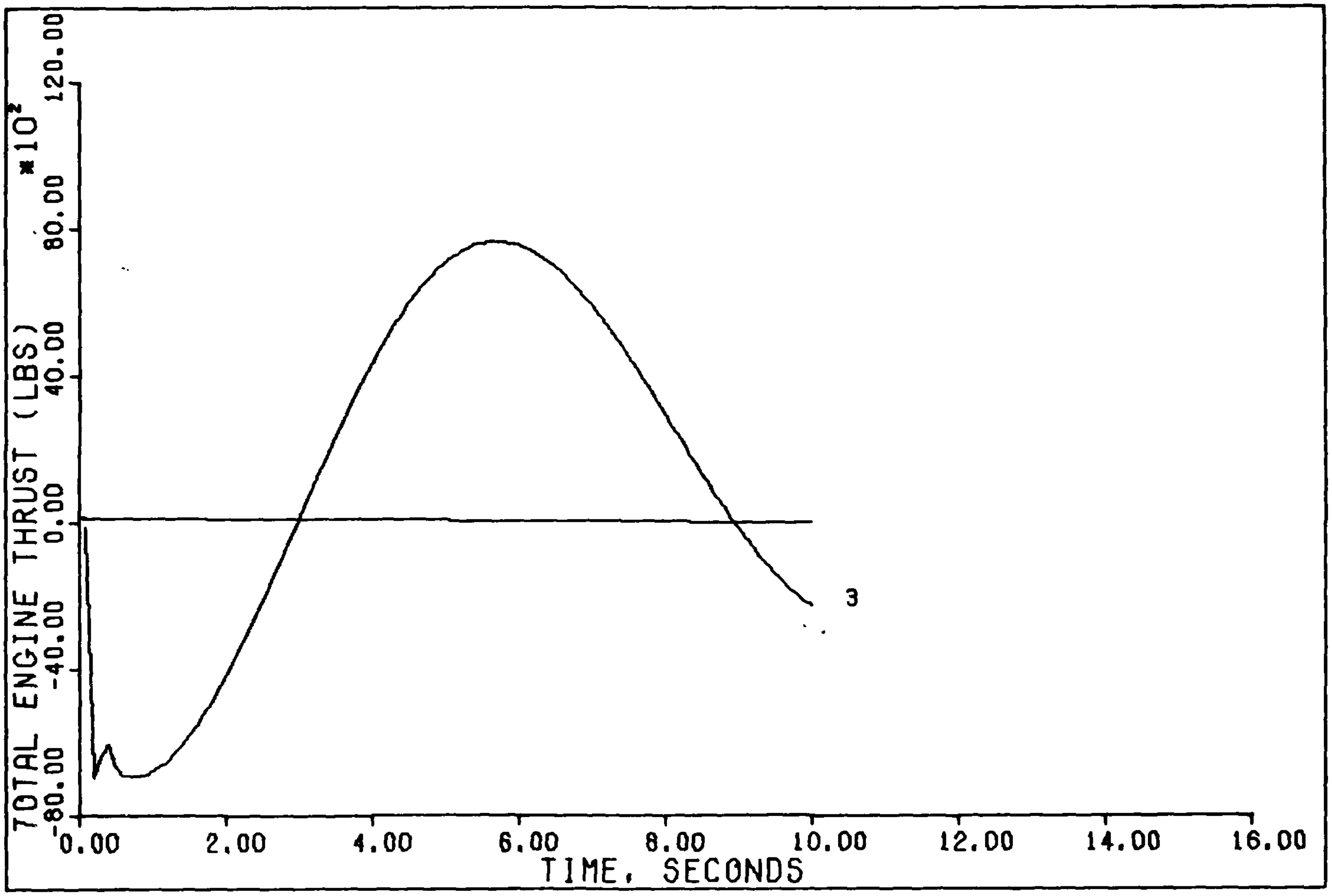
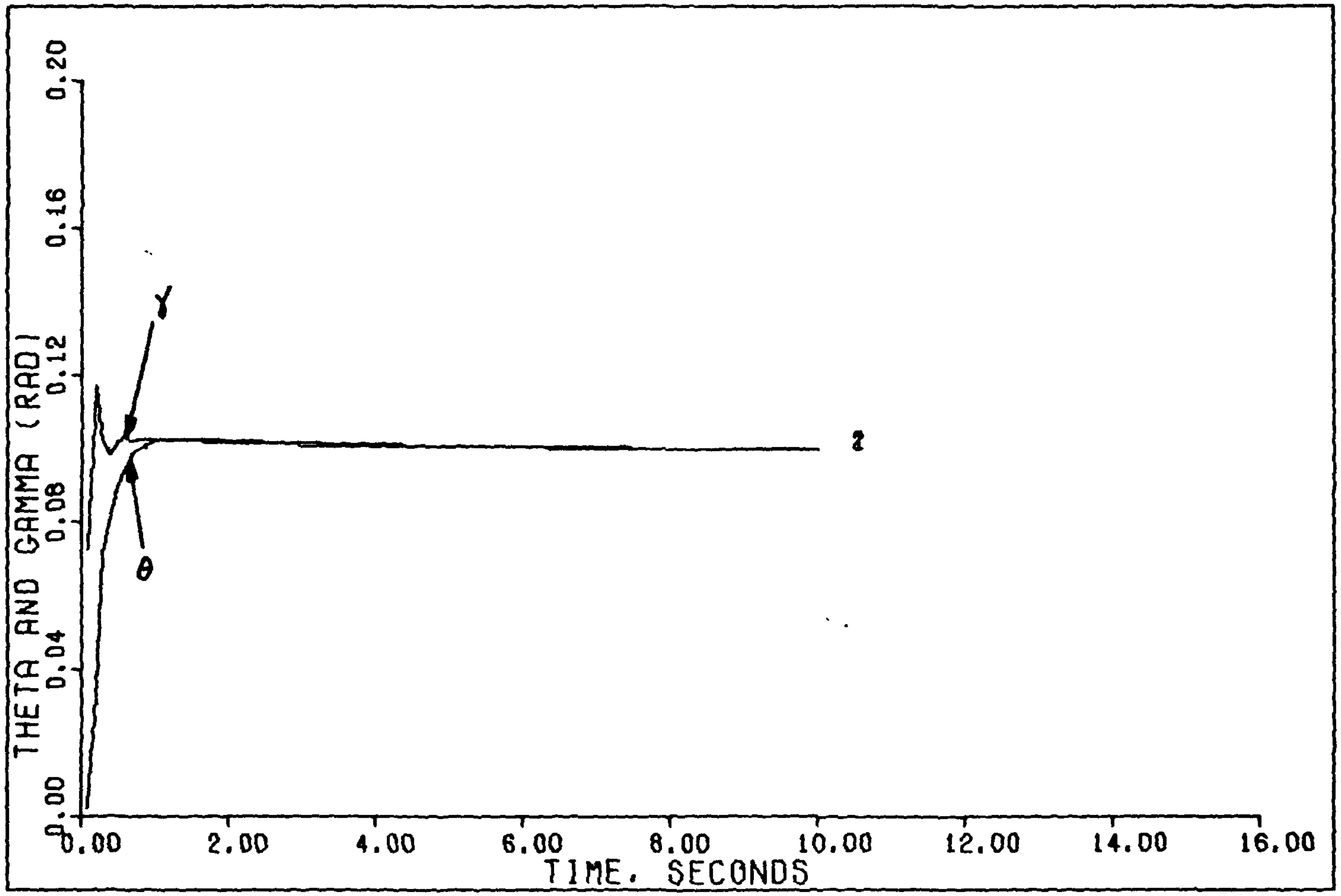


Figure 14b. Lift Flap Deflection



STRAIGHT CLIMB COMMAND (SUPERSONIC FLIGHT CONDITION)

Figure 14c. Total Engine Thrust



STRAIGHT CLIMB COMMAND (SUPERSONIC FLIGHT CONDITION)

Figure 14d. Pitch and Flight Path Angles

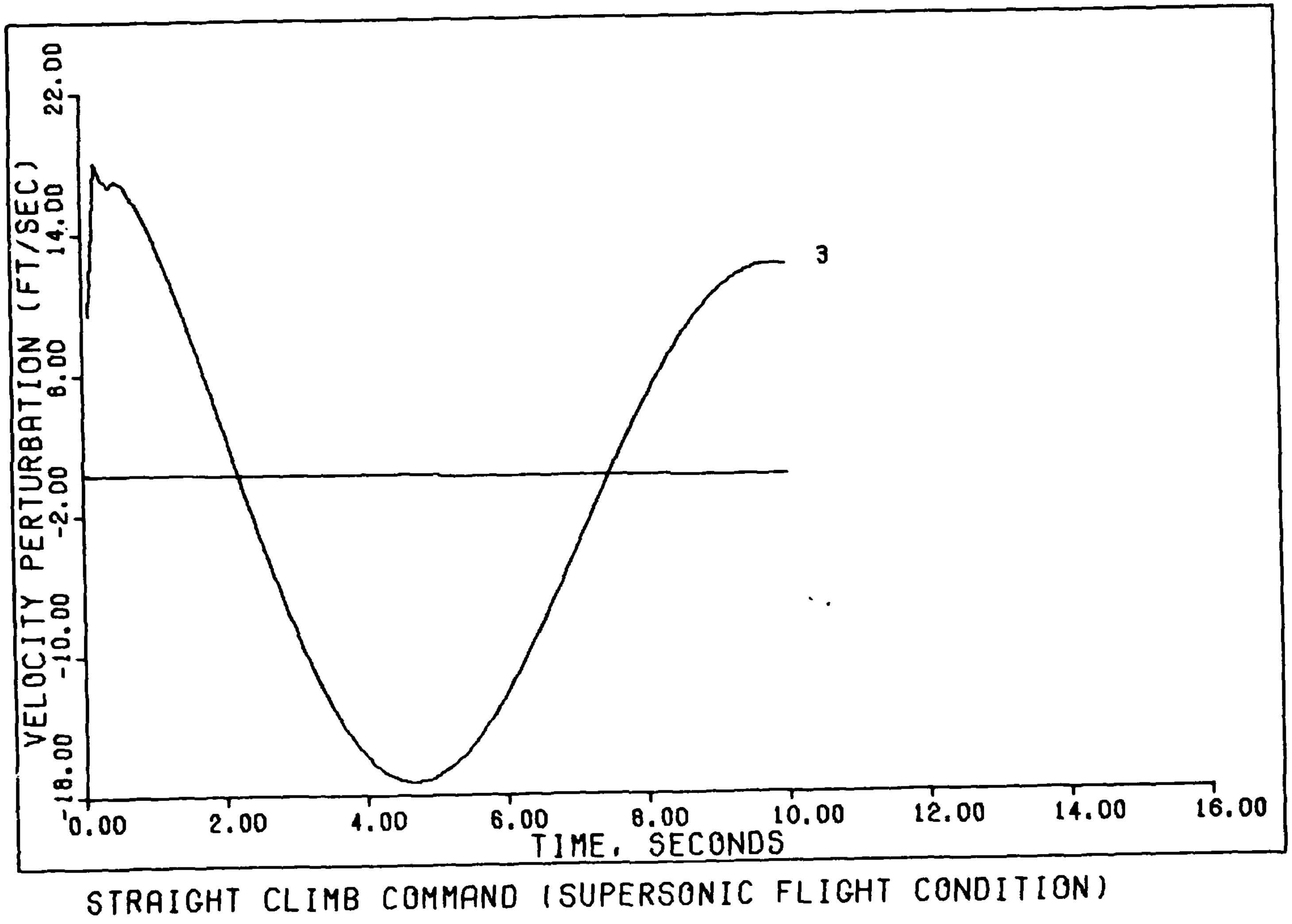


Figure 14e. Velocity Perturbation

variation from the nominal value and would be imperceptable to the pilot.

The steady state values for the inputs are:

CANARD: 0^{o}

LIFT FLAP: 0^{o}

The engine is still oscillatory at T=10 secs. The thrust perturbation; -7,000 lb to +7,600 lb from the nominal, is well within the limits of the engine. This would indicate that a 0.1 radian straight climb is quite feasible at this flight condition.

CONCLUSION

The results of this chapter confirm Porter's claim of robustness of the design method. The only limitation on performance is the physical limits of the aircraft. The use of a single controller could prove to be a radical departure from the current practice of gain-scheduling for the controller elements based on flight conditions.

A few words are in order concerning the size of the transients of the inputs, particularly in the vertical translation and straight climb modes. The lift flap initial deflections may reach 300^{o} in some instances, obviously out of the normal operating range. This is caused primarily by the command vector step inputs. These transients could be reduced or even eliminated by shaping orramping the inputs to the final value. This could be a final design parameter to tailor the input response to an acceptable form while maintaining output fidelity.

CHAPTER VI

EFFECTS OF INDIVIDUAL DESIGN PARAMETERS

INTRODUCTION

The design of the longitudinal tracker control law for the transonic flight condition presented in Chapter V is the result of a lengthy iterative design procedure. This chapter is written to assist any designer who wishes to develop a digital control law using the singular perturbation methods discussed in Chapter II. It is assumed that the designer has determined which type of plant and design he will use. It is further assumed that a first-cut design may have objectionable characteristics, such as excessive overshoot, and that the designer needs guidance in selectiting the various design parameters. The parameter changes presented here are compared to the base line figures of merit of the design presented in Chapter V. Since the design is based on an irregular plant, a designer using a regular or an unknown plant design does not need the measurement matrix information section. The conclusions concerning the other parameters is the same for all three design methods, except as noted.

One of the biggest factors in the design procedure is the presence of the servos. The servo dynamics are not included in the design due to the resulting complexity and difficulty of a proper choice of a measurement matrix. By excluding the servo dynamics in the design, the response is made slow enough so that the servo effects can not be seen (in the form of rapid transients) in the inputs and outputs. This is quite difficult with the engine model "servo" whose time constant is 1.77 sec.

EFFECTS OF DIFFERENT SAMPLING TIMES

In all three design methods, a smaller sampling time gives tighter control - less overshoot and shorter settling time. Three sample times are examined:

BASELINE $(T=0.01)$

SMALL $(T=0.005)$

LARGE $(T=0.05)$

In this design, however, smaller sample times resulted in an increase in the frequency and magnitude of the transients. This is unacceptable in terms of engine fatigue, since the number of peak thrust transients correlate to the decrease in engine life. The large sampling time of 0.05 sec causes the aircraft to go unstable for all three command vectors. Since the aircraft is unstable in the pitch axis, this sampling time appears to be too large to counter-act the instability. The figures of merit for all 3 sampling times and for all 3 command vectors are presented in Table V. Figure 15 is a plot of pitch and flight path angles for a positive pitch pointing command vector, $T = 0.005$ sec. Fig. 16 is the same plot but for $T = 0.05$ sec. These can be compared to the baseline in Fig. 6d.

EFFECT OF DIFFERENT MEASUREMENT MATRICES

In the irregular design, a proper choice of a measurement matrix yields satisfactory Z_2 roots as discussed in Chapter II. For the 4 state, 3 input, 3 output model developed for the transonic flight condition, a measurement matrix of the form:

$$M = [M_1, 0, 0]^T \qquad (80)$$

yields a single Z_2 root:

TABLE V EFFECTS OF DIFFERENT SAMPLING TIMES

COMMAND	INPUT OR	FIGURES OF	BASELINE	SMALLER	LARGER
VECTOR	OUTPUT	MERIT	Tsamp=0.01	Tsamp=0.005	Tsamp=0.05
POSITIVE PITCH POINTING	THETA	OVERSHOOT	17%	8%	Unstable
		SETTLING TIME	5.9sec	4.5 sec	Unstable
		# OF PEAKS	1 pk	1 pk	Unstable
	GAMMA	INTERACTION	12%	6%	Unstable
		# OF PEAKS	1 pk	1 pk	Unstable
	VELOCITY	MAX VALUES	-2.5 to 1.3	-1.5 to 0.7	Unstable
	CANARD	MAX VALUES	-3° to -16°	-3° to -28°	Unstable
		OF PEAKS	3 pks	6 pks	Unstable
	LIFT FLAP	MAX VALUES	1° to -75°	-2.5° to -73°	Unstable
		# OF PEAKS	1 pk	3 pks	Unstable
	F_{TOTAL}	MAX VALUES	1400 to -750	4080 to -1900	Unstable
		# OF PEAKS	3 pks	7 pks	Unstable
VERTICAL TRANS-LATION	GAMMA	OVERSHOOT	3%	9%	Unstable
		SETTLING TIME	3.9 sec	2.4 sec	Unstable
		# OF PEAKS	2 pks	2 pks	Unstable
	THETA	INTERACTION	-20%	-9%	Unstable
		# OF PEAKS	1 pk	1 pk	Unstable
	VELOCITY	MAX VALUES	-2.4 to 1.4	-2.3 to 1.7	Unstable
	CANARD	MAX VALUES	39° to 10°	26° to -6.7°	Unstable
		# OF PEAKS	2 pks	4 pks	Unstable
	LIFT FLAP	MAX VALUES	300° to 15°	180° to -105°	Unstable
		# OF PEAKS	2 pks	4 pks	Unstable
	F_{TOTAL}	MAX VALUES	6700 to -1400	5000 to -1500	Unstable
		# OF PEAKS	3 pks	6 pks	Unstable
STRAIGHT CLIMB	THETA	OVERSHOOT	Ø	Ø	Unstable
		SETTLING TIME	0.7sec	0.7 sec	Unstable
		# OF PEAKS	1 pk	1 pk	Unstable
	GAMMA	OVERSHOOT	9%	11%	Unstable
		SETTLING TIME	1.7 sec	0.5 sec	Unstable
		# OF PEAKS	2 pks	3 pks	Unstable
	VELOCITY	MAX VALUES	-4.1 to 2	-3 to 1.5	Unstable
	CANARD	MAX VALUES	36° to -9°	-101° to 100°	Unstable
		# OF PEAKS	2 pks	3 pks	Unstable
	LIFT FLAP	MAX VALUES	298° to -25°	124° to -168°	Unstable
		# OF PEAKS	4 pks	4 pks	Unstable
	F_{TOTAL}	MAX VALUES	7000 to -570	5400 to -3500	Unstable
		# OF PEAKS	4 pks	6 pks	Unstable

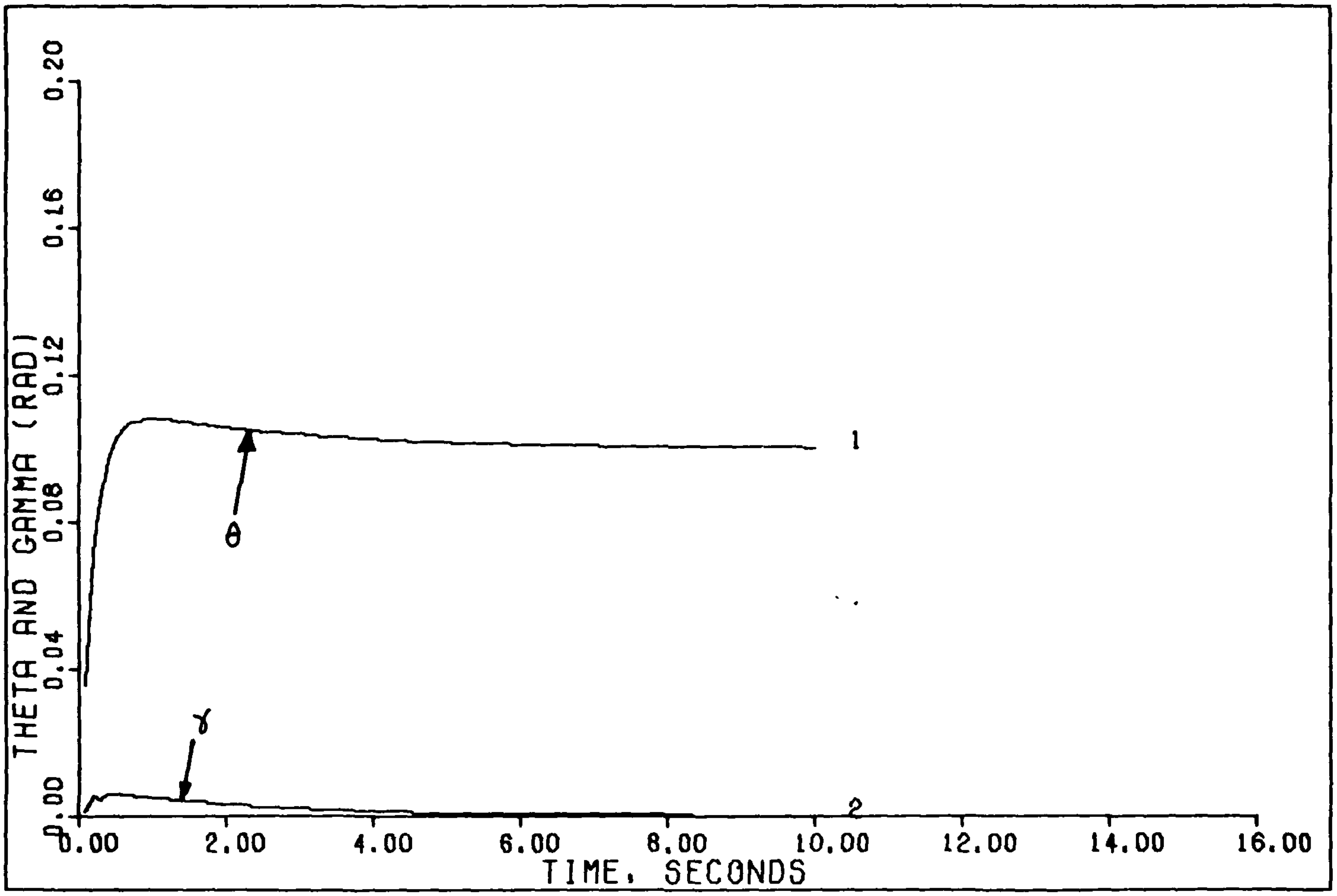


Figure 15. Pitch and Flight Path Angles, Positive Pitch Pointing Command (T=0.005)

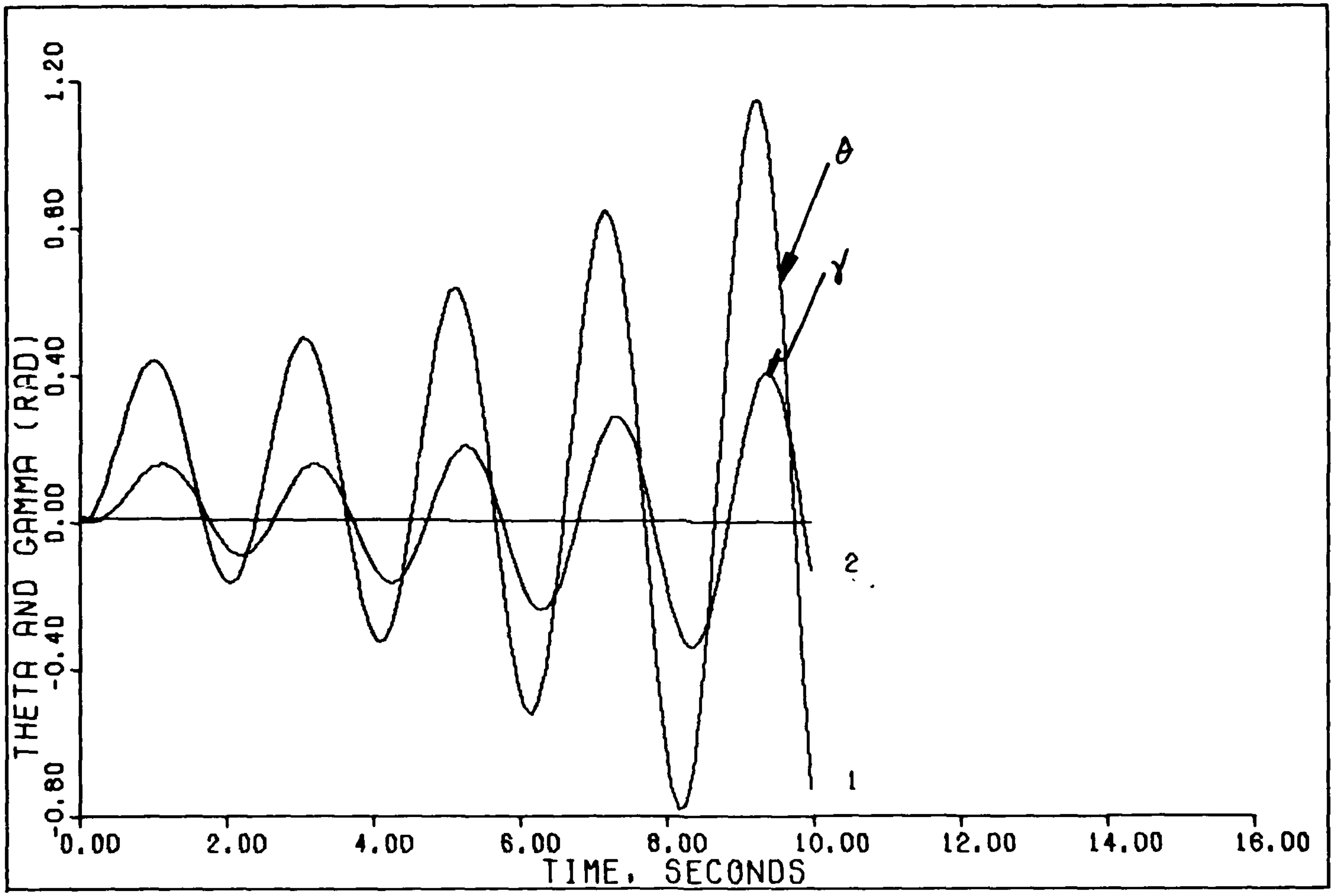


Figure 16. Pitch and Flight Path Angles, Positive Pitch Pointing Command (T=0.05)

$$Z_2 = \left[1 - \frac{Tsamp}{M_1} \right] \quad (81)$$

The baseline measurement matrix is:

$$M = [0.2,0,0]^{-T}$$

for T =0.01 sec, $Z_2 = 0.95$.

Two other measurement matrices were examined. Measurement matrix A is:

$$M_A = [0.25,0,0]^T$$

This yields $Z_2 = 0.96$ (for Tsamp = 0.1)

Measurement matrix B is:

$$M_B = [0.15,0,0]^T$$

which yields $Z_2 = 0.933$.

This tends to indicate that M_A will gives a slower response than the baseline. This is confirmed in the results of Table VI. Measurement matrix A produces a slower response with larger overshoots and settling times than the baseline response. Measurement matrix B gives marginally better responses than the baseline but at the expense of increased transients.

When choosing a measurement matrix, the designer may wish to place his Z_2 roots farther into the unit circle than the values used here. This can lead to stability problems in some cases. Due to the reciprocal nature of Eq. 81, a small value for M_1 places the Z_2 root close to the origin in the unit circle. This may be desirable, but the problem remains that the extra measurements generated by the measurement matrix may also be too small to be significant. Thus, the designer must optimize his choice of measurement matrix based on the "region of activity" in the Z-plane, usually close to the Z=1 point for small

TABLE VI EFFECTS OF DIFFERENT MEASUREMENT MATRICES

COMMAND VECTOR	INPUT OR OUTPUT	FIGURES OF MERIT	BASELINE $M=(.2,0,0)^T$	M_A $M=(.25,0,0)^T$	M_B $M=(.15,0,0)^T$
POSITIVE PITCH POINTING	THETA	OVERSHOOT	17%	22%	15%
		SETTLING TIME	5.9 sec	6.3 sec	5.4 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	INTERACTION	12%	12%	15%
		# OF PEAKS	1 pk	1 pk	1 pk
	VELOCITY	MAX VALUES	-2.5 to 1.3	-2.6 to 1.4	-2.5 to 1.2
	CANARD	MAX VALUES	-3° to -16°	-2° to -15°	-5° to -25°
		# OF PEAKS	3 pks	3 pks	5 pks
	LIFT FLAP	MAX VALUES	1° to -75°	-1° to -80°	-1° to -80°
		# OF PEAKS	1 pk	1 pk	3 pks
	F_{TOTAL}	MAX VALUES	1400 to -750	980 to -560	2380 to -1100
		# OF PEAKS	3 pks	2 pks	5 pks
VERTICAL TRANS-LATION	GAMMA	OVERSHOOT	3%	3%	3%
		SETTLING TIME	3.9 sec	3.8 sec	4.0 sec
		# OF PEAKS	2 pks	2 pks	2 pks
	THETA	INTERACTION	-20%	-25%	-18%
		# OF PEAKS	1 pk	1 pk	4 pks
	VELOCITY	MAX VALUES	-2.4 to 1.4	-2.3 to 1.4	-2.5 to 1.5
	CANARD	MAX VALUES	39° to 10°	39° to 9°	40° to 10°
		# OF PEAKS	2 pks	2 pks	2 pks
	LIFT FLAP	MAX VALUES	300° to 15°	300° to 15°	300° to 15°
		# OF PEAKS	2 pks	2 pks	2 pks
	F_{TOTAL}	MAX VALUES	6700 to -1400	6700 to -100	6700 to -210
		# OF PEAKS	3 pks	3 pks	4 pks
STRAIGHT CLIMB	THETA	OVERSHOOT	Ø	Ø	Ø
		SETTLING TIME	0.7 sec	0.9 sec	0.5 sec
		# OF PEAKS	1 pk	1 pk	2 pks
	GAMMA	OVERSHOOT	9%	8%	11%
		SETTLING TIME	1.7 sec	2.1 sec	1.4 sec
		# OF PEAKS	2 pks	2 pks	1 pk
	VELOCITY	MAX VALUES	-4.1 to 2	-4 to 1.9	-4.1 to 2.0
	CANARD	MAX VALUES	36° to -9°	37° to -3°	34° to -13°
		# OF PEAKS	2 pks	2 pks	2 pks
	LIFT FLAP	MAX VALUES	298° to -25°	297° to -15°	300° to -43°
		# OF PEAKS	4 pks	2 pks	2 pks
	F_{TOTAL}	MAX VALUES	7000 to -570	6900 to 160	7400 to -570
		# OF PEAKS	4 pks	4 pks	4 pks

sampling times. Fig. 17 shows the pitch and flight path angles for a vertical translation command vector and measurement matrix M_A. Fig. 18 is the same response but for measurement matrix M_B. These can be compared with Fig. 7d for the baseline response. These figures illustrate the effects of different measurement matrices.

EFFECTS OF DIFFERENT ALPHA PARAMETERS

Alpha is the proportion of direct to integral feedback. In the continuous domain, SISO case, alpha dictates the position of the zero in the lag-lead network formed by the proportional-plus-integral controller. In the digital domain, alpha has a primary influence on settling time with a secondary influence on overshoots. As α is increased, integral action is diminished and settling time increases. This is coupled with slightly lower overshoots in some cases. The reverse is also true. As α is decreased, settling time decreases but the lag contributed by increased integral action contributes to larger overshoots.

The choice of alpha in this design must be constrained. If α is chosen less than 2, the velocity grows as an exponential sinusoid. This may be due to the lag introduced by the engine model. Choosing alpha as 2.5 maintains stability and still achieves adequate performance. Two test cases are run, one with $\alpha = 1$ and the second with $\alpha = 4$. The results are presented in Table VII.

For $\alpha=1$, the overshoots are much larger than for the baseline results ($\alpha = 2.5$). The settling times are smaller and the velocity perturbation is a growing exponential sinusoid.

For $\alpha = 4$, the settling times are larger and most overshoots are smaller than the baseline results. Thus, the choice of $\alpha = 2.5$ seems to

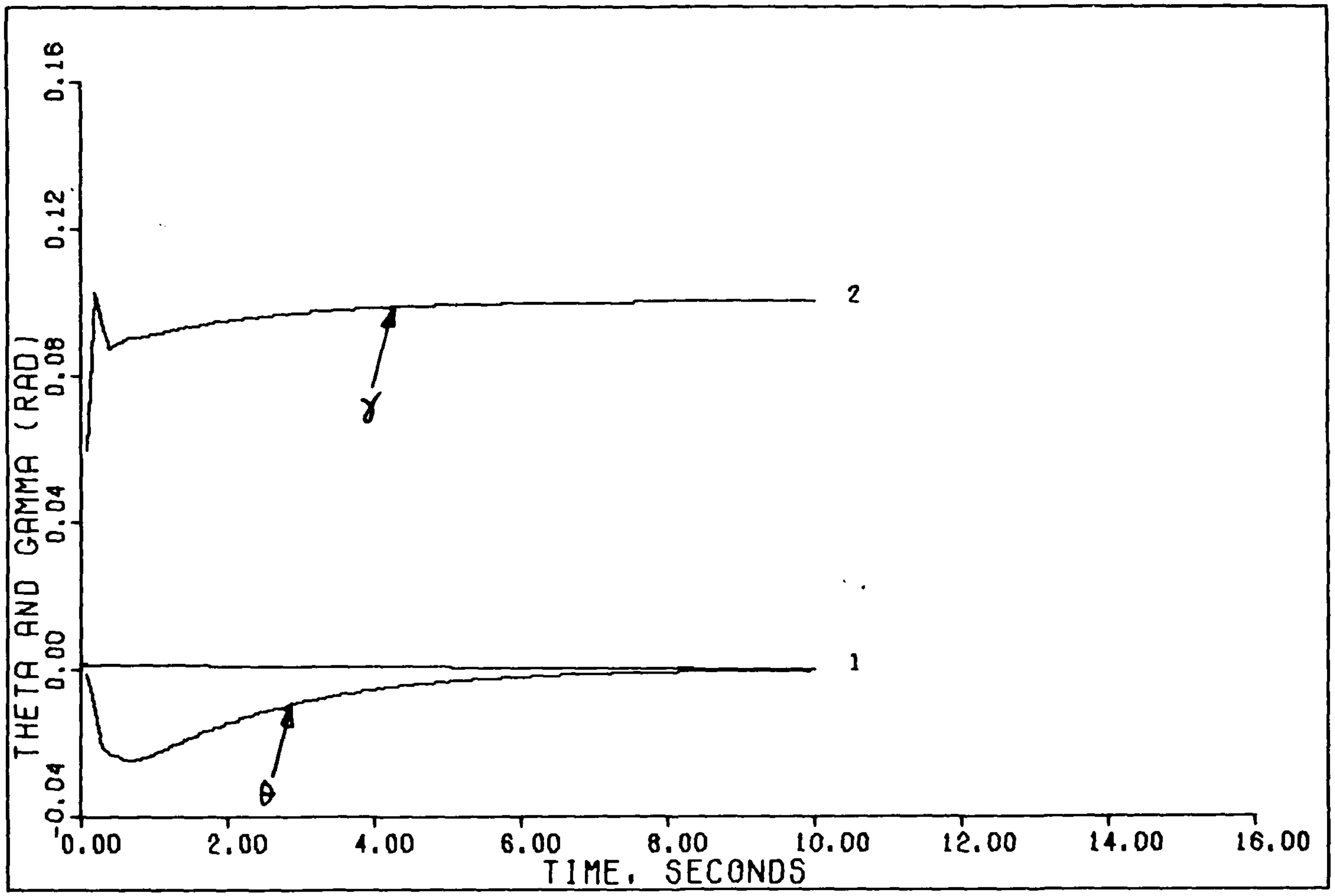


Figure 17. Pitch and Flight Path Angles, Vertical Translation Command ($M=[0.25,0,0]^T$)

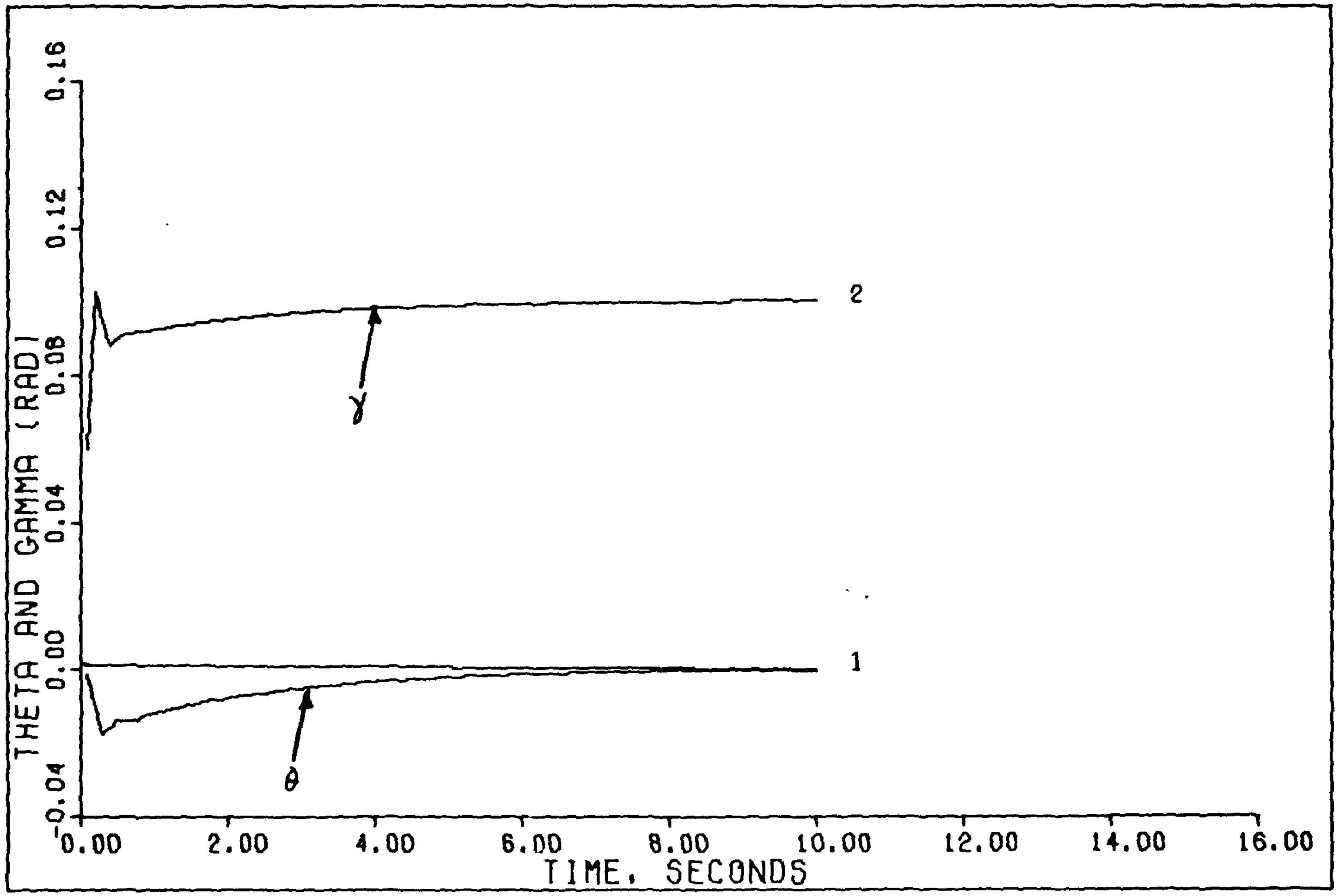


Figure 18. Pitch and Flight Path Angles, Vertical Translation Command ($M=[0.15,0,0]^T$)

TABLE VII EFFECTS OF ALPHA PARAMETER CHANGES

COMMAND VECTOR	INPUT OR OUTPUT	FIGURES OF MERIT	BASELINE α=2.5	LOWER α=1.0	HIGHER α=4.0
POSITIVE PITCH POINTING	THETA	OVERSHOOT	17%	68%	11%
		SETTLING TIME	5.9 sec	3.2 sec	7.8 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	INTERACTION	12%	36%	8%
		# OF PEAKS	1 pk	1 pk	6 pks
	VELOCITY	MAX VALUES	-2.5 to 1.3	Unstable	-2 to 0.3
	CANARD	MAX VALUES	-3° to -16°	9.6° to -19°	18° to -10°
		# OF PEAKS	3 pks	1 pk	1 pk
	LIFT FLAP	MAX VALUES	1° to -75°	-2° to -102°	0.3° to -74°
		# OF PEAKS	1 pk	1 pk	1 pk
	F_{TOTAL}	MAX VALUES	1400 to -750	1260 to -1180	2800 to -670
		# OF PEAKS	3 pks	3 pks	5 pks
VERTICAL TRANSLATION	GAMMA	OVERSHOOT	3%	1 %	15%
		SETTLING TIME	3.9 sec	1.7 sec	5 sec
		# OF PEAKS	2 pks	1 pk	2 pks
	THETA	INTERACTION	-20%	-58%	-13%
		# OF PEAKS	1 pk	1 pk	1 pk
	VELOCITY	MAX VALUES	-2.4 to 1.4	Unstable	-2.6 to 1.1
	CANARD	MAX VALUES	39° to 10°	24° to 14°	37° to -10°
		# OF PEAKS	2 pks	2 pks	4 pks
	LIFT FLAP	MAX VALUES	300° to 15°	140° to 70°	252° to -153°
		# OF PEAKS	2 pks	2 pks	3 pks
	F_{TOTAL}	MAX VALUES	6700 to -1400	4700 to 300	5700 to -3400
		# OF PEAKS	3 pks	4 pks	7 pks
STRAIGHT CLIMB	THETA	OVERSHOOT	Ø	12%	Ø
		SETTLING TIME	0.7 sec	1.4 sec	0.7 sec
		# OF PEAKS	1 pk	2 pks	1 pk
	GAMMA	OVERSHOOT	9 %	15%	21%
		SETTLING TIME	1.7 sec	3 sec	0.7 sec
		# OF PEAKS	2 pks	1 pk	3 pks
	VELOCITY	MAX VALUES	-4.1 to 2	Unstable	-3.4 to 0.6
	CANARD	MAX VALUES	36° to -9°	28° to 0°	18° to -20°
		# OF PEAKS	2 pks	2 pks	4 pks
	LIFT FLAP	MAX VALUES	298° to -25°	188° to -0.5°	255° to -204°
		# OF PEAKS	4 pks	2 pks	3 pks
	F_{TOTAL}	MAX VALUES	7000 to -570	3450 to -150	8500 to -3900
		# OF PEAKS	4 pks	2 pks	5 pks

be the best trade off between settling times, overshoots and stability for the system. Figs. 19 and 20 show flight path aud pitch angles, for a positive pitch pointing command vector (for $\alpha = 1$ and $\alpha = 4$). These can be compared to the baseline performance seen in Fig. 6d.

EFFECT OF DIFFERENT EPSILON PARAMETERS

Epsilon is a scalar parameter that multiplies each element in the output weighting marix. By increasing ϵ, the entire Σ weighting matrix is increased, leading to quicker and tighter control: faster setting times, lower overshoots. If ϵ is decreased, the control responses are characterized by slower settling times and larger overshoots.

Two values of epsilon are used, a larger value, ($\epsilon = 0.2$), and a smaller value, ($\epsilon = 0.05$), than the baseline value, ($\epsilon = 0.1$). The results of the two trials are compared to the baseline results in Table VIII. For $\epsilon = 0.2$, the result is tighter control in the form of lower overshoots, interaction and settling times. This is coupled with the previously mentioned problems associated with tighter control: an increase in the frequency and magnitude of initial transients. For epsilon = 0.05, looser control is seen in the form of large overshoots and larger settling times. These results are seen in Figs. 21 and 22. Fig. 21 shows flight path and pitch angles, for a positive pitch pointing command vector and $\epsilon = 0.2$. Fig. 22 is the same, but for $\epsilon = 0.05$. These figures can be compared to the baseline plot of Fig. 6d.

EFFECTS OF SIGMA MATRIX ELEMENT CHANGES

The sigma matrix determines the position of the closed loop fast roots (Z_3) discussed in Chapter II. Each element on the diagonal has a direct influence on the response of its corresponding output. In a fast

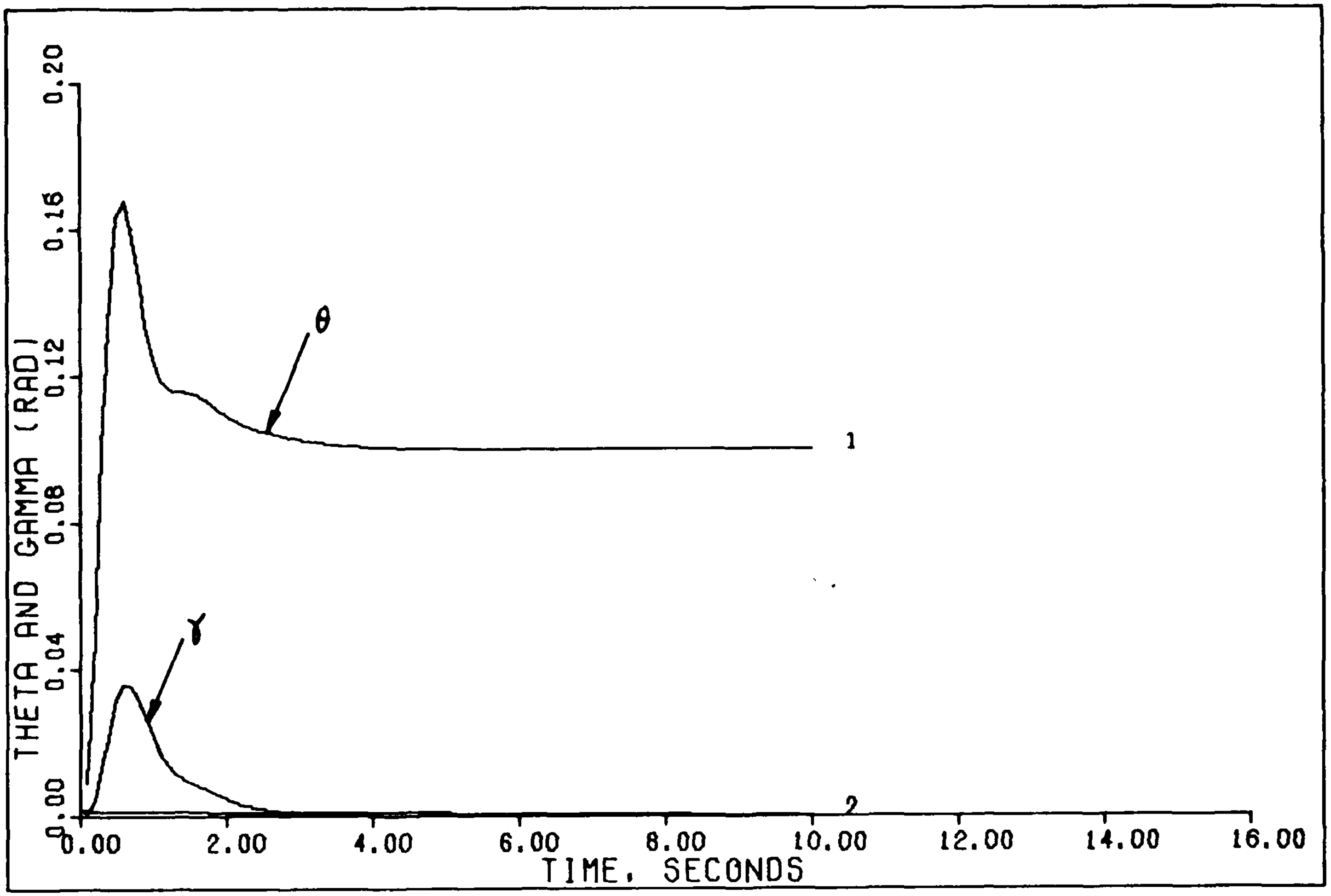


Figure 19. Pitch and Flight Path Angles, Positive Pitch Pointing Command (α=1)

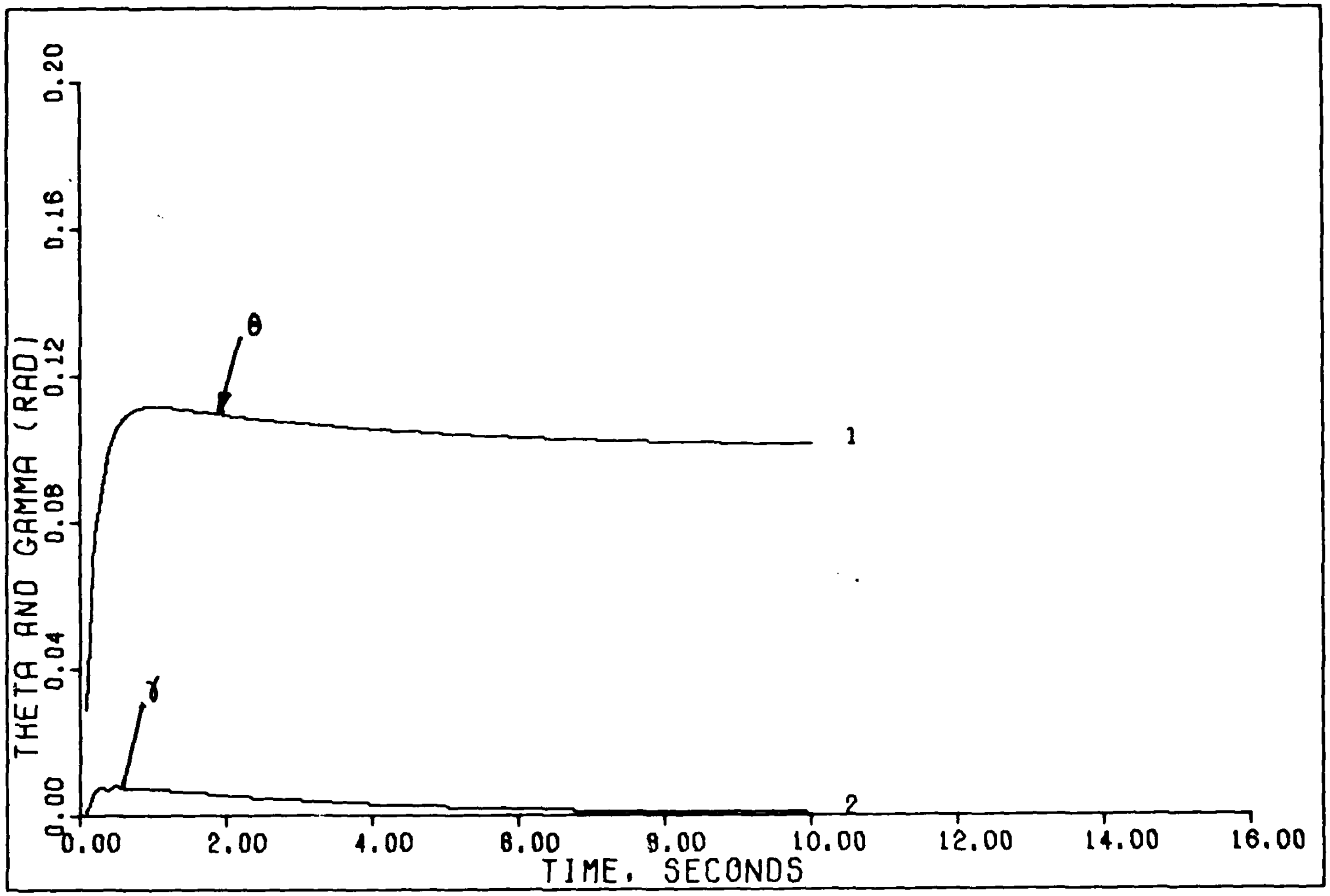


Figure 20. Pitch and Flight Fath Angles, Positive Pitch Pointing Commands (α=4)

TABLE VIII EFFECTS OF EPSILON PARAMETER CHANGES

COMMAND VECTOR	INPUT OR OUTPUT	FIGURES OF MERIT	BASELINE	LARGER	SMALLER
			∈ =0.1	∈ =0.2	∈ =0.05
POSITIVE PITCH POINTING	THETA	OVERSHOOT	17%	8%	49%
		SETTLING TIME	5.9 sec	4.5 sec	7.1 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	INTERACTION	12%	6.3%	30%
		# OF PEAKS	1 pk	1 pk	1 pk
	VELOCITY	MAX VALUES	-2.5 to 1.3	-1.5 to 0.7	-4.1 to 1.7
	CANARD	MAX VALUES	-3° to -16°	-2° to -30°	4° to -17°
		# OF PEAKS	3 pks	5 pks	1 pk
	LIFT FLAP	MAX VALUES	1° to -75°	1° to -75°	-1.8° to -88°
		# OF PEAKS	1 pk	5 pks	1 pk
	F_{TOTAL}	MAX VALUES	1400 to -750	4400 to -2300	900 to -840
		# OF PEAKS	3 pks	5 pks	3 pks
VERTICAL TRANSLATION	GAMMA	OVERSHOOT	3%	11%	Ø
		SETTLING TIME	3.9 sec	2.4 sec	4.9 sec
		# OF PEAKS	2 pks	5 pks	2 pks
	THETA	INTERACTION	-20%	-10%	-48%
		# OF PEAKS	1 pk	3 pks	1 pk
	VELOCITY	MAX VALUES	-2.4 to 1.4	-2.5 to 1.7	-2.2 to 0.7
	CANARD	MAX VALUES	39° to 10°	34° to -21°	27° to 13°
		# OF PEAKS	2 pks	6 pks	2 pks
	LIFT FLAP	MAX VALUES	300° to 15°	253° to -209°	212° to 70°
		# OF PEAKS	2 pks	5 pks	2 pks
	F_{TOTAL}	MAX VALUES	6700 to -1400	6700 to -3900	4760 to 810
		# OF PEAKS	3 pks	5 pks	2 pks
STRAIGHT CLIMB	THETA	OVERSHOOT	Ø	Ø	Ø
		SETTLING TIME	0.7 sec	0.7 sec	1 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	OVERSHOOT	9%	18%	5%
		SETTLING TIME	1.7 sec	0.7 sec	3.6 sec
		# OF PEAKS	2 pks	5 pks	1 pk
	VELOCITY	MAX VALUES	-4.1 to 2	-3.1 to 1.5	-5.1 to 2.4
	CANARD	MAX VALUES	36° to -9°	16° to -20°	31° to 0°
		# OF PEAKS	2 pks	5 pks	2 pks
	LIFT FLAP	MAX VALUES	298° to -25°	196° to -272°	210° to -.02°
		# OF PEAKS	4 pks	6 pks	2 pks
	F_{TOTAL}	MAX VALUES	7000 to -570	7900 to -6200	4025 to 530
		# OF PEAKS	4 pks	5 pks	2 pks

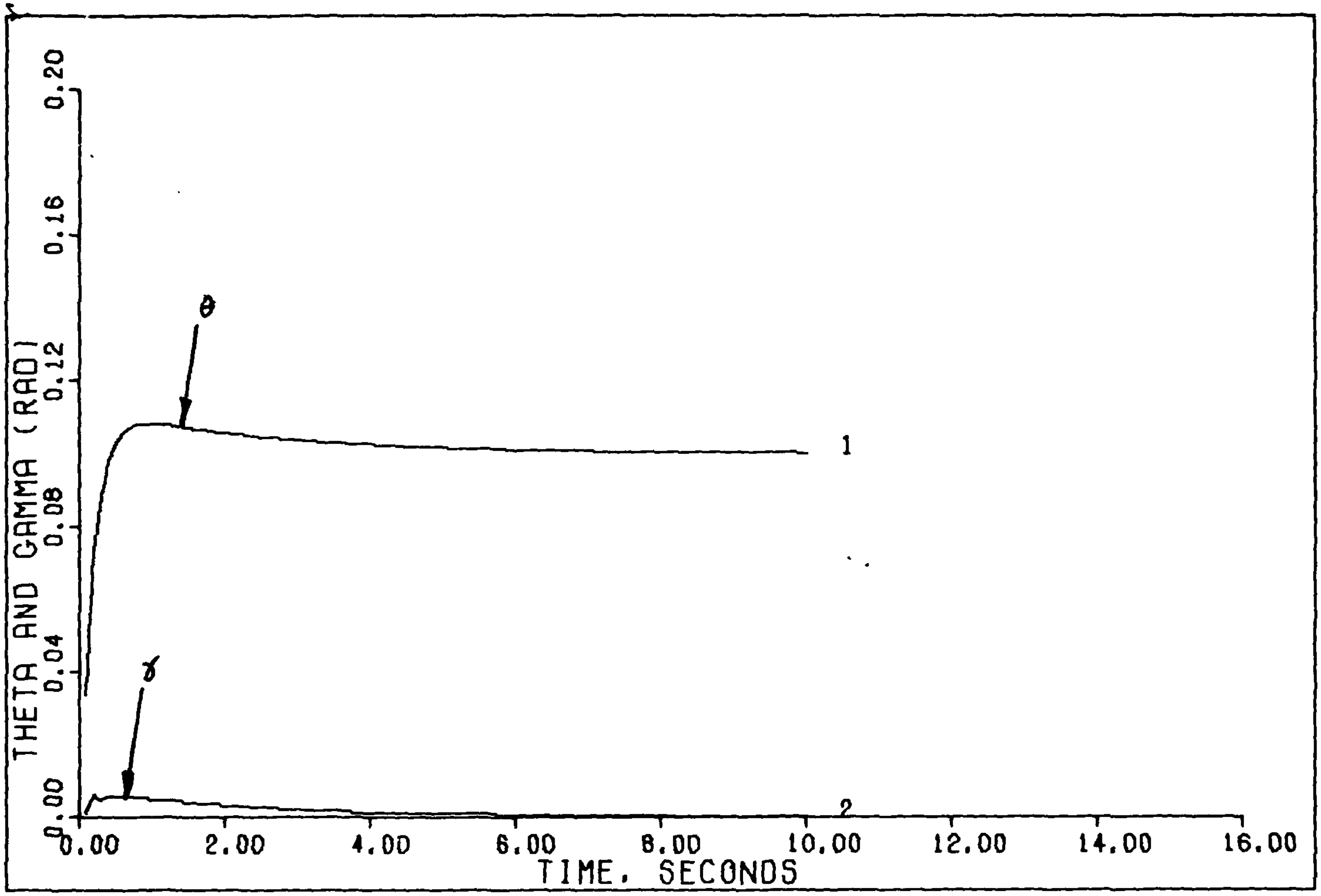


Figure 21. Pitch and Flight Path Angles, Positive Pitch Pointing Command (ϵ =0.2)

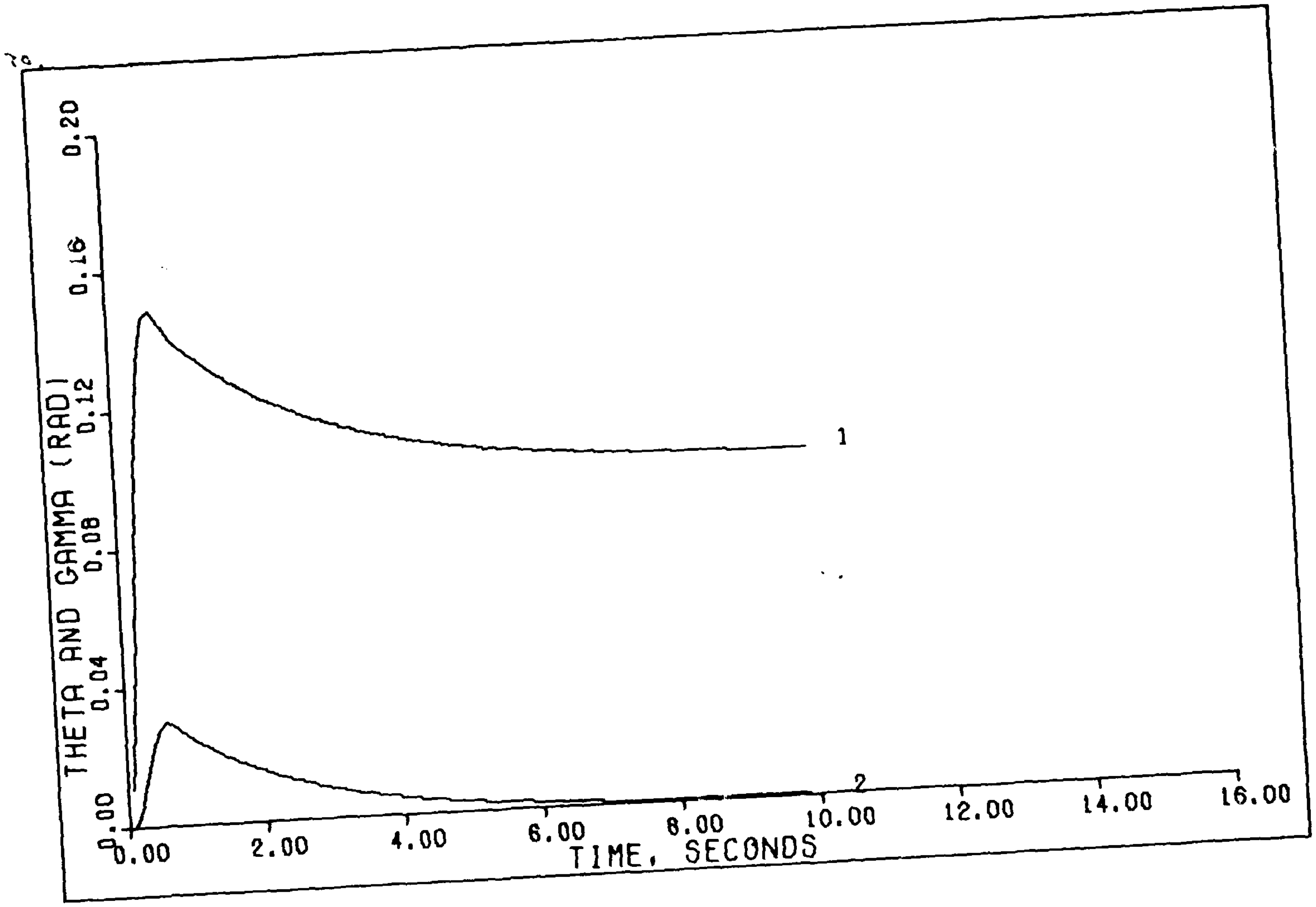


Figure 22. Pitch and Flight Path Angles, Positive Pitch Pointing Command (ϵ =0.05)

sampling environment there is a minimum effect on the other outputs due to the decoupling at high sample rates.

If a single element in the Σ matrix is changed, the corresponding output responds in a similar manner discussed previously in the epsilon parameter change section. If the element, σ, is increased, tighter control and faster response occurs. If the element is decreased, a slower response will occur. Changing an individual element also changes the input that has the most direct influence on the specific output. This is illustrated by changing the third element, σ_3, in the matrix. Two values are used to determine the effects on the system - higher value, ($\sigma_3 = 0.06$), and a lower value, ($\sigma_3 = 0.015$), than the baseline value, ($\sigma_3 = 0.03$). The results are presented in Table IX.

It is readily apparent that the σ_3 element affects only its corresponding output (velocity). The only input affected is the thrust. This indicates a high degree of decoupling for the system. For $\sigma_3 = 0.06$, the velocity perturbation is smaller in all cases, coupled with an increase in initial transients in the Ftotal input. This agrees with the previous statements concerning tighter control and fast response. For $\sigma_3 = 0.015$, the velocity perturbations are larger and thrust transients are smaller. This also agrees with the previous conclusions concerning looser control and slower response. These results are seen graphically in Figs. 23 and 24. Fig. 23 is the velocity response for a straight climb vector and $\sigma_3 = 0.06$. Figure 24 is the same response but for $\sigma_3 = 0.015$. These can be compared to the baseline in Fig. 8e.

CONCLUSION

The design process to develop a digital control design using

TABLE IX EFFECTS OF SIGMA MATRIX ELEMENT CHANGES

COMMAND VECTOR	INPUT OR OUTPUT	FIGURES OF MERIT	BASELINE σ_3=0.03	HIGHER σ_3=0.06	LOWER σ_3=0.015
POSITIVE PITCH POINTING	THETA	OVERSHOOT	17%	18%	18%
		SETTLING TIME	5.9 sec	6 sec	6 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	INTERACTION	12%	12%	12%
		# OF PEAKS	1 pk	1 pk	1 pk
	VELOCITY	MAX VALUES	-2.5 to 1.3	-1.6 to 0.8	-3.8 to 1.1
	CANARD	MAX VALUES	-3° to -16°	-3° to -16°	-3° to -16°
		# OF PEAKS	3 pks	3 pks	3 pks
	LIFT FLAP	MAX VALUES	1° to -75°	-1° to -77°	-1° to -77°
		# OF PEAKS	1 pk	2 pks	2 pks
	F_{TOTAL}	MAX VALUES	1400 to -750	1400 to -590	1500 to -800
		# OF PEAKS	3 pks	6 pks	5 pks
VERTICAL TRANSLATION	GAMMA	OVERSHOOT	3%	3%	3%
		SETTLING TIME	3.9 sec	3.9 sec	3.9 sec
		# OF PEAKS	2 pks	2 pks	2 pks
	THETA	INTERACTION	-20%	-20%	-20%
		# OF PEAKS	1 pk	1 pk	1 pk
	VELOCITY	MAX VALUES	-2.4 to 1.4	-2.3 to 1.6	-2.8 to 1.4
	CANARD	MAX VALUES	39° to 10°	39° to 10°	39° to 10°
		# OF PEAKS	2 pks	2 pks	2 pks
	LIFT FLAP	MAX VALUES	300° to 15°	300° to 14°	300° to 15°
		# OF PEAKS	2 pks	2 pks	2 pks
	F_{TOTAL}	MAX VALUES	6700 to -1400	7000 to 160	6500 to -540
		# OF PEAKS	3 pks	4 pks	4 pks
STRAIGHT CLIMB	THETA	OVERSHOOT	Ø	Ø	Ø
		SETTLING TIME	0.7 sec	0.7 sec	0.7 sec
		# OF PEAKS	1 pk	1 pk	1 pk
	GAMMA	OVERSHOOT	9%	9%	9%
		SETTLING TIME	1.7 sec	1.8 sec	1.8 sec
		# OF PEAKS	2 pks	2 pks	2 pks
	VELOCITY	MAX VALUES	-4.1 to 2	-3.1 to 1.6	-5.7 to 2.4
	CANARD	MAX VALUES	36° to -9°	36° to -9°	-36° to -8°
		# OF PEAKS	2 pks	2 pks	2 pks
	LIFT FLAP	MAX VALUES	298° to -25°	298° to -27°	298° to -27°
		# OF PEAKS	4 pks	2 pks	2 pks
	F_{TOTAL}	MAX VALUES	7000 to -570	7300 to 300	6900 to -525
		# OF PEAKS	4 pks	4 pks	4 pks

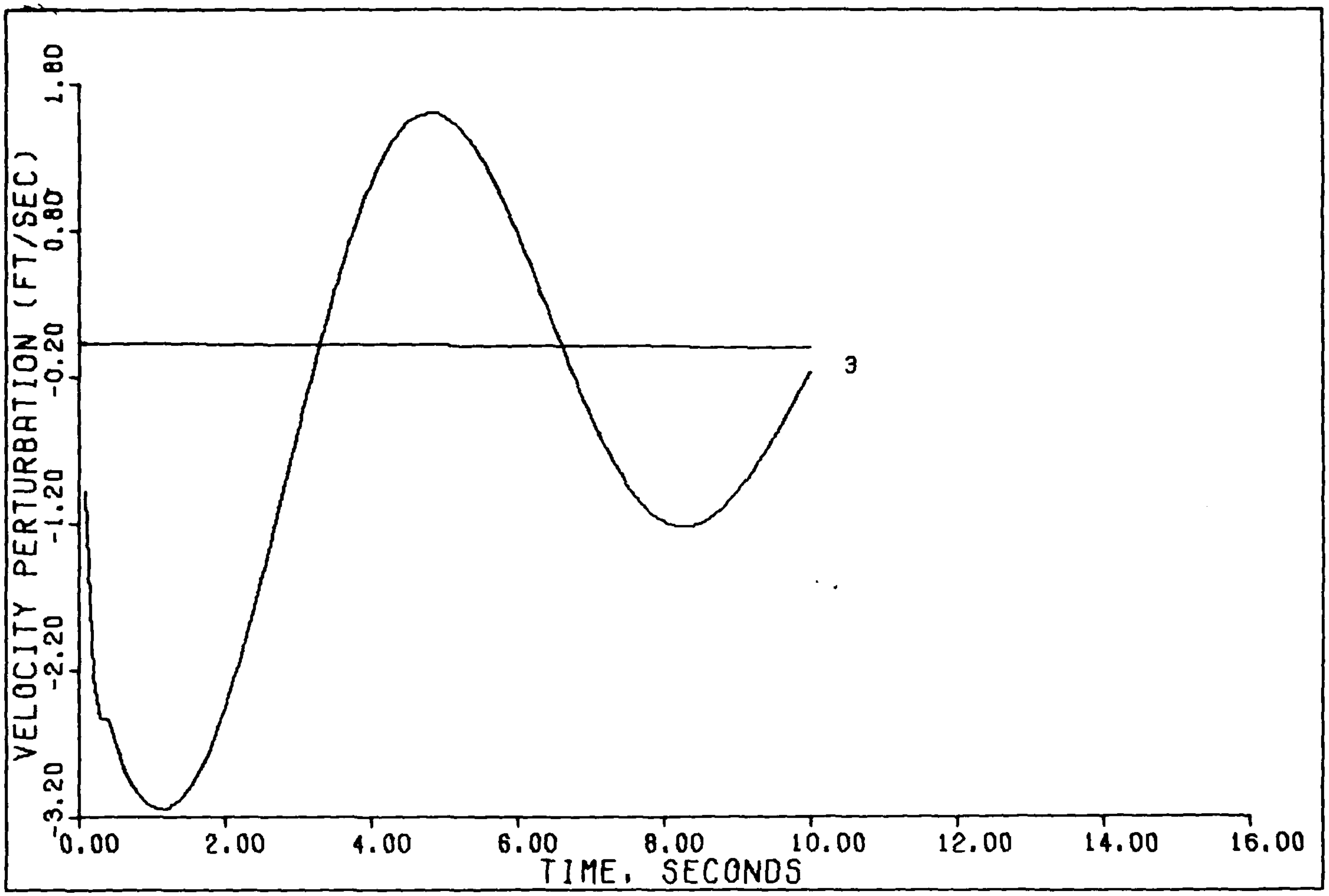


Figure 23. Velocity Perturbation, Straight Climb Command (σ_3=0.06)

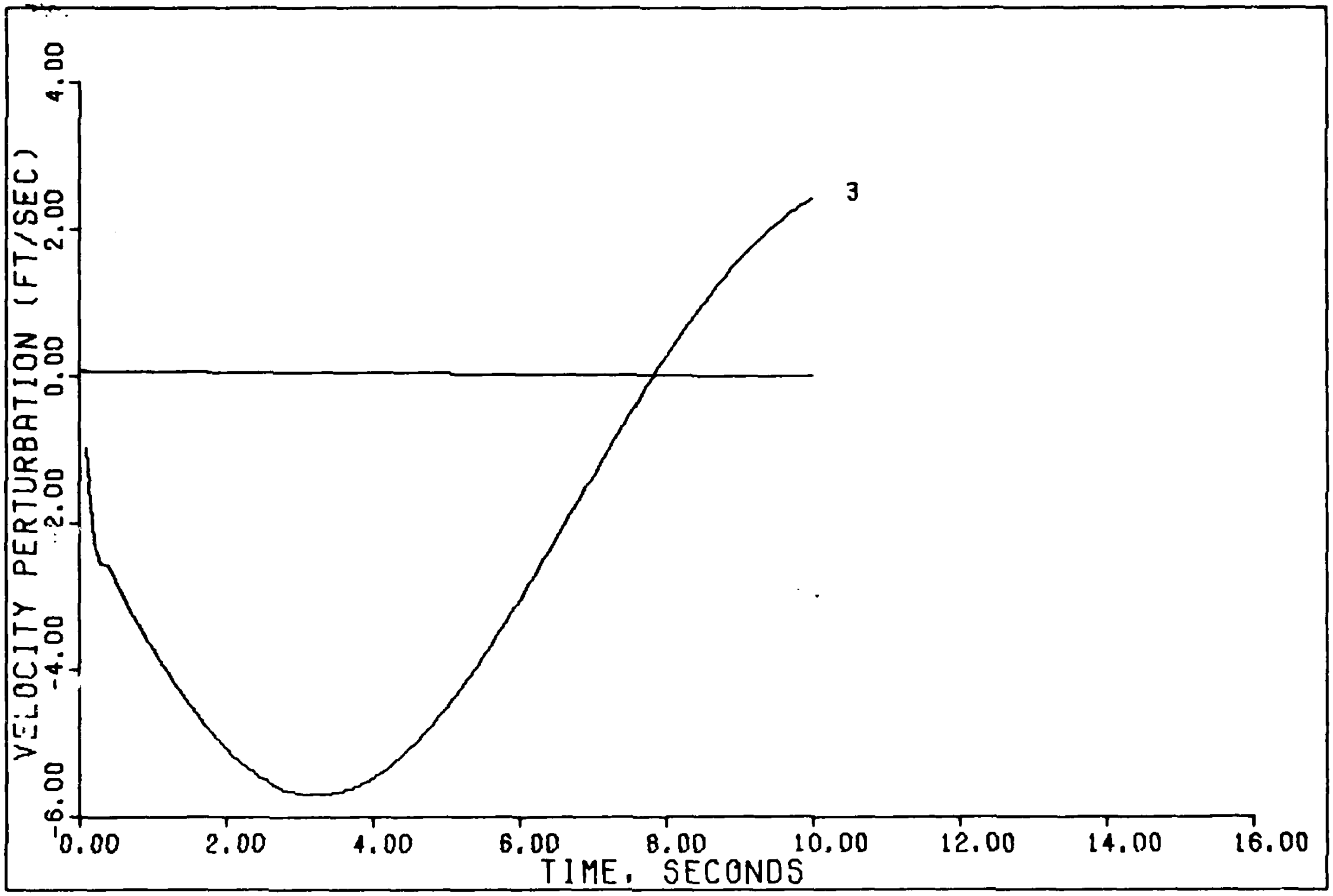


Figure 24. Velocity Perutrbation, Straight Climb Command (σ_3 =0.015)

singular perturbation methods is inherently iterative. The designer needs a grasp of the influence of parameter changes on the design with respect to figures of merit for outputs and limitations on the inputs. Table X provides the designer with this information. Experience has shown that the designer must look at all the required commands to the system, such as pitch pointing, vertical translation, and climb, rather than just optimizing around a single input vector.

TABLE X SUMMARY OF PARAMETER CHANGE INFLUENCE

PARAMETER	CHANGE	OVERSHOOT	SETTLING TIME	TRANSIENTS
SAMPLING TIME	INCREASE	INCREASE (P)	INCREASE (P)	DECREASE (P)
	DECREASE	DECREASE (P)	DECREASE (P)	INCREASE (P)
MEASUREMENT MATRIX	INCREASE ELEMENT	INCREASE (P)	INCREASE (S)	DECREASE (P)
	DECREASE ELEMENT	DECREASE (P)	DECREASE (S)	INCREASE (P)
ALPHA	INCREASE	DECREASE (S)	INCREASE (P)	INCREASE (P)
	DECREASE	INCREASE (S)	DECREASE (P)	DECREASE (P)
EPSILON	INCREASE	DECREASE (P)	DECREASE (P)	INCREASE (P)
	DECREASE	INCREASE (P)	INCREASE (P)	DECREASE (P)
SIGMA MATRIX	INCREASE ELEMENT	DECREASE (P-I)	DECREASE (P-I)	INCREASE (P-I)
	DECREASE ELEMENT	INCREASE (P-I)	INCREASE (P-I)	DECREASE (P-I)

(P)= PRIMARY INFLUENCE
(S)= SECONDARY INFLUENCE
(P-I)= PRIMARY INFLUENCE-INDIVIDUAL OUTPUT

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

An interactive computer program, MULTI, was written and used to develop and test a longitudinal tracker for the FPCC aircraft. Chapter II provides a brief summary of three papers presented by Porter that form the basis for the design methods (Ref. 1 to 3). A flow chart is given at the end of that chapter to assist the reader in choosing the proper design method based upon the constraints imposed by each of the three design methods: known-regular plants, known-irregular plants and unknown plants.

The computer program MULTI, an interactive user-oriented computer aided design tool, was developed to satisfy the requirements listed in Chapter III. The chapter also contains a brief outline of the program's structure and flow. A much more detailed description is found in Appendix B. Appendix C contains a user's manual that assists the MULTI user in applying the design methods. These two appendices are from Capt. Porter's thesis and are included for completeness (Ref. 11). The program developement was a joint effort between Capt. Porter and the author.

A four-state, 3 input and 3 output model for the FPCC aircraft was developed for 3 flight conditions, subsonic (Mach 0.6, 0 ft), transonic (Mach 0.9, 30,000 ft) and supersonic (Mach 2.3, 40,000 ft). Chapter IV describes the model evolution from the original 6 state, 6 input and 6 output model to its current version. Appendix A contains the non-linear equations that are used by the FPCC SIM program to derive the state equations.

Chapter V contains the results from a design based upon the transonic flight-condition. Three command vectors are applied to the system: positive pitch pointing, vertical translation and straight climb. Very good results are obtained. The physical limits on the lift flap are exceeded in some cases, but this can be remedied by applying a ramped-up step input in order to avoid large initial transients. The design is applied to both the subsonic and supersonic models to determine robustness. Very good results are obtained, including the finding that either a 0.1 radian vertical translation or a 0.1 radian positive pitch pointing maneuver is too severe at Mach 2.3 for the engine to sustain zero velocity change.

Chapter VI is essentially a sensitivity study. Each of the design parameters is raised or lowered individually to determine the overall effect on the system. This also has the secondary purpose of validating the original design. A table is presented at the end of the chapter to provide the designer with a quick overview of design parameter influence.

RECOMMENDATIONS

A limitation exists in the method for choosing a measurement matrix for the irregular design. Once a systematic approach is developed, this design method should find universal acceptance.

Any future follow-on effort in designing a flight control system for the FPCC aircraft should include the lateral dynamics as well as the longitudinal. This will contain 4 additional states, 3 inputs and 3 outputs. The feasibility of developing separate longitudinal and lateral control systems can also be studied.

The computer program MULTI provides the designer with a powerful tool. The usefulness may be improved by inclusion of the following points:

(1) The use of random access files;

(2) The ability to identify transmission zeros;

(3) The ability to calculate the closed-loop asymptotic roots;

(4) The ability to determine open-loop stability;

(5) The use of a more rigorous error protection scheme;

(6) The ability to derive figures of merit.

BIBLIOGRAPHY

1. Porter, B. "Design of Set-Point Tracking and Disturbance Rejection Controllers for Unknown Multivariable Plants," IEE Workshop on the Theory and Application of Self-Tuning and Adaptive Control, Oxford England, March 1981. Report No. VSAME/DL/101/81. Salford, England: University of Salford, January 1981.

2. Bradshaw, A. and Porter, B. "Singular Perturbation Methods in the Design of Tracking-Systems Incorporating Fast-Sampling Error-Actuated Controllers," International Journal of Systems Science, Vol. 12. pp. 1181 - 1192, October 1981.

3. Bradshaw, A. and Porter, B. "Singular Perturbation Methods in the Design of Tracking-Systems Incorporating Fast-Sampling Error-Actuated Controllers," International Journal of Systems Science, Vol. 12. pp. 1207 - 1220, October 1981.

4. Hemani, A. Package 200 - Program for Computing the Transient Behavior of Tracking Systems Incorporating Unknown Plants, Actuator and Sensor Dynamics, and Digital Controllers. Salford, England: University of Salford, 1981.

5. Honeywell Inc. Systems and Research Center. Users Manual for the Program FPCCSIM. MR-12369. Minneapolis, Minnesota, July 1975.

6. Houtz, J. E. FPCC Linear Analysis Digital Computer Program. AFFDL/FGL-TM-77-82-FGL. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, OH, November 1977.

7. Porter, B. and D'Azzo, J. J. "Transmission Zeros of Linear Multivariable Continuous-Time Systems," Electronics Letters, 13: pp. 753 - 755, November 1977.

8. Ridgely, D. B. Use of Entire Eigenstructure Assignment With High Gain Error Actuated Flight Control Systems. Master Thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1981.

9. Kennedy, T. A. CESA - An Interactive Computer Program for Complete Eigenstructure Assignment to Aid in Designing State-Space Control Laws for MIMO Systems. Master Thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, March 1979.

10. Porter, B. and Bradshaw, A. "Design Direct Digital Flight-Mode Control Systems for High-Performance Aircraft;" IEEE 1981 National Aerospace and Electronics Conference (NAECON 1981) May 19 - 21, 1981, Dayton, OH, 1292 - 1298.

11. Porter, D. Design and Analysis of a Multivariable, Digital Controller for the A-7D Digital II Aircraft and the Development of an Interactive Computer Design Program. Master Thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1981.

12. Paschall, R. Design of a Multivariable Tracker Control Law for the A-7D Digital II Aircraft, Master Thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1981.

13. Larimer, S. J. TOTAL - An Interactive Computer Aided Design Program for Digital Continuous Control System Analysis and Synthesis. Master Thesis. Air Force Institute of Technology, Wright-Patterson AFB, OH, March 1978.

14. Control Data Corp., Fortran Version 5 Reference Manual (Revision C). Sunnyvale, CA: Control Data Corporation, 1980.

15. Heimbold, R. L., et al. Flight Propulsion Control Coupling (FPCC) and Dynamic Interaction Investigation, Volumes I and II. AFFDL-TR-75-153. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, OH, September 1975.

16. Heimbold, R. L., et al. Flight Propulsion Control Coupling (FPCC) and Dynamic Interaction Investigation, Volume III. AFFDL-TR-77-41. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, OH, June 1977.

17. Control Data Corp., CYBER Loader Version 1 Reference Manual. Sunnyvale, CA: Control Data Corporation, 1979.

18. International Mathematical and Statistical Libraries, Inc. IMSL Library 3, Edition 6, (Fortran) CDC 6000/7000, CYBER 70/170 Series. Texas: International Mathematical and Statistical Libraries, Inc., 1977.

19. ASD Computer Center, CALCOMP Plotter Guide. Revision D. ASD Computer Center, Wright-Patterson AFB, OH, August 1978.

20. Nickolai, P. J. and Clemm, D. S. Solution of Ordinary Differential Equations of the CDC 6600/CYBER 74 Processors. AFFDL-TM-76-FBR. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, OH, January 1977.

21. Stamm, M. Users Guide to Subroutine PLOTIT. Unpublished Users Manual. Air Force Institute of Technology, Wright-Patterson AFB, OH, June 1980.

APPENDIX A: DEVELOPMENT OF EQUATIONS OF MOTION FOR THE FPCC AIRCRAFT

The longitudinal equations of motion for the aircraft in body axes are presented.

TRANSLATIONAL EQUATIONS

$$\dot{u}_I = \frac{F_{x_B}}{m} + rV_I - qw_I - g \sin \theta \qquad \text{(A-1)}$$

$$\dot{w}_I = \frac{F_{z_B}}{m} + pV_I + qu_I + g \cos \theta \cos \phi \qquad \text{(A-2)}$$

ROTATIONAL COMPONENTS

$$\dot{q} = \frac{M_B}{I_{yy}} + \frac{(I_{zz} - I_{xx})}{I_{yy}} rp + \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \qquad \text{(A-3)}$$

EULER RATES

$$\dot{\theta} = q \cos \phi - r \sin \phi \qquad \text{(A-4)}$$

EXTERNAL FORCES IN BODY AXES

$$F_{x_B} = X_s \cos \alpha - Z_s \sin \alpha + X_{eng_1} + X_{eng_2} \qquad \text{(A-5)}$$

$$F_{z_B} = Z_s \cos\alpha + X_s \sin\alpha + Z_{eng_1} + Z_{eng_2} \qquad \text{(A-6)}$$

MOMENTS IN BODY AXES

Δx and Δz represent c.g. offset from m.r.p.

$$M_B = M_s + F_{x_B} \Delta z + F_{z_B} \Delta x + X_{eng_1} z_{eng_1} + X_{eng_2} z_{eng_2}$$

$$- Z_{eng_1} x_{eng_1} - Z_{eng_2} x_{eng_2} \qquad \text{(A-7)}$$

AERODYNAMIC FORCES AND MOMENTS IN STABILITY AXES

$$X_s = -\frac{\rho}{2} V_T^{\,2} \, S \, C_D - \sum_{k=1}^{2} D_{R_K} \qquad \text{(A-8)}$$

$$D_{R_K} = \frac{w_{c_1} * V^T}{g} \qquad \text{(A-9)}$$

$$Z_s = -\frac{\rho}{2} V_T^{\,2} \, S \, C_L \qquad \text{(A-10)}$$

$$M_s = \frac{\rho}{2} V_T^{\,2} \, S \, \bar{c} \, C_m \qquad \text{(A-11)}$$

DIRECT PROPULSION FORCES IN BODY AXES

$$X_{eng_1} = \eta_F \, F_1 \cos\theta_j \cos\Psi_{eng_1} - C_{D_{I_1}} \frac{1}{2} \rho V_T^{\,2} \, S \qquad \text{(A-12)}$$

$$X_{eng_2} = \eta_F \, F_2 \cos\theta_j \cos\Psi_{eng_2} - C_{D_{I_2}} \frac{1}{2} \rho V_T^{\,2} \, S \qquad \text{(A-13)}$$

$$Z_{eng_1} = -\eta_F \, F_1 \sin\theta_j \qquad \text{(A-14)}$$

$$Z_{eng_2} = -\eta_F \, F_2 \sin\theta_j \qquad \text{(A-15)}$$

POWER CIRCULATION EFFECTS ON STABILITY AXES

$M < 1$

$$C_{L_\Gamma} = \frac{1}{2} \sum_{K=1}^{2} (G_K - 1)\, C_{\mu_K} \sin \delta_{jf_K} \qquad \text{(A-16)}$$

$$C_{M_\Gamma} = C_{L_\Gamma} \frac{(X_{MRP} - X_{CP})}{c} \qquad \text{(A-17)}$$

$$C_{D_\Gamma} = \frac{(C_{L_{jo}} - C_{L_\Gamma})^2 - C_{L_{jo}}^{\;2}}{\pi\, A\, R\, e_\Gamma} \qquad \text{(A-18)}$$

$$C_{L_{\delta_{jf_K}}} = \frac{1}{2} (G_K - 1)\, C_{\mu_K} \cos \delta_{jf_K} \qquad \text{(A-19)}$$

$$C_{M_{\delta_{jf_K}}} = C_{L_{\delta_{jf_K}}} \frac{(X_{MRP} - X_{CP})}{\bar{c}} \qquad \text{(A-20)}$$

$$C_{D_{\delta_{jf_K}}} = C_{L_{\delta_{jf_K}}} \frac{(\;2\, C_L\;)}{\pi\, AR\, e_\Gamma} \qquad \text{(A-21)}$$

$$C_{\mu_K} = \frac{4\, F_K}{\rho\, V_T^{\;2}\, S} \qquad \text{(A-22)}$$

$$C_{L_{jo}} = C_L - C_{L_\Gamma} \qquad \text{(A-23)}$$

$$G_K = \frac{1.281}{(C_{\mu_K})^{.392}} \qquad \text{(A-24)}$$

$$e_\Gamma = 0.75 \qquad \text{(A-25)}$$

$M > 1$

$$C_{L_{\delta_{jf_K}}} = C_{M_{\delta_{jf_K}}} = C_{D_{\delta_{jf_K}}} = 0 \qquad \text{(A-26)}$$

AERODYNAMIC FORCES

$$C_L = (C_L)_{\alpha} + C_{L_{\delta_c}} \delta_c + \Delta C_{L_{MF}} + C_{L_{\delta_{jf_1}}} \delta_{jf_1} + C_{L_{\delta_{jf_2}}} \delta_{jf_2} + \frac{\bar{c}}{2V^T} \left(C_{L_q} q + C_{L_{\dot{\alpha}}} \dot{\alpha} \right) \qquad \text{(A-27)}$$

$$C_D = C_{D_{MIN}} + k \left(C_{L_{\alpha}} \alpha - C_{L_k} \right)^2 + C_{D_{\delta_c}} \left| \delta_c \right| + K_{DF} \left(\Delta C_{L_{MF}} \right)^2 + C_{D_{\delta_{jf_1}}} \delta_{jf_1} + C_{D_{\delta_{jf_2}}} \delta_{jf_2} \qquad \text{(A-28)}$$

AERODYNAMIC MOMENTS

$$C_M = C_{M_{\alpha}} \alpha + C_{M_{\delta_{MF}}} \delta_{MF} + C_{M_{\delta_c}} \delta_c + C_{M_{\delta_{jf_1}}} \delta_{jf_1} + C_{M_{\delta_{jf_2}}} \delta_{jf_2} + \frac{\bar{c}}{2 V_T} \left(C_{M_q} q + C_{M_{\dot{\alpha}}} \dot{\alpha} \right) \qquad \text{(A-29)}$$

MASS PROPERTIES

Airplane mass at combat weight

$M = 902$ slugs (A-30)

Moments of Inertia

$I_{xx} = 57{,}500$ slug-ft^2 (A-31)

$I_{yy} = 130{,}000$ slug-ft^2 (A-32)

$I_{zz} = 175{,}000$ slug-ft^2 (A-33)

$I_{xz} = 5{,}400$ slug-ft^2 (A-34)

MOMENT REFERENCE POINT (0.25 MAC)

$$X_{MRP} = 3.5 \text{ ft} \qquad \text{(A-35)}$$

C.G. OFFSET FROM MRP

$$\Delta x = \Delta y = \Delta z = 0 \qquad \text{(A-36)}$$

THRUST GEOMETRY

$$x_{eng_1} = x_{eng_2} = -12.5 \text{ ft} \qquad \text{(A-37)}$$

$$y_{eng_1} = -5.16 \text{ ft} \qquad \text{(A-38)}$$

$$y_{eng_2} = 5.16 \text{ ft} \qquad \text{(A-39)}$$

$$z_{eng_1} = z_{eng_2} = 0 \qquad \text{(A-40)}$$

THRUST LINE TOW-IN ANGLE

$$\psi = 0^{\circ} \qquad \text{(A-41)}$$

THRUST ADJUSTMENT FACTOR

$$\eta_F = 0.98 \qquad \text{(A-42)}$$

APPENDIX B

PROGRAMMER'S MANUAL FOR PROGRAM MULTI

PROGRAMMER'S MANUAL FOR PROGRAM MULTI

1. Introduction

This guide provides the documentation needed for future modifications to MULTI. It is also intended to help a programmer analyze the flow structure of the program so that MULTI can be modified if unexpected errors occur. This manual describes the overlay structure, local and attached subroutines and the names of all elements of the program.

MULTI is written in FORTRAN V and is fully documented by COMMENT statements throughout the program. The program structure is such that programmers familiar with FORTRAN V can understand MULTI's operation. The programmer should have a full, working knowledge of the theory behind the design of discrete-time, error-actuated controllers for linear, multivariable plants as developed by Professor Brian Porter, University of Salford, England (Ref. 1, 2, 3). The programmer should also have used MULTI interactively.

MULTI retains the code from the University of Salford which deals with measurement matrices for the C and G(0) matrices. In order to avoid confusion, these matrices are not utilized in the current design code, and thus throughout MULTI, the following matrix identities exist:

CM (I,J) = C (I,J)

GMO (I,J) = GO (I,J)

The reader should obtain a MULTI source listing before continuing with this guide and have a CDC FORTRAN V REFERENCE MANUAL (Ref. 14) and a CDC LOADER REFERENCE MANUAL (Ref. 17) available. Other programming guides which may be helpful to a potential MULTI programmer include:

IMSL Library Reference Manual (Ref. 18)

ASD Computer Center CALCOMP Plotting Guide (Ref. 19)

Solution of Ordinary Differential Equations of the CDC 6600/CYBER 74 Processors (Ref. 20)

Computer Printout for Subroutine PLOTIT (Ref. 21)

2. Description of Overall Structure

MULTI is written to provide a user with an interactive design tool for the design of control laws needed to attain tracking and disturbance rejection in a multi-variable plant. It is necessary for the program to have the capability to evaluate the input and output responses after the control law is simulated. In addition, it is necessary for an iterative design method that the program retain all parameter values between designs. Finally, the program must be fully interactive.

MULTI must fulfill all of these requirements and be able to operate in the limited $65{,}000_8$ words of memory core that is available on AFIT's INTERCOM. In its original form MULTI required in excess of $110{,}000_8$ words of memory. Thus it was necessary to redesign MULTI using an overlay structure and labeled common block.

2.1 Overlays

Overlays are used to reduce the storage requirements of large programs by dividing the program into modules. All modules are separate programs in their own right and are linked together by the use of a main executive module.

The main executive module and all common variables of the program constantly remain in operational core. The main executive directs the program flow by calling the primary overlays into operational core as they are needed.

The data from common variables is passed between the main and primary overlays by declaring the variable in COMMON statements in the main executive overlay.

The CDC FORTRAN V REFERENCE MANUAL and the CDC LOADER REFERENCE MANUAL for NOS and NOS/BE can provide a programmer with more detailed information on overlays and COMMON statements.

2.2 MULTI's Overlay Structure

MULTI is composed of one main overlay and 13 primary overlays.

The main overlay provides the executive directing function of the program. As an executive, the overlay initializes and stores data in specified common blocks. When an option number is entered by the user, the main overlay checks an IF/ELSEIF structure to attach the primary overlay needed to satisfy the option request. The main overlay also contains subroutines which are used by more than one primary overlay.

A main overlay description is given by Fig. B-1.

MAIN OVERLAY

Overlay #	Program Name	Options Included
(0,0)	MULTI	ALL

Overlay #	Subroutines	Purpose
(0,0)	MATPR	Prints matrices
	QPRINT	Asks if data is to be printed
	ANSWER	Asks if data is correct
	INVERT	Inverts matrices

Fig. B-1 Description of Main Overlay

Initially there were three primary overlays; each was chosen to conform to three developmental aspects of controller design: data input, control law systhesis and simulation. However, to further reduce memory core requirements, the overlays and related subroutines are now organized and defined as shown in Fig. B-2.

PRIMARY OVERLAYS

Overlay #	Program Name	Options Included
(1,0)	OPT0	#0 - #9
(2,0)	OPT10	#10 - #13, #15 - #17, #19
(3,0)	OPT14U	#14 - Unknown Plants
(4,0)	OPT14R	#14 - Regular Plants
(5,0)	OPT14I	#14 - Irregular Plants
(6,0)	OPT18	#18
(7,0)	OPT20	#20 - #29
(10,0)	OPTPLT	#30 - #39
(11,0)	OPT31	#31 - #33
(12,0)	OPT34	#34 - #36
(13,0)	ERROR	ALL
(14,0)	MEMORY	#99
(15,0)	PRINT	#100 - #130

Overlay #	Subroutines	Purpose
(7,0)	CLPASS	Forms different equations
	TOUT	Calculates output values

Fig. B-2 Description of Primary Overlays

2.3 <u>MULTI's Data Elements</u>

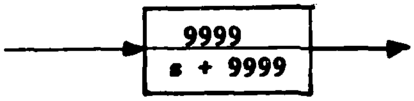
Labeled common blocks are used in MULTI to transfer data values between overlays. These data values are also retained in memory and determine the basic need for memory core space. Each COMMON block is selected so that only the data required by the primary attached to the main overlay is transferred into the primary overlay. In addition, all character variables must be in separate blocks and not in numerical COMMON blocks.

The numerical sequence of the blocks corresponds to the overlay structure of the program. In COMMON block 7, the postscripts "A" and "S" refer to actuator and sensor. Figure B-3 shows which common blocks are used in each overlay.

Arrays and matrix elements which are defined in the labeled common blocks are dimensioned in the same statements rather than with an additional set of DIMENSION statements. Arrays and matrices which are not common to more than one overlay are dimensioned in the individual overlay.

Initializations for the common blocks is accomplished via DATA statements in the executive overlay. All values are set equal to zero except as follows:

Actuators and sensors are set to be the transfer function...



Control limits are set equal to.

Minimum value — -1.0×10^{10}

Maximum value — 1.0×10^{10}

Initial values for all other arrays and matrices are entered in the individual overlays by the use of DO loops or DATA statements.

2.4 Program Labels

The program labels are selected to reflect certain operations in the program flow. Each label quickly identifies the code associated with the label and provides a method for easy editing and error checking. The MULTI program labels are defined as follows:

	OVERLAY													
COMMON BLOCK	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	(6,0)	(7,0)	(10,0)	(11,0)	(12,0)	(13,0)	(14,0)	(15,0)
B 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
B 2	X	X	X	X	X	X	X	X					X	X
B 3	X	X		X									X	X
B 4	X	X		X	X	X	X	X					X	X
B 5	X	X		X				X					X	X
B 6	X	X						X					X	X
B 7A	X	X						X					X	X
B 7S	X	X						X					X	X
B 8	X		X	X	X	X							X	X
B 8A	X		X	X	X	X		X					X	X
B 9	X		X	X	X	X		X					X	X
B10	X		X					X					X	X
B11	X		X			X	X	X					X	X
B12	X							X					X	
B12A	X							X					X	X
B12B	X							X	X				X	X
B13	X							X	X	X	X			X
B13A	X							X	X					X
B13B	X							X						X
B14	X								X	X	X			X
B14A	X								X	X	X			X
B15	X								X	X	X			X
B16	X								X	X	X			X
B17	X								X		X			X

X denotes that common block is used in overlay

Figure B-3 COMMON Block Usage in MULTI's Overlays

LABEL #	PURPOSE
0000 - 1999	Assigned sequentially in the program to label general operations.
2000 - 2999	Assigned to correspond to individual option transfers (i.e. OPTION #1 is labeled 2001).
3000 - 3999	Assigned to error statements in Overlay (13,0).
4000 - 4999	Assigned to FORMAT statements.
8000 - 8999	Assigned to error-free exit from overlays.
9000 - 9999	Assigned to CONTINUE statements for erroneous exit from overlays.

2.5 Overlay Structure

Each of MULTI's overlays are structured in the same basic format. The format has several elements beginning with overlay identification and ending with an END statement. Each overlay, however, may not contain all elements. The format used in MULTI's structure is as follows:

A. Overlay Beginning, including...
- Overlay Identification
- Program Name
- CHARACTER, INTEGER, REAL Declarations
- COMMON Blocks
- DIMENSION Statements
- DATA Statements

B. Option Flag Checking

C. Option Routing

D. Option/Overlay Code
- Option Flag Initializations

E. FORMAT Statements

F. Overlay Ending, including...
- Error Flag Initializations
- END Statement

Comment cards are used throughout the program to help a programmer interpret the overlays. In general, four lines of asterisks box an overlay-heading comment card, two rows of asterisks lead all subroutines and format blocks and a single line of asterisks separate the options in each overlay. Finally, comment cards with dashed lines explain various operations or the next lines of code, or the dashed lines are used to separate sections of the program.

2.6 Overlay Calls

The main overlay is identified with a name and numerical designator:

```
OVERLAY (MULTI,0,0)
```

The primary overlays are identified only by numerical descriptions:

```
OVERLAY (n,0)
```

where "n" is an octal number.

Each overlay is called by designating the main overlay name, MULTI, and the numerical identification of the primary overlay with "n" in decimal. For example, to attach Overlay (11,0) the program call statement is:

```
CALL OVERLAY (MULTI,9,0)
```

where 9 is the decimal equivalent of the octal number 11.

This aspect of overlay usage is critical and is not defined well in the CDC FORTRAN V REFERENCE MANUAL.

2.7 Option Flags and Error Flags

MULTI uses a simple, but effective, method to assure that options are accomplished in the correct order. This is necessary since program flow must begin with data input and proceed to simulation. As each option is accomplished, a corresponding option flag is set before program flow leaves the option.

To check program flow, the operational code beginning each

The lines of code described above are only executed once; this occurs when the program is first begun. After MULTI is opened, the program flow remains in the option request-loop which begins at statement #9000.

3.2 Error Checking

All program-determined errors set an error flag denoting the type of error. When the program variable, IERR, is not equal to zero, Overlay (13,0) is called to warn the user of the error condition.

This error checking is the first operational line of the option request-loop code.

3.3 Option Request-Loop Code

The option request-loop begins with the statement:

```
PRINT '(//A)', 'OPTION, PLEASE #'
```

and is used to obtain the option number request from the user and direct the program flow accordingly. The routing is accomplished by checking a series of IF/ELSEIF statements.

There are two additional checks accomplished when OPTION #14 is selected or if any plotting options are chosen.

If OPTION #14 is requested, the program asks the user to select the type of design that is to be accomplished. The single-character variable, METHOD, is used to further route the program flow to obtain the overlay needed for the computations.

When OPTIONS #30-39 are chosen, Overlay (10,0) is called to form a plotting matrix for terminal plots or CALCOMP plots. Upon completion of OVERLAY (10,0), the program returns to the main executive for more option checks. If a terminal plot is requested, Overlay (11,0) is called to accomplish the terminal plot. If OPTIONS #34-36 are requested to obtain a CALCOMP plot, Overlay (12,0) is brought into operational core to produce the PLOT file.

3.4 Main Executive Overlay Subroutines

There are four subroutines attached to the main overlay. These subroutines are used by more than one overlay and therefore remain in operational core at all times.

SUBROUTINE MATPR (TR,IR,IC) Subroutine MATPR is used for printing all matrices. The subroutine has three parameters as described below:

TR - A real matrix of maximum dimension 10x10.
IR - An integer value denoting the # of rows of the matrix.
IC - An integer value denoting the # of columns of the matrix.

SUBROUTINE QPRINT (CHAR,*) This subroutine is used to ask if data should be printed. There are two parameters used in the subroutine call:

CHAR - Character string with maximum of 30 characters.
*** - Line number denoting where program flow should go if data is not to be printed.**

If the data is to be printed, the code directing the printing should follow immediately after the subroutine call.

SUBROUTINE ANSWER (*,*) ANSWER is used after the program echoes input data to the user and asks if the data is correct. If the answer from the user is affirmative, the program continues with the statements following the subroutine call. If the data is not correct or the option is to be aborted, the subroutine parameters direct the program flow as follows:

1st * - Line number denoting incorrect data directing return to data input point.
2nd * - Line number denoting option abort directing return to end of option.

For the second asterisk, the line number should be selected so as to set IERR equal to zero and so that the option flag is not set.

SUBROUTINE INVERT (A,AINV,N,IA,*) This subroutine is used to invert a matrix. The calling parameters are:

A - A real matrix to be inverted of maximum dimension 10x10.
AINV - The resulting real, inversed matrix.
N - The # of rows in the A matrix.
IA - The maximum row dimension of A matrix as described by the external program.
*** - Line number denoting where program flow should be directed if the A matrix cannot be inverted.**

The line number parameter is selected to route program control to a point where an error flag is set corresponding to a program-directed error statement. This error statement indicates which matrix is not invertable.

Subroutine INVERT accomplishes the matrix inversion by use of the IMSL surbroutine, LINV2F (Ref. 18). The error flag from Subroutine LINV2F indicates if the inversion cannot be obtained. Print statements relating the problem are contained in Subroutine INVERT so that the user does not have to refor to the IMSL directives.

4. Description of MULTI's Primary Overlays

Program MULTI contains 13 primary overlays. The program code is written in FORTAN V code and is easy to understand. Comments are included in some portions of the code to help explain the function of certain sections of the program.

Each overlay follows the structure described in Section 2.5. An additional structural aspect at the beginning of Overlays (1,0), (2,0), (7,0), (10,0) and (15,0) is the routing code to the individual options. Each overlay contains lines similar to the following line taken from Overlay (7,0):

GO TO (2021,2022,2023,2024,2025,2026,2027,2028,2029) NOPT

This line is used to route OPTIONS #20 to #29 to the lines beginning each option. NOPT is an integer variable which is equal to the option number the user requests. Upon entry in to Overlay (7,0), NOPT's value is change to NOPT-20. For example, if NOPT originally was 23, it is changed to equal 3, and the program flow is directed to label 2023 which is the third label listed in the GO TO statement. Option routing in the other overlays is similar. If NOPT's modified value becomes zero, program flow is to the statements directly following the GO TO statement. NOPT must be returned to its original value before leaving an overlay in case an error flag has been set during the overlay's operation.

The reader can review the source listing for a description of each primary overlay. An overview of each of the overlays follows in this section. Section 5. contains a listing of pre-requisites for each option.

4.1 Overlay (1,0) - Options #0 - #9

This overlay is used for entering data values which describe the plants. The codefor the option in this overlay consists mainly of READ and PRINT statements.

OPTION #3 is used to enter the plant A, B, C and D matrices. It is noteworthy that the code is currently written so that if a matrix data entry is made incorrectly, the user must re-enter the entire matrix rather than change the individual element value. For this reason each matrix is echoed back to the user for checking by the calls to Subroutines MATPR and ANSWER.

OPTIONS #4 and #5 include code to reset actuator and sensor values to the transfer function:

$$\longrightarrow \boxed{\dfrac{9999}{s + 9999}} \longrightarrow$$

in order to eliminate their effect in the simulation.

OPTIONS #8 and #9 read plant data from a local file. The data is read by opening a file, LFN, specified by the user. LFN is a variable that can be up to 30 characters in length. The file is first opened on a program-selected unit number, the data is read, and the unit is then closed. It is essential that the unit be closed at the end of the option so that multiple entries to the option can occur. This might be desired if the user wishes to introduce new plant data values for a different flight condition.

4.2 Overlay (2,0) - Options #10 - #19

This overlay contains the code for options #10 through #19, except that the code for OPTIONS #14 and #18 are contained in subsequent overlays. OPTIONS #15 thru #17 are not used.

The code for the options in this overlay is mostly composed of PRINT and READ statements pertaining to design parameters.

OPTION #19 contains the FORTRAN lines to read design parameters from a local file which must be specified by the user. The operation of data file opening and closing is given in the discussion of Overlay (1,0).

4.3 Overlay (3,0) - Option #14 - Unknown Plants

The controller matrices, K0 and K1, for unknown plants are calculated in Overlay (3,0). In addition to the COMMON blocks of the overlay, there are four additional scratch matrices (CM, VV, WW and ZZ) dimensioned and initialized. The matrix values for these matrices are not retained after the overlay is terminated.

The unknown design can be accomplished by use of the plant G(0) matrix or by forming this matrix from the plant's A, B, C and D matrices. In the latter case the calculations are:

Program Calculation	Actual Calculation
$VV=A^{-1}$	$VV=A^{-1}$
ZZ=VV*B	$ZZ=A^{-1}*B$
GO=D - C*ZZ	$G(0)=D-*A^{-1}*B$
WW=D - C*ZZ	$GM(0)=D-C*A^{-1}*B$

At this point either the calculated G(0) matrix or the user-originated matrix is used to create the controller matrices. The calsulations are:

Program Calculation	Actual Calculation
$WW=GO^{-1}$	$WW=G(0)^{-1}$
S=WW*GAMA	
K1=S*EPSLON	$K1=G(0)^{-1}*GAMA*EPSILON$
K0=ALPHA*S*EPSLON	$K0=ALPHA*G(0)^{-1}*GAMA*EPSILON$

The single character variable, METHOD, is set at the end of the overlay such that:

METHOD = 'U'

denoting unknown plant design for normal overlay termination or:

METHOD = 'X'

denoting abnormal overlay termination.

4.4 Overlay (4.0) - Option #14 - Regular Plants

This overlay calculates controller matrices, K0 and K1, for regular plants. There are three scratch matrices (VV, WW and ZZ) which are dimensioned and initialized at the beginning of the overlay. The matrices are destroyed when the overlay is completed.

After pre-requisite checks, the controller matrices are formed by the following calculations:

Program Calculation	Actual Calculation
VV=C*B	VV=C*B
$ZZ=(VV)^{-1}$	$ZZ=(C*B)^{-1}=B^{-1}*C^{-1}$
WW=ZZ*GAMA	$WW=B^{-1}*C^{-1}*GAMA$
K0=WW*ALPHA*EPSLON	$K0=B^{-1}*C^{-1}GAMA*ALPHA*EPSILON$
K1=WW*EPSLON	$K1=B^{-1}*C^{-1}*GAMA*EPSILON$

For a normal termination to the overlay, the single character variable, METHOD, is set equal to "R" denoting a regular plant design. If abnormal termination occurs, METHOD is set equal to "X".

4.5 Overlay (5,0) - Option #14 - Irregular Plants

An irregular plant design of the controller matrices, K0 and K1, is accomplished in Overlay (5,0). Thre are four temporary matrices (VV, WW, ZZ and VWZ) which are dimensioned and initialized for use only while this overlay is attached to operational core.

For this type of design the program requires a measurement matrix, MM, to augment the rank deficiencies of the plant matrices. After pre-requisite checks, the next block of program code determines if the measurement matrix already exists, if the user must initilize MM, or if the user desires to re-initialize the values of the matrix. The measurement matrix values are originally all set equal to zero in the main overlay. If the user has not initialized the values by selecting

OPTIONS #9 or #18, the matrix values must be entered throught this overlay.

The following equations then occur:

<u>Program Calculation</u>	<u>Actual Calculation</u>
$WW=C + MM*A$	$F=C + M*A$
$VV=WW_2$	
$ZZ=WW_2$	$F_2=C_2 + M*A_{12}$
$WW=ZZ^{-1}$	$WW=F_2^{-1}$
$ZZ=WW*GAMA$	$ZZ=F_2^{-1}*GAMA$
$WW=B_2$	$WW=B_2$
$VWZ=WW^{-1}$	$VWZ=B_2^{-1}$
$WW=VWZ*ZZ$	$WW=B_2^{-1}*F_2^{-1}*GAMA$

The controller matrices are then calculated as follows:

<u>Program Calculation</u>	<u>Actual Calculation</u>
$K0=WW*ALPHA*EPSLON$	$K0=B_2^{-1}*F_2^{-1}*GAMA*ALPHA*EPSILON$
$K1=WW*EPSLON$	$K1=B_2^{-1}*F_2^{-1}*GAMA*EPSILON$

The overlay terminates by setting by setting the single character variable, METHOD, equal to "I", denoting an irregular plant design, if the overlay terminates normally. It is set equai to "X" if the overlay terminates abnormally.

VMULFF, a high precision, IMSL Library subroutine, is used in this overlay to multiply two matrices together. When using this subroutine, the second matrix which is entered into the subroutine is overwritten during the matrix multiplication process. Thus, the contents of this matrix are destroyed.

4.6 Overlay (6,0) - Option #18

Overlay (6,0) is provided for two purposes. One purpose is to create a measurement matrix for use in an irregular plant controller design, and the other purpose is to provide a tool for row and column operations on any matrix.

This overlay uses four scratch matrices (VV, WW, ZZ and VWZ) during its computations. These matrices are discarded when the option ends.

The integer variable, ISKIP, is set equal to 1, 2 or 3 depending on the user's choice of measurement matrix creation, row and column operations on [C * B], or row and column operations on any matrix of interest. The routing for the different choices is determined by an IF/ELSEIF block of code.

Before creating a measurement matrix, the user must obtain the feedback gain matrix, K1, from CESA OPTION #38 (Ref. 9). After this matrix is read into the program as matrix, VWZ, and is verified by the user, the measurement matrix is calculated.

Program Calculations	Actual Calculations
$MM=C_1$	$MM=C_1$
$WW=C_2$	$WW=C_2$
$ZZ=MM - WW*VWZ$	$ZZ=C_1 - C_1*K1$
$MM=A_{12}$	$MM=A_{11}$
$WW=A_{12}$	$WW=A_{12}$
$VV=WW*VWZ - MM$	$VV=A_{12}K1 - A_{11}$

$$WW=VV^{-1} \qquad WW=(A_{12}*K1 - A_{11})^{-1}$$

$$MM=ZZ*WW \qquad MM=(C_1 - C_2*K1)*(A_{12}*K1 - A_{11})^{-1}$$

The program code for performing row and column operations on [C * B] and performing row and column operations on any other matrix is identical. The only difference between the two choices is the matrix that is used in the operations. In the first case, the [C * B] matrix is automatically formed by the program without any input from the user. In the second case, the user must supply a matrix of maximum dimension 10x10.

The program provides for four choices of operations on a matrix. The user's choice of operations, which is identified by the integer variable, ISKIP, directs program flow through an IF/ELSEIF block. When a new matrix is formed, it is echoed back to the user. The program then returns to the operation-selection point.

4.7 <u>Overlay (7,0) - Options #20 - #29</u>

All aspects of the system simulation are contained in this overlay. The overlay also contains two subroutines, CLPASS and YOUT, which are used in OPTION #26. Since subroutine CLPASS is used as a parameter in a CALL statement, it must be declared as an EXTERNAL in the beginning of the overlay. There are two scratch matrices (CM and F) and eight scratch arrays (IWORK, WORK, X, Z, Y, E, AX and MMAX) that are dimensioned, initialized, used and discarded as the overlay operates. The F matrix of this option should not be confused with the F matrix generated by OPTION #14 for irregular plants. There is also a character variable called STRING hich is declared at the beginning of the overlay.

Before the variable NOPT routes the program flow to the different options of the overlay, the CM and F matrix values must be set. Although these matrices are only superficial, it is necessary that they be equal to these values:

CM (I,J) = C (I,J)

F (I,J) = Identity Matrix

These two matrices are found in the original program code obtained from the University of Salford and are used in OPTION #26.

OPTION #20 to #25 and OPTION #27 are mainly FORTRAN READ and PRINT statements which are used in obtaining state and integrator initial values, the output command vector, simulation time parameters and the control input limitations. OPITON #27 also contains the code to remove control limits by setting the control input minimum value to -1.0×10^{10} and the control input maximum to 1.0×10^{10}.

OPTION #29 opens unit number 40 to read the design parameters from a user-specified local file. The operation of this option is the same as the operation of OPTION #9 which is discussed in Section 4.1.

The heart of the simulation is contained in the OPTION #26 code. As previously stated, this code is nearly identical to the code received from the University of Salford. The option begins by checking for option flags which may not be set and by checking for the presence of actuators, sensors and contol limits. Next, the total number of states is determined by adding the number of states from the sensors, actuators and plant matrices.

There are three major loops in this option. The outer loop for each sampling time is blocked by label #1285; the middle loop based on sampling period is blocked by label #1250; the inner loop based on step time is blocked by label #1240.

OUTER LOOP The outer loop occurs since the program allows the simulation to be run with more than one sampling time. Each time a diferent sampling time is run, the program must initialize several variable and array values. A calculation of the number of incremental points is also made based upon the total simulation time, TT, the sampling time, SAMT, and the step time, ST. The following formulas determine the number of total time increments, NT:

$$\begin{array}{lll} \mathrm{NIN} = \text{Integer Value of } (\mathrm{TT} / \mathrm{SAMT} + 0.5) & & \text{(B-1)} \\ \mathrm{NTT} = \text{Integer Value of } (\mathrm{SAMT} / \mathrm{ST} = 0.5) & & \text{(B-2)} \\ \mathrm{NT} = \mathrm{NIN} * \mathrm{NTT} & & \text{(B-3)} \end{array}$$

If the step time is chosen to be greater than the sampling time, NTT is set to a value of one prior to the calculation of total number ot time increments in Eq. (B-3). The diagram in Fig. B-4 shows the relationship between TT, SAMP and ST.

If the total number of time increments is greater than 100, the simulation exceeds the dimension of the matrices which hold the input and output values. Thus, an integer variable, IPACK, is calculated to determine the rate at which data is packed into a matrix with 100 rows.

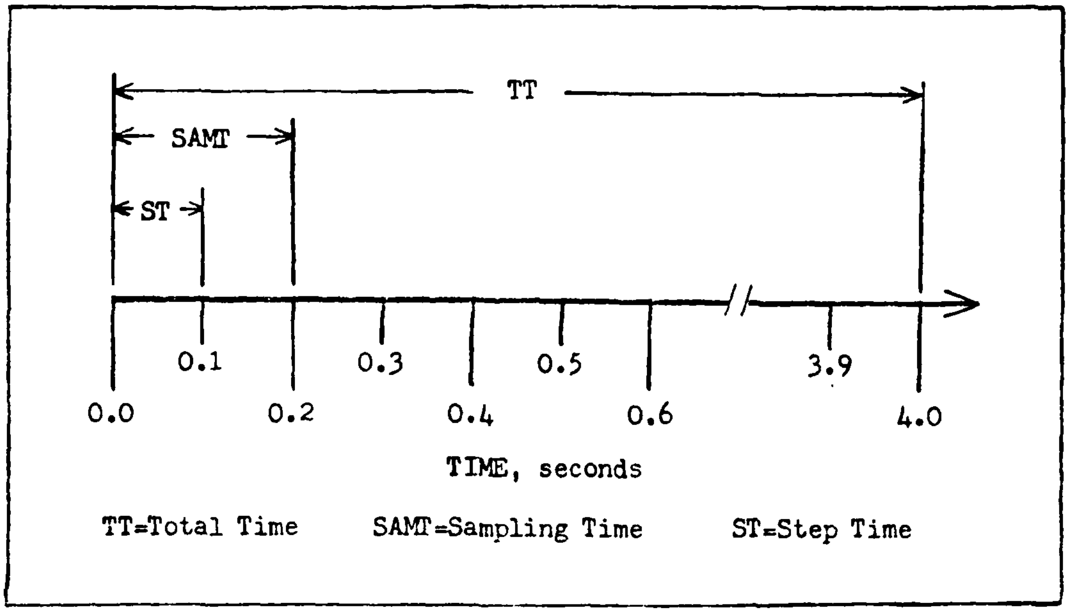


Fig. B-4 Relationship Between Total Time, Sampling Time and Step Time

If there are 125, or less, time increments, the data from the first 100 increments is retained. If there are more than 125 increments, IPACK is calculated as:

$$\text{IPACK} = [\text{ Integer Value of } (NT / 100)] + 1 \qquad \text{(B-4)}$$

As an example of the packing process, if there are 300 time increments, IPACK is calculated as three, and every third input and output value is retained.

The last part of the code for the outer loop sets the integer variable, ISKIP, which is used to suppress the time-sequential printout of input and output values as the solutions to the state differential equations are found.

MIDDLE LOOP The middle loop is passed through NIN times as calculated in Eq. (B-1). First, the ouput measurement vector, YM, is formed. In MULTI this vector is equivalent to the output vector, Y, and is calculated from:

$$\underline{Y} = \underline{CM} * \underline{X}$$

The error vectors are then formed as follows:

For Irregular Plants...

Program Calculation	Actual Calculation
$AX=A_1*X$	$AX=A_1*X$
$MMAX=MM_1*AX$	$MMAX=MM_1*A_1*X$
$S=F*Q$	$S=Q$
$E=V - S - MMAX$	$E=V - Q - MM_1*A_1*X$

For Unknown Plants and Regular Plants...

Program Calculations	Actual Calculations
$S=F*Q$	$S=Q$
$E=V - S$	$E=V - Q$

For unknown plants the controller is then formed from:

$$\underline{U} = (\underline{KO} * \underline{E} + \underline{K1} * \underline{Z}) / SAMT \qquad (B\text{-}6)$$

and for all othe plants the controller is formed from:

$$\underline{U} = (\underline{KO} * \underline{E} + \underline{K1} * \underline{Z}) * SMAT \qquad (B\text{-}7)$$

However, before these inputs are used in the simulation they are compared to the current input control limitations and modified if necessary.

A call to Subroutine YOUT is the final line of code in the middle loop. This subroutine call determines the controlled output vector from the states of the system at any sspecified time and is described at the end of this section.

INNER LOOP The inner loop is passed through NTT times as calculated by Eq. (B-2). The first section of the loop is concerned with the data packing procedure described earlier and sequentially sets the row indices for the input and output matrices, UP and YP. These two matrices are

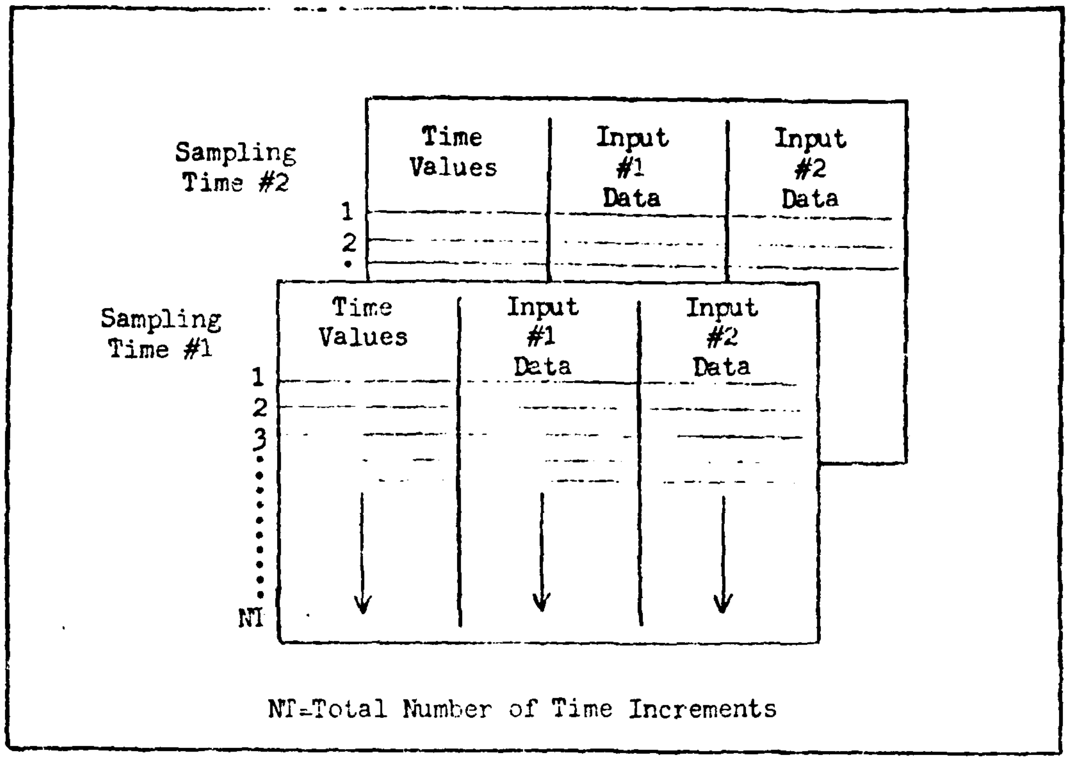


Fig. B-5 Representation of the 3-D Input Matrix, UP

retained throughout MULTI and contain the input and output data as calculated by OPTION #26. The matrices are 3-dimensional where the row dimension is equivalent to an input or output vector at each time increment The first column holds the time increment values, T, while the other columns hold the sequential data for each input or output. The third dimension is equivalent to the sampling time. See the diagram presented in Fig. B-5.

MULTI uses Subroutine ODE (Ref. 20) from the ASD CC6600 Library to solve the set of differential equations formed by Subroutine CLPASS. The CLPASS subroutine is discussed at the end of this section. Currently the precision limits for ODE's computations are set at 1.0 x 10^{-4}. If an error occurs during the solution process an error message is printed to alert the user of the condition. The program provides the error flag number to the user and states that the ODE Manual is to be referenced.

The inner loop is ended by a call to Subroutine YOUT to form a new output vector, Y, from the values returned from the ODE subroutine. When the inner loop is complete, a new integrator vector is calculated and program flow returns to the middle loop.

When all time increments for the total simulation time have been processed through the inner and middle loops, the program code for printing control limit information is entered. A "simulation complete" message is then provided and the outer loop is run again until each sampling time is simulated.

Before the option is completed, the integer variable NT is set equal to the actual number of data points in the input and output matrices of the last simulation. This is necessary so that the plotting routines receive the correct number of data values in the matrices. However, since NT does not have a value for each sampling time and an input/output matrix can be formed where the first sampling time dimension has less data values than the second sampling time dimension, a termination error can result unless MULTI operates under one of the following procedures:

(a) MULTI should be run with all sampling times entered in order of increasing value, and step time should be set so as to be equal, or less than, the smallest sampling time.

or **(b) MULTI should be run using only one sampling time.**

<u>SUBROUTINE CLPASS (T,X,XDOT)</u>

This subroutine is nearly identical to the CLPASS subroutine attached to the PAK200 program (Ref. 4). Minor modifications are made in some parts of the code so that it may be used as an external program in Subroutine ODE.

This subroutine forms the differential equations from the plant, actuator, sensor and error integral matrices. The parameters are:

T - Time
X - State Vector
XDOT - Derivative of State Vector, X

CLPASS defines the values of the array, XDOT, at time T. Array X includes all the states of the composite system. This array contains the plant states first, followed by the actuator states and then by the sensor states.

The system input, U, is applied to the actuators. The output of the actuators, W, provides the input to the plant. Finally, the plant output, YM, is the input to the sensors, and the sensor output, Q, is the actual system output. Fig. B-6 helps the reader visualize this arrangement.

SUBROUTINE YOUT (X,Y,W,C,D)

Subroutine YOUT is a modified version of the YOUT subroutine provided with PAK200. MULTI's YOUT subroutine uses a COMMON block to pass data into the subroutine rather than passing the data through the calling parameters.

This subroutine determines the controller output vector from the states of the system at any time. The parameters in the subroutine call are defined as:

X - System State Vector
Y - Controlled Output Vector
W - Actuator Output Vector

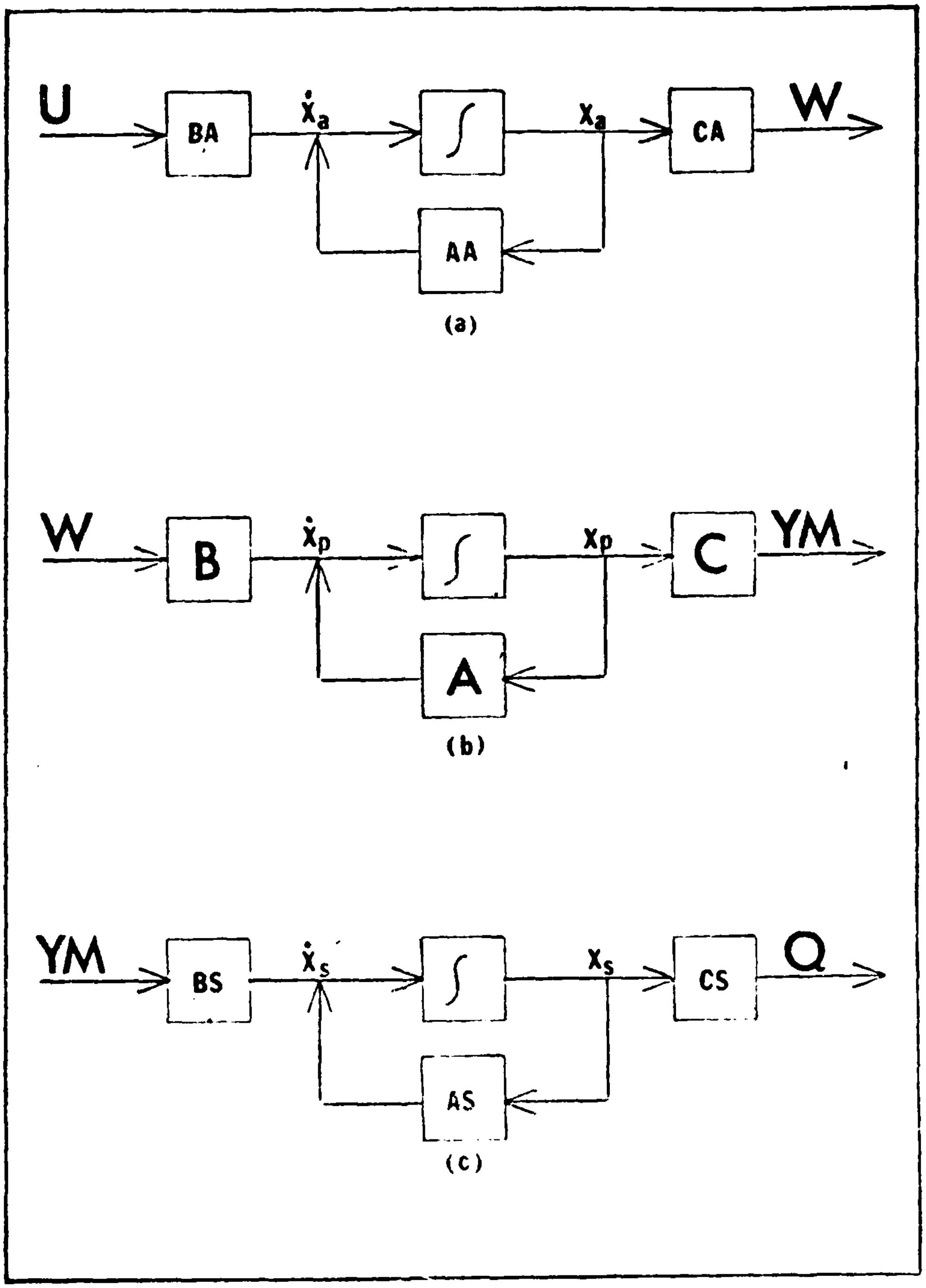


Figure B-6 Block Diagrams for the (a) actuators, (b) plant and (c) sensors

C - System Output Matrix
D - System Direct Output Matrix

4.8 Overlay (10,0) - Options #30 - #39

When any type of plot is to be generated, MULTI uses Overlay (10,0) to generate a plotting matrix, PLMAT, which is filled with data values from the input and/or output value matrices, UP and YP. After the PLMAT is formed the actual terminal plot is generated by Overlay (12,0). These two overlays are discussed in the next two sections.

It is pointed out to the reader that COMMON blocks B13 and B13A change the character representations of the integer variable, NT, and the matrices, YP and UP. In this overlay and Overlays (11,0) and (12,0), these variable names are referred to as N, Y and P, respectively.

The FORTRAN coding in this overlay is very condensed and the program flow is very complex. The coding is condensed so that the overlay can operate successfully while using minimum core space. The core space requirement is critical since this overlay uses the three largest matrices of the program, YP, UP and PLMAT. The program flow is complex since MULTI provides (1) terminal plots and CALCOMP plots, (2) four types of each kind of plot, (3) a long version and two shortened versions of the plot request. All these provisions are completed in this single overlay. In addition, if PLMAT for different flight conditions is generated, the program flow loops between this overlay and the other options of the program until PLMAT is completed. In this case, the overlay is required to keep a count of the number of times that the plot sequence has been entered.

To correctly complete a plotting matrix, PLMAT, this overlay extracts columns of data from the U and Y matrices. The columns are transferred to PLMAT in a pre-designated order after the user enters the values for the variables ICODE, IFLTCN, LINES, NDEPTH, IDEPTH, CHOICE, NUM, ICOLMN and ICLM. The values for these variables may be set by the program, rather than by the user, depending on the type of plot requested. The definitions for these variables are found in the last section of this guide.

After the PLMAT matrix is complete, the integer variables, NUM and ICLM, are reset to the values required for correct operation in Overlays (11,0) and (12,0). This ends the overlay.

4.9 Overlay (11,0) - Terminal Plot

MULTI has the capability of producing terminal plots by calling upon Subroutine PLOTIT. PLOTIT is a subroutine designed by Major Michael R. Stamm of the Department of Physics at the Air Force Institute of Technology (Ref. 21).

This overlay begins with the addition of a new COMMON block called SCALIT which is common only to the PLOTIT subroutine. This COMMON block provides for the initialization of the terminal plot's calling parameters. As discussed in Section 4.8, this overlay also renames the variables, NT, UP and YP as N, U and Y, respectively.

Overlay (11,0) must also provide the PLOTIT subroutine with the independant and dependant data arrays, XAXIS and YAXIS, plus the minimum and maximum values of these arrays. The XAXIS array is always filled with values from the first column of the output matrix, Y. The YAXIS is sequentially set equivalent to the columns of the PLMAT matrix as the PLOTIT subroutine is called. PLOTIT is called until all the data in the PLMAT matrix is scanned and entered into the terminal plot routine. The terminal plot is then generated.

The overlay is ended with a series of PRINT statements which correspond to symbols on the terminal plot curves to the input/output numbers of the plant states.

4.10 Overlay (12,0) - CALCOMP Plot

MULTI has the capability of producing CALCOMP plots that can be routed to the CALCOMP plotter upon program termination. The various CALCOMP subroutinees used are well documented in the CALCOMP user guide (Ref. 19). Only five plots may be put on a plot file at one time. Once this maximum is reached, the user must exit MULTI and route the plot file to the plotter. MULTI keeps count on the number of plots on file and prompts the user when the maximum of five is reached.

The overlay begins by incrementing the plot number counter. This is followed by setting the XAXIS equal to the first column in the YP matrix which contains the time increment listing. The PLMAT matrix is then scanned for its minimum and maximum values in order to accurately set the YAXIS range. A title with a 20 character maximum is then entered.

The CALCOMP subroutines are then begun. The XAXIS and YAXIS arrays are scaled for their minimum and maximum values. The proper columns from the PLMAT array are chosen and plotted. Finally, a box is drawn around the entire plot.

The last section of Overlay (12,0) is the warning message concerning the current number of plots on file.

4.11 Overlay (13,0) - Error Statements

This overlay is used to print error messages for all operational failures detected by the program. The operational code of the overlay consists entirely of PRINT statements.

The first 30 error statements refor to option errors. The number of the error statement relates directly to the number of an error flag which, in turn, corresponds directly to an option number. Error statements #31 thru #39 provide information about matrix errors. Error statements #40 thru #44 give miscellaneous error information. When an option does not exist, a PRINT statement is still included to provide an operational check of MULTI. These error statements should never be printed.

As a working example, consider a prerequisite check at the beginning of an option. OPTION #3 requires the option flag from OPTION #2 to be set before the plant matrices can be entered. If IFLAG(2) is not set to one prior to selecting OPTION #3, IERR is set equal to two and Overlay (13,0) is called. A value of two for IERR routes the program flow to Statement #3002 which prints:

```
# OF STATUS, INPUTS & OUTPUTS MISSING...SEE OPTION #2
```

4.12 <u>Overlay (14,0) - Option #99 - Memory Files</u>

When the user selects OPTION #99 to end program MULTI, Overlay (14,0) is called to create the data memory files. There are three memory files created.

MEMO is created to hold plant data as directed by the integer variable, IPLANT. If IPLANT equals "1", all state, sensor and actuator matrices are saved; if IPLANT equals "2", the plant G(0) matrix is copied to the file. If IPLANT equals "0", plant data from OPTIONS #2 to #5 are still transferred to the memory file, and, when the file is read back into MULTI, the IPLANT value alerts the user that the file's data may be incomplete.

MEM10 is used to save all design data, and MEM20 is opened to contain the simulation data.

The overlay ends with several PRINT statements relating the file names that have been created and the data that each contains.

4.13 Overlay (15.0) - Options #100 - #139

MULTI's last overlay provides data value printing for most of the options. If an option is not concerned with data input or data creation, the program uses error flag #43 to print:

THERE ARE NO VALUES IN MEMORY CORE FOR OPTION #

If data is available, the values are printed by routing the program flow to the proper section of the overlay. All printing options are numbered by adding a value of 100 to the related option number of the main program. For example, data values entered or created by OPTION #12 are printed by OPTION #112.

The data values from some options are combined to form a single block of data information. The data from OPTIONS #11 and #13 is generated when either OPTION #111 or #113 is selected. The same occurs with data from OPTIONS #21 and #22; the data is printed by using either OPTION #121 or #122. And finally, all data values from OPTIONS #23 to #25 are provided by selecting any option number from 123 to 125.

The code for the overlay is simple and self-explanatory. The reader is directed to the MULTI source listing for further detail on this overlay's operation.

5. **Option Pre-requisites**

The following list provides the pre-requisite requirementε for MULTI's options.

OPTION #	PRE-REQUISITE OPTION #
3	2
4	2
5	2
12	2
14	2, 3, 11 TO 13
18 (entries 1 & 2)	5
21	2
22	2
26	14, 21 to 25
29	2
31	26
32	26
33	31 or 32
34	26
35	26
36	34 or 25

APPENDIX C

USER'S MANUAL FOR PROGRAM MULTI

USER'S MANUAL FOR PROGRAM MULTI

Overview of MULTI

MULTI is an interactive program which enables a computer user to design and simulate digital, multivariable control laws for discrete, linear, multiple-output systems. The three design methods incorporated in this program were developed by Professor Brian Porter of the University of Salford, England. To help the user become quickly familiar with the program, the following overview is provided.

MULTI is able to design and simulate control laws for three types of plants:

(1) Regular - The linear, multivariable plant dynamics are described by the usual state and output equations, and the first Markov parameter, CB, has full rank (Ref. 2).

(2) Irregular - The plant dynamics are described by state and output equations, and the first Markov parameter does not have full rank. Thus, the system must be augmented by a measurement matrix so that a control law may be developed (Ref. 3).

(3) Unknown - The plant, state and output equations and unknown, but the steady state transfer function matrix, G(0), is obtainable from off-line tests if the open-loop plant is stable. This method is also applicable to known plants (Ref. 1).

All control law designs may be evaluated by a discrete-time simulation of the system, actuator, and sensor equations. The user may elect to obtain a terminal plot or a Calcomp plot of the system input and/or output responses.

MULTI focuses on Porter's digital design as opposed to a continuous design method. At various points during the design and simulation process, the program can provide intermediate information about system matrices or input/output values.

USER'S MANUAL FOR PROGRAM MULTI

Overview of MULTI

MULTI is an interactive program which enables a computer user to design and simulate digital, multivariable control laws for discrete, linear, multiple-output systems. The three design methods incorporated in this program were developed by Professor Brian Porter of the University of Salford, England. To help the user become quickly familiar with the program, the following overview is provided.

MULTI is able to design and simulate control laws for three types of plants:

(1) Regular - The linear, multivariable plant dynamics are described by the usual state and output equations, and the first Markov parameter, CB, has full rank (Ref. 2).

(2) Irregular - The plant dynamics are described by state and output equations, and the first Markov parameter does not have full rank. Thus, the system must be augmented by a measurement matrix so that a control law may be developed (Ref. 3).

(3) Unknown - The plant, state and output equations and unknown, but the steady state transfer function matrix, G(0), is obtainable from off-line tests if the open-loop plant is stable. This method is also applicable to known plants (Ref. 1).

All control law designs may be evaluated by a discrete-time simulation of the system, actuator, and sensor equations. The user may elect to obtain a terminal plot or a Calcomp plot of the system input and/or output responses.

MULTI focuses on Porter's digital design as opposed to a continuous design method. At various points during the design and simulation process, the program can provide intermediate information about system matrices or input/output values.

Input, design and simulation data is automatically stored on separate memory files for future use. This prevents the time consuming task of re-entering data when the same system is under study.

The program performs full error detection and diagnostics when there are deficiencies in input data, design data, plotting data, or if the plant equations are not sufficient for the type of design being tried. Input error detection and recovery is limited, relying mainly on the automatic CYBER error detection capability. The user should be wary of entering values inconsistant with the limitations described in this guide, since exceeding these limits may cause unexpected termination of the program. If this occurs the memory files are not generated and all data is lost.

It must be stated that MULTI is not as dynamic in its ability to receive input data as other programs with which the user may be familiar (i.e. TOTAL [Ref. 13], CESA [Ref. 9]). At present, character inputs to display current matrix values or system design values are not available except by requesting the proper option number. Also at present, entering a "$" symbol, rather than actual data, to abort an option, is not recognized unless specifically noted in the program instructions below.

1. Introduction to MULTI

The MULTI computer package contains approximately 40 ordered options which give the user an interactive, iterative approach in the design and simulation of control laws for linear, multivariable plants. The control law assures that for constant commands the output tracks the input and that disturbance rejection is accomplished. The method can be used only if (1) the introduction of integral action preserves stabilizability, (2) the number of outputs is equal to or less that the number of inputs, and (3) the forward transfer function has no zero-valued transmission zeros.

Once the model is entered and the parameters for the control law are selected, a simulation can be run which provides a full time-sequential listing of input/output values and a tabular listing of the input/output values, if the user desires. The following discussion should give the user all the information for optimal use of the computer program package called MULTI.

To attach and run MULTI, after LOGIN, the user enters:

```
COMMAND - CONNECT,INPUT,OUTPUT
COMMAND - ATTACH,MULTI,ID=AFIT,SN=AFIT
COMMAND - SCREEN,FULL
COMMAND - (ATTACH DATA FILES)
COMMAND - MULTI
```

1.1 MULTI's Input Modes

MULTI has three input modes in which it requests input information from the user. These modes are called the OPTION Mode, the DATA Mode and the QUERY Mode. Each mode has its own restrictions on allowable input and its own method of requesting information.

OPTION MODE The OPTION Mode is the executive command mode for MULTI. It has the following prompt message:

```
OPTION, PLEASE>
```

Once the program is in this mode, the user is allowed to select any option referring to input, design, simulation, plotting, or program termination. This mode is the main input mode of MULTI since it allows the user complete access to the program and data files.

DATA Mode MULTI enters the data mode when information is needed to perform an option. Each data input request has a specific statement associated with it, and each request is terminated by the symbol, ">". Normally, data input must be numerical and separated by either a comma, a space, or the return key of the terminal. There are other miscellaneous data mode options which require character entries. All replies can be accomplished by a single character except when the program asks the user to enter the choice of "INPUT" of "OUTPUT". In this case the program requires the entire word, spelled correctly, to continue properly.

QUERY Mode MULTI uses the QUERY Mode an an input checking tool and to suppress date printing. The name QUERY Mode suggests that the program is asking the user if the data is correct or is to be printed.

The messages is this mode are either of the form:

```
ENTER "0" TO SKIP DATA PRINTOUT
ENTER "1" TO OBTAIN DATA PRINTOUT...
```

or

```
IS THIS CORRECT...YES,NO,$...
```

In the latter case, a "NO" reply will return the user to the data input point while a "$" reply will terminate the option. Although the option is terminated, the values just entered are placed in memory. The user can also enter a simple "Y" of "N".

1.2 MULTI's Options

MULTI contains 40 main options that allow the user versatility in the input, design, simulation and analysis of control laws for unknown, regular or irregular plants. The options are grouped together into four major option blocks as follows:

BLOCK 1	**OPTIONS # 0 thru # 9**	**Plant Input Options**
BLOCK 2	**OPTIONS #10 thru #19**	**Design Parameter Options**
BLOCK 3	**OPTIONS #20 thru #29**	**Simulation Options**
BLOCK 4	**OPTIONS #30 thru #39**	**Plotting Options**

Option BLOCK 5 is composed of printing commands which print the data entered into the program from the related main option. That is, to print the data which was entered into the program by using Option #5, the user selects Option #105.

Although the selectiion of a sampling time is actually a design parameter, BLOCK 2 is used only to form the control law matrices which are independant of sampling time. Thus, the sampling time selection is accomplished in the simulation option block where the actual control is used.

2. Complete Description of MULTI's Options

In order to utilize MULTI's assets fully, it is necessary to have a complete understanding of the 40 main options available in MULTI. This manual is intended to provide all of the information needed to accomplish this requirement. Each of the following options described may be selected by simply typing in the option number while the program is in the option mode.

2.1 BLOCK 1 - Plant Input Options

Since a control law design cannot be started without the knowledge of the plant model or steady state transfer function, it is necesary to enter some representation of the system for which the control law is to be designed. Thus BLOCK 1 is an integral part of the MULTI program. Plant data may be entered via G(0) matrix (OPTION #1) or by entering the individual matrices of the state-space representation of the plant (OPTIONS #2 and #3).

Although the steady state transfer function, G(0), can be used to obtain control law matrices, it does not provide sufficient information to run a simulation and obtain plots of the responses by using option BLOCKS 3 and 4.

The state and output equations of the plant follow the form:

$$\underline{x}\,(kT + T) = A\,\underline{x}\,(kT) + B\,\underline{u}\,(kT) \qquad \text{(C-1)}$$

$$\underline{y}\,(kT) = C\,\underline{x}\,(kT) + D\,\underline{u}\,(kT) \qquad \text{(C-2)}$$

where

A = sampled-data plant matrix

B = sampled-data control input matrix

C = system output matrix

D = feed-forward matrix

Note that there is not provision for *a disturbance* matrix in Eqn. C-1.

The plant input option block also allows the user to input actuator and sensor state equations into the program. These equations are of the same format as shown for the system state equations, however there is no feed-forward matrix, D. If the actuator and sensor values are not entered by the user, they are set equal to single-order servos with 1 X 10^{-4} time constants.

The program accepts a maximum of ten states, ten inputs and ten outputs. The program only allows for 2nd-order actuators and sensors, thus they may have to be approximated by a 1st-order or 2nd-order equivalent servo.

OPTION #0 - List Options 0 thru 9

This option lists the plant input options from 0 to 9.

OPTION #1 - Enter G(0) Matrix

This option enters the steady state transfer function matrix, G(0). The user is asked to supply the number of inputs, M, and the number of outputs, P, thus setting the dimension of the matrix. The data is then entered by row following each prompt message.

OPTION #2 - Enter Number of States, Inputs, and Outputs

This option asks the user to enter the number of states, N, the number of inputs, M, and the number of outputs, P. These values must be entered sequentially as N, M, and P.

OPTION #3 - Enter A, B, C & D Plant Matrices

This option enters the plant A, B, C and D matrices. Each input has the same format and the user enters the data by rows after the prompt. After each matrix is entered, its values are automatically echoed back to the user for checking. After the A, B and C matrices are entered, the program asks if there is a D matrix. If the reply is negative, the option is terminated, and the program sets the D matrix values to zero.

OPTIONS #4 and #5 - Enter actuator/Sensor State Equation Matrix Data

These options enter or eliminate the actuator and sensor state equation A, B, and C matrices. When these options are selected the following prompt will appear:

ENTER "0" TO ELIMINATE ACTUATORS/SENSORS

ENTER "1" TO SET ACTUATOR/SENSOR VALUES...

If "1" is entered, the user is first asked to enter the number of states of each actuator/sensor. The input sequence must correspond to the sequence of the actuators/sensors. The user then inputs the A matrix values by rows, the B column values and the C row values after each prompt.

OPTIONS #6 and #7 - Reserved Options

OPTION #8 - Copy G(0) Data from Local File

An alternate data input mode from OPTION #1 is to use OPTION #8. The data may be contained in any local file and need not contain *EOR or *EOF statements. The file should not be a permanent file called MEMO and if it is a local file by that name, its contents will be overwritten upon normal program termination. The file must be in the proper format as shown below:

```
100 = 2                 Indicates G(0) information
110 = 3 3               M,P values
120 = -1. 5. 0.         G(0) matrix values by row
130 = 1. 2. 7.
140 = 1.2 4. .123E01
```

The matrix data values can be entered in any format. The line numbers above indicate that the data file may be created in the CYBER "EDITOR", however the line numbers must be suppressed when the file is saved into a local file.

When using this option, the program asks the user to specify and verify the name of a local file which holds the data. The data is then read into memory and the user can verify the data entries by using OPTIONS #101 and #102.

OPTION #9 - Copy Plant, Actuator and Sensor Info from Local File

This option copies plant state equations, actuator and sensor information from a local data file into computer storage location. The file name restrictions mentioned in the OPTION #8 discussion also apply to this option. This option is the alternate to OPTIONS #2 through #5. The *following* lines *show* the exact format for the data file:

```
100 = 1                Indicates state equation information
110 = 3 2 2            N,M,P values
120 = 1. 5.5 7E-02     A matrix values by row
130 = .2 -1.2 0
140 = .5 .005 99.1
150 = 4. 0.            B matrix values by row
160 = 1. 1.
170 = 0. 1.
180 = 0. 1 1.          C matrix values by row
190 = 1. 1. 1.
200 = N                Indicates no D matrix
210 = N                Indicates no actuators
220 = Y                Indicates no sensor data follows
230 = 1 2              States sensor #1 and sensor #2
240 = -9999.           Sensor #1 A matrix value
250 = 9999.            Sensor #1 B matrix value
260 = 1.               Sensor #1 C matrix value
270 = 0. 1.            Sensor #2 A matrix valuess by row
280 = -888.063 -147.2
290 = 0. 293.141       Sensor #2 B column values
300 = 1. 0.            Sensor #2 C row values
```

The matrix data values can be entered in any format. The line number above indicate that the data file may be created in the CYBER "EDITOR", however the line numbers must by suppressed when the file is saved into a local file. The *EOR and *EOF entries at the end of the listing are not required

When using OPTION #9, the program asks the user to specify and verify the name of a local file which holds the data. The data is then read into memory and the user can verify the data entries, if desired, by using OPTIONS #102 thru #105.

2.2 BLOCK 2 - Design Input Options

MULTI's control law development is based upon forming the controller matrix, K, from proper partitions of the G(0) or A, B, C and D matrices of the plant. In turn, KO and K1 matrices are created using the scalar variables ALPHA and EPSILON. The equations are:

for unknown plants...

$$u\ (kT) = T\ [\ KO * e\ (kT) = K1 * z\ (kT)] \qquad (C\text{-}3)$$

for regular or irregular plants...

$$u\ (kT) = f\ [KO * e\ (kT) + K1 * z\ (kT)] \qquad (C\text{-}4)$$

where

$$KO = ALPHA * EPSILON * K \qquad (C\text{-}5)$$

$$K1 = EPSILON * K \qquad (C\text{-}6)$$

$$f = 1\ /\ T \qquad (C\text{-}7)$$

In the above formulas,

u = input vector

e = proportional error vector

z = integral error vector

T = sampling period

The K matrix above is equivalent to $G(0)^{-1}$ * SIGMA, where SIGMA is a diagonal output weighting matrix, and

for unknown plants...

$$G(0) = D - [\ C * A^{-1} * B\ ] \qquad (C\text{-}8)$$

for regular plants...

$$G(0) = [\ C * E\ ] \qquad (C\text{-}9)$$

for irregular plants...

$$G(0) = [F_2 * B_2\ ] \qquad (C\text{-}10)$$

In the irregular case, the F_2 matrix is a partition of the measurement equation matrix which is formed as follows:

$$w = [\ F_1\ ,\ F_2\ ] = [C_1 + M * A_{11}\ ,\ C_2 + M * A_{12}\ ] \qquad (C\text{-}11)$$

AD-A118 117 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 1/3
DIGITAL FLIGHT CONTROL SYSTEM DESIGN USING SINGULAR PERTURBATIO--ETC(U)
DEC 81 J S SMYTH
UNCLASSIFIED AFIT/EE/GE/81D-55 NL

3 OF 3
AD
A118117

END
DATE
FILMED
09-82
DTIC

The program refers to the matrix, M, in Eqn. (C-11) as the measurement matrix.

OPTION #10 - List Options 10 thru 19

This option lists the design parameter input options 10 to 19.

OPTION #11 - Enter ALPHA

This option is used to set the proportion of direct to integral feedback, such that:

$$KO = ALPHA * K1 \qquad (C-12)$$

OPTION #12 - Enter SIGMA Weighting Matrix

In this option the SIGMA weighting matrix diagonal elements are entered. This matrix is used in forming the controller matrix, K, and has the effect of weighting the different inputs and outputs.

OPTION #13 - Enter EPSILON (SIGMA Weighting Matrix Multiplier)

In this option, EPSILON is entered. This scalar parameter is the SIGMA weighting matrix multiplier. It is multiplied to each diagonal element in the SIGMA matrix.

OPTION #14 - Run Design...Unknown, Regular & Irregular Plants

In this option the KO and K1 matrices are formed. The user is asked to enter the type of design to be accomplished and the reply can be either the full word or a single character.

When using an unknown design with state equation input from OPTION #3, the user can opt to print out the G(0) matrix after it is formed.

If the regular design is used, the program asks if the user wishes to see the C * B matrix after it is formed.

The first step in using the irregular design is to provide the program with a measurement matrix. If the measurement matrix is formed in OPTION #18 or previously entered, the user need not re-initialize the values unless they are to be changed. The rows of the matrix are entered after the prompt is given. When all matrix values are received the program enters the QUERY Mode. When forming the K0 and K1 matrices, the user has the option of printing the [F_1 , F_2] matrix.

When the matrices, K0 and K1, are formed, MULTI gives the user the message:

K0 & K1 MATRICES FORMED

and the specific matrix values can be checked by using OPTION #114.

OPTIONS #15 thru #17 - Reserved Options

OPTION #18 - Measurement Matrix Formation...or Row & Column Operations on C * B or other Matrix

The option has two functions as noted by the title.

If forming a measurement matrix to be used in the irregular plant controller design of OPTION #14, the user must enter the CESA feedback gain matrix, K1, from CESA Option #38 (Ref. 9). MULTI advises when the measurement matrix has been formed. It can be printed out by using OPTION #118.

This option can also be used to perform simple row and column operations on any matrix. There is provision for the user to perform these operations on the [C * B] matrix without entering or altering the actual matrix values. Otherwise the user must enter the matrix size and then the row values after each prompt. The matrix of interest can be no larger than 10 X 10. Ihe addition/subtraction and multiplication/division operations are defined as follows:

Enter X, Y, Z such that...

(X,row) * (Y,factor) + (Z,row) = (New Z,row)

(X,column) * (Y,factor) + (Z,column) = (New Z,column)

(X,row) * (Y,factor) = (New X,row)

or **(X,column) * (Y,factor) = (New X,column)**

After each operation the resulting matrix is printed so that the next operation can be determined. The final matrix is not stored into any memory location for future use in other options and is destroyed at the termination of the option.

OPTION #19 - Copy Design Parameter Data from Local File

This option is used as an alternative to OPTIONS #11 thru #14 and copies design parameter data from a local data file. As it is mandatory that the data file be in the proper format, the following example file is provided:

```
100=U or R or I          Indicates Unknown, Regular, Irregular
110= 2                   ALPHA value
120= .1 1 1              SIGMA matrix diagonal elements
130= .5                  EPSILON value
140= 4.6 4.6 4.6         KO matrix values by row
150= -4.6 -4.6 -4.6
160= 1. 1. 6.
170= 2.3 2.3 2.3         Kl matrix values by row
180= -2.3 -2.3 -2.3
190= .5 .5 3.
200= .25                 Measurement matrix values by row
210= 0.
220= 0.
```

If the ALPHA value is 1, the file must not contain Kl matrix values since Kl is identical to KO. The measurement matrix values are only read if the file contains data for am irregular plant design. The line numbers above indicate only that the data file may be created in the CYBER "EDITOR". However, the line numbers must not appear in the actual data file. The file terminators, *EOR and *EOF, are optional.

Pror to using OPTION #19, either OPTION #2 or OPTION #9 must be used to set the values for the number of inputs and outputs. To use OPTION #19, the user must supply and verify the name of the local file which holds the design parameter data. The program reads the data into the proper memory locations and the user can verify the values by choosing OPTIONS #111 to #114. Refer to OPTION #8 for a complete description of the restrictions that apply to local data file names.

2.3 BLOCK 3 - Simulation Options

The simulation options of this block are used to evaluate the controller matrices designed in OPTION #14. The simulation is performed after the user provides the initial values of the states and integrators, the command input vector , and the necessary time parameters. The actual simulation run is accomplished in OPTION #26. The simulation is obtained by numerically solving a set of ordinary differential equations formed from the plant, actuator and sensor state equation matrices and is run from time zero.

The ASD Library Subroutine ODE (Ref. 20) is the basis for the simulation. The user should be familiar with this subroutine, or as a minimum, have access to ODE's error descriptions.

During the simulation the control input is held constant over each sampling period. Since it may be advantageous to observe the system output between sampling periods, the user can specify a step time less than the sampling time.

As the simulation is run, a time-sequential printout of input and output values for every time interval of the simulation run can be selected. This printout includes actual input and controlled input values if the user has limited the inputs to certain values. If limitations are applied to the input values, the program generates a tabular listing of how often the limits are exceeded during the simulation.

The simulation data is then stored in a matrix for later printout or for plotting.

OPTION #20 - List Options 20 thru 29

This option lists the simulation options from 20 to 29.

OPTION #21 - Set State & Integrator Initial Values

In this option the user specifies the state and integrator values at time T(0). At present, all simulations must begin at zero time and proceed forward. The inital condition values for the states are entered after the first prompt message, and the integrator values are entered after the second prompt message. The values for each must be sequential.

OPTION #22 - Set Input Command Vector, V

This option is used to select the specific magnitude for each of the commanded inputs. At present, all inputs are step values. The entries for the input command vector are to be made sequentially.

OPTION #23 - Enter Sampling Times

The sampling times are entered in this option. There may be up to two different sampling times entered. The program first asks how many sampling times will be entered, and then prompts the user to enter the same number of sampling time values.

OPTION #24 - Enter Simulation Time, TT

OPTION #24 is used to enter the time length for the simulation run. The user is cautioned to select this time with care since the combination of simulation time/sample time or simulation time/step time determines the solution interval time. If the simulation time is too large, Subroutine ODE may not be able to finish a simulation run before the CYBER CP time limit. In this case, MULTI will be terminated without warning and no memory files will be generated.

OPTION #25 - Enter Calculation Step Size, ST

As mentioned in the introduction to this section, the system input or output values between sampling times may be of interest. This option allows the user to select the step time between each sampling period. If the step time is chosen larger than the sampling time, the program proceeds using the sampling time as the discrete time interval. For example, if sampling time is 0.02 seconds, a step size of 0.01 would give two sets of data during the sampling interval. If the step size was greater then .02 seconds, it is set to sampling time automatically.

OPTION #26 - Run Simulation

When OPTION #26 is chosen, MULTI forms and solves the set of ordinary differential equations formed from the plant, actuator and sensor state equations. Very fast actuators and sensors are approximated if no actual values have been entered and the user receives messages to indicate that no values have been set. The user also receives a message that no control limits have been applied to the inputs, if such is the case.

MULTI runs a simulation for each sampling time selected in OPTION

#23. As the program proceeds through each sampling time, the user is first given the length of the run, the step time and the sampling time. The user is then told how many time increments are generated during the simulation run. The user can opt to suppress the time-sequential listing of input/output values. The mesage:

—CALCULATIONS IN PROGRESS—

is provided if the time-sequential listing is suppressed, since the solution, in some cases, may take several seconds.

If the number of time increments over the simulation time interval is greater than 100, the input/output data must be "packed" into a matrix for later use in plotting or providing a tabular listing of the input/output values. When this occurs, missing time increments are noted when the tabular data of input/output values are listed. However, the user can always obtain a full list of these values by choosing the time-sequential listing.

At the conclusion of each simulation run, the program provides a table telling show many times the input values exceeded the limitations chosen in OPTION #27. This listing is not retained in memory and can only be accessed once.

The tabular listing of the input/output data points for each sampling time can be obtained from OPTION #126.

OPTION #27 - Set Control Input Limits

In this option the user can set input control limitations. The program supplies the correct directive on how to enter the limits and provides proper messages when the limitations are to be entered. The user tell which input is to limited and then gives the minimum and maximum values of the limitation.

This option can also be used to eliminate control input limitations if they are previously entered.

To obtain a listing of the current input limits, Option #127 is used.

OPTION #28 - Reserved Option

OPTION #29 - Copy Simulation Parameters for a Local File

This single option is used to enter state and integrator initial values, command vector values, sampling times, simulation run length and interval step time. It can be used as an alternate to OPTIONS #21 through #25. The reader should refer to the discussion of OPTION #8 for file name restrictions.

The data file must follow the format shown below:

100= 0. 0. 0.	Initial state values
110= 0. 0.	Initial integrator values
120= 1. -1.	Command vector values (transposed)
130= 2 .01 .02	# of sampling times, sampling times
140= 3	Simulation run time
150= .01	Step time

As with the other data file examples, the line numbers above only indicate that the data matrix may be created in the CYBER "EDITOR", and must be suppressed when the file is saved as a local file. The *EOR and *EOF terminators are optional.

Prior to using this option, the number of states, inputs and outputs must be set by using OPTION #2 or OPTION #9. To use OPTION #29 the user must enter and verify the name of the local file which holds the simulation data. After the program reads the data into memory, the user can verify the entries by using OPTIONS #21 to #25.

2.4 BLOCK 4 - Plotting Options

An integral aspect of any design is the ability to quickly interpret and analyze the results. The quickest method of analyzing results is to view them in either tabular or graphical form. MULTI provides the user with two types of graphical interpretation tools.

The first and fastest tool is a quick sketch at the user's terminal of any combination of input/output responses. The second tool is a CALCOMP plot. Each has its own advantages and limitations. The use of both is explained in this section.

To provide a plot MULTI requires several questions to be answered concerning what data is to be plotted. After using OPTION #31 and OPTION #34, the user becomes aware that an easier method of requesting plots is required. In this light, the short version of requesting plots is available by using OPTIONS #32 and #35. Finally, if the user desires the same information to be plotted after another simulation (for example: OUTPUT #2 vs. INPUT #2) and has already entered these choices by using OPTION #31, #32, #34 or #35, an immediate plot can be generated by using OPTION #33 or OPTION #36.

When OPTION #26 has been completed, MULTI has formed and stored a three-dimensional matrix of the data values. For this discusson it is only necessary to explain that the row dimension is equivalent to the input or output values at each time increment, the column dimension is equivalent to the input or output number, and the third dimension is equivalent to the sampling time. A plot is generated after a user chooses the type of plot to be generated, the sampling time of interest, input or output, the number of inputs or outputs and the particular inputs or outputs of interest. After this information is received, MULTI can proceed with plotting the correct data matrix rows and columns.

There are four basic types of plots that can be obtained. For a single sampling time, the user can select a plot of any output versus any input, a plot of one to four inputs or outputs on a single graph, or a plot of the same input or output for up to four different flight conditions. For the latter type of plot, the user returns to the main program to run another simulation and then re-enters the plotting routine. The last type of plot is for comparing inputs and outputs between different sampling times. When planning to plot data from two different sampling times, it is mandatory that the step time, for the simulation runs, be equal to or less than the smallest sampling time. If this restriction is not followed, MULTI is terminated early and no data files are generated by the program.

OPTIONS #31 through #33 pertain to terminal plots, or quick sketches at the user's terminal. OPTIONS #34 through #36 are for CALCOMP plots. As mentioned above, there are shortened versions of both the teiminal plot and CALCOMP plot requests. The same plots are obtained with these shortened versions as are produced by using the full versions. The difference lies in entering the data for the plot.

OPTION #30 - List Options 30 thru 39

This option lists the simulation options from 30 to 39.

OPTION #31 - Quick Sketch at User's Terminal

This option provides the user with 4 plotting choices. The choices are presented in a menu at the start of the option. Each choice has a set of questions associated with it. For example, choice #2 requires the user to enter the sampling time number, the number of inputs and outputs, and the specific inputs or outputs to be plotted. These questions are presented and need to be answered each time this option is used. Since this can be a tedious task for more that a few plots, the user may choose the shorter versions of this option, OPTION #32 or OPTION #33, when applicable.

OPTION #32 - Quick Sketch - Short Version

If the user is experienced in the use of MULTI, he may elect this option rather than OPTION #31. The required plotting data is entered in one data string as follows:

TYPE #1 - Input/Output Pairs

ENTER...# of pairs (OPTION #35 only), sampling time #, input #s, output #s

TYPE #2 - Inputs or Outputs

ENTER...sampling time #, # of inputs/outputs, input/output #s

TYPE #3 - Multiple Flight Conditions

ENTER...# of flight conditions, sampling time #, input/output #

TYPE #4 - Multiple Sampling Times

ENTER...# of sampling times, sampling time #s, input/output #

OPTION #33 - Quick Sketch - Retaining Same Plotting Choices

This option is self-explanatory. Once the information for a plot has been entered by using OPTIONS #31 or #32, the user may change a design parameter, run a simulation, and choose OPTION #33 to obtain a plot. This plot of the new simulation data is a plot of the same type as previously generated.

OPTION #34 - Calcomp Plot

This option provides the user with the same 4 plotting choices as in OPTION #31. The menu presented at the start of the option is the same, along with the questions associated with each menu choice. These questions must be answered each time this option is chosen and can become a tedious task. The experienced MULTI user may choose the shorter version, OPTION #35, or when applicable, OPTION #36. The program requests two titles from the user. The first is a 30 character maximum string that is put on the Y-axis of the plot. The second is a 60 character maximum strint that is put below the plot as a full title. The program then requests a plot size factor. Nominally, this is set at 0.9 for a Gx9 plot.

OPTION #35 - Calcomp Plot - Short Version

This options is essentially the same as OPTION #32 but is used for CALCOMP plots. The plotting data is entered as a data string with the format listed in OPTION #32's description.

OPTION #36 - Calcomp plot - Retaining Same Plotting Choices

This option is self-explanatory. Once the information for a plot has been entered by using OPTIONS #34 or #35, the user may change a design parameter, run a new simulation, and choose this option to obtain a plot. This plot of the simulation data is a plot of the same type as previously generated by OPTION #34 or OPTION #35.

The CALCOMP plots are stored in a local file called PLOT which the user can route to the plotter after MULTI is terminated. Local restrictions permit no more than five plots on one plotting file, thus, MULTI advises the user when five plots have been generated. All CALCOMP plots are 5 by 8 inches in size. The user is also required to provide a title for the plot. This title is placed along the Y-axis of the plot; the X-axis is always labeled "TIME". Titles can be a maximum of 20 characters in length.

2.5 BLOCK 5 - Printing Options

All available options in this block print data which the user has entered using the options in BLOCKS 1 to 4.

OPTION #100 - List Options 100 thru 130

This option lists the printing options from 100 to 130.

OPTIONS #101 to #129 - Print Data Entered in OPTIONS #1 to #29

As explained in Section 1.2, the options in this range are related to the sub-100 numbered options. To print out the data entered by OPTION #1, the user selects OPTION #101; to print out the data entered by OPTION #2, the user selects OPTION #102; etc.

OPTION #103 prints out the data from OPTION #2 and then gives the user the option of selecting the A, B, C or D matrix from OPTION #3.

ALPHA and EPSILON are combined in one printout and obtained by selecting OPTIONS #111 or #113.

OPTION #114 prints out the control matrices, K0 and K1, from OPTION #14 and also prints out a heading explaining which type of plant the user chose in generating the matrices. If this heading is suppressed, the program is unsure of how the matrices are generated, but the values displayed are still correct K0 and K1 matrices.

The date from OPTIONS #21 and #22 are also combined into one printout and the user obtains both sets of data when selecting OPTION #121 or OPTION #122.

The same is true for sampling times, simulation time and calculations step size. They form one printout and can be printed by selecting OPTION #123, #124 or #125.

When OPTION #126 is entered, the user is required to choose the tabular listing for the input or the output. The full word, INPUT or OUTPUT, must be entered and spelled correctly for the program to continue properly.

OPTION #130 - List Plotting Selections

This single option is used to determine the current plotting selections from the last entry to OPTIONS #31 to #36. When chosen, the program tells the user if the data is for a terminal plot or a CALCOMP plot and then gives the current plotting choices.

2.6 MULTI Generated Memory Files

When OPTION #99 is used to end MULTI, there are three data files generated. A file called MEMO is formed from the data entered from option BLOCK 1; a file called MEM10 is created from the data entered from option BLOCK 2; and a file called MEM20 is opened to hold the data entered in option BLOCK 3. These files are local files and should be rewound before they are store into permanent file space, edited, or routed to a line printer. Another filed called PLOT is also generated if the user selects any of the CALCOMP plotting options.

The routing command for MEMO, MEM10 and MEM20 is:

```
COMMAND - ROUTE,MEMO/MEM10/MEM20.DC=PR,TID=91,FID=XXXXX,ST=CSB
```

and the routing command for PLOT is:

```
COMMAND - ROUTE,PLOT,DC=PT,TID=91,FID=XXXXX,ST=CSB
```

These routing commands send the files to the facilities at AFIT.

The user can choose any 5 letter "flag ID" designated as "XXXXX" above.

3. Summary of MULTI's OPTIONS

MULTI contains four main options BLOCKS which have ten options each, and 30 printing options. The MULTI options are summarized below.

BLOCK 1 - Plant Input Options

Option # 0.	List Options 0 thru 9
Option # 1.	Enter G(0) Matrix
Option # 2.	Enter # of States, Inputs & Outputs (N,M,P)
Option # 3.	Enter Plant A, B, C & D Matrices
Option # 4.	Enter Actuator State Equation Matrix Data
Option # 5.	Enter Sensor State Equation Matrix Data
Option # 6.	Option Reserved
Option # 7	Option Reserved
Option # 8.	Copy G(0) Info from Local File
Option # 9.	Copy Plant, Actuator & Sensor Info from Local File

BLOCK 2 - Design Parameter Input Options

Option #10.	List Options 10 thru 19
Option #11.	Enter ALPHA
Option #12.	Enter SIGMA Weighting Matrix
Option #13.	Enter EPSILON (SIGMA Matrix Multiplier)
Option #14.	Run Design...Unknown, Regular & Irregular Plants
Option #15.	Option Reserved
Option #16.	Option Reserved
Option #17.	Option Reserved
Option #18.	Measurement Matrix Formation...or Row & Column Operations on C*B or other Matrix
Option #19.	Copy Design Parameters from Local File

BLOCK 3 - Simulation Options

Option #20.	List Options 20 thru 29
Option #21.	Set State and Integrator Initial Values, X(0) & Z(0)
Option #22.	Set Input Comand Vector V
Option #23.	Enter Sample Times
Option #24.	Enter Simulation Time
Option #25.	Enter Calculation Step Size
Option #26.	Run Simulation
Option #27.	Set Control Input Limits
Option #28.	Option Reserved
Option #29.	Copy Simulation Parameter from a Local File

BLOCK 4 - Plotting Options

Option #30.	List Opitons 30 thru 30

Option #31.	**Quick Sketch at User's Terminal**
Option #32.	**Quick Sketch - Short Version**
Option #33.	**Quick Sketch - Retaining Same Plotting Choices**
Option #34.	**CALCOMP Plot**
Option #35.	**CALCOMP Plot - Short Version**
Option #36.	**CALCOMP Plot - Retaining Same Plotting Choices**
Option #37.	**Option Reserved**
Option #38.	**Option Reserved**
Option #39.	**Option Reserved**

BLOCK 5 - Printing Options

All **100-Series Options Print Data Values...**
For Values Set in Option #1... Use Option #101
For Values Set in Option #2... Use Option #102
Etc.
For Plotting Selections........Use Option #130

To obtain a controller design and simulation these option numbers, in the following order, are generally used:

For Unknown Plants...

1 (or 9); 11, 12, 13, 14U

or **2 & 3 (or 9); 11 to 14U (or 19); 21 to 25 (or 29), 26; 31 to 36**

For Irregular Plants...

2 & 3 (or 9); 11 to 14I (or 19); 21 to 25 (or 29), 26; 31 to 36

If a measurement matrix is to be generated for the irregular case, OPTION #18 should be used prior to OPTION #14. In all cases, OPTIONS #4, #5 and #27 which set specific actuator, sensor and input control limits should be used, if desired, prior to OPTION #26.

4. **Example for Unknown Plants (Ref. 1)**

This appendix is concluded with a computer generated design and simulation using an unknown plant design defined by the following state equations:

$$\underline{x} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \underline{u}$$

$$\underline{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \underline{x}$$

with

ALPHA	= 0.9
SIGMA	= diag [2 1]
EPSILON	= 60
All Initial Values	= 0
Command Vector, V	= $[\ 3\ -1\ ]^T$
Sampling Time	= .02
Simulation Time	= 4
Calculation Step Size	= .02

This design includes no actuators, no sensors and no input control limitations.

```
COMMAND- ATTACH,MULTI,ID=TS20109
 PFN IS
 MULTI
 AT CY= 001 SN=AFIT
COMMAND- CONNECT,INPUT,OUTPUT
COMMAND- SCREEN,80
COMMAND- MULTI


WELCOME TO MULTIVARIABLE DESIGN
C 1981    PORTER,SMYTH,PASCHALL

THIS PROGRAM USES THE DESIGN TECHNIQUES DEVELOPED BY
PROFESSOR BRIAN PORTER, UNIV. OF SALFORD, ENGLAND


OPTION, PLEASE > #0

PLANT INPUT OPTIONS:
      0.   LIST OPTIONS 0 THRU 9
      1.   ENTER G(0) MATRIX
      2.   ENTER # OF STATES, INPUTS & OUTPUTS (N,M,P)
      3.   ENTER PLANT A, B, C, & D MATRICES
      4.   ENTER ACTUATOR STATE EQUATION MATRIX DATA
      5.   ENTER SENSOR STATE EQUATION MATRIX DATA
      6.   OPTION RESERVED
      7.   OPTION RESERVED.
      8.   COPY G(0) INFO FROM LOCAL FILE
      9.   COPY PLANT, ACTUATOR & SENSOR INFO FROM LOCAL FILE


OPTION, PLEASE > #2

THIS OPTION SETS THE NUMBER OF STATES, INPUTS & OUTPUTS

ENTER NUMBER OF STATES, INPUTS AND OUTPUTS >3,2,2


OPTION, PLEASE > #3

THIS OPTION ENTERS THE PLANT A,B,C, AND D MATRICES

ENTER "A" MATRIX...3 ROWS WITH 3 ELEMENTS EACH
ROW 1 >-1 0 0
ROW 2 >0 -2 0
ROW 3 >0 0 -3

PLANT MATRIX A...

  -.1000E+01    0.            0.
  0.            -.2000E+01    0.
  0.            0.            -.3000E+01
```

```
IS THIS CORRECT...YES,NO,$...>YES

ENTER "B" MATRIX...3 ROWS WITH 2 ELEMENTS EACH
ROW 1 >1 -1
ROW 2 >1 1
ROW 3 >1 0

PLANT MATRIX B...

   .1000E+01   -.1000E+01
   .1000E+01    .1000E+01
   .1000E+01   0.


IS THIS CORRECT...YES,NO,$...>Y

ENTER "C" MATRIX...2 ROWS WITH 3 ELEMENTS EACH
ROW 1 >1 0 0
ROW 2 >0 1 1

OUTPUT MATRIX C...

   .1000E+01   0.            0.
  0.            .1000E+01     .1000E+01


IS THIS CORRECT...YES,NO,$...>NO

ENTER "C" MATRIX...2 ROWS WITH 3 ELEMENTS EACH
ROW 1 >1 0 0
ROW 2 >0 1 0

OUTPUT MATRIX C...

   .1000E+01   0.            0.
  0.            .1000E+01    0.


IS THIS CORRECT...YES,NO,$...>YES

IS THERE A "D" MATRIX...YES OR NO...>NO


OPTION, PLEASE > #103

THERE ARE 3 STATES, 2 INPUTS & 2 OUTPUTS
ENTER...A,B,C,D...FOR PRINTOUT >A

PLANT MATRIX A...

  -.1000E+01   0.            0.
  0.           -.2000E+01    0.
  0.           0.            -.3000E+01
```

```
OPTION, PLEASE > #103

THERE ARE 3 STATES, 2 INPUTS & 2 OUTPUTS
ENTER...A,B,C,D...FOR PRINTOUT >B

PLANT MATRIX B...

   .1000E+01   -.1000E+01
   .1000E+01    .1000E+01
   .1000E+01   0.


OPTION, PLEASE > #103

THERE ARE 3 STATES, 2 INPUTS & 2 OUTPUTS
ENTER...A,B,C,D....FOR PRINTOUT >C

OUTPUT MATRIX C...

   .1000E+01   0.            0.
  0.             .1000E+01   0.


OPTION, PLEASE > #10

DESIGN PARAMETER INPUT OPTIONS:
     10.  LIST OPTIONS 10 THRU 19
     11.  ENTER ALPHA
     12.  ENTER SIGMA WEIGHTING MATRIX
     13.  ENTER EPSILON (SIGMA MATRIX MULTIPLIER)
     14.  RUN DESIGN...UNKNOWN, REGULAR & IRREGULAR PLANTS
     15.  OPTION RESERVED
     16.  OPTION RESERVED
     17.  OPTION RESERVED
     18.  MEASUREMENT MATRIX FORMATION...OR
          ROW & COLUMN OPERATIONS ON C*B OR OTHER MATRIX
     19.  COPY DESIGN PARAMETERS FROM LOCAL FILE

OPTION, PLEASE > #11

THIS OPTION SETS THE PROPORTION OF INTEGRAL AND DIRECT FEEDBACK

ENTER ALPHA >0.9


OPTION, PLEASE > #12

THIS OPTION SETS THE WEIGHTING BETWEEN OUTPUT CHANNELS

ENTER SIGMA WEIGHTING MATRIX...2 DIAGONAL ELEMENTS ONLY
DIAGONAL ELEMENTS >2,1
```

```
OPTION, PLEASE > #13

THIS OPTION SETS THE WEIGHTING MATRIX MULTIPLIER

ENTER SIGMA MATRIX SCALAR MULTIPLIER, EPSILON >60


OPTION, PLEASE > #14

ENTER DESIGN METHOD...UNKNOWN,REGULAR,IRREGULAR...>UNKNOWN

THIS OPTION COMPUTES K0 & K1 FOR UNKNOWN PLANTS

ENTER "0" TO SKIP G(0) MATRIX PRINTOUT
ENTER "1" TO OBTAIN THIS DATA PRINTOUT...>1

G(0) MATRIX...

   .1000E+01   -.1000E+01
   .5000E+00    .5000E+00


K0 & K1 MATRICES FORMED


OPTION, PLEASE > #114

CONTROL MATRICES ARE FOR PLANTS WHICH ARE UNKNOWN

K0 MATRIX...

   .5400E+02    .5400E+02
  -.5400E+02    .5400E+02


K1 MATRIX...

   .6000E+02    .6000E+02
  -.6000E+02    .6000E+02



OPTION, PLEASE > #20

SIMULATION OPTIONS:
     20.  LIST OPTIONS 20 THRU 29
     21.  SET STATE & INTEGRATOR INITIAL VALUES, X(0) & Z(0)
     22.  SET INPUT COMMAND VECTOR, V
     23.  ENTER SAMPLE TIMES
     24.  ENTER SIMULATION TIME
     25.  ENTER CALCULATION STEP SIZE
     26.  RUN SIMULATION
     27.  SET CONTROL INPUT LIMITS
     28.  OPTION RESERVED
     29.  COPY SIMULATION PARAMETERS FROM LOCAL FILE
```

```
OPTION, PLEASE > #21

THIS OPTION SETS THE INITIAL CONDITION VECTORS FOR THE STATES & INTEGRATORS

ENTER THE X(0) VECTOR OF 3 ELEMENTS
>0,0,0

ENTER THE Z(0) VECTOR OF 2 ELEMENTS
>0,0


OPTION, PLEASE > #22

THIS OPTION SETS THE INPUT COMMAND VECTOR, V

ENTER THE V VECTOR OF 2 ELEMENTS
"V" COLUMN >3,-1


OPTION, PLEASE > #23

THIS OPTION SETS THE SAMPLING TIME FOR EACH RUN

ENTER NUMBER (MAX OF 2) OF SAMPLING TIMES >1
ENTER 1 SAMPLING TIME(S) >.02


OPTION, PLEASE > #24

THIS OPTION SETS THE TOTAL SIMULATION TIME

ENTER TOTAL TIME >4


OPTION, PLEASE > #25

THIS OPTION SETS THE CALCULATION STEP SIZE

ENTER STEP SIZE >.02


OPTION, PLEASE > #26

THIS OPTION RUNS THE SIMULATION IF ALL DATA IS AVAILABLE

SIMULATION INCLUDES NO ACTUATORS
SIMULATION INCLUDES NO SENSORS
SIMULATION INCLUDES NO CONTROL LIMITS


RUN TIME=4.  STEP TIME=.02  SAMPLE TIME=.02

THERE ARE 200 TIME INCREMENTS
   ENTER "0" TO SKIP SEQUENTIAL LISTING
   ENTER "1" TO OBTAIN THIS DATA...>0

----CALCULATIONS IN PROGRESS----


SIMULATION FOR SAMPLING TIME #1 COMPLETE
```

```
OPTION, PLEASE > #126

ENTER...INPUT,OUTPUT...FOR DATA MATRIX >OUTPUT
ENTER SAMPLING TIME OF INTEREST AS...1,2,3, ETC...>1

OUTPUT DATA MATRIX...

     TIME              1             2
   .2000E-01     .1277E+00    -.4214E-01
   .6000E-01     .3798E+00    -.1229E+00
   .1000E+00     .6113E+00    -.1942E+00
   .1400E+00     .8236E+00    -.2572E+00
   .1800E+00     .1018E+01    -.3129E+00
   .2200E+00     .1196E+01    -.3622E+00
   .2600E+00     .1360E+01    -.4061E+00
   .3000E+00     .1509E+01    -.4453E+00
   .3400E+00     .1646E+01    -.4802E+00
   .3800E+00     .1772E+01    -.5116E+00
   .4200E+00     .1886E+01    -.5397E+00
   .4600E+00     .1991E+01    -.5651E+00
   .5000E+00     .2087E+01    -.5881E+00
   .5400E+00     .2175E+01    -.6090E+00
   .5800E+00     .2255E+01    -.6280E+00
   .6200E+00     .2328E+01    -.6454E+00
   .6600E+00     .2394E+01    -.6613E+00
   .7000E+00     .2455E+01    -.6760E+00
   .7400E+00     .2511E+01    -.6895E+00
   .7800E+00     .2561E+01    -.7021E+00
   .8200E+00     .2607E+01    -.7138E+00
   .8600E+00     .2649E+01    -.7246E+00
   .9000E+00     .2687E+01    -.7348E+00
   .9400E+00     .2721E+01    -.7444E+00
   .9800E+00     .2753E+01    -.7534E+00
   .1020E+01     .2781E+01    -.7619E+00
   .1060E+01     .2807E+01    -.7699E+00
   .1100E+01     .2831E+01    -.7775E+00
   .1140E+01     .2852E+01    -.7848E+00
   .1180E+01     .2871E+01    -.7917E+00
   .1220E+01     .2888E+01    -.7983E+00
   .1260E+01     .2904E+01    -.8045E+00
   .1300E+01     .2918E+01    -.8106E+00
   .1340E+01     .2931E+01    -.8163E+00
   .1380E+01     .2942E+01    -.8219E+00
   .1420E+01     .2952E+01    -.8272E+00
   .1460E+01     .2961E+01    -.8324E+00
   .1500E+01     .2969E+01    -.8373E+00
   .1540E+01     .2977E+01    -.8421E+00
   .1580E+01     .2983E+01    -.8467E+00
   .1620E+01     .2989E+01    -.8511E+00
   .1660E+01     .2994E+01    -.8554E+00
   .1700E+01     .2998E+01    -.8596E+00
   .1740E+01     .3002E+01    -.8636E+00
   .1780E+01     .3006E+01    -.8675E+00
   .1820E+01     .3009E+01    -.8713E+00
   .1860E+01     .3011E+01    -.8749E+00
```

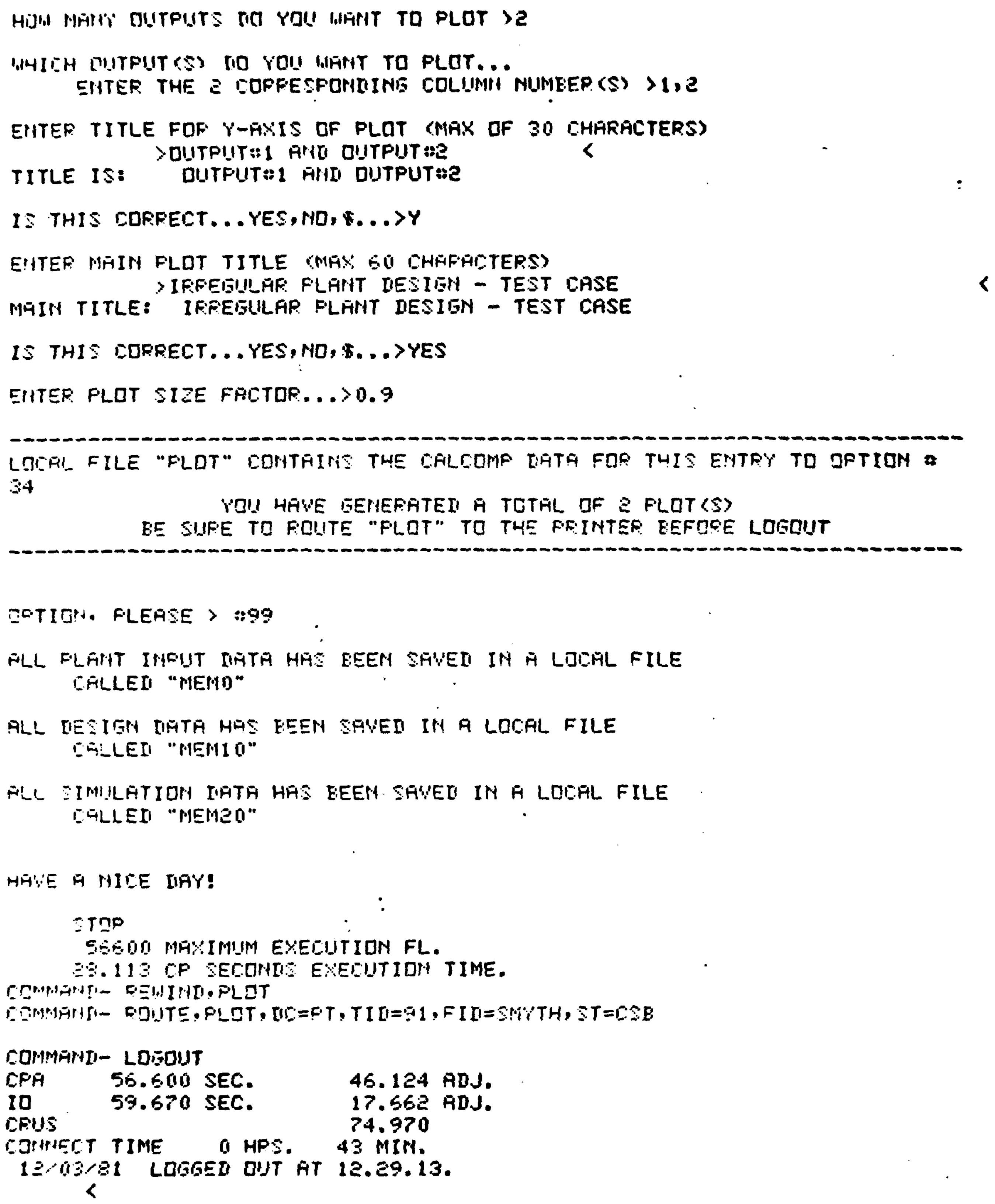
```
.1900E+01     .3014E+01     -.8785E+00
.1940E+01     .3015E+01     -.8819E+00
.1980E+01     .3017E+01     -.8853E+00
.2020E+01     .3018E+01     -.8885E+00
.2060E+01     .3019E+01     -.8916E+00
.2100E+01     .3020E+01     -.8947E+00
.2140E+01     .3021E+01     -.8976E+00
.2180E+01     .3022E+01     -.9005E+00
.2220E+01     .3022E+01     -.9033E+00
.2260E+01     .3022E+01     -.9060E+00
.2300E+01     .3022E+01     -.9086E+00
.2340E+01     .3022E+01     -.9111E+00
.2380E+01     .3022E+01     -.9136E+00
.2420E+01     .3022E+01     -.9160E+00
.2460E+01     .3022E+01     -.9184E+00
.2500E+01     .3022E+01     -.9207E+00
.2540E+01     .3021E+01     -.9229E+00
.2580E+01     .3021E+01     -.9250E+00
.2620E+01     .3021E+01     -.9271E+00
.2660E+01     .3020E+01     -.9291E+00
.2700E+01     .3020E+01     -.9311E+00
.2740E+01     .3019E+01     -.9331E+00
.2780E+01     .3019E+01     -.9349E+00
.2820E+01     .3018E+01     -.9367E+00
.2860E+01     .3018E+01     -.9385E+00
.2900E+01     .3017E+01     -.9402E+00
.2940E+01     .3017E+01     -.9418E+00
.2980E+01     .3016E+01     -.9435E+00
.3020E+01     .3016E+01     -.9451E+00
.3060E+01     .3015E+01     -.9466E+00
.3100E+01     .3015E+01     -.9481E+00
.3140E+01     .3014E+01     -.9495E+00
.3180E+01     .3014E+01     -.9509E+00
.3220E+01     .3013E+01     -.9523E+00
.3260E+01     .3013E+01     -.9537E+00
.3300E+01     .3012E+01     -.9549E+00
.3340E+01     .3012E+01     -.9562E+00
.3380E+01     .3011E+01     -.9574E+00
.3420E+01     .3011E+01     -.9586E+00
.3460E+01     .3011E+01     -.9597E+00
.3500E+01     .3010E+01     -.9609E+00
.3540E+01     .3010E+01     -.9620E+00
.3580E+01     .3009E+01     -.9630E+00
.3620E+01     .3009E+01     -.9640E+00
.3660E+01     .3009E+01     -.9650E+00
.3700E+01     .3008E+01     -.9660E+00
.3740E+01     .3008E+01     -.9669E+00
.3780E+01     .3008E+01     -.9679E+00
.3820E+01     .3007E+01     -.9688E+00
.3860E+01     .3007E+01     -.9696E+00
.3900E+01     .3007E+01     -.9705E+00
.3940E+01     .3006E+01     -.9713E+00
.3980E+01     .3006E+01     -.9721E+00
```

```
OPTION, PLEASE > #30

PLOTTING OPTIONS:
     30.  LIST OPTIONS 30 THRU 39
     31.  QUICK SKETCH AT USER'S TERMINAL
     32.  QUICK SKETCH--SHORT VERSION
     33.  QUICK SKETCH--RETAINING SAME PLOTTING CHOICES
     34.  CALCOMP PLOT
     35.  CALCOMP PLOT--SHORT VERSION
     36.  CALCOMP PLOT--RETAINING SAME PLOTTING CHOICES
     37.  OPTION RESERVED
     38.  OPTION RESERVED
     39.  OPTION RESERVED


OPTION, PLEASE > #31

THIS OPTION PRODUCES A PLOT AT YOUR TERMINAL

PLEASE CHOOSE ONE OF THE FOLLOWING:

FOR A SINGLE SAMPLING TIME
       1...A PLOT OF UP TO 2 INPUT AND OUTPUT PAIRS
       2...A PLOT OF UP TO 4 INPUTS OR OUTPUTS
       3...A PLOT OF UP TO 4 DIFFERENT FLIGHT CONDITIONS
               (FOR ANY SINGLE INPUT OR OUTPUT)

OR FOR UP TO 4 DIFFERENT SAMPLING TIMES
       4...A PLOT OF ANY SINGLE INPUT OR OUTPUT

ENTER CHOICE DESIRED >2

CHOICE #2...YOU'VE CHOSEN TO PLOT INPUTS OR OUTPUTS

YOU MIGHT HAVE SELECTED TO RUN SEVERAL SAMPLING TIMES...
DESIGNATE THESE SAMPLING TIMES AS...1,2,3, ETC...
      ENTER THE 1 SAMPLING TIME(S) OF INTEREST >1

ENTER...INPUT OR OUTPUT...FOR YOUR PLOT >INPUT

HOW MANY  INPUTS DO YOU WANT TO PLOT >2

WHICH  INPUT(S) DO YOU WANT TO PLOT...
      ENTER THE 2 CORRESPONDING COLUMN NUMBER(S) >1,2

FOR NO GRID ON PLOT ENTER "0", FOR A GRID ENTER "1" >1
```

```
  2.54    -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
           XXX                                                         I
  1.81    -+ XXX +     +     +     +     +     +     +     +     +     +
           I    XXXX                                                   I
  1.03    -+     + XXXXXXX   +     +     +     +     +     +     +     +
           I              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX             I
  .353    -+     +     +     +     +     +     +     +    XXXXXXXXXXXXXX
           I                                                           I
 -.375    -+     +     +     +     +     +     +     +     +     +     +
           I                                                           I
 -1.10    -+     +     +     +     +     +     +     +     +     +     +
           I                                                           I
 -1.83    -+     +     +     +     +     +     +     +     +     +     +
           I                                                           I
 -2.56    -+     +     OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
           I      OOOOO                                                I
 -3.29    -+    OO     +     +     +     +     +     +     +     +     +
           I  OO                                                       I
 -4.02    -+ O   +     +     +     +     +     +     +     +     +     +
           OO                                                          I
 -4.75    -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

            0.                                                3.98

CURVE X ABOVE IS  INPUT 1
CURVE O ABOVE IS  INPUT 2


OPTION, PLEASE > #31

THIS OPTION PRODUCES A PLOT AT YOUR TERMINAL

PLEASE CHOOSE ONE OF THE FOLLOWING:

FOR A SINGLE SAMPLING TIME
      1...A PLOT OF UP TO 2 INPUT AND OUTPUT PAIRS
      2...A PLOT OF UP TO 4 INPUTS OR OUTPUTS
      3...A PLOT OF UP TO 4 DIFFERENT FLIGHT CONDITIONS
              (FOR ANY SINGLE INPUT OR OUTPUT)

OR FOR UP TO 4 DIFFERENT SAMPLING TIMES
      4...A PLOT OF ANY SINGLE INPUT OR OUTPUT

ENTER CHOICE DESIRED >2

CHOICE #2...YOU'VE CHOSEN TO PLOT INPUTS OR OUTPUTS

YOU MIGHT HAVE SELECTED TO RUN SEVERAL SAMPLING TIMES...
DESIGNATE THESE SAMPLING TIMES AS...1,2,3, ETC...
     ENTER THE 1 SAMPLING TIME(S) OF INTEREST >1

ENTER...INPUT OR OUTPUT...FOR YOUR PLOT >OUTPUT
```

```
HOW MANY OUTPUTS DO YOU WANT TO PLOT >2

WHICH OUTPUT(S) DO YOU WANT TO PLOT...
     ENTER THE 2 CORRESPONDING COLUMN NUMBER(S) >1,2

FOR NO GRID ON PLOT ENTER "0", FOR A GRID ENTER "1" >1



  3.22    -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
           I                  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  2.78    -+     +     + XXXXX     +     +     +     +     +     +     +
           I          XXX                                              I
  2.34    -+     +  XX +     +     +     +     +     +     +     +     +
           I       X                                                   I
  1.90    -+     XX    +     +     +     +     +     +     +     +     +
           I    XX                                                     I
  1.46    -+   XX+     +     +     +     +     +     +     +     +     +
           I  X                                                        I
  1.03    -+  X  +     +     +     +     +     +     +     +     +     +
           I X                                                         I
  .596    -+ X   +     +     +     +     +     +     +     +     +     +
           IX                                                          I
  .146    -X     +     +     +     +     +     +     +     +     +     +
           OO                                                          I
 -.293    -+ OO  +     +     +     +     +     +     +     +     +     +
           I   OOOOO                                                   I
 -.732    -+     +  OOOOOOOOOOOOOOO+     +     +     +     +     +     +
           I                       OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
 -1.17    -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

            0.                                                 3.98

CURVE X ABOVE IS OUTPUT 1
CURVE O ABOVE IS OUTPUT 2


OPTION, PLEASE > #34

THIS OPTION PRODUCES A CALCOMP PLOT

PLEASE CHOOSE ONE OF THE FOLLOWING:

FOR A SINGLE SAMPLING TIME
       1...A PLOT OF UP TO 2 INPUT AND OUTPUT PAIRS
       2...A PLOT OF UP TO 4 INPUTS OR OUTPUTS
       3...A PLOT OF UP TO 4 DIFFERENT FLIGHT CONDITIONS
               (FOR ANY SINGLE INPUT OR OUTPUT)

OR FOR UP TO 4 DIFFERENT SAMPLING TIMES
       4...A PLOT OF ANY SINGLE INPUT OR OUTPUT

ENTER CHOICE DESIRED >2

CHOICE #2...YOU'VE CHOSEN TO PLOT INPUTS OR OUTPUTS
```

```
YOU MIGHT HAVE SELECTED TO RUN SEVERAL SAMPLING TIMES...
DESIGNATE THESE SAMPLING TIMES AS...1,2,3, ETC...
     ENTER THE 1 SAMPLING TIME(S) OF INTEREST >1

ENTER...INPUT OR OUTPUT...FOR YOUR PLOT >INPUT

HOW MANY  INPUTS DO YOU WANT TO PLOT >2

WHICH  INPUT(S) DO YOU WANT TO PLOT...
     ENTER THE 2 CORRESPONDING COLUMN NUMBER(S) >1,2

ENTER TITLE FOR Y-AXIS OF PLOT (MAX OF 30 CHARACTERS)
          >INPUT#1 AND  INPUT #2          <
TITLE IS:    INPUT#1 AND  INPUT #2

IS THIS CORRECT...YES,NO,$...>Y

ENTER MAIN PLOT TITLE (MAX 60 CHARACTERS)
          >IRREGUCBR PLANT DESIGN- TEST CASE                               <
MAIN TITLE:  IRREGULAR PLANT DESIGN- TEST CASE

IS THIS CORRECT...YES,NO,$...>Y

ENTER PLOT SIZE FACTOR...>.9

------------------------------------------------------------
LOCAL FILE "PLOT" CONTAINS THE CALCOMP DATA FOR THIS ENTRY TO OPTION #
34
               YOU HAVE GENERATED A TOTAL OF 1 PLOT(S)
          BE SURE TO ROUTE "PLOT" TO THE PRINTER BEFORE LOGOUT
------------------------------------------------------------


OPTION, PLEASE > #34

THIS OPTION PRODUCES A CALCOMP PLOT

PLEASE CHOOSE ONE OF THE FOLLOWING:

FOR A SINGLE SAMPLING TIME
     1...A PLOT OF UP TO 2 INPUT AND OUTPUT PAIRS
     2...A PLOT OF UP TO 4 INPUTS OR OUTPUTS
     3...A PLOT OF UP TO 4 DIFFERENT FLIGHT CONDITIONS
            (FOR ANY SINGLE INPUT OR OUTPUT)

OR FOR UP TO 4 DIFFERENT SAMPLING TIMES
     4...A PLOT OF ANY SINGLE INPUT OR OUTPUT

ENTER CHOICE DESIRED >2

CHOICE #2...YOU'VE CHOSEN TO PLOT INPUTS OR OUTPUTS

YOU MIGHT HAVE SELECTED TO RUN SEVERAL SAMPLING TIMES...
DESIGNATE THESE SAMPLING TIMES AS...1,2,3, ETC...
     ENTER THE 1 SAMPLING TIME(S) OF INTEREST >1

ENTER...INPUT OR OUTPUT...FOR YOUR PLOT >OUTPUT
```



```
HOW MANY OUTPUTS DO YOU WANT TO PLOT >2

WHICH OUTPUT(S) DO YOU WANT TO PLOT...
     ENTER THE 2 CORRESPONDING COLUMN NUMBER(S) >1,2

ENTER TITLE FOR Y-AXIS OF PLOT (MAX OF 30 CHARACTERS)
           >OUTPUT#1 AND OUTPUT#2          <
TITLE IS:    OUTPUT#1 AND OUTPUT#2

IS THIS CORRECT...YES,NO,$...>Y

ENTER MAIN PLOT TITLE (MAX 60 CHARACTERS)
           >IRREGULAR PLANT DESIGN - TEST CASE                         <
MAIN TITLE:  IRREGULAR PLANT DESIGN - TEST CASE

IS THIS CORRECT...YES,NO,$...>YES

ENTER PLOT SIZE FACTOR...>0.9

--------------------------------------------------------------------
LOCAL FILE "PLOT" CONTAINS THE CALCOMP DATA FOR THIS ENTRY TO OPTION #
34
                YOU HAVE GENERATED A TOTAL OF 2 PLOT(S)
         BE SURE TO ROUTE "PLOT" TO THE PRINTER BEFORE LOGOUT
--------------------------------------------------------------------


OPTION, PLEASE > #99

ALL PLANT INPUT DATA HAS BEEN SAVED IN A LOCAL FILE
     CALLED "MEM0"

ALL DESIGN DATA HAS BEEN SAVED IN A LOCAL FILE
     CALLED "MEM10"

ALL SIMULATION DATA HAS BEEN SAVED IN A LOCAL FILE
     CALLED "MEM20"


HAVE A NICE DAY!

     STOP
      56600 MAXIMUM EXECUTION FL.
     29.113 CP SECONDS EXECUTION TIME.
COMMAND- REWIND,PLOT
COMMAND- ROUTE,PLOT,DC=PT,TID=91,FID=SMYTH,ST=CSB

COMMAND- LOGOUT
CPA      56.600 SEC.        46.124 ADJ.
IO       59.670 SEC.        17.662 ADJ.
CRUS                        74.970
CONNECT TIME     0 HRS.   43 MIN.
 12/03/81  LOGGED OUT AT 12.29.13.
     <
```

VITA

Joseph S. Smyth was born on November 26, 1958 in Hicksville, New York. He graduated from Chaminade - Julienne High School in June, 1976. In September, 1976 he entered Tulane University School of Engineering. He then received a full three and one-half year AFROTC scholarship. He graduated Magna Cum Laude with a Bachelor of Science Degree in Electrical Engineering and a commission in the United States Air Force in May, 1980. He entered the School of Engineering, Air Force Institute of Technology, in May, 1980. His next assignment is with the Armament Division, Air Force Systems Command, Eglin Air Force Base, Florida.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/GE/EE/81D-55

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
DIGITAL FLIGHT CONTROL SYSTEM DESIGN
USING SINGULAR PERTURBATION METHODS

5. TYPE OF REPORT & PERIOD COVERED
MS Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Joseph S. Smyth
2Lt USAF

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology (AFIT-EN)
Wright-Patterson AFB,Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
December,1981

13. NUMBER OF PAGES
222

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

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

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE; IAW AFR 190-17

Lynn Wolaver
LYNN E. WOLAVER
Dean for Research and
Professional Development

23 JUL 1982

AIR FORCE INSTITUTE OF TECHNOLOGY (ATC)
WRIGHT-PATTERSON AFB, OH 45433

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

Digital Flight Control Systems,singular perturbation methods, Flight Propulsion Control Coupling (FPCC), positive pitch pointing,vertical translation, straight climb command, variable sampling times,fast sampling error actuated control, flight control system robustness,control configured vehicle, computer-aided-design program.

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

In this report a single longitudinal tracker is developed for the aircraft for three differentflight conditions. The method used is the singular perturbation method applied to fast-sampling,output-feedback digital control. Each flight condition has three command modes: positive pitch pointing, vertical translation and straight climb. A sensitivity study is performed to validate the design and illustrate design parameter influences on system response. A computer-aided-design program , MULTI , is developed to assist

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

in the iterative design process. The program is fully interactive, user-oriented, and provides error protection. The program allows complete design and simulation of three types of control law designs: known-regular plants, known irregular plants, and unknown plants. The report contains a brief but complete summary of each of these control law design methods.

A user's manual and a programmer's manual are provided for further development of the program.

FILMED

9-8