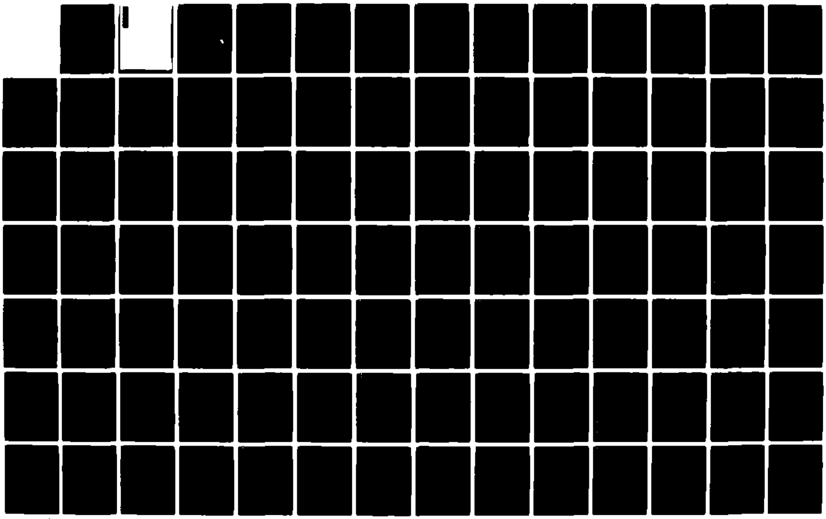
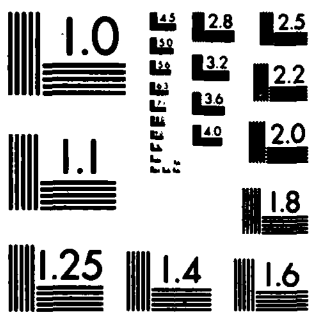
AD-A123 945 A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM (U) ILLINOIS UNIV AT URBANA DECISION AND CONTROL LAB V R SAKSENA APR 80 DC-37 N00014-79-C-0424 1/2

UNCLASSIFIED F/G 1/3 NL





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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A123945	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER R-878(DC-37); UILU-ENG-80-2210
7. AUTHOR(s) VIKRAM RAJ SAKSENA		8. CONTRACT OR GRANT NUMBER(s) AFOSR-78-3633 N00014-79-C-0424
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program		12. REPORT DATE April 1980
		13. NUMBER OF PAGES 89
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

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

Microprocessor control
Singular perturbation
Optimal output regulator

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

This report is concerned with the real-time control of an aircraft using a microcomputer system. The applicability of two optimal control theories--singular perturbation theory and output regulator theory--to this specific problem has been tested. Simulation results indicate that for systems possessing a two-time-scale property, such as an aircraft, singular perturbation theory provides a better solution than output regulator theory, and is also computationally more efficient.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM

BY

VIKRAM RAJ SAKSENA

B. Tech., Indian Institute of Technology, 1978

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1980

Thesis Advisor: Professor J. B. Cruz, Jr.

Urbana, Illinois

A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM

Vikram Raj Saksena, M.S.
Coordinated Science Laboratory and
Department of Electrical Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Abstract

This report is concerned with the real-time control of an aircraft using a microcomputer system. The applicability of two optimal control theories--singular perturbation theory and output regulator theory--to this specific problem has been tested. Simulation results indicate that for systems possessing a two-time-scale property, such as an aircraft, singular perturbation theory provides a better solution than output regulator theory, and is also computationally more efficient.

UILU-ENG-80-2210

A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM

by

Vikram Raj Saksena

This work was supported in part by the U. S. Air Force under Grant AFOSR-78-3633 and in part by the Joint Services Electronics Program under Contract N00014-79-C-0424.

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

March, 1980

WE HEREBY RECOMMEND THAT THE THESIS BY

VIKRAM RAJ SAKSENA

ENTITLED A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

Jose B Cruz Jr

Director of Thesis Research

Head of Department

Committee on Final Examination†

Chairman

† Required for doctor's degree but not for master's.

O-517

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his advisor, Professor J. B. Cruz, Jr., for his excellent guidance throughout the work. In addition, he would also like to thank Professor P. V. Kokotovic for helpful discussions and the CSL Computer Services staff for extensive programming assistance. Special thanks go to Ms. Rose Harris for typing the manuscript.

TABLE OF CONTENTS

1. INTRODUCTION

The need for more sophisticated digital flight controllers has become more apparent in recent years. With the advent and perfection of microcomputer systems, digital flight control systems have become extremely feasible for controlling and maneuvering the complex motions of a modern aircraft.

The works of Daly [1] and Jackson [2] have shown the merits of minicomputer based flight control systems. But from a practical viewpoint, microcomputer systems are more attractive for reasons of compactness. Particularly in recent years, with rapid advances in the LSI technology, more and more of the sophisticated features of a minicomputer are being incorporated into a microcomputer, without increasing its size. Reliability considerations also dictate the use of a multiple number of dedicated controllers, rather than a single large controller performing all the control operations. The present day microcomputer systems are ideally suited for dedicated controller applications as in an aircraft.

Optimal control techniques have been extensively applied for the design of flight control systems, due to the need for control and trajectory optimization. The dynamical equations of an aircraft being highly nonlinear, the direct application of these techniques is computationally involved. No closed form solution is available for such problems, and one has to resort to numerical methods, which might prove too slow for high speed real-time applications like in an aircraft. Hence, for practical reasons, the plant equations are linearized around equilibrium points corresponding to different flight conditions, and the standard results of linear regulator theory are applied for designing PID controllers. This has been attempted before by Daly [1] and Jackson [2].

A major restriction, from a practical viewpoint, of the optimal linear regulator theory is that the solution is obtained in a state feedback form. In most practical cases, such a control is difficult to implement due to the inaccessibility of all the state variables for feedback. In such cases, the optimal state regulator is implemented by generating the inaccessible states using a state observer. Adding a state observer increases the order of the system, and may result either in an increased cost if implemented in hardware, or an increased controller execution time if implemented in software. This may be unavoidable if the plant is not stabilizable without feedback from such inaccessible states. But in many cases, the plant can be stabilized and a satisfactory performance achieved, by suitably designing a linear output regulator.

Until recently, no systematic procedure had been formulated for designing an optimal output regulator. The works of Medanic, [3] and [4], now provide an efficient computational method for the design of static and dynamic output regulators.

It has been widely acknowledged that dynamic models of many physical systems possess a two-time-scale property, i.e., have 'slow' and 'fast' states. Singular perturbation theory [5], [6], [7], [8] exploits this property of systems to provide us with computationally efficient tools for designing controllers based on reduced-order models.

It has been noticed that linearized models of many aircrafts possess a two-time-scale property--pitch angle, velocity and altitude being the 'slow' variables, and angle of attack and pitch rate being the 'fast' variables.

Moreover, the 'fast' state variables are stable. It is also known that the 'fast' variables are more difficult to measure than the 'slow' variables which are directly available to the pilot on his control panel. Hence, the simplest controller design would involve only a knowledge of the three 'slow' states. Therefore, from the very nature of the problem, it is evident that both singular perturbation theory and output regulator theory can be directly applied to solve the aircraft control problem. The design based on singular perturbation theory would involve neglecting the 'fast' dynamics, and obtaining a reduced order model based only on the 'slow' variables. A state regulator would then be designed based on this reduced order model. The design based on output regulator theory would consider the 'slow' variables as the plant outputs. These outputs would then be used to design an optimal static output feedback. These two design methodologies can be easily extended to design dynamic PI controllers as well.

In this thesis, flight control systems have been designed based on singular perturbation theory and output regulator theory. The relative merits and demerits of these two design techniques has been examined based on their real time implementation on a Z-80 based microcomputer system.

2. AIRCRAFT MODELING

In order to proceed with any meaningful control system design, a mathematical description of the plant dynamics is first required. This is generally obtained in the form of a set of first-order ordinary differential equations. In this thesis, a simplified model of an airplane's longitudinal equations of motion is used.

2.1. Dynamical Equations

The dynamical equations for the aircraft model are derived based on a rigid body assumption (ignoring aeroelasticity etc.). In general an airplane coordinate system can be assumed to have the configuration as shown in Figure 2.1 where the symbols refer to the quantities as given in Table 2.1 [9]. For the types of aircrafts as the one studied here, the angle of attack (α) is usually small, and therefore small angle approximations can be made. This leads to the following

$$\sin\alpha \approx \alpha$$

$$\cos\alpha \approx 1$$

$$u = v\cos\alpha \approx v$$

$$\dot{u} = \dot{v}\cos\alpha - v\dot{\alpha}\sin\alpha \approx \dot{v} - v\dot{\alpha}\alpha \approx \dot{v}$$

$$\omega = v\sin\alpha \approx v\alpha$$

$$\dot{\omega} = \dot{v}\sin\alpha + v\dot{\alpha}\cos\alpha = \dot{v}\alpha + v\dot{\alpha} \approx v\dot{\alpha}$$

$$\begin{aligned}\sin\theta = \sin(\nu+\alpha) &= \sin\nu\cos\alpha + \sin\alpha\cos\nu \\ &\approx \sin\nu + \alpha\cos\nu \\ &\approx \sin\nu\end{aligned}$$

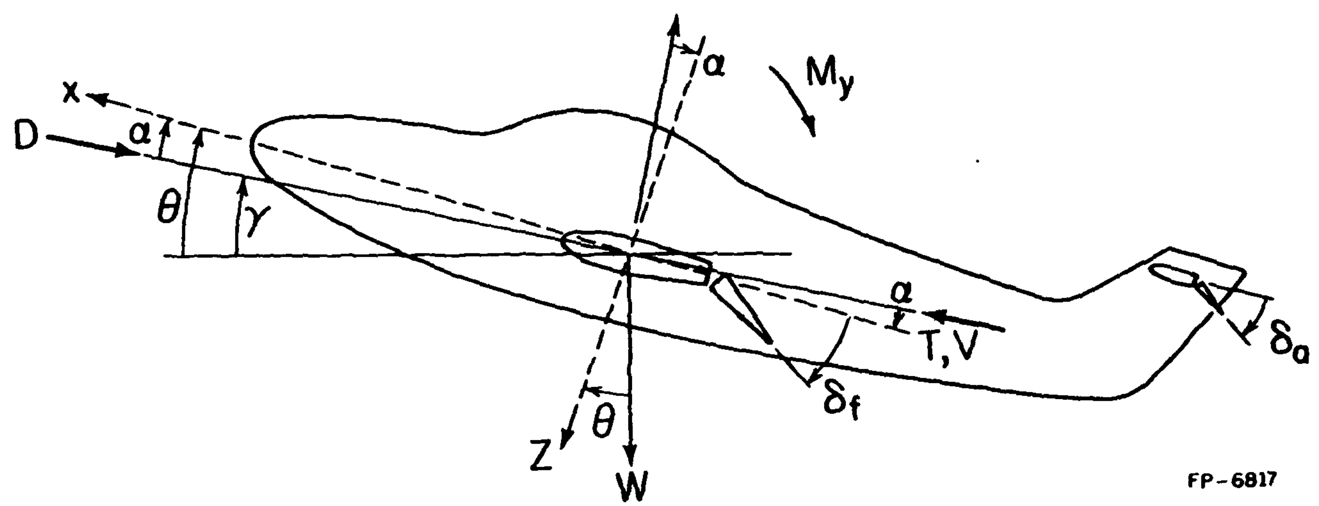


Figure 2.1. Airplane configuration.

Table 2.1. Definition of symbols used

α:	angle of attach
θ:	pitch angle
ν:	flight path angle
M:	mass of the aircraft
V:	velocity
H:	altitude
W:	weight of the aircraft
I_{YY}:	moment of inertia
X_{CG}:	center of gravity
S:	wing surface area
ρ:	air density
C:	chord length
X:	body axis
Z:	verticle normal to body axis
L:	lift force
D:	drag force
T:	thrust

$$\cos\theta = \cos(\nu+\alpha) = \cos\nu\cos\alpha - \sin\nu\sin\alpha$$
$$\approx \cos\nu - \alpha\sin\nu$$
$$\approx \cos\nu.$$

Summing the forces in the x-direction

$$-M\dot{u} - W\sin\theta + L\sin\alpha - (D-T)\cos\alpha = 0.$$

Now, using these approximations

$$-M\dot{V} - W\sin\nu - D + T \approx 0$$

or,

$$\dot{V} = \frac{1}{M}\,[T-D-W\sin\nu]. \tag{2.1}$$

Summing the forces in the Z-direction

$$-M(\dot{w}-u\theta) + W\cos\theta - L\cos\alpha - (D-T)\sin\alpha = 0.$$

Again, using the above approximations

$$-MV(\dot{\alpha}-\dot{\theta}) + W\cos\nu - L \approx 0$$

or,

$$\dot{\alpha} = \dot{\theta} - \frac{1}{MV}\,[L-W\cos\nu]. \tag{2.2}$$

Summing the moment in the Y-direction

$$I_{YY}\ddot{\theta} = M_y. \tag{2.3}$$

Also, for the rate of change of altitude we have

$$\dot{H} = V\sin\nu. \tag{2.4}$$

The lift, drag, and moment can be written as

$$\begin{aligned} L &= \frac{1}{2}\rho V^2 S C_\ell \\ D &= \frac{1}{2}\rho V^2 S C_d \\ M_y &= \frac{1}{2}\rho V^2 S c C_m \end{aligned} \tag{2.5}$$

where the coefficients C_ℓ, C_m, and C_d depend on wing plan form used and placement of the wing (and sometimes placement of the engines). All the coefficients in these equations can be found for any size airplane using the specified configuration and by looking up the wing specifications. These equations are generally simplified for mach numbers less than 1.0 by

$$\begin{aligned} C_\ell &= C_{\ell o} + C_{\ell a}\alpha + C_{\ell f}\delta_f \\ C_d &= C_{do} + C_\ell^2 + C_{df}\delta_f \\ C_m &= C_{mo} + C_{mc\ell}C_\ell + C_{me}\delta_e + C_{mf}\delta_f - \frac{C}{2V}(\dot\alpha + \dot\theta) \end{aligned} \tag{2.6}$$

where δ_f: flap deflection

δ_e: aileron deflection

δ_t: throttle position.

Any airplane can now be simulated, perhaps with minor modifications due to engine placement, tail configuration or Mach number. For simplicity, the coefficients of the GAT II simulation as described in Daly's thesis [1] are used with minor revisions.

Thrust is a more complicated subject. It is highly dependent on Mach number, altitude and the type of engine used (turboprop, turbofan, propeller, etc.). In general there are no easily found formulae for thrust. For simplicity, the thrust formulation (propeller) used in Daley's thesis was adopted, which is

$$\begin{aligned} Map &= C_{Po} + C_{pn}H + C_{pn}N + C_{pnt}N\delta_t \\ Bhp &= C_{bo} + C_{bn}N + C_{bp}Map + C_{bh}H \\ T &= Ne\ Bhp(C_{to} + C_{tv}V + C_{th}H + C_{tvh}VH) \end{aligned} \tag{2.7}$$

where Map: manifold pressure

N: RPM

Bhp: brake horsepower

Ne: number of engines.

The values of the various coefficients defined above are listed in Table 2.2.

The equilibrium flight conditions used are

$$V_o = 190\cdot 66 \text{ ft/sec.}$$
$$H_o = 2000 \text{ ft} \tag{2.8}$$
$$\theta_o = \dot{\theta}_o = \alpha_o = 0.$$

Now, we define the states x_1-x_5 and controls u_1-u_3 as

$$\begin{aligned} x_1 &= \alpha & u_1 &= \delta_e \\ x_2 &= V & u_2 &= \delta_f \\ x_3 &= \theta & u_3 &= \delta_t \\ x_4 &= \dot{\theta} & & \\ x_5 &= H. & & \end{aligned}$$

Combining equations (2.1)-(2.7) yields the fifth-order nonlinear system below

$$\dot{x}_1 = x_4 - \frac{1}{Mx_2} [\frac{1}{2} \rho x_2^2 S(C_{\ell o} + C_{\ell a} x_1 + C_{\ell f} u_2) - W \cos (x_3 - x_1)]$$

$$\begin{aligned} \dot{x}_2 = \frac{1}{M} [&-W \sin (x_3 - x_1) - \frac{1}{2} \rho x_2^2 S(C_{do} + C_{dc\ell}(C_{\ell o} + C_{\ell a} x_1 + C_{\ell f} u_2)^2 + C_{df} u_2) \\ &+ Ne(C_{to} + C_{tv} x_2 + C_{th} x_5 + C_{tvh} x_2 x_5)(C_{bo} + C_{bn} N + C_{bh} x_5 \\ &+ C_{bp}(C_{po} + C_{ph} N + C_{pn} x_5 + C_{pnt} N u_3))] \end{aligned}$$

Table 2.2. Aircraft parameters

$C_o = 0.0765$	$C_{po} = 29.92$
$C_a = 4.62$	$C_{ph} = -0.0009$
$C_f = 0.365$	$C_{pn} = -0.00076$
$C_{do} = 0.026$	$C_{pnt} = -0.0165$
$C_{dc} = 0.062$	$C_{bo} = -352.3$
$C_{df} = 0.021$	$C_{bn} = 0.1155$
$C_{mo} = 0.1$	$C_{bp} = 10.8$
$C_{mc} = -0.0529 + x_{cg}/c$	$C_{bh} = 0.0025$
$C_{me} = -0.0354$	$C_{to} = 3.5$
$C_{mf} = -0.0368$	$C_{tv} = -0.00642$
$C_{bhp} = 2.11$	$C_{th} = -4.73 \times 10^{-5}$
	$C_{tvh} = 8.7 \times 10^{-8}$

$N = 2500$ rpm

$Ne = 2$

$\rho = 0.004842$ slugs/ft^3

$S = 180$ ft^2

$x_{cg} = 0.2$ ft

$c = 5$ ft

$W = 4000$ lbs

$I_{YY} = 2050$ slugs ft^2

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{1}{I_{YY}} (\tfrac{1}{2} \rho x_2^2 Sc)[C_{mo} + C_{mc\ell}(C_{\ell o}+C_{\ell a}x_1+C_{\ell f}u_2) + C_{me}u_1 + C_{mf}u_2 - \frac{C}{2x_2} (\dot{x}_1+x_4)]$$

$$\dot{x}_5 = x_2 \sin (x_3-x_1). \tag{2.9}$$

2.2. Linearization

To get the linearized plant equations, we must first find the equilibrium point. The equilibrium states are specified by (2.8). To obtain the equilibrium controls, one must solve the system of equations

$$\dot{x}_e = f(x_e, u_e).$$

From $\dot{x}_1 = 0$ we obtain

$$u_{2e} = \frac{1}{C_{\ell f}} [\frac{W}{QS} - C_{\ell o} - C_{\ell a}\alpha_o]. \tag{2.10}$$

From $\dot{x}_4 = 0$ we obtain

$$u_{1e} = \frac{-1}{C_{me}} [C_{mo} + C_{mc\ell}C_{\ell} + C_{mf}u_{2e}]. \tag{2.11}$$

From $\dot{x}_2 = 0$ we obtain

$$u_{3e} = \frac{-1}{C_{bp}C_{pnt}N} [C_{bo} + C_{bp}C_{po} + N(C_{bn}+C_{bp}C_{pn}) + (C_{bh}+C_{bp}C_{ph})Ho - \frac{QS}{T_{bhp}} C_d] \tag{2.12}$$

where

$$Q = \tfrac{1}{2} \rho x_2^2$$

$$C_{\ell} = C_{\ell o} + C_{\ell a}x_1 + C_{\ell f}u_2$$

$$C_d = C_{do} + C_{dc\ell}C_{\ell}^2 + C_{df}u_2$$

$$Tbhp = Ne(C_{to} + C_{tv}x_2 + C_{th}x_5 + C_{tvh}x_2x_5).$$

Plugging in the values of the equilibrium states from (2.8) and the various coefficients from Table 2.2, we get the equilibrium controls as

$$
\begin{aligned}
u_{1e} &= 2.2344 \\
u_{2e} &= 0.4798 \\
u_{3e} &= 0.1816.
\end{aligned}
\tag{2.13}
$$

The linearized system is now obtained using the first order perturbation techniques.

Given the nonlinear system

$$\dot{x} = f(x,u).$$

Its linearized representation about the equilibrium point (x_e,u_e) is given by

$$\dot{x} = Ax + Bu$$

where

$$A = \left.\frac{\partial f}{\partial x}\right|_{x_e,u_e}$$

$$B = \left.\frac{\partial f}{\partial u}\right|_{x_e,u_e} .$$

The elements of the A and B matrices are listed in Tables 2.3 and 2.4 respectively.

Plugging in the numerical values from (2.8), (2.13), and Table 2.2, the linearized representation of the airplane model is obtained as

$$\dot{x} = \begin{bmatrix} -3.1 & -0.18 & 0 & 1 & 0 \\ 0.14 & -0.07 & -0.32 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -0.74 & 0.09 & 0 & -1.02 & 0 \\ -1.91 & 0 & 1.91 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & -0.25 & 0 \\ 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ -1.37 & -1.49 & 0 \\ 0 & 0 & 0 \end{bmatrix} u.$$

The states x_2 and x_5 have been scaled down by factors of 100 and 1000 respectively to facilitate implementation.

Table 2.3. Linearized 'A' coefficients

$$a_{11} = -\frac{QS}{MX_{2e}} C_{\ell a}$$

$$a_{12} = -\frac{\rho S}{2M} C_{\ell} - \frac{W}{MX_{2e}^{2}}$$

$$a_{14} = 1$$

$$a_{21} = g - 2Q\frac{S}{M} C_{dc\ell} C_{\ell a} C_{\ell}$$

$$a_{22} = -X_{2e} C_d \frac{S\rho}{M} + N_e (C_{tv} + C_{tvh} x_{5e}) \frac{Bhp}{M}$$

$$a_{23} = -g$$

$$a_{25} = \frac{N_e}{M} [(C_{th} + C_{tvh} x_{2e}) Bhp + Tbhp (C_{bh} + C_{bp} C_{ph})]$$

$$a_{41} = \frac{QSc}{I_{YY}} (C_{mc\ell} C_{\ell a} - \frac{c}{2x_{2e}} a_{11})$$

$$a_{42} = \frac{\rho X_{2e} Sc}{I_{YY}} [C_{mo} + C_{mc\ell} C_{\ell o} + C_{mc\ell} C_{\ell a} x_{1e} + C_{me} u_{1e}$$

$$+ (C_{mc\ell} C_{\ell f} + C_{mf}) u_{2e}] + \frac{Sc^{2}\rho}{4 I_{YY}} (-2x_{4e} + \frac{\rho X_{2e}}{M} SC_{\ell})$$

$$a_{43} = -\frac{QSc^{2}}{2I_{YY} x_{2e}} a_{13}$$

$$a_{44} = -\frac{QSc^{2}}{I_{YY} x_{2e}}$$

$$a_{51} = -x_{2e}$$

$$a_{53} = x_{2e}$$

$$a_{13} = a_{15} = a_{24} = a_{31} = a_{32} = a_{33} = a_{35} = a_{45} = a_{52}$$

$$= a_{54} = a_{55} = 0$$

Table 2.4. Linearized 'B' coefficients

$$b_{12} = -\frac{QS}{MX_{2e}} C_{\ell f}$$

$$b_{22} = -\frac{SQ}{M}(2C_{dc\ell}C_{\ell f}C_{\ell} + C_{df})$$

$$b_{23} = \text{Thbp } C_{bp}C_{pnt} \cdot N/M$$

$$b_{41} = \frac{QSc}{I_{YY}} C_{me}$$

$$b_{42} = \frac{QSc}{I_{YY}}\left(C_{mc\ell}C_{\ell f} + C_{mf} - \frac{c}{2x_{2e}} b_{12}\right)$$

$$b_{11} = b_{13} = b_{21} = b_{31} = b_{32} = b_{33} = b_{43} = b_{51} = b_{52} = b_{53} = 0$$

3. CONTROLLER DESIGN

In this section, a controller is designed for the aircraft, applying the techniques of optimal control theory. Two design methodologies--singular perturbation theory and output regulator theory--are studied and applied for designing the aircraft control system. Here, while discussing the two techniques, only the main results directly applicable to our design problem are given. The details are in references [4]-[8].

3.1. Singular Perturbation Theory

First, the general design steps are given, and then, these are directly applied to the aircraft control problem.

3.1.1. General problem

The problem considered here is not the most general problem which has been solved in singular perturbation literature. This is a more specific case which is directly applicable to our aircraft control problem.

Given a system which can be described by a set of differential equations of the following form

$$\begin{aligned} \dot{z}_1 &= A_{11}z_1 + A_{12}z_2 + B_1u; \qquad z_1(0) = z_{10} \\ \mu\dot{z}_2 &= A_{21}z_1 + A_{22}z_2 + B_2u; \qquad z_2(0) = z_{20} \end{aligned} \tag{3.1}$$

where $z_1 \in R^{n_1}$, $z_2 \in R^{n_2}$, $u \in R^m$, and $0 < \mu \ll 1$

and the performance index

$$J = \frac{1}{2}\int_0^\infty (z_1'Q_1z_1 + u'Ru)dt \tag{3.2}$$

where

$$Q_1 = Q' \geq 0 \quad \text{and} \quad R = R' > 0.$$

It is desired to obtain a feedback control $u = Fz$, such that the performance index (3.2) is minimized and the closed loop is asymptotically stable. It is assumed that the matrix A_{22} is stable.

The reduced order model, or the 'slow subsystem' is obtained by setting $\mu = 0$

$$\dot{Z}_s = A_o Z_s + B_o u_s; \qquad Z_s(0) = Z_{10}$$

$$\bar{Z}_2 = -A_{22}^{-1}(A_{21}Z_s + B_2 u_s) \tag{3.3}$$

where,

$$A_o = A_{11} - A_{12}A_{22}^{-1}A_{21}$$

$$B_o = B_1 - A_{12}A_{22}^{-1}B_2$$

$$J_s = \frac{1}{2}\int_o^\infty (z_s' Q z_s + u_s' R u_s)dt. \tag{3.4}$$

It is well known from optimal control theory, that the optimal control for (3.3), (3.4) is given by

$$u_s = -R^{-1}B_o' K_s Z_s \tag{3.5}$$

where K_s is the positive definite solution of the algebraic Riccati equation

$$A_o' K_s + K_s A_o + Q - K_s B_o R^{-1} B_o' K_s = 0. \tag{3.6}$$

Moreover, the control (3.5) when applied to the system (3.3) makes it asymptotically stable.

Singular perturbation theory goes on to show that if we apply the control

$$u = -R^{-1}B_o' K_s Z_1 = FZ_1 \tag{3.7}$$

to the system (3.1), then provided A_{22} is stable, there exists a $0 < \mu^* \ll 1$ such that the closed-loop system is asymptotically stable for any $\mu \in [0, \mu^*]$, and also

$$J_s(\text{opt}) = J(\text{opt}) + 0(\mu). \tag{3.8}$$

The solution to (3.1), with the control (3.7), is approximated for all finite $t \geq 0$ by

$$Z_1(t) = \exp[(A_o + B_o F)t] Z_s(0) + 0(\mu)$$

$$Z_2(t) = -A_{22}^{-1}(A_{22} + B_2 F)\exp[(A_o + B_o F)t] Z_s(0) + \exp[A_{22} t/\mu] Z_f(0) + 0(\mu)$$

where,

$$Z_s(0) = Z_{10}$$

$$Z_f(0) = Z_{20} - \bar{Z}_2(0). \tag{3.9}$$

3.1.2. Aircraft controller design

The linearized plane equations as given by (2.14) are

$$\dot{x} = \begin{bmatrix} -3.1 & -0.18 & 0 & 0 & 0 \\ 0.14 & -0.07 & -0.32 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.74 & 0.09 & 0 & -1.02 & 0 \\ -1.91 & 0 & 1.91 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & -0.25 & 0 \\ 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ -1.37 & -1.49 & 0 \\ 0 & 0 & 0 \end{bmatrix} u. \tag{3.10}$$

The eigenvalues of the open loop system are

$$0, \quad -0.02 \pm j0.18, \quad -1.52, \quad -2.62.$$

This indicates that (3.10) possesses a two-time-scale property. Hence we can represent (3.10) in the form (3.1).

An examination of the zero-input response of (3.10) indicates that the states x_2, x_3, and x_5 can be considered as 'slow' variables, and the states x_1 and x_4 can be considered as 'fast' variables. Introducing a fictitious parameter $\mu = 0.05$, the system (3.10) can be put in the form (3.1) as follows

$$\dot{z}_1 = \begin{bmatrix} -0.07 & -0.32 & 0 \\ 0 & 0 & 0 \\ 0 & 1.91 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 0.14 & 0 \\ 0 & 1 \\ -1.91 & 0 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u$$

$$\mu \dot{z}_2 = \begin{bmatrix} -0.009 & 0 & 0 \\ 0.0045 & 0 & 0 \end{bmatrix} z_1 + \begin{bmatrix} -0.155 & 0.05 \\ -0.037 & -0.051 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.125 & 0 \\ -0.0685 & -0.0745 & 0 \end{bmatrix} u$$

where,

$$Z_1 = [x_2 \quad x_3 \quad x_5]'$$

$$Z_2 = [x_1 \quad x_4]'. \tag{3.11}$$

The performance index is chosen to be

$$J = \frac{1}{2} \int_0^\infty (Z_1' Q Z_1 + u'Ru)\,dt$$

$$Q = R = I^{3\times 3}. \tag{3.12}$$

Letting $\mu \to 0$, we obtain the slow subsystem as

$$\dot{z}_s = \begin{bmatrix} -0.07 & -0.32 & 0 \\ 0.11 & 0 & 0 \\ 0.05 & 1.91 & 0 \end{bmatrix} z_s + \begin{bmatrix} -0.05 & -0.1 & -0.16 \\ -1.09 & -1.14 & 0 \\ 0.67 & 0.85 & 0 \end{bmatrix} u_s. \tag{3.13}$$

The solution of the algebraic Riccati equation (3.6) is obtained as

$$K_s = \begin{bmatrix} 4.29 & 0.27 & 0.71 \\ 0.27 & 2.75 & 1.6 \\ 0.71 & 1.6 & 1.49 \end{bmatrix} \tag{3.14}$$

Hence, from (3.7) we obtain

$$u = \begin{bmatrix} 0.03 & 1.93 & 0.78 \\ 0.14 & 1.79 & 0.62 \\ 0.69 & 0.04 & 0.11 \end{bmatrix} z_1 \tag{3.15}$$

Therefore, the partial state feedback to be applied to the original nonlinear plane (2.9) is given by

$$\begin{aligned} U_1 &= 2.2344 + 0.03(x_2-x_{2s}) + 1.93(x_3-x_{3s}) + 0.78(x_5-x_{5s}) \\ U_2 &= 0.4798 + 0.14(x_2-x_{2s}) + 1.79(x_3-x_{3s}) + 0.62(x_5-x_{5s}) \\ U_3 &= 0.1816 + 0.69(x_2-x_{2s}) + 0.04(x_3-x_{3s}) + 0.11(x_5-x_{5s}). \end{aligned} \tag{3.16}$$

The closed loop eigenvalues of the linearized system (3.10) with the control (3.15) are

$$-0.17, \quad -0.28 \pm j1.98, \quad -1.34, \quad -2.23.$$

For $x_o' = [1 \;\; 0 \;\; 1 \;\; 1 \;\; 0]$, the value of the performance index (3.12) with the control (3.15) is obtained as

$$J_s = 6.53.$$

This is to be compared with the optimal cost obtained on solving the full state regulator problem,

$$J(opt) = 6.27.$$

The controller designed above is alright if the airplane trajectory is to be regulated to the equilibrium flight conditions given by (2.8) in the absence of any disturbances. If there are any disturbances present, then satisfactory regulation will not be achieved in general. Also with the above controller, it is not possible to 'force' the desired states to any other set points.

In order to account for constant disturbances and to be able to regulate the states to other set points, an integral controller is to be incorporated.

Since the states of interest are the velocity, pitch angle, and altitude, three new states are defined as

$$\begin{aligned} \dot{x}_6 &= x_2 - v_{ref} \\ \dot{x}_7 &= x_3 - \theta_{ref} \\ \dot{x}_8 &= x_5 - H_{ref}. \end{aligned} \tag{3.17}$$

These new states are also considered as slow variables. The augmented system put in the form (3.1) is (with $\mu = 0.05$),

$$\dot{z}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.07 & -0.32 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.91 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.14 & 0 \\ 0 & 1 \\ -1.91 & 0 \end{bmatrix} z_2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u$$

$$\mu \dot{z}_2 = \begin{bmatrix} 0 & 0 & 0 & -0.009 & 0 & 0 \\ 0 & 0 & 0 & 0.0045 & 0 & 0 \end{bmatrix} z_1 + \begin{bmatrix} -0.155 & 0.05 \\ -0.037 & -0.051 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.125 & 0 \\ -0.0685 & -0.0745 & 0 \end{bmatrix} u$$

where,

$$\begin{aligned} z_1 &= [x_6 \quad x_7 \quad x_8 \quad x_2 \quad x_3 \quad x_5]' \\ z_2 &= [x_1 \quad x_4]'. \end{aligned} \tag{3.18}$$

The performance index is chosen to be

$$J = \frac{1}{2} \int_0^\infty (z_1' Q z_1 + u' R u) dt$$

where

$$Q = I^{6\times 6} \qquad R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}. \tag{3.19}$$

Letting $\mu \to 0$, we obtain the slow subsystem as

$$\dot{z}_s = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.07 & -0.32 & 0 \\ 0 & 0 & 0 & 0.11 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 1.91 & 0 \end{bmatrix} z_s + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0.05 & -0.1 & -0.16 \\ -1.09 & -1.14 & 0 \\ 0.67 & 0.85 & 0 \end{bmatrix} u_s . \tag{3.20}$$

Based on this reduced order model, the near-optimal control is obtained as

$$u = \begin{bmatrix} -0.47 & -0.28 & 3.11 & -0.01 & 9.44 & 6.64 \\ 0.11 & 1.4 & 0.14 & 0.16 & 0.86 & -0.12 \\ 3.12 & -1.28 & 0.44 & 6.67 & 0.49 & 1.29 \end{bmatrix} z_1 . \tag{3.21}$$

The eigenvalues of the linearized closed loop system with the control of (3.21) are

$$-0.11, \quad -0.23 \pm j3.3, \quad -0.56 \pm j0.43, \quad -0.89, \quad -1.34 \pm j1.04.$$

For $x_o' = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]$, the value of the performance index (3.19) with the control (3.21) is obtained as

$$J_s = 16.54.$$

This is to be compared with the optimal cost obtained on solving the full state regulator problem

$$J(\text{opt}) = 15.89.$$

The partial state feedback to be applied to the original nonlinear plant (2.9), (3.17) is given by

$$U_1 = 2.2344 - 0.01(x_2 - x_{2s}) + 9.44(x_3 - x_{3s}) + 6.64(x_5 - x_{5s}) - 0.46x_6 - 0.28x_7 + 3.11x_8$$
$$U_2 = 0.4798 + 0.16(x_2 - x_{2s}) + 0.86(x_3 - x_{3s}) - 0.12(x_5 - x_{5s}) + 0.11x_6 + 1.4x_7 + 0.14x_8$$
$$U_3 = 0.1816 + 6.67(x_2 - x_{2s}) + 0.49(x_3 - x_{3s}) + 1.29(x_5 - x_{5s}) + 3.12x_6 - 1.28x_7 + 0.44x_8 \tag{3.22}$$

3.2. Output Regulator Theory

Here again, the general design steps are given first and then these are directly applied to the aircraft control problem.

3.2.1. General problem

Given the system

$$\dot{z}_1 = A_{11}Z_1 + A_{12}Z_2 + B_1u; \qquad Z_1(0) = Z_{10}$$
$$\dot{Z}_2 = A_{21}Z_1 + A_{22}Z_2 + B_2u; \qquad Z_2(0) = Z_{20}$$
$$y = Z_1$$

where

$$Z_1 \in R^n, \quad Z_2 \in R^r, \quad u \in R^m \tag{3.23}$$

and the performance index,

$$J = \frac{1}{2}\int_0^\infty (z_1'Qz_1 + u'Ru)dt$$

where

$$Q = Q' \geqslant 0 \quad \text{and} \quad R = R' > 0. \tag{3.24}$$

It is desired to find a control

$$u = Ky$$

which minimizes (3.24). In order to find K, we proceed as follows. First, the full state regulator problem for (3.23), (3.24) is solved. Define $S = BR^{-1}B'$ and $F = A - SM_c$, where M_c is the positive definite solution of the algebraic Riccati equation

$$A'M_c + M_cA + Q - M_cBR^{-1}B'M_c = 0 \tag{3.28}$$

and

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad ; \; B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}.$$

Let $x = \begin{vmatrix} Y \\ Z \end{vmatrix}$, $Y \in R^{r\times r}$ consist of the subset of r eigenvectors of F associated with a particular subspectrum Λ_r that we wish to retain in the output regulator.

It has been shown in [3] that, if, for some Λ_r, the matrix $A_r = A_{22} - NA_{12}$, where $N = ZY^{-1}$, is stable; then there exists a unique output feedback gain matrix K such that the closed loop system A_c is asymptotically stable, and

$$\Lambda(A_c) = \Lambda_r \cup \Lambda(A_r).$$

The optimal control is given by

$$u = -R^{-1}B'M_cPy \tag{3.26}$$

where

$$P = \begin{vmatrix} I \\ N \end{vmatrix}.$$

The cost matrix associated with the control (3.26) is

$$M_o = M_c + V'D_oV \tag{3.27}$$

where

$$V = [-N \quad I]$$

and D_o is the unique positive definite solution of the Lyapunov equation

$$A_r'D_o + D_oA_r + G_o = 0 \tag{3.28}$$

where

$$G_o = [0 \quad I]M_cSM_c[0 \quad I]'.$$

3.2.2. Aircraft controller design

The linearized plant equations gi ·n by (2.14) are put in the form (3.23),

$$\dot{z}_1 = \begin{bmatrix} -0.07 & -0.32 & 0 \\ 0 & 0 & 0 \\ 0 & 1.91 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 0.14 & 0 \\ 0 & 1 \\ -1.91 & 0 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u$$

$$\dot{z}_2 = \begin{bmatrix} -0.18 & 0 & 0 \\ 0.09 & 0 & 0 \end{bmatrix} z_1 + \begin{bmatrix} -3.1 & 1 \\ -0.74 & -1.02 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.25 & 0 \\ -1.37 & -1.49 & 0 \end{bmatrix} u$$

$$y = z_1$$

where

$$z_1 = [x_2 \quad x_3 \quad x_5]'$$
$$z_2 = [x_1 \quad x_4]'. \tag{3.29}$$

The performance index is chosen to be

$$J = \frac{1}{2}\int_0^\infty (Z_1'QZ_1 + u'Ru)dt$$
$$Q = R = I^{3\times 3}. \tag{3.30}$$

Solving (3.25) we obtain the cost for the full state regulator problem as

$$M_c = \begin{bmatrix} 4.28 & 0.27 & 0.68 & -0.24 & 0.02 \\ 0.27 & 6.75 & 2.78 & -1.86 & 1.83 \\ 0.68 & 2.78 & 1.87 & -1 & 0.57 \\ -0.24 & -1.86 & -1 & 0.61 & -0.44 \\ 0.02 & 1.83 & 0.57 & -0.44 & 0.64 \end{bmatrix}$$

$$F = A-BR^{-1}B'M_c = \begin{bmatrix} -0.19 & -0.42 & -0.04 & 0.17 & -0.03 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1.91 & 0 & -1.91 & 0 \\ -0.22 & -0.57 & -0.16 & -2.97 & 0.79 \\ -0.16 & -6.82 & -1.99 & 0.87 & -3.47 \end{bmatrix}.$$

The eigenvalues of F are

$$-0.17, \quad -1.03 \pm j1.22, \quad -1.81, \quad -2.59.$$

It was found that the only set of 3 eigenvalues which can be retained while satisfying the sufficient condition for output stabilizability are

$$-1.03 \pm j1.22, \quad -1.81.$$

The components of the corresponding eigenvectors are

$$Y = \begin{bmatrix} 6.53 & -0.69 & 3.98 \\ 18.36 & -19.49 & 13.28 \\ -39.59 & 18.68 & -34.11 \end{bmatrix} \quad ; \quad Z = \begin{bmatrix} 9.03 & 15.84 & -19.02 \\ 4.95 & 42.39 & -24.03 \end{bmatrix}$$

$$N = ZY^{-1} = \begin{bmatrix} 14.25 & 1.3 & 2.73 \\ 17.54 & -0.25 & 2.65 \end{bmatrix}$$

$$A_r = A_{22}-NA_{12} = \begin{bmatrix} 0.11 & -0.3 \\ 1.87 & -0.77 \end{bmatrix}$$

$$\Lambda(A_r) = -0.33 \pm j0.6.$$

Hence, A_r being stable the sufficient condition is satisfied

$$P = \begin{bmatrix} I \\ N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 14.25 & 1.3 & 2.73 \\ 17.54 & -0.25 & 2.65 \end{bmatrix}.$$

Hence, the output feedback gain matrix is

$$K = -R^{-1}B'M_cP = \begin{bmatrix} 6.67 & 1.5 & 1.43 \\ 7.47 & 1.39 & 1.43 \\ 0.2 & -0.01 & 0.01 \end{bmatrix}$$

$$G_o = [0 \quad I]M_cSM_c[0 \quad I]' = \begin{bmatrix} 0.64 & -0.97 \\ -0.97 & 1.47 \end{bmatrix}.$$

The solution of (3.28) is obtained as

$$D_o = \begin{bmatrix} 4.87 & -0.46 \\ -0.96 & 1.13 \end{bmatrix}$$

$$V = [-N \quad I] = \begin{bmatrix} -14.25 & -1.3 & -2.73 & 1 & 0 \\ -17.54 & 0.25 & -2.65 & 0 & 1 \end{bmatrix}.$$

Hence, from (3.27) we obtain

$$M_o = M_c + V'D_oV = \begin{bmatrix} 1104 & 75.88 & 201.9 & -60.97 & -13.35 \\ 75.87 & 15.27 & 17.89 & -8.24 & 2.7 \\ 201.9 & 17.89 & 39.17 & -12.96 & -1.188 \\ -60.97 & -8.24 & -12.96 & 5.43 & -0.895 \\ -13.35 & 2.7 & -1.188 & -0.895 & 1.77 \end{bmatrix}.$$

Therefore, from (3.26) we obtain

$$u = \begin{bmatrix} 6.67 & 1.5 & 1.43 \\ 7.47 & 1.39 & 1.43 \\ 0.2 & -0.01 & 0.01 \end{bmatrix} y. \qquad (3.31)$$

The eigenvalues of the linearized closed loop system are

$$-0.33 \pm j0.6, \quad -1.63 \pm j1.22, \quad -1.81.$$

For $x_o' = [1 \ \ 0 \ \ 1 \ \ 1 \ \ 0]$, the optimal cost with full state feedback is

$$J(opt) = 6.27.$$

The cost with the control (3.31) is

$$J = 1405.$$

It is to be noted here that the difference in the two costs is more when the controller is designed based on output regulator theory as compared with the difference when it is designed based on singular perturbation theory. This is explained later after studying their performance in real-time implementation.

The partial state feedback to be applied to the original nonlinear plant (2.9) is given by

$$\begin{aligned} U_1 &= 2.2344 + 6.67(x_2 - x_{2s}) + 1.5(x_3 - x_{3s}) + 1.43(x_5 - x_{5s}) \\ U_2 &= 0.4798 + 7.47(x_2 - x_{2s}) + 1.39(x_3 - x_{3s}) + 1.43(x_5 - x_{5s}) \\ U_3 &= 0.1816 + 0.2(x_2 - x_{2s}) - 0.01(x_3 - x_{3s}) + 0.01(x_5 - x_{5s}). \end{aligned} \qquad (3.32)$$

As before, a PI controller is now designed by augmenting the plant with the three new states defined by (3.17). The augmented system put in the form (3.23) is

$$\dot{z}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.07 & -0.32 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.91 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.14 & 0 \\ 0 & 1 \\ -1.91 & 0 \end{bmatrix} z_2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u$$

$$\dot{z}_2 = \begin{bmatrix} 0 & 0 & 0 & -0.18 & 0 & 0 \\ 0 & 0 & 0 & 0.09 & 0 & 0 \end{bmatrix} z_1 + \begin{bmatrix} -3.1 & 1 \\ -0.74 & -1.02 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.25 & 0 \\ -1.37 & -1.49 & 0 \end{bmatrix} u$$

$$y = z_1$$

where

$$z_1 = [x_6 \quad x_7 \quad x_8 \quad x_2 \quad x_3 \quad x_5]'$$

$$z_2 = [x_1 \quad x_4]'. \tag{3.33}$$

The performance index is chosen to be

$$J = \frac{1}{2} \int_0^\infty (z_1' Q z_1 + u' R u) dt \tag{3.34}$$

where

$$R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad \text{and } Q = I^{6\times6}.$$

On solving the state regulator problem, the closed loop eigenvalues are obtained as

$$-0.1, \quad -0.56 \pm j0.43, \quad -0.99, \quad -1.43 \pm j1.82, \quad -2.32 \pm j0.23.$$

Retaining the first six eigenvalues in the output regulator, we get

$$N = ZY^{-1} = \begin{bmatrix} -0.75 & 1.05 & 5.83 & -0.24 & 8.1 & 8.54 \\ -1.4 & 1.87 & 10.7 & -0.41 & 12.43 & 14.88 \end{bmatrix}$$

$$A_r = A_{22} - NA_{12} = \begin{bmatrix} 12.35 & -7.1 \\ 27.75 & -13.45 \end{bmatrix}$$

$$\Lambda(A_r) = -0.1 \pm j4.35$$

$$P = \begin{vmatrix} I \\ N \end{vmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -0.75 & 1.05 & 5.83 & -0.24 & 8.1 & 4.54 \\ -1.4 & 1.87 & 10.7 & -0.41 & 12.43 & 14.88 \end{bmatrix}$$

$$K = -R^{-1}B'M_cP = \begin{bmatrix} -1.07 & 0.37 & 7.68 & -0.14 & 12.5 & 11.97 \\ -0.49 & 2.23 & 4.92 & 0 & 6.79 & 6.58 \\ 3.45 & -0.8 & -2.2 & 6.74 & -3.19 & -2.56 \end{bmatrix}.$$

Therefore, from (3.26), we obtain

$$u = \begin{bmatrix} -10.7 & 0.37 & 7.68 & -0.14 & 12.6 & 11.97 \\ -0.49 & 2.23 & 4.92 & 0 & 6.79 & 6.58 \\ 3.45 & -0.8 & -2.2 & 6.74 & -3.19 & -2.56 \end{bmatrix} z_1. \tag{3.35}$$

The eigenvalues of the linearized closed loop system are

$$-0.1, \quad -0.1 \pm j4.35, \quad -0.56 \pm j0.43, \quad -0.99, \quad -1.43 \pm j1.82.$$

For

$$x_o' = [1 \ \ 0 \ \ 1 \ \ 1 \ \ 0 \ \ 0 \ \ 0 \ \ 0],$$

the optimal cost with full state feedback is

$$J(\text{opt}) = 15.89.$$

The cost with the control (3.35) is

$$J = 23.56.$$

It is to be noted here that the difference in the two costs is not so much as was in the previous case with no integral control. This is because now we were able to retain all the 'small' eigenvalues in the output regulator as opposed to the last design where this could not be possible.

The partial state feedback to be applied to the original nonlinear plant (2.9), (3.17) is given by

$$
\begin{aligned}
U_1 &= 2.2344 - 0.14(x_2 - x_{2s}) + 12.6(x_3 - x_{3s}) + 11.97(x_5 - x_{5s}) - 1.07x_6 + 0.37x_7 + 7.68x_8 \\
U_2 &= 0.4798 + 6.79(x_3 - x_{3s}) + 6.58(x_5 - x_{5s}) - 0.49x_6 + 2.23x_7 + 4.92x_8 \\
U_3 &= 0.1816 + 6.74(x_2 - x_{2s}) - 3.19(x_3 - x_{3s}) - 2.56(x_5 - x_{5s}) + 3.45x_6 - 0.8x_7 - 2.2x_8.
\end{aligned}
\tag{3.36}
$$

The controllers have been designed based on a continuous-time model of the plant as opposed to a discrete model which would have been more appropriate. This was done because it was not known beforehand what sampling period would be used; and also due to the fact that when sampled fast enough, the response from real-time implementation would closely approximate the response from simulation of the continuous-time system.

4. REAL TIME IMPLEMENTATION

4.1. Simulation

All preliminary simulation, to get the analytical results for all the controllers just derived, was done on the CYBER 175 digital computer. Computer programs had to be written to perform all of the integrations and other related operations needed. Because of the size of the program and the need for versatility of input data, an interactive format was utilized. This method of having the operator respond to different options (e.g. initial conditions) helped facilitate debugging of the program also. Furthermore, this made it possible to study any flight condition by a simple response to a parameter change option. The only true shortcoming involved here was that the program did not have the option of generating feedback matrices (these were obtained beforehand using the LINSYS [10] and LAS packages) so the responses to different conditions (other than the initially chosen one) were suboptimal in some sense.

All of the interactive programming and condition organization was done with one main program. This program would ask for the desired flight conditions and would then make calls to the various subprograms needed to facilitate these. The subprograms would then execute the different commands such as for integration or plots. Integrations were performed using subroutines from IBM's IMSL package and the plots were obtained using the CALCOMP plotting package.

4.2. Implementation

The AD-5 analog computer had the nonlinear aircraft plant equations patched onto it, thus simulating the dynamics of a real time airplane. This required a lot of manipulation and scaling due to the limited amount of hardware available, and due to saturation restrictions.

To help set up and test this, several PDP-11 programs were used. Again, here, the programs were set up interactively, so any flight conditions could be simulated. But again due to scaling and hardware limitations, there was actually only a limited range of variations possible. For accuracy and speed of setting up, a subroutine was written to calculate and set all values, automatically, according to what parameters were desired. The analog diagram is shown in Figure 4.1.

The software for the digital controller was written in Z-80 assembly language. The program was assembled on the DEC-10 and the code was downloaded directly into the specified RAM area of the microcomputer. The microcomputer itself was interfaced with the AD-5 through a set of A/D and D/A converters. There were 8 ports (of 8 bits each) of A/D and D/A converters used for inputing the desired states and outputing the control signals. The sampling period was set at 1 msec. This was done by writing an interrupt routine which used the internal clock of the system to interrupt the A/D ports every 1 msec to read the input data. To obtain the plots, the PDP-11 - DEC-10 system was used. The PDP-11 would sample and store the desired response values (states and controls) every 1 msec. These were later transferred to the DEC-10 so that the AG210 subroutines could be used to plot the data.

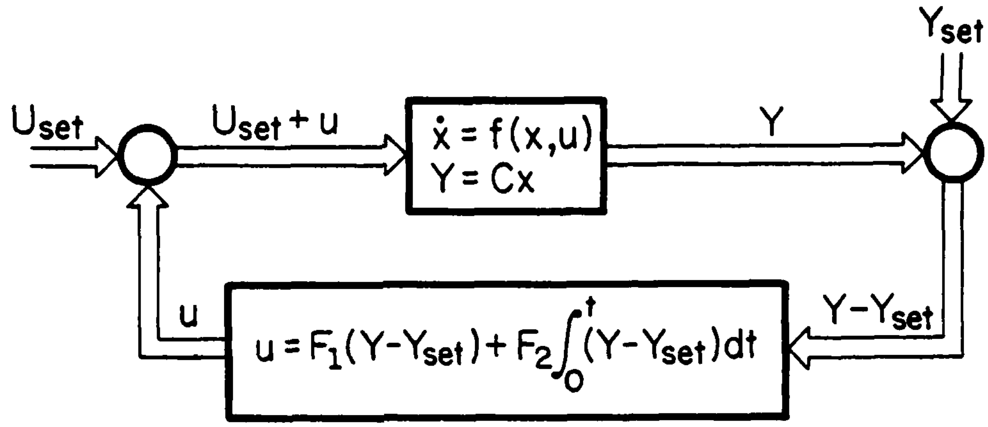


Figure 4.1a. Mathematical block diagram of the test system.

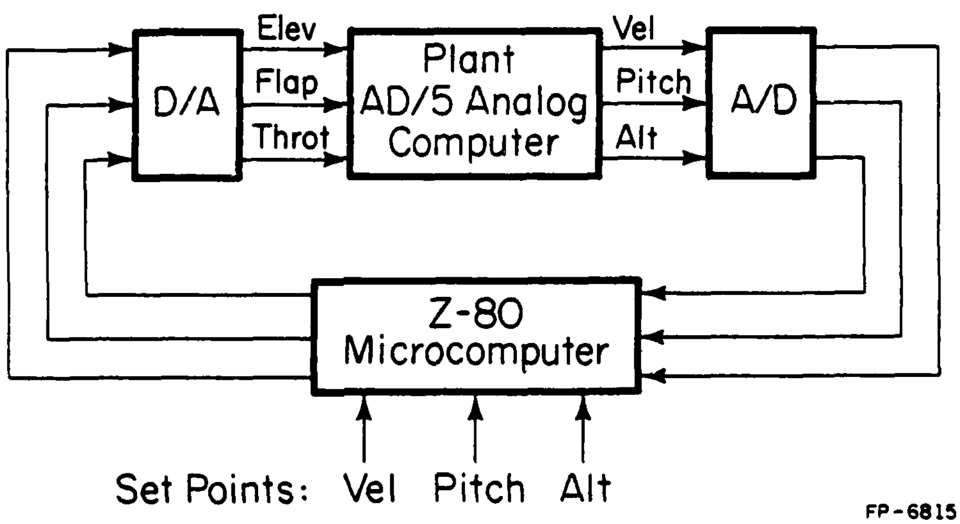


Figure 4.1b. Functional Block diagram of the test system.

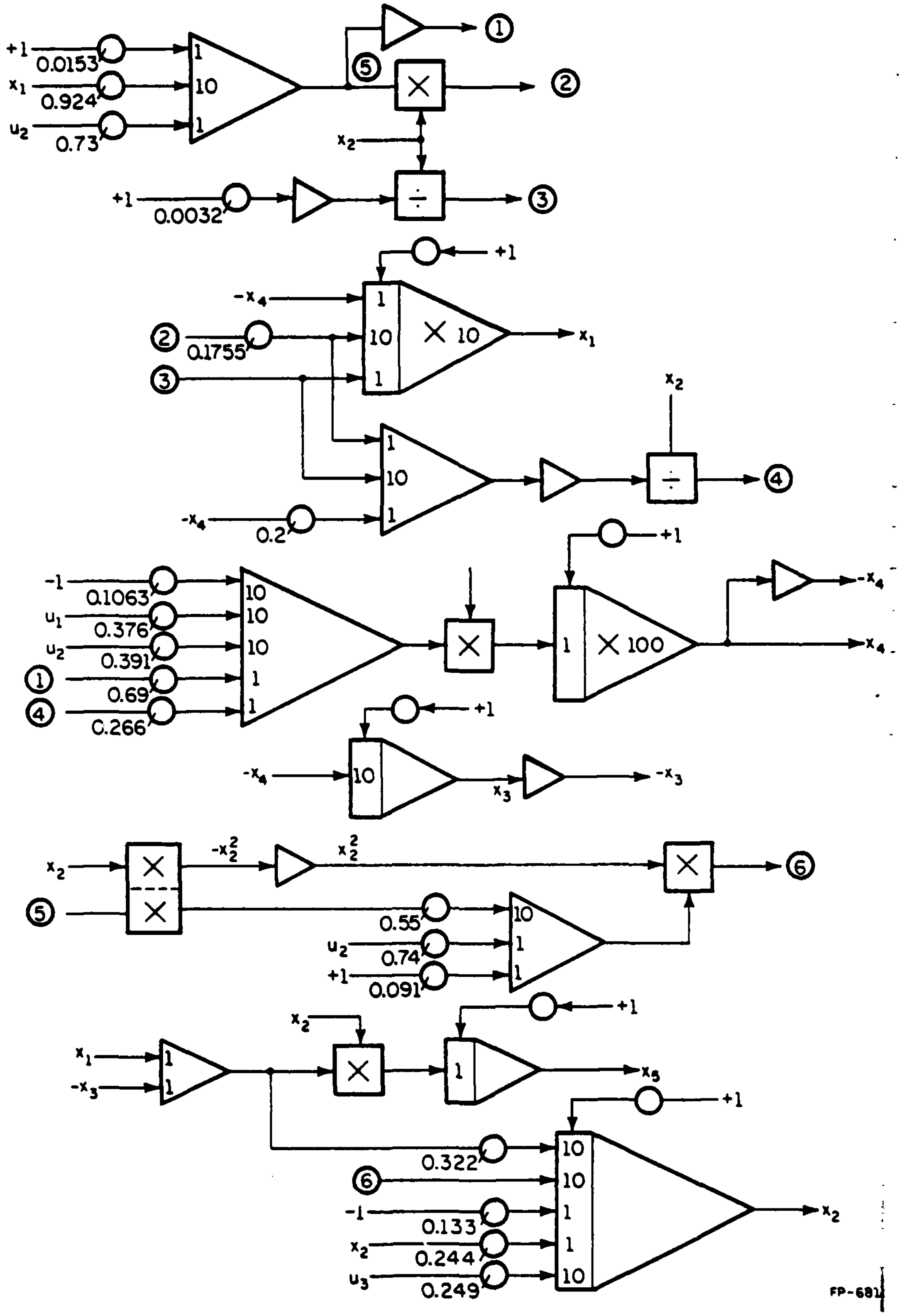


Figure 4.1c. Analog patch diagram of the aircraft model.

4.3. Results and Discussions

Four sets of curves are plotted for each of the two controllers. The first is just the proportional controller at the nominal operating point; the second is the PI-controller at the nominal operating point; and the third and fourth are PI-controllers at two different set points. These curves are shown in Figures 4.2-4.5.

In the discussions below, the controller designed via singular perturbation theory is referred to as controller A, while the controller designed via output regulator theory is referred to as controller B.

Figure 4.2 shows the system response with the proportional controller. A quick examination of the curves indicates that controller B performs much poorer than controller A. The state responses with controller B are more oscillatory and take a longer time to reach the steady state as compared to the state responses with controller A. Moreover, the stability region around the nominal flight trajectory is much smaller with controller B than with controller A. It was found that with controller B, the system would go unstable if the initial velocity lies outside 180-215 ft/sec, or if the initial pitch angle lies outside $\pm 0.6^{\circ}$, or if the initial altitude lies outside 1880-2100 ft. The corresponding ranges with controller A were found to be 150-250 ft/sec, $\pm 1.4^{\circ}$, 1500-2500 ft. In terms of the control effort, all the three controls fluctuate more rapidly with controller B than with controller A. The poorer performance of controller as compared to controller A was to be expected because of the ill-conditioning of the output regulator design in this case (as noted in the last chapter).

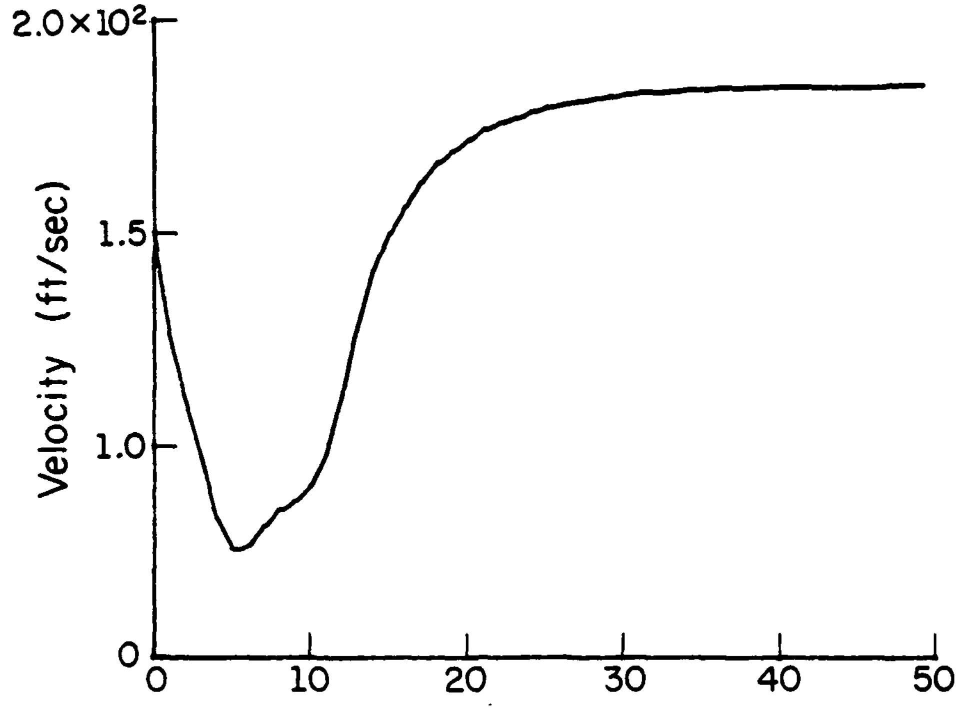


Figure 4.2.1a. Singular perturbation design.

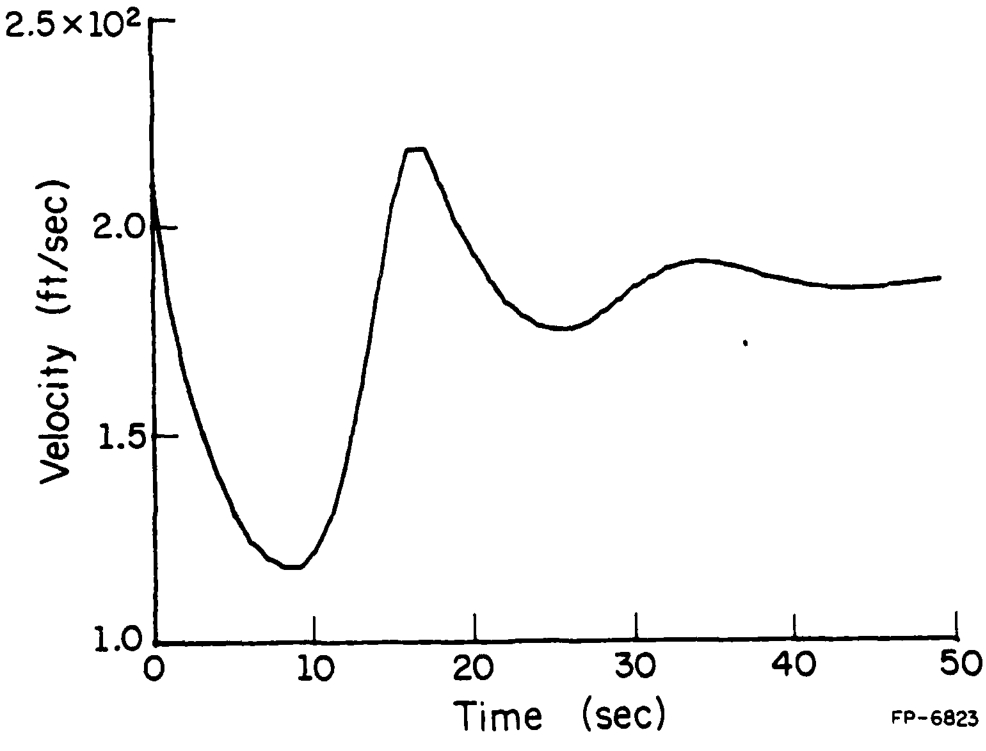


Figure 4.2.1b. Output regulator design.

Figure 4.2. Proportional controller.

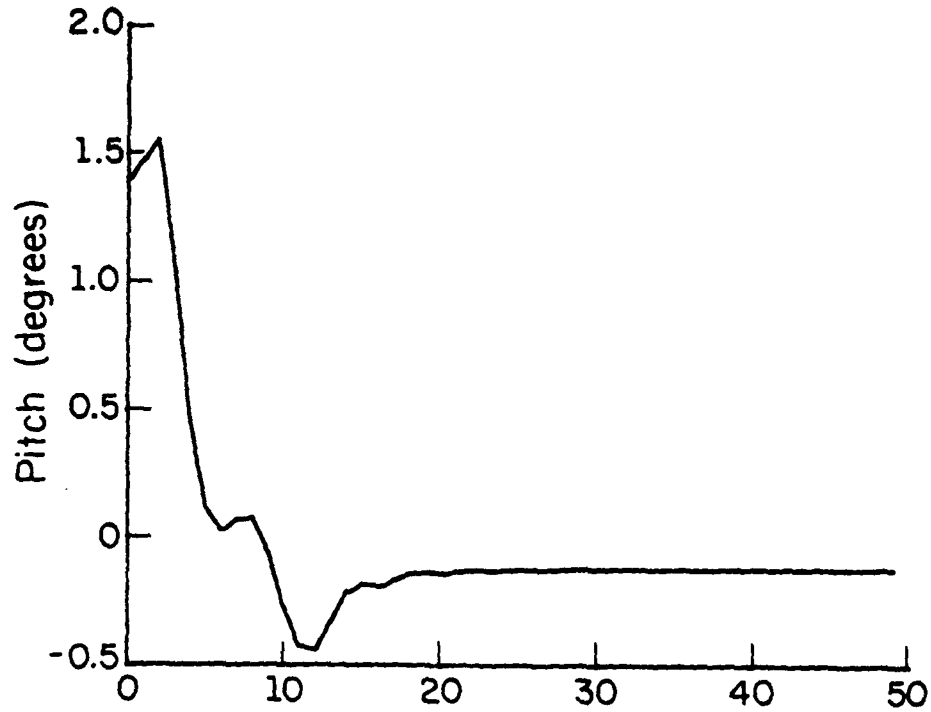


Figure 4.2.2a. Singular perturbation design.

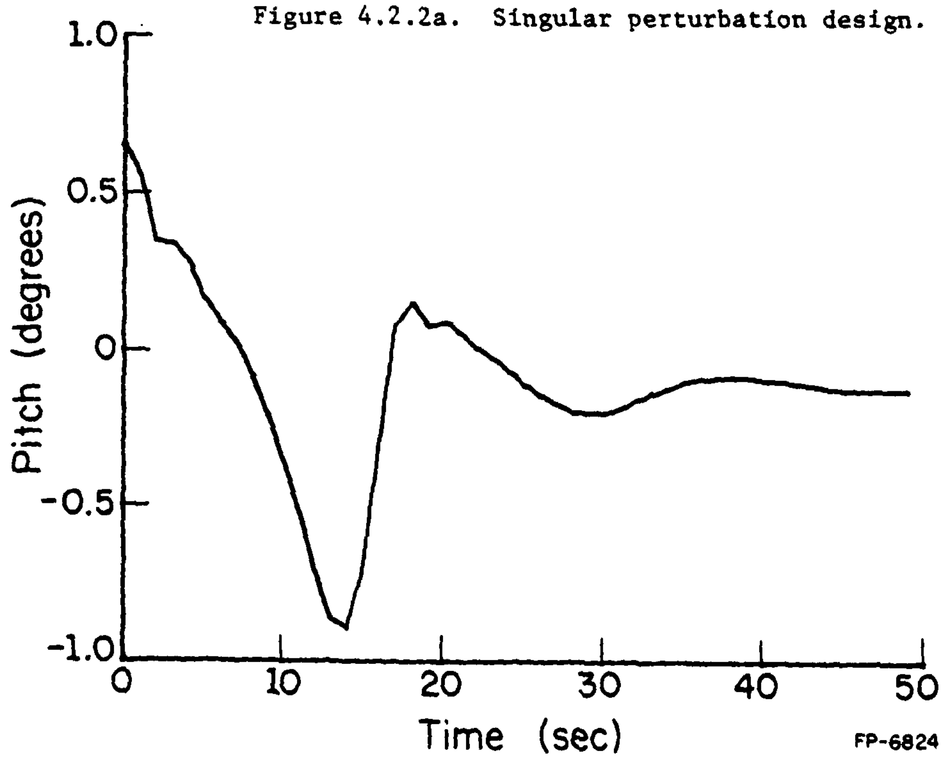


Figure 4.2.2b. Output regulator design.

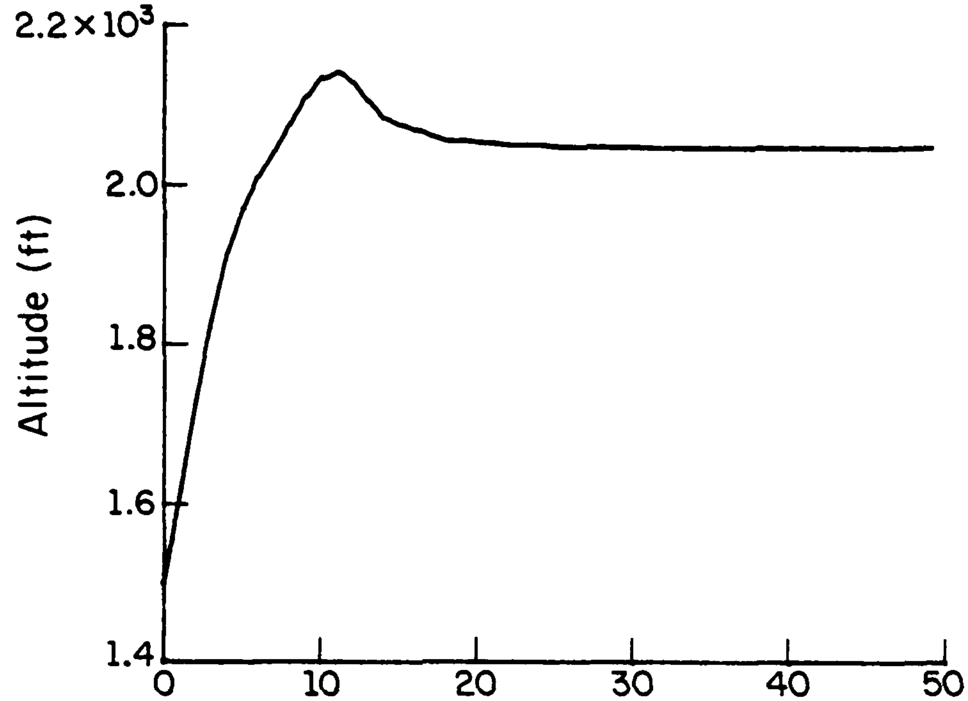


Figure 4.2.3a. Singular perturbation design.

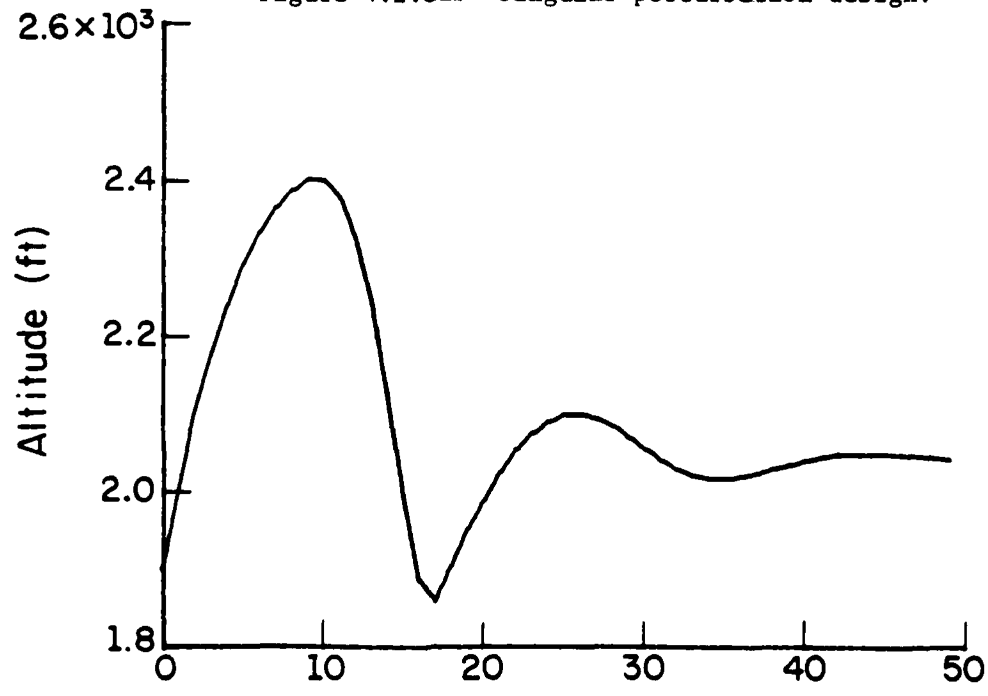


Figure 4.2.3b. Output regulator design. FP-6825

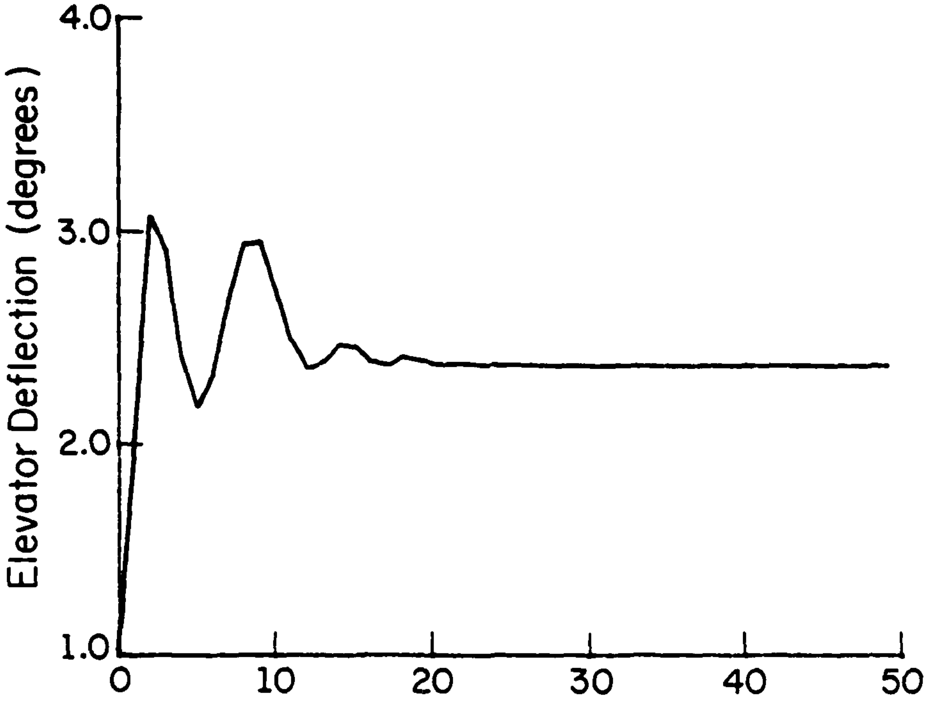


Figure 4.2.4a. Singular perturbation design.

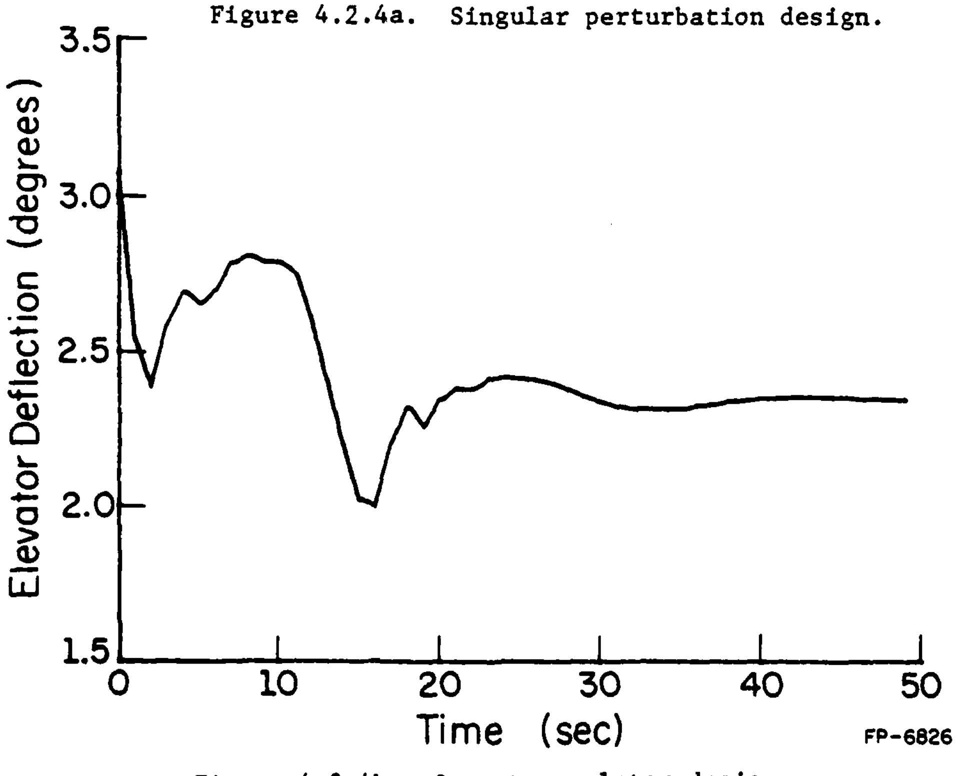


Figure 4.2.4b. Output regulator design.

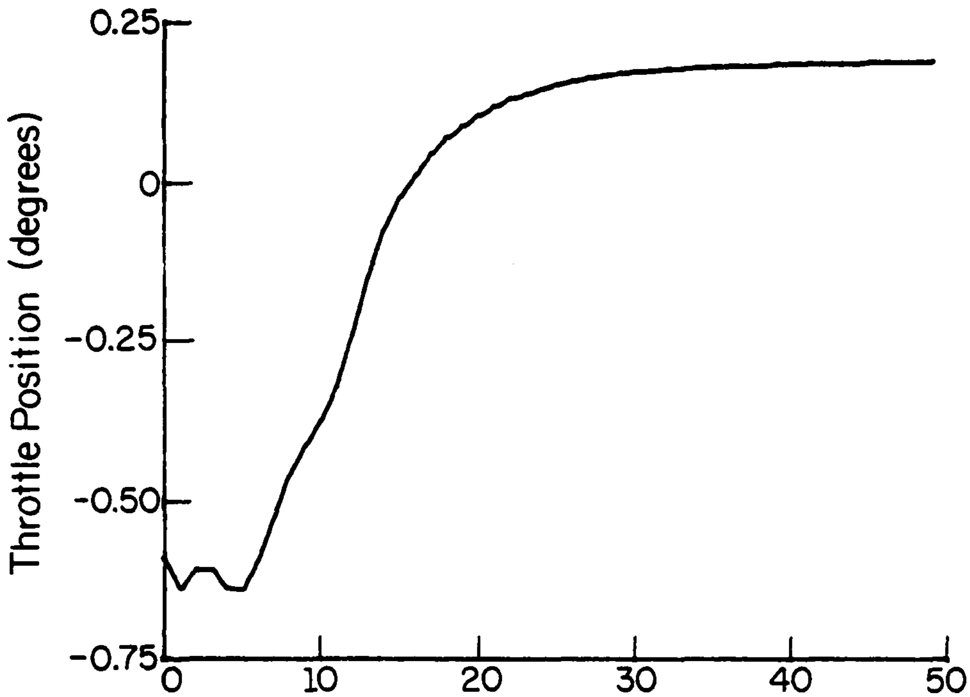


Figure 4.2.5a. Singular perturbation design.

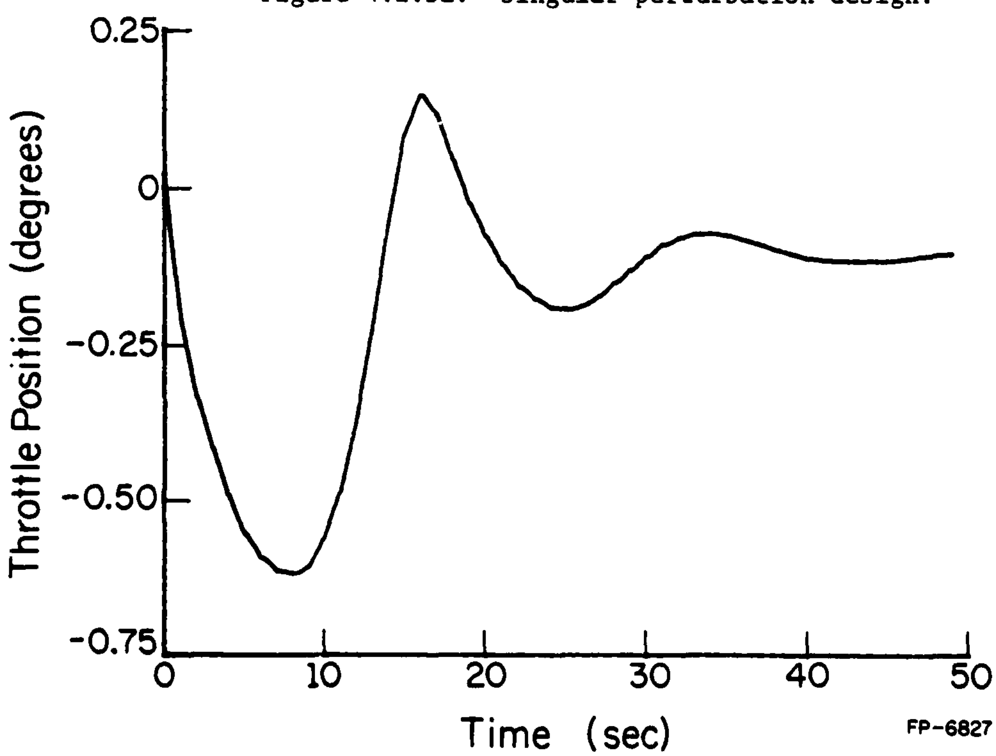


Figure 4.2.5b. Output regulator design

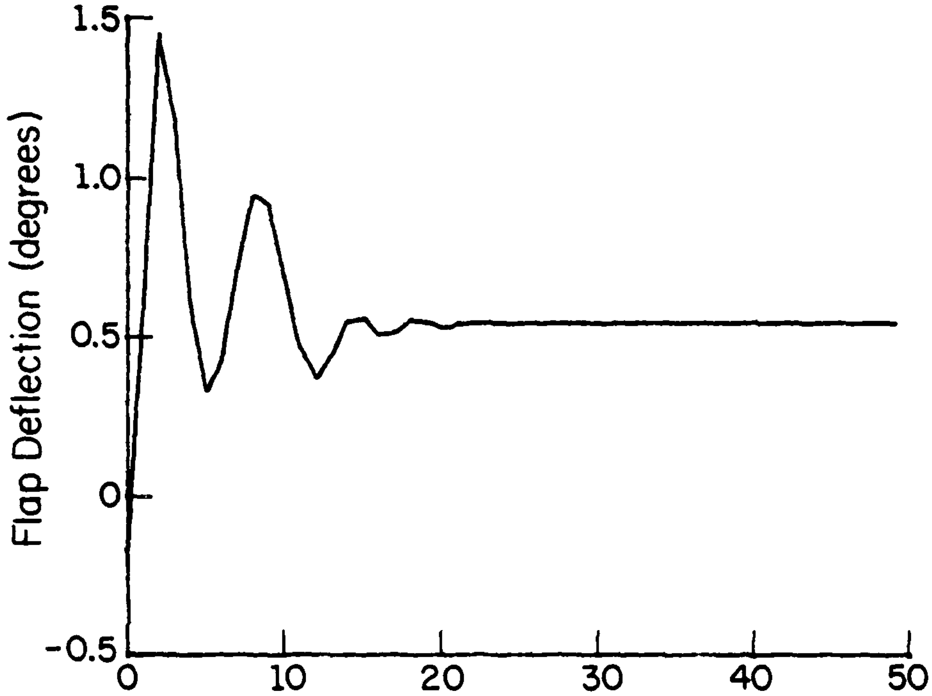


Figure 4.2.6a. Singular perturbation design.

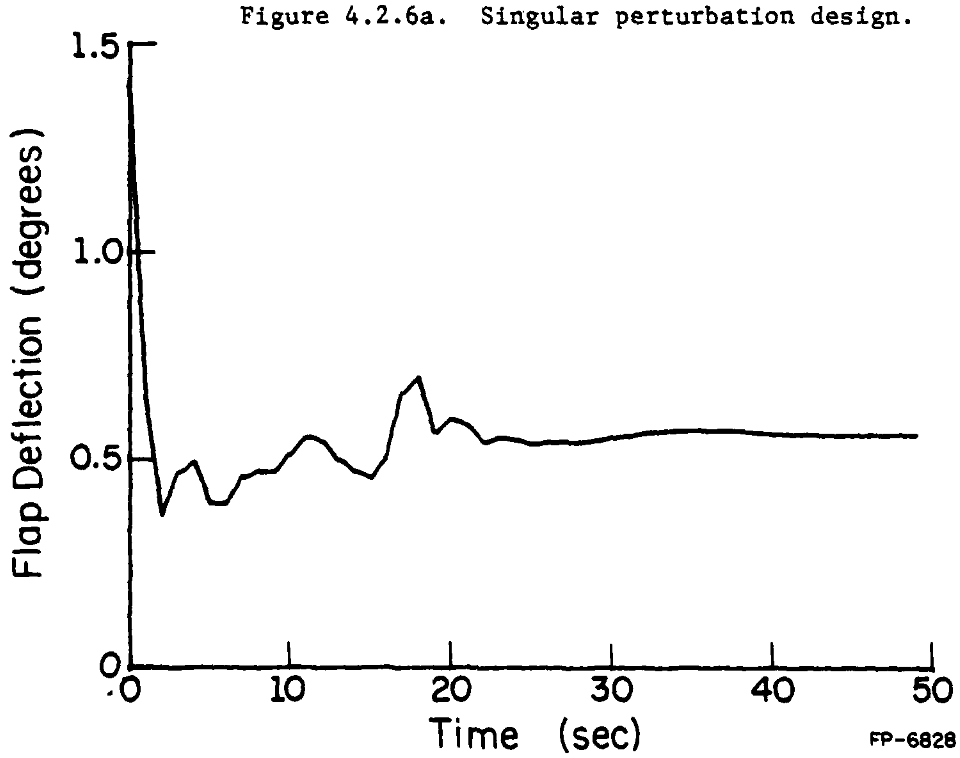


Figure 4.2.6b. Output regulator design.

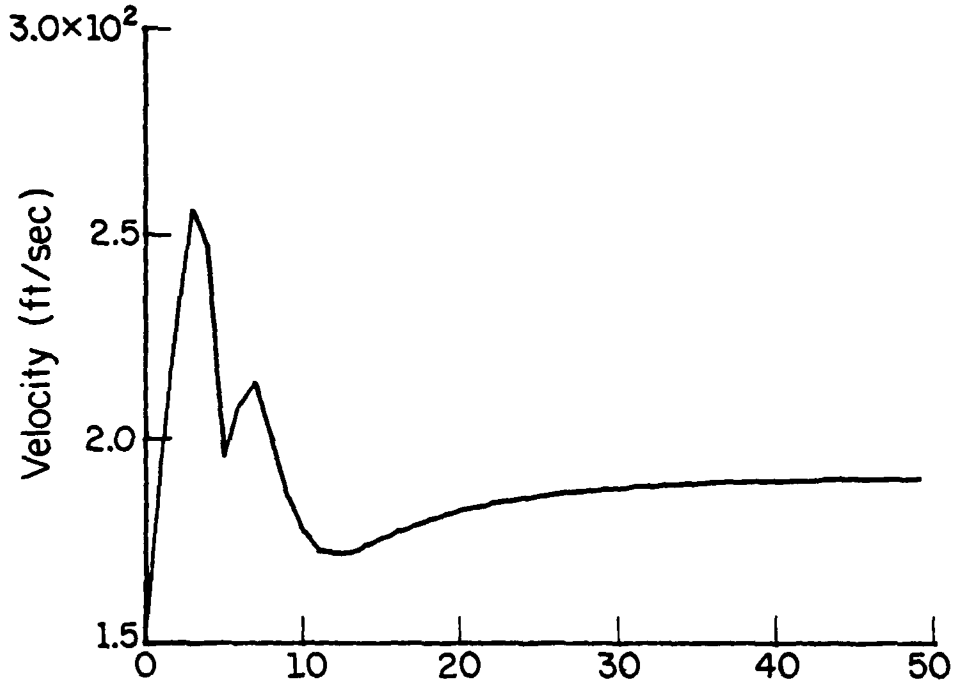


Figure 4.3.1a. Singular perturbation design.

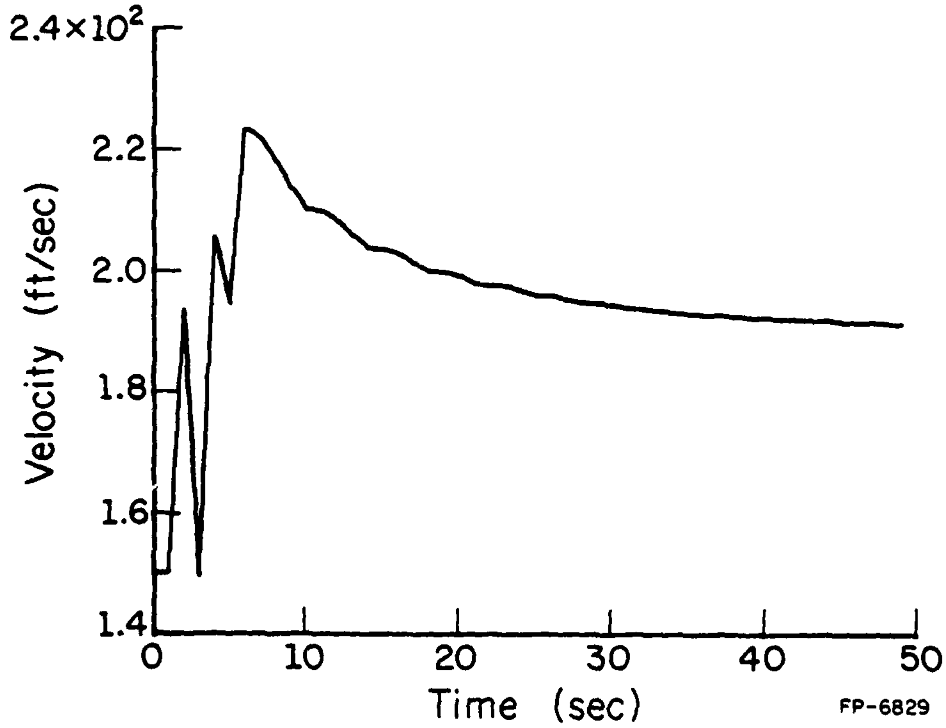


Figure 4.3.1b. Output regulator design.

Figure 4.3. PI-controller at nominal set point.

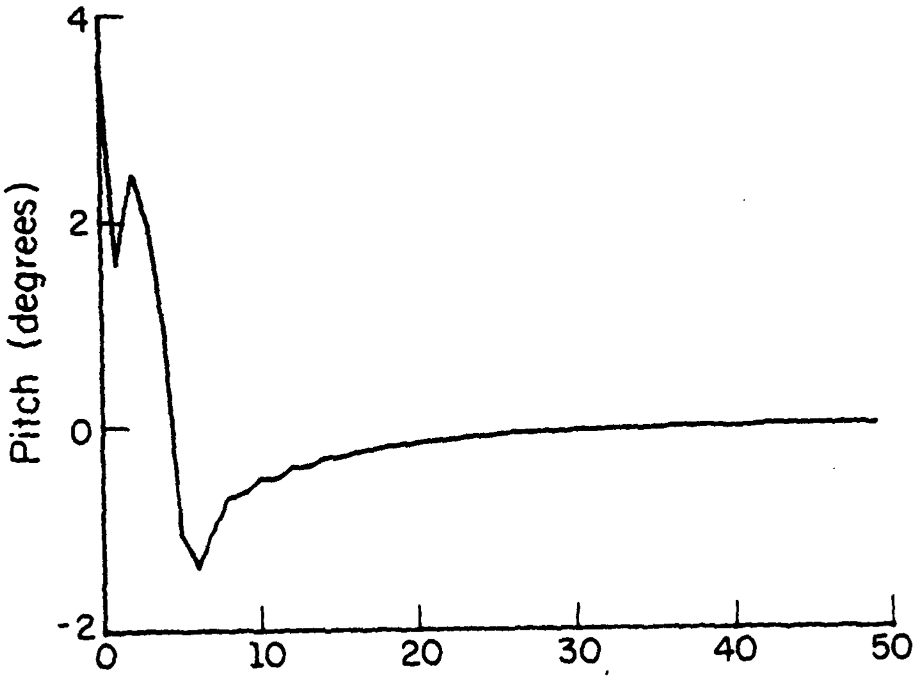


Figure 4.3.2a. Singular perturbation design.

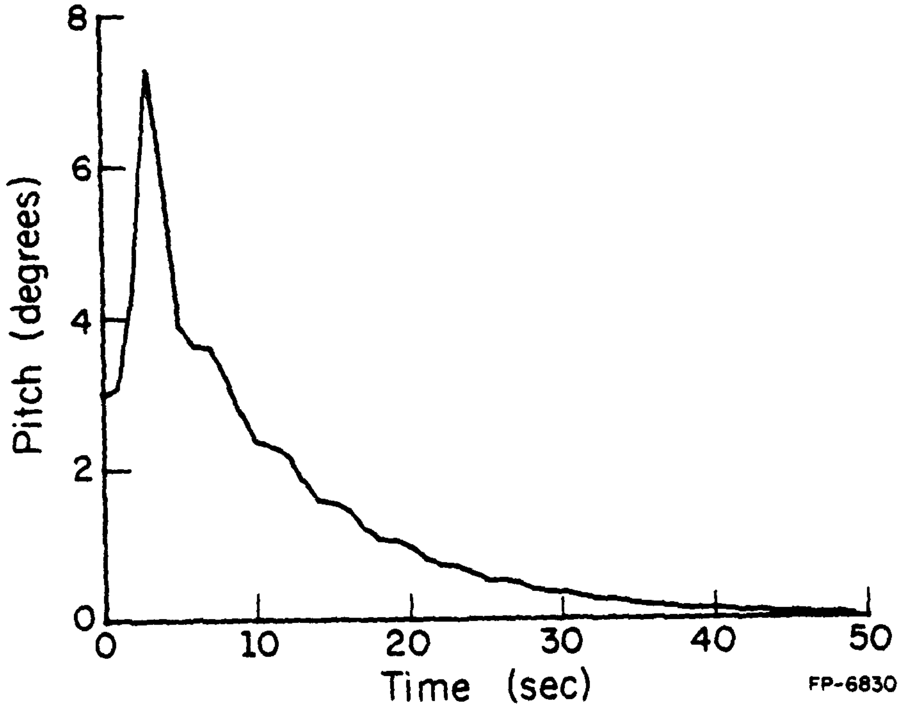


Figure 4.3.2b. Output regulator design.

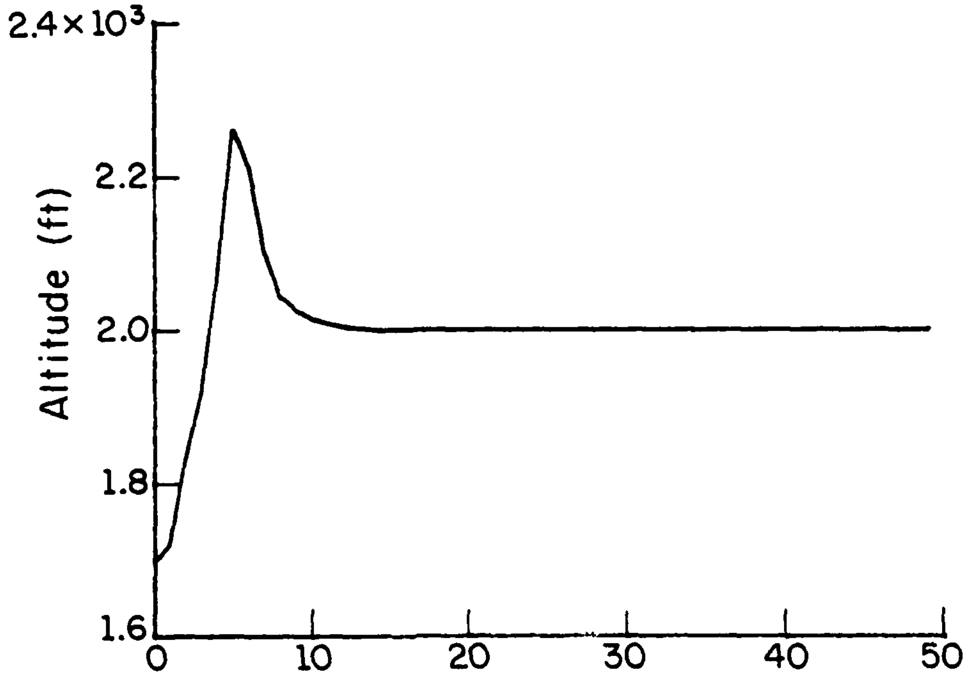


Figure 4.3.3a. Singular perturbation design.

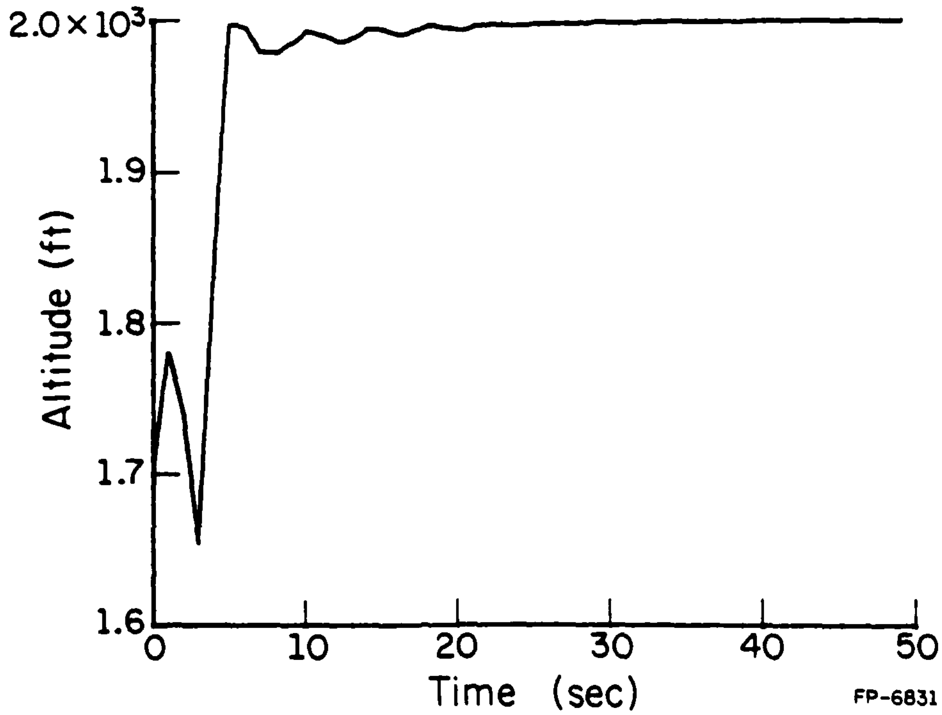


Figure 4.3.3b. Output regulator design.

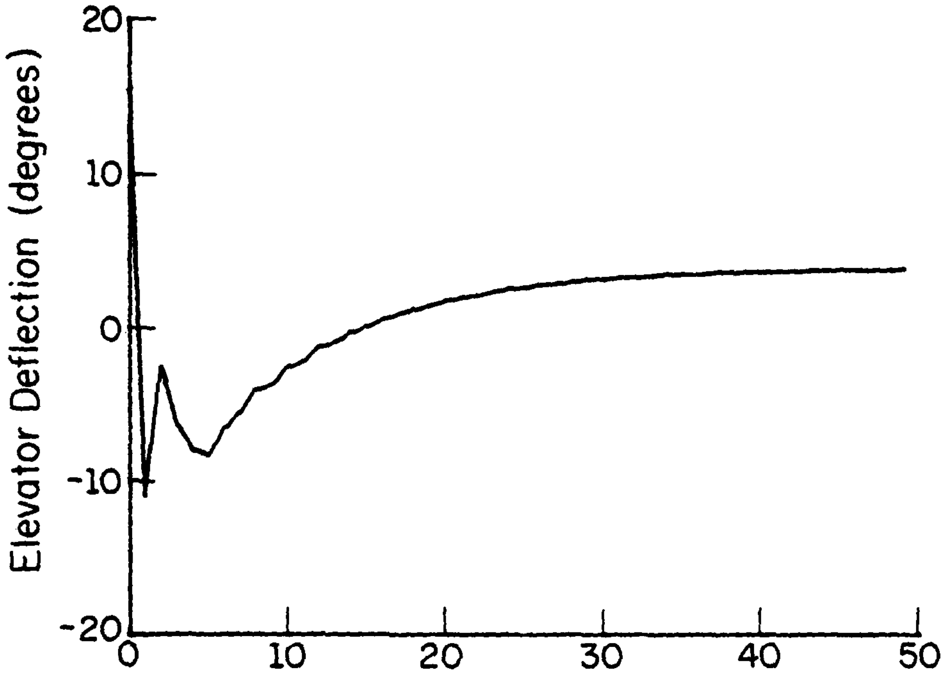


Figure 4.3.4a. Singular perturbation design.

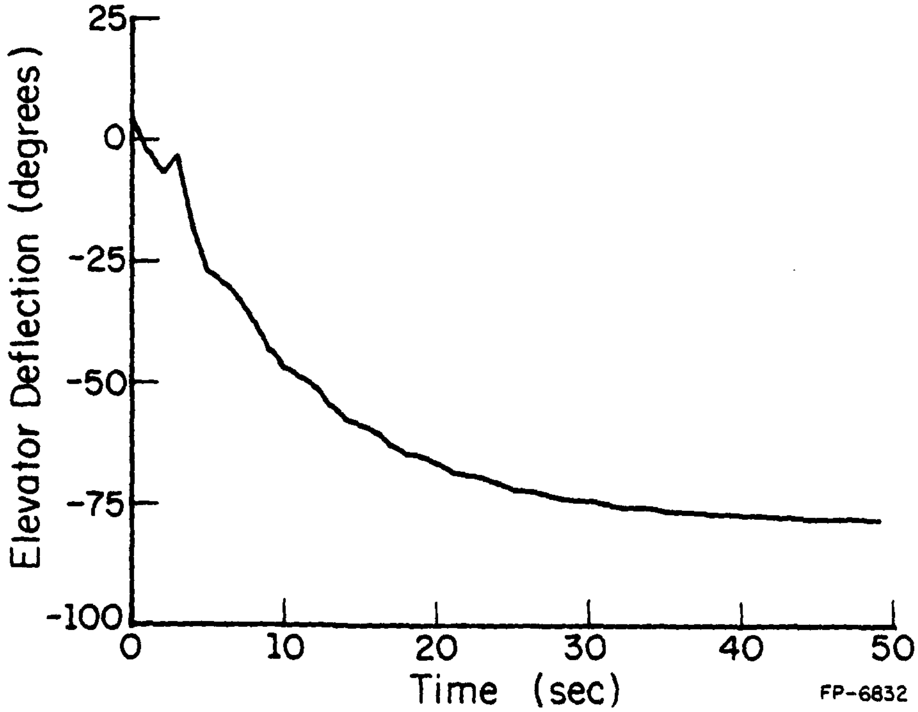


Figure 4.3.4b. Output regulator design.

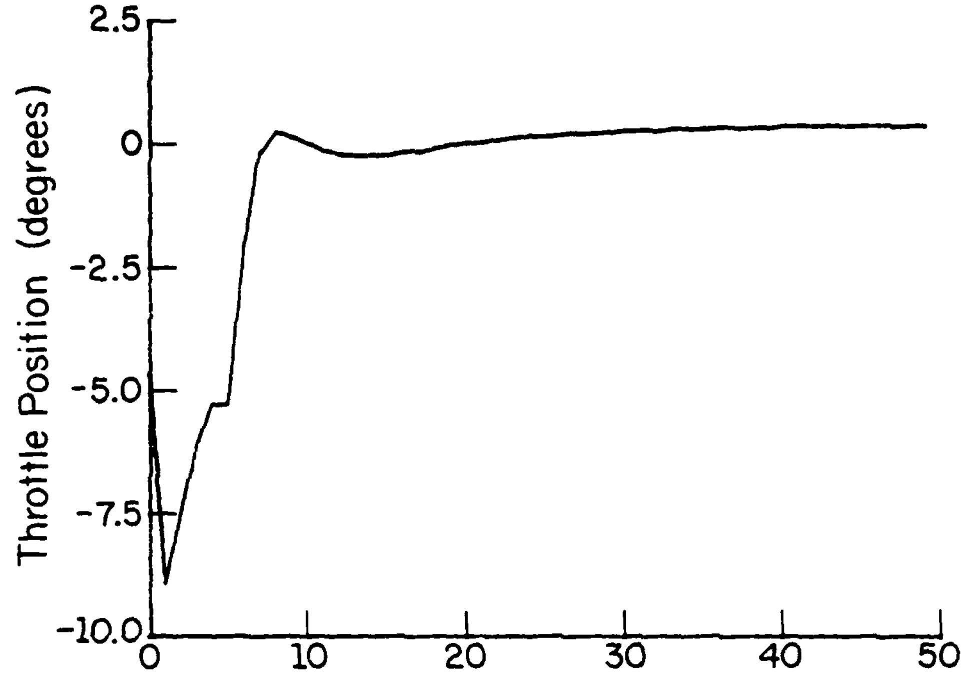


Figure 4.3.5a. Singular perturbation design.

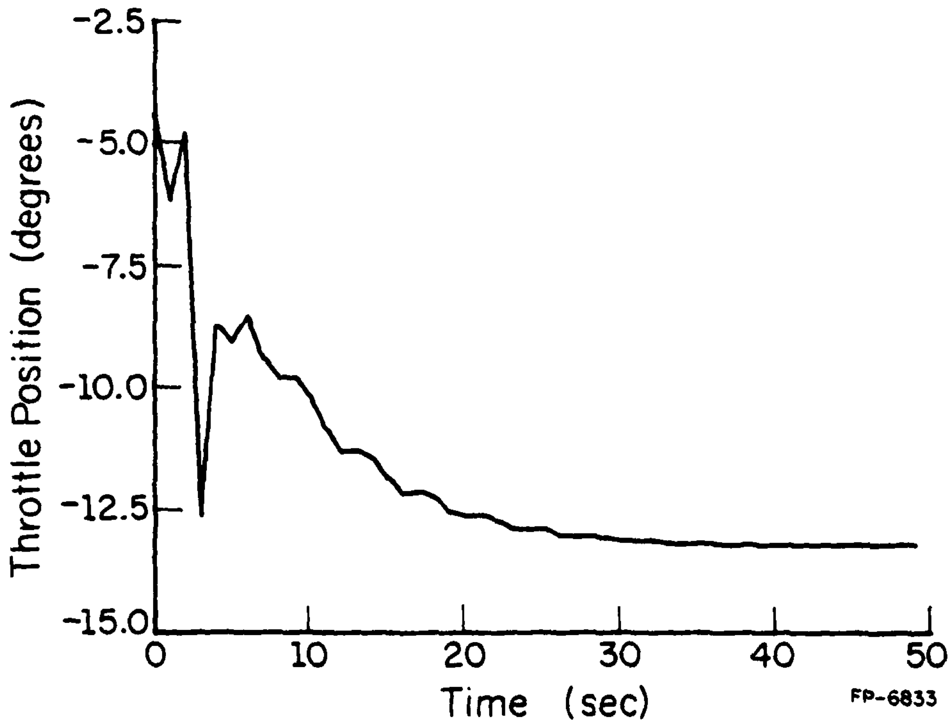


Figure 4.3.5b. Output regulator design.

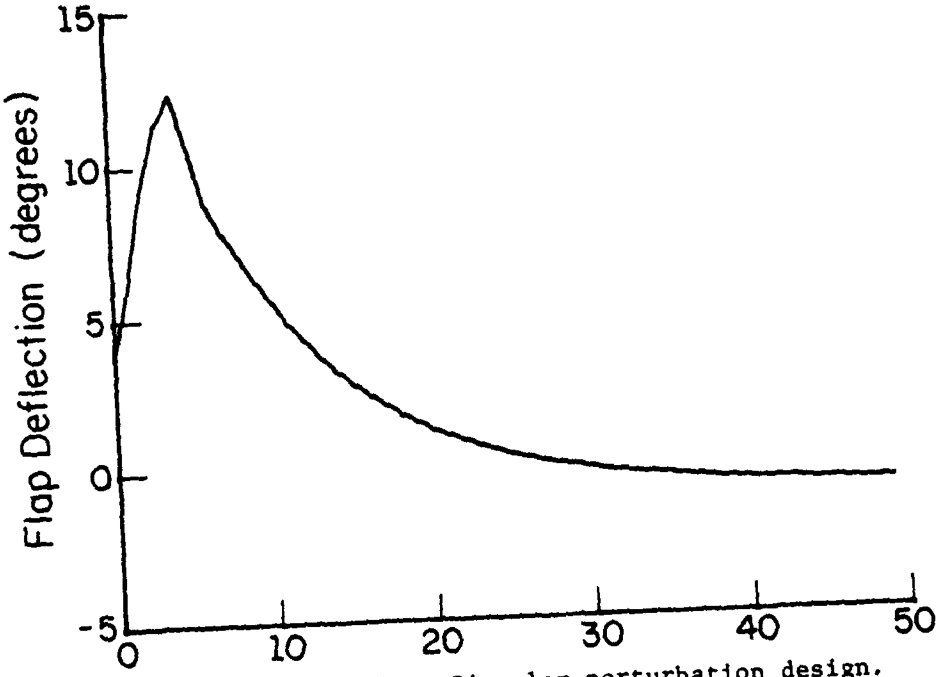


Figure 4.3.6a. Singular perturbation design.

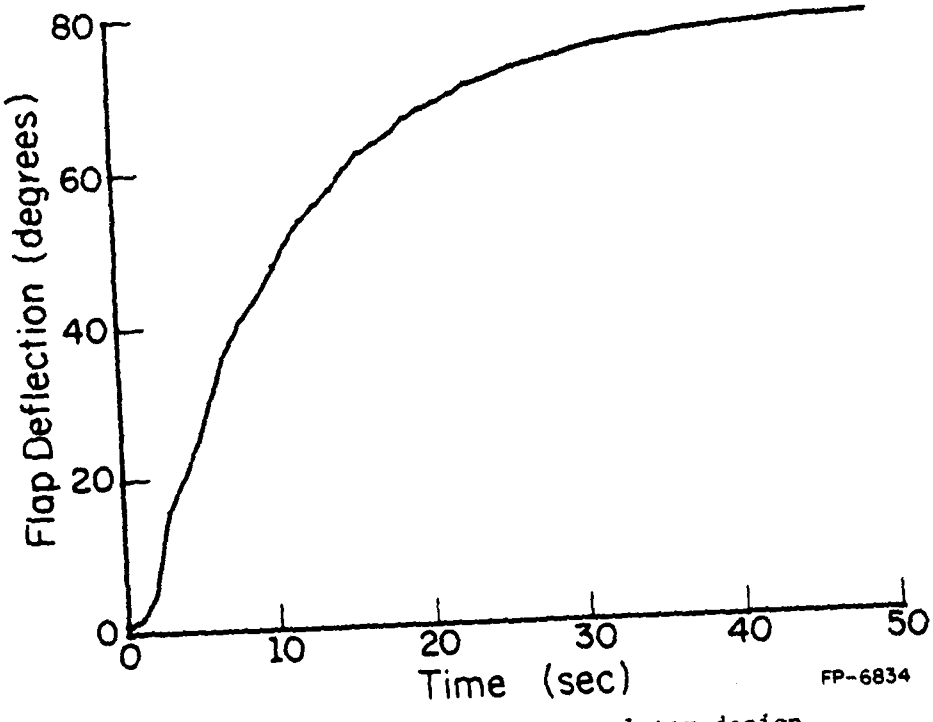


Figure 4.3.6b. Output regulator design.

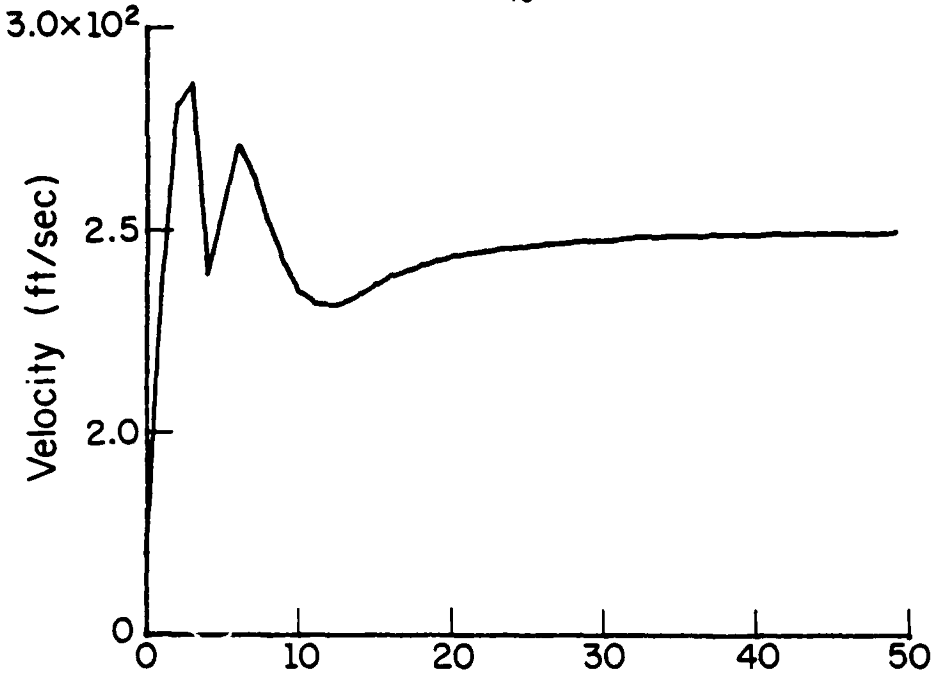


Figure 4.4.1a. Singular perturbation design.

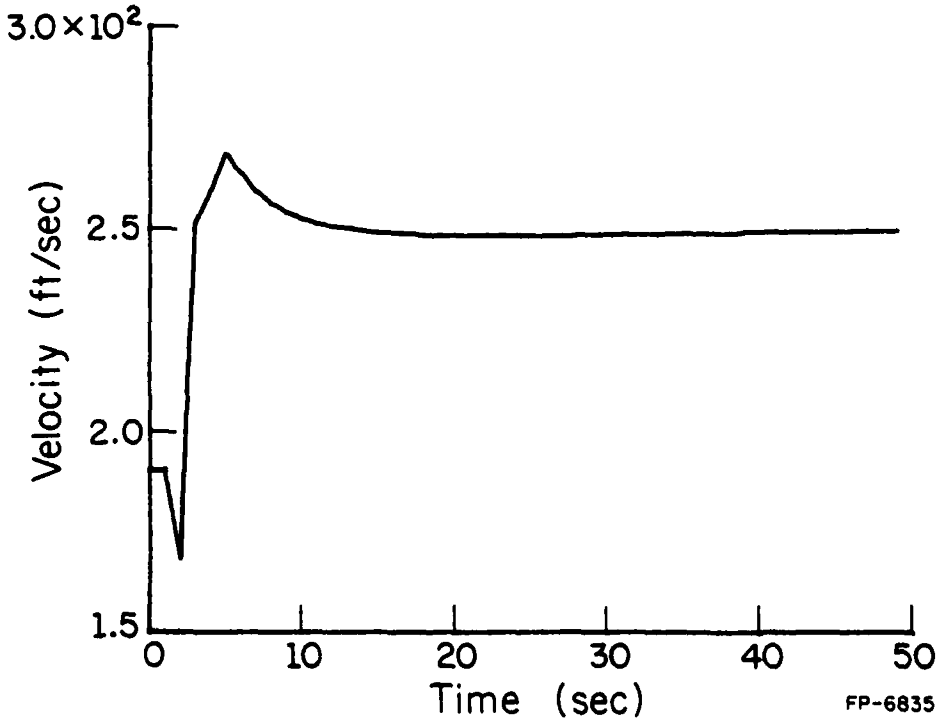


Figure 4.4.1b. Output regulator design.

Figure 4.4. PI-controller. Set point: Velocity = 250 ft/sec
Pitch = 0.5°
Altitude = 2300 ft

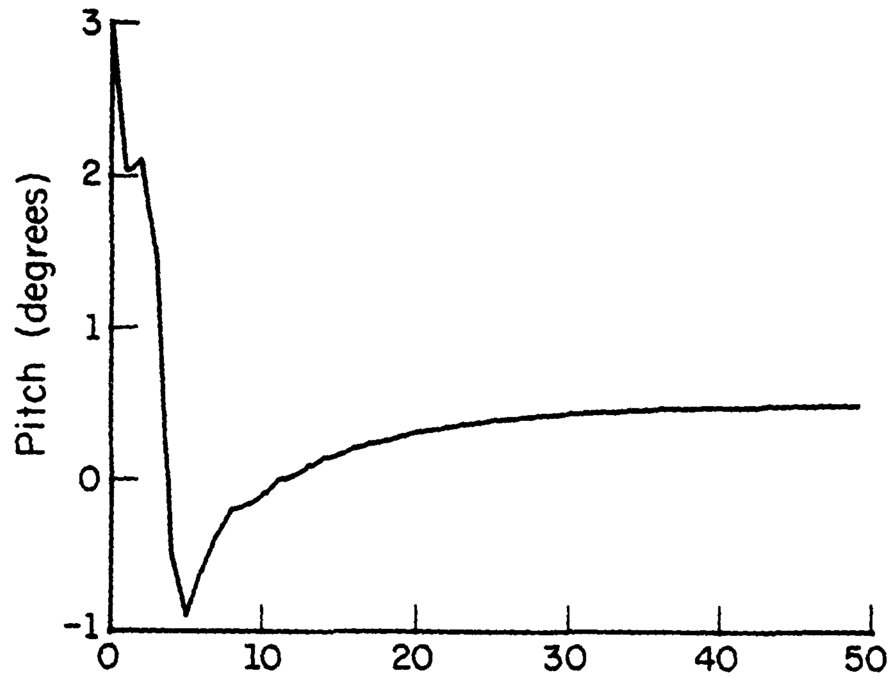
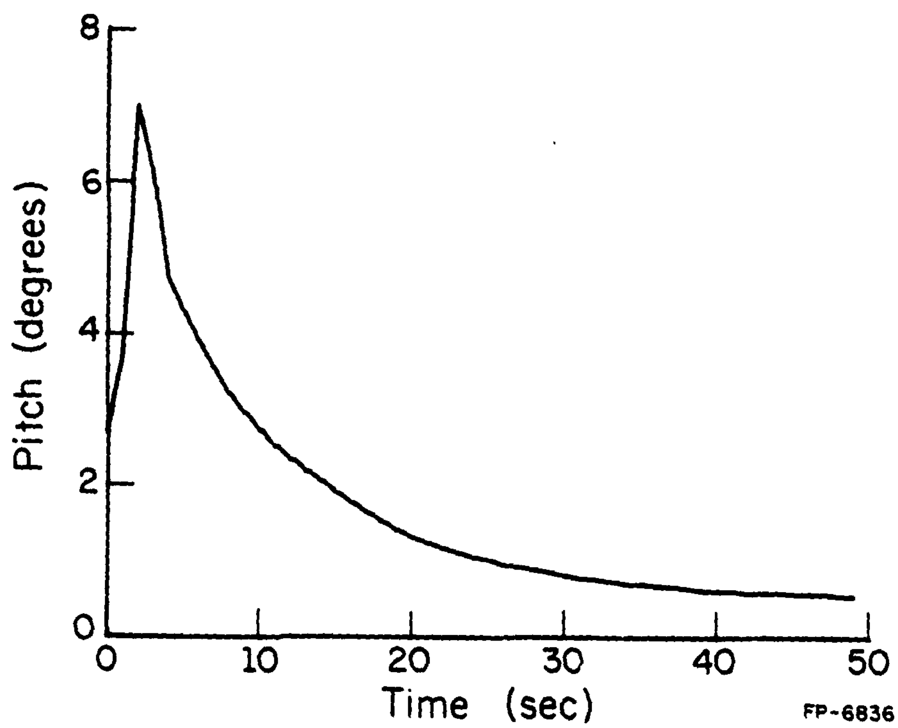


Figure 4.4.2a. Singular perturbation design.



FP-6836

Figure 4.4.2b. Output regulator design.

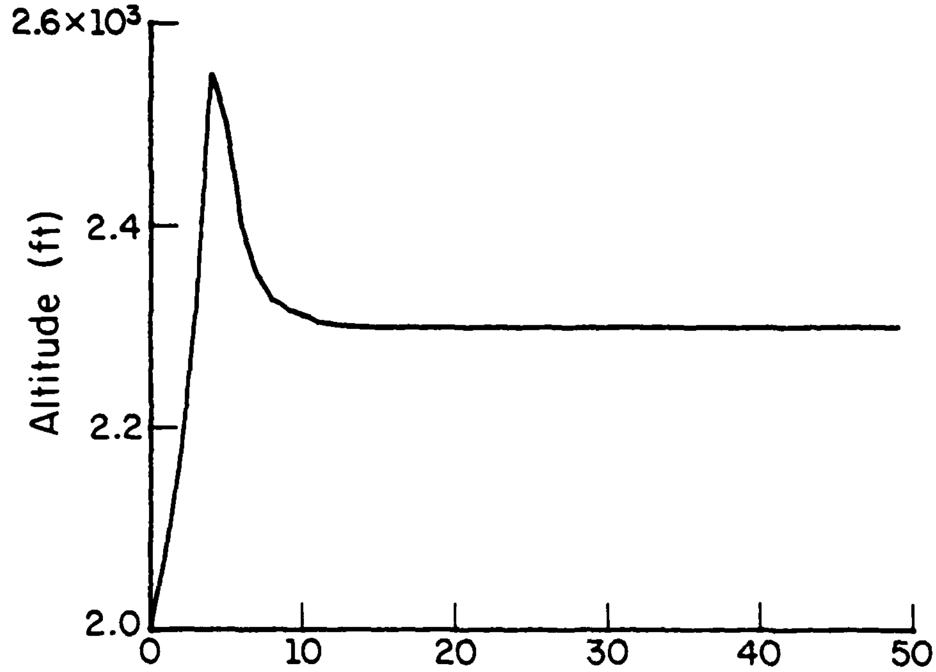


Figure 4.4.3a. Singular perturbation design.

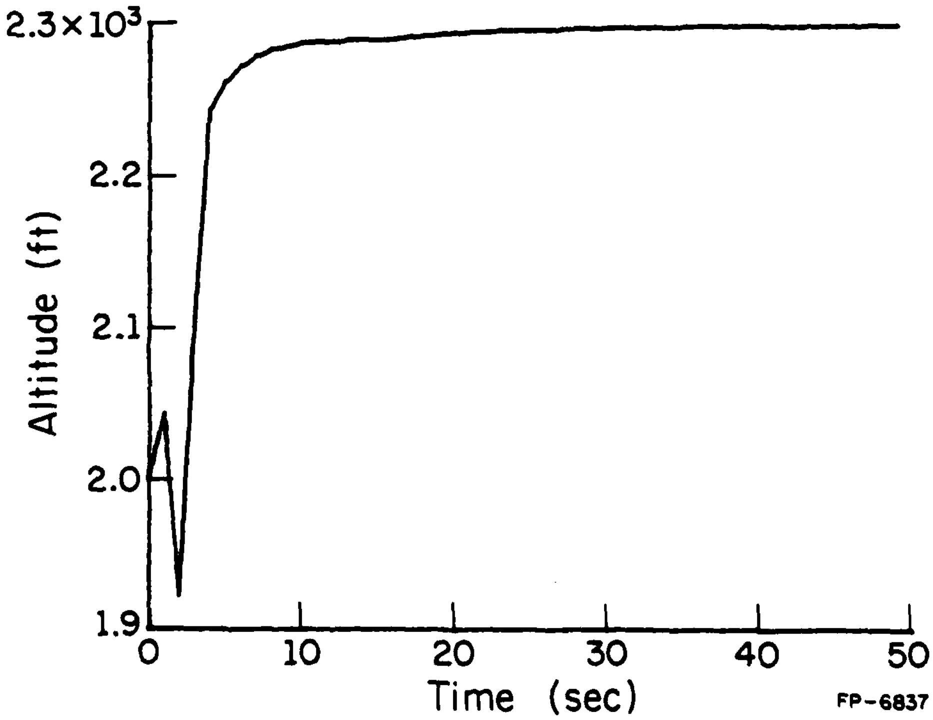


Figure 4.4.3b. Output regulator design.

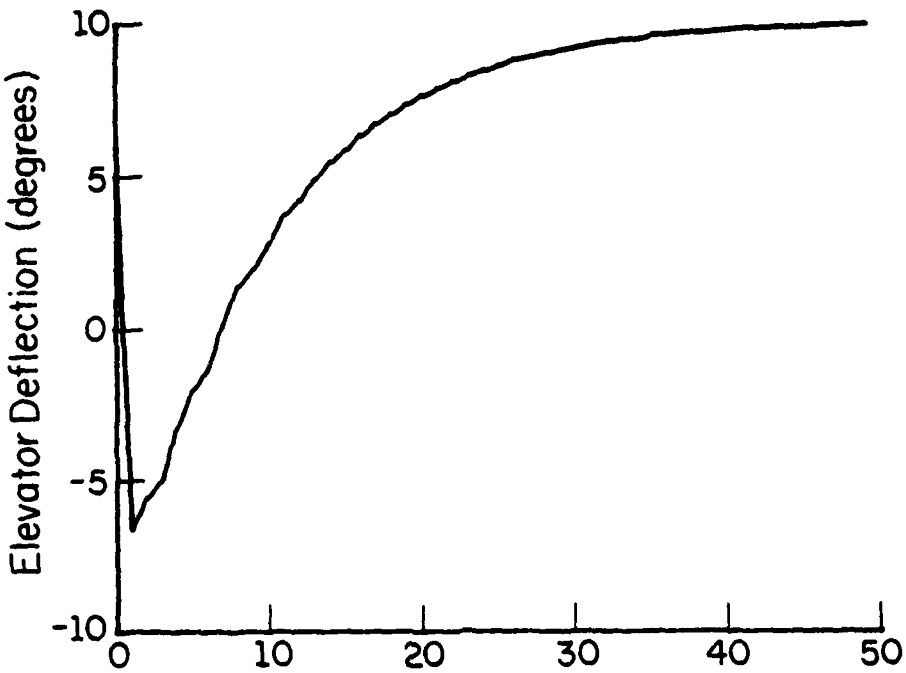


Figure 4.4.4a. Singular perturbation design.

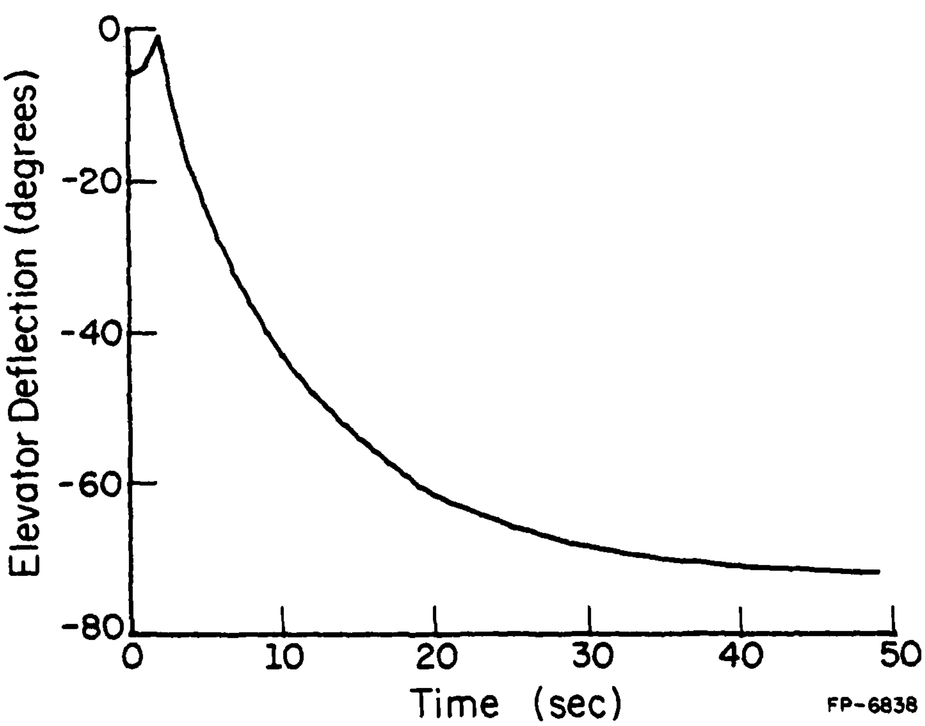


Figure 4.4.4b. Output regulator design.

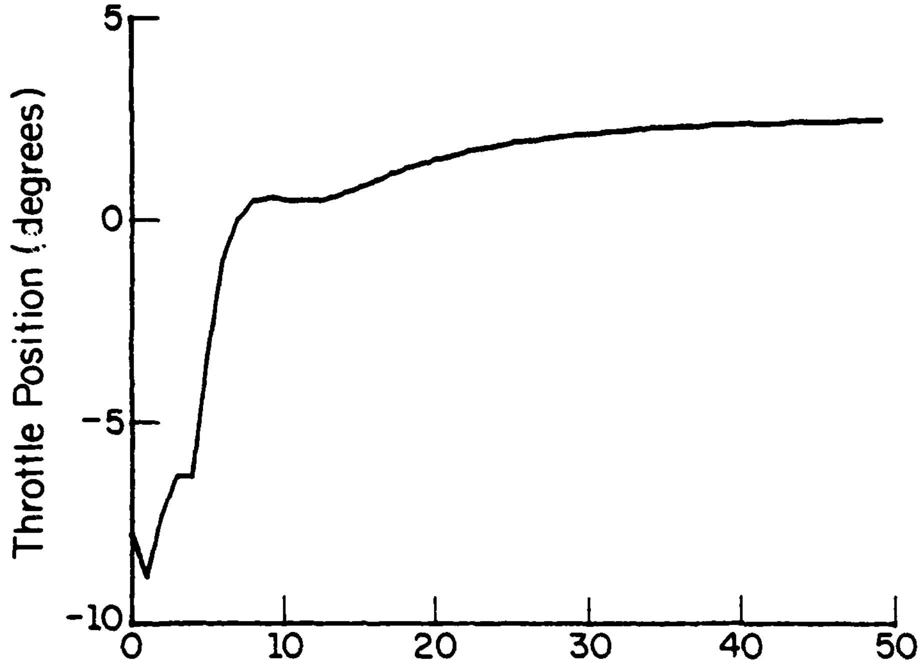


Figure 4.4.5a. Singular perturbation design.

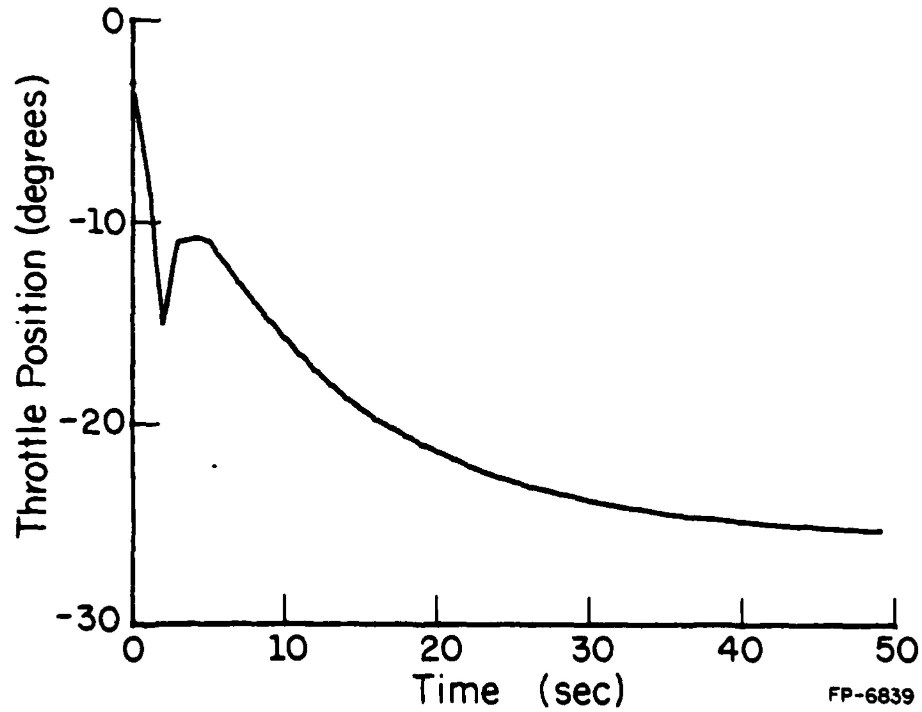


Figure 4.4.5b. Output regulator design.

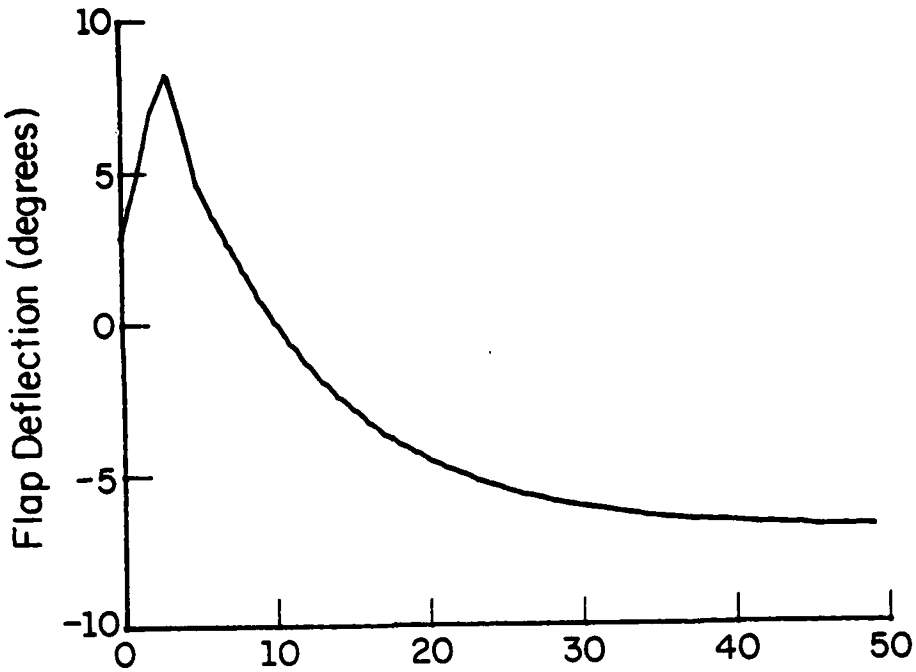


Figure 4.4.6a. Singular perturbation design.

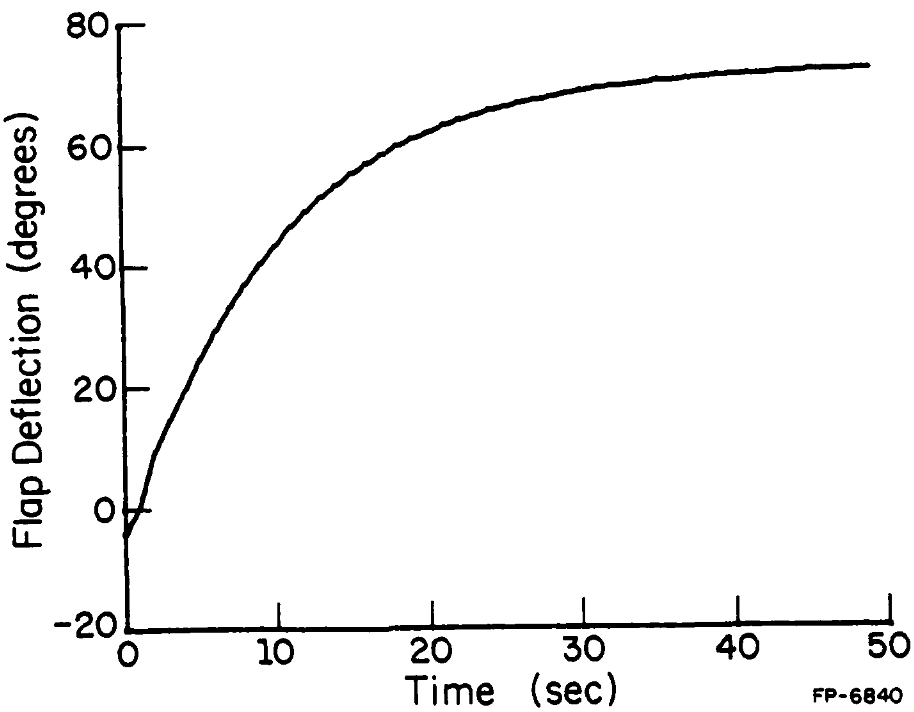


Figure 4.4.6b. Output regulator design.

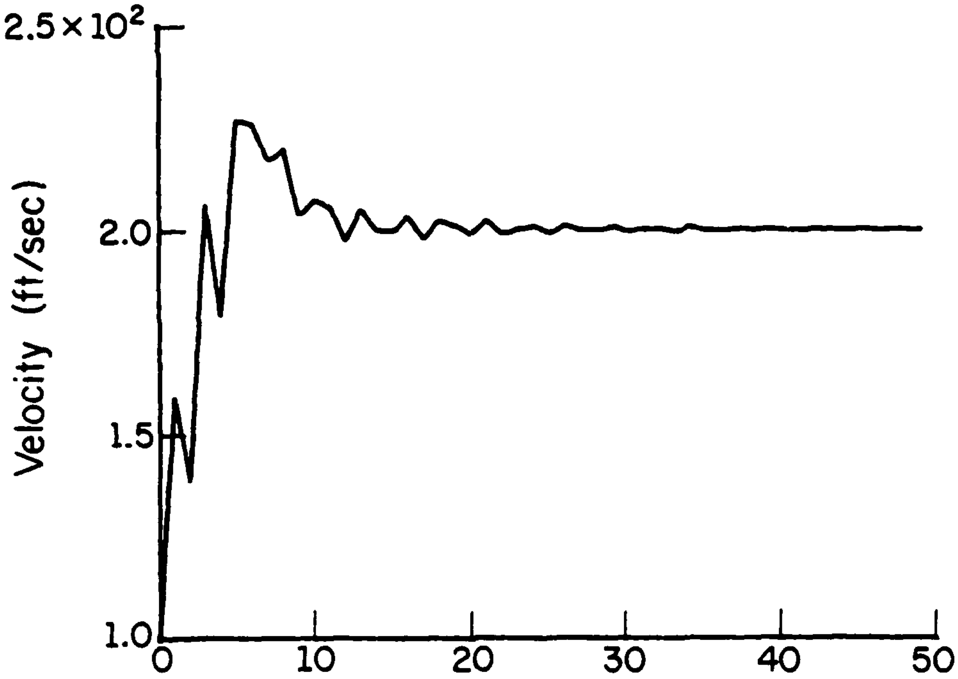


Figure 4.5.1a. Singular perturbation design.

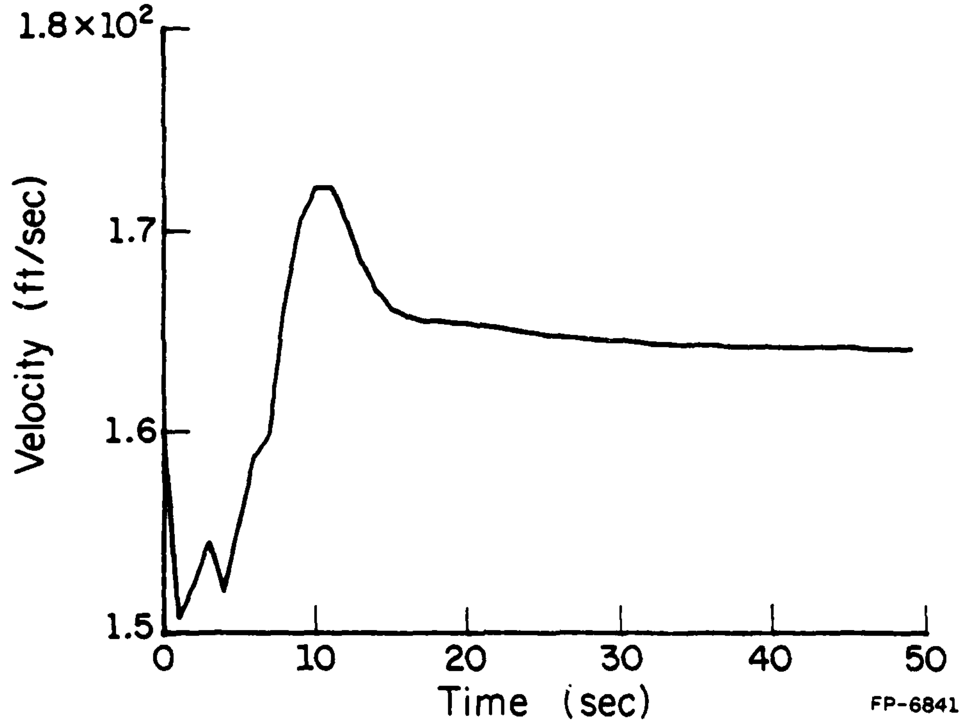


Figure 4.5.1b. Output regulator design.

Figure 4.5. PI-controller. Set point: Velocity = 170 ft/sec
Pitch = 0°
Altitude = 1800 ft

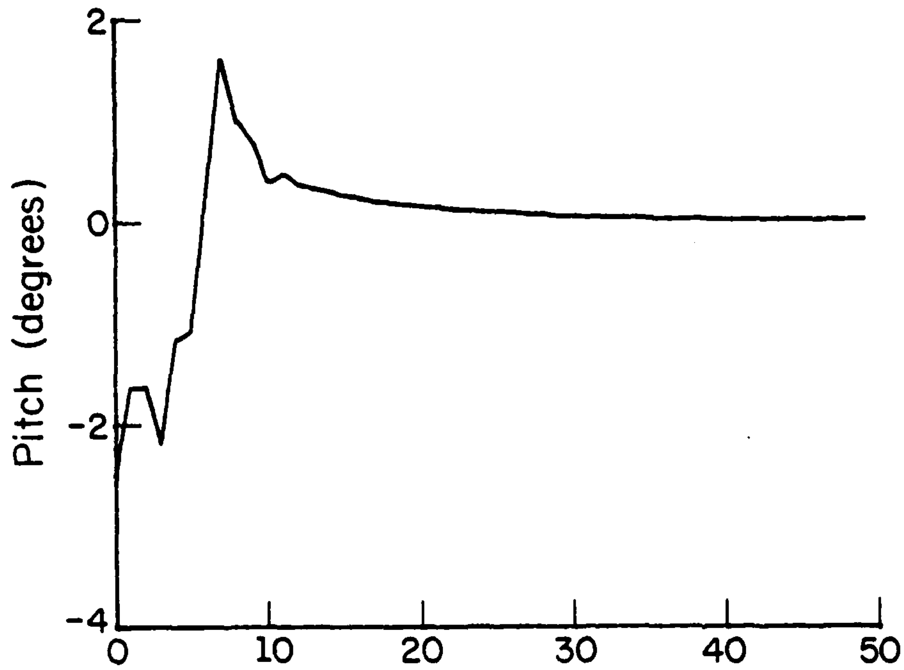


Figure 4.5.2a. Singular perturbation design.

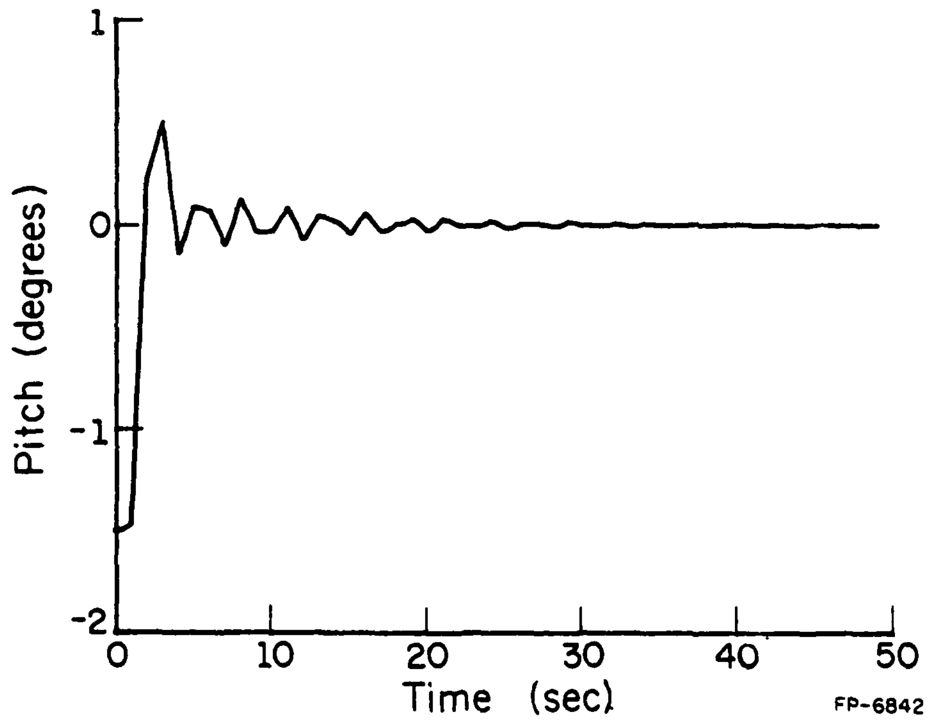


Figure 4.5.2b. Output regulator design.

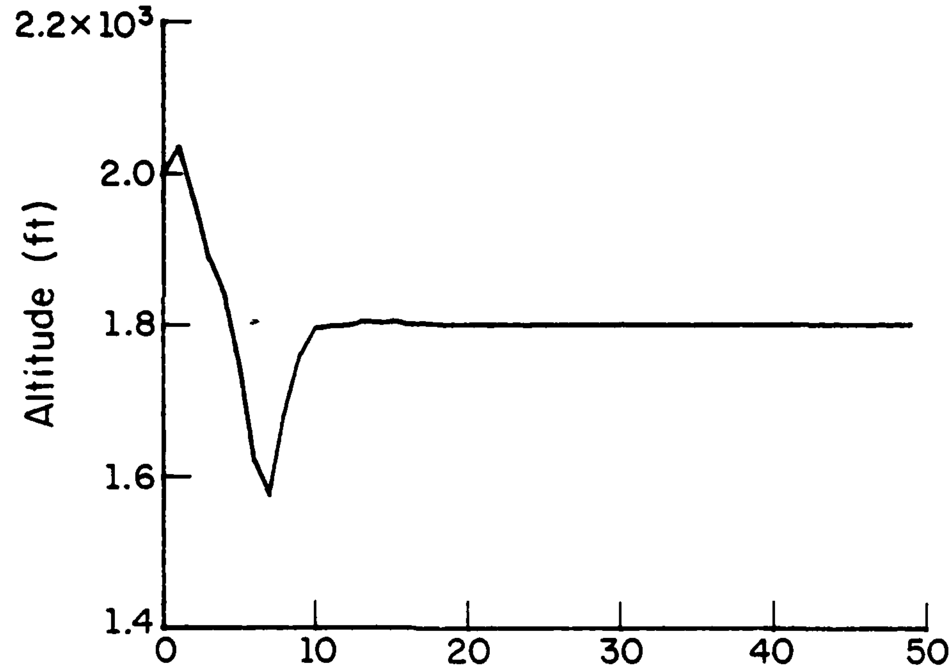
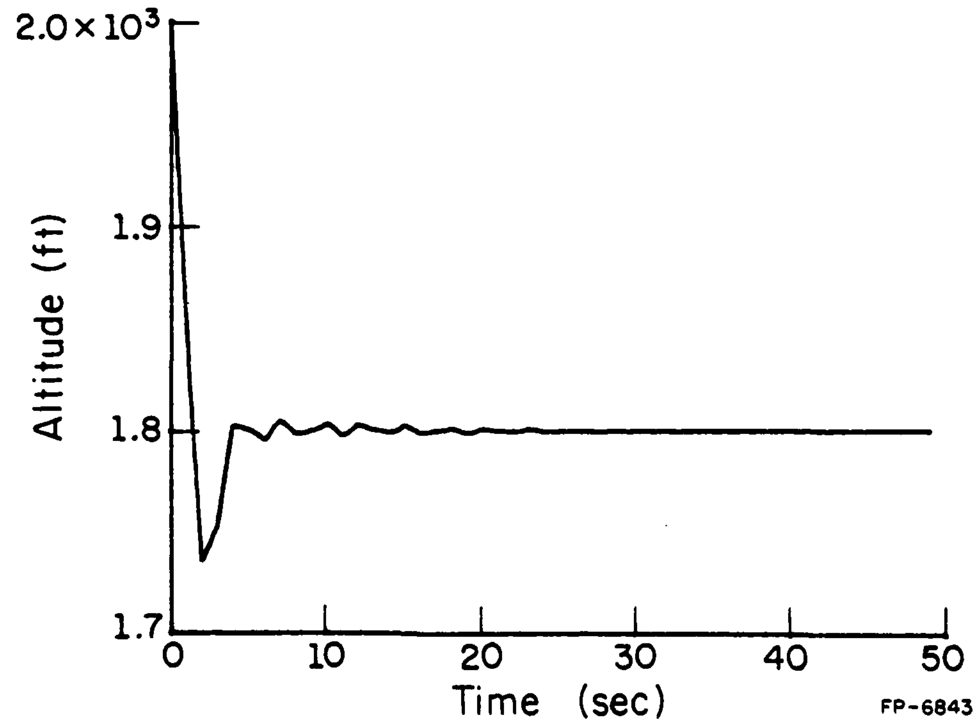


Figure 4.5.3a. Singular perturbation design.



FP-6843

Figure 4.5.3b. Output regulator design.

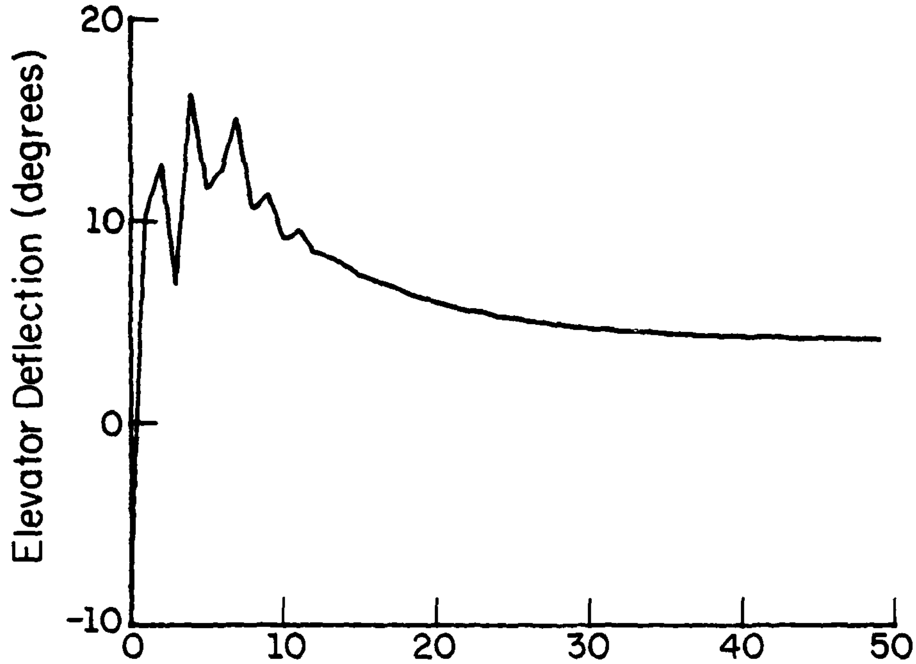


Figure 4.5.4a. Singular perturbation design.

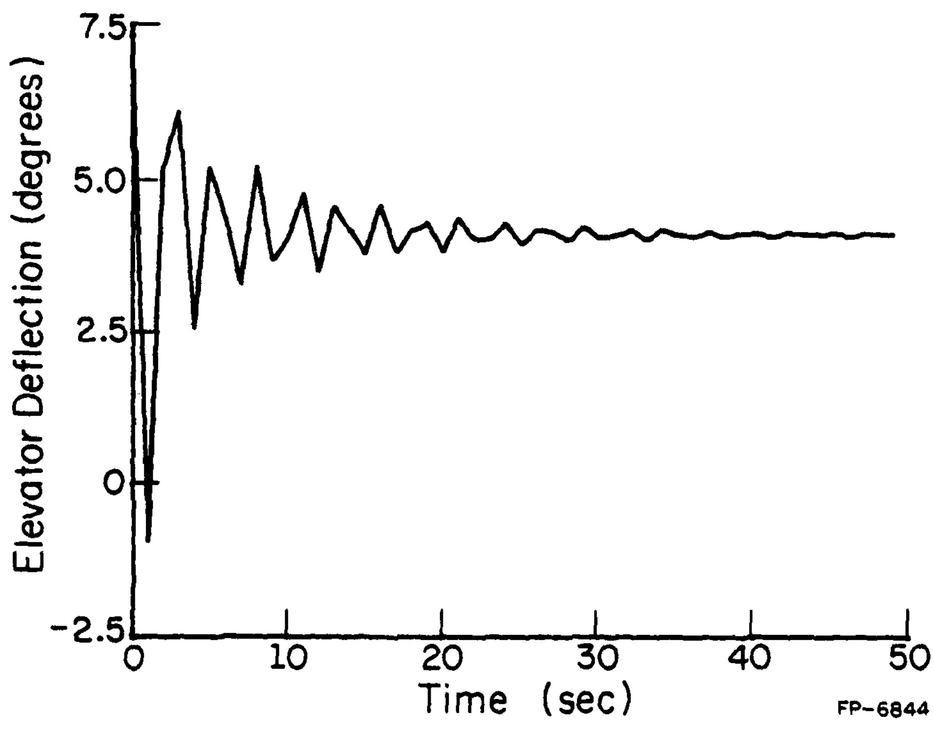


Figure 4.5.4b. Output regulator design.

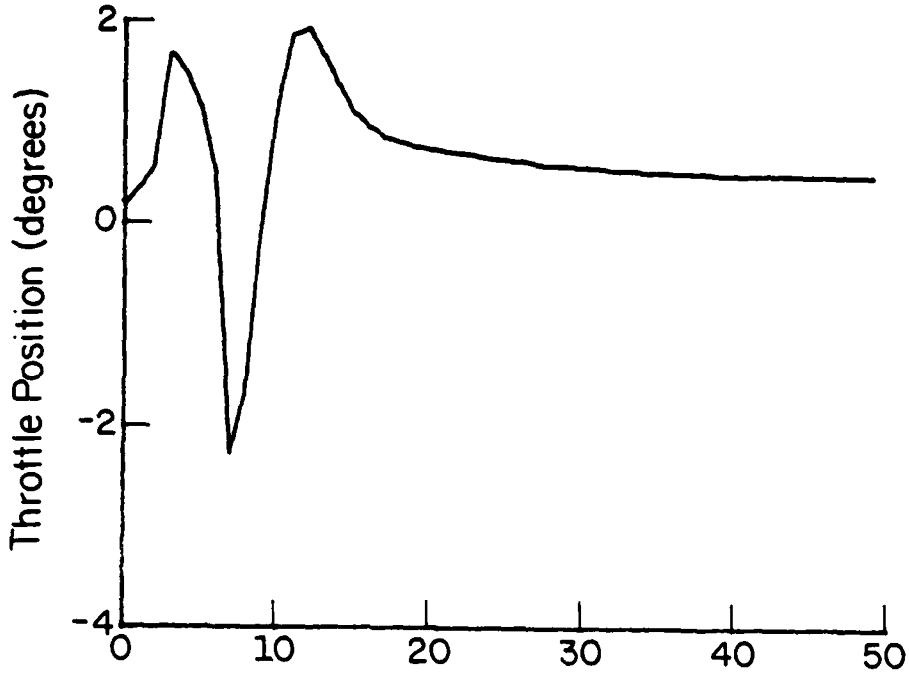


Figure 4.5.5a. Singular perturbation design.

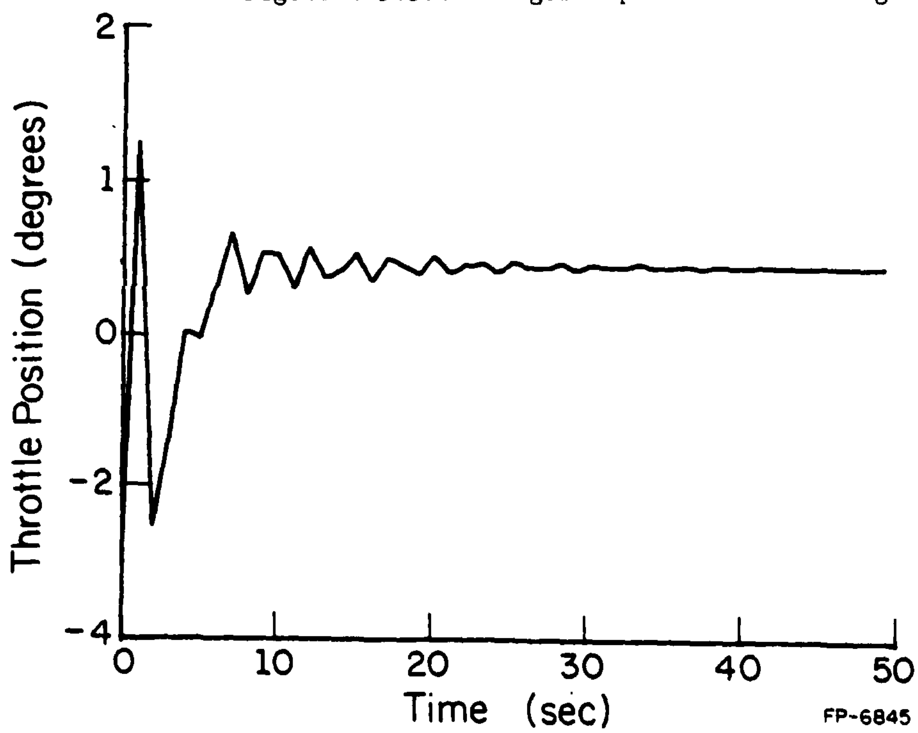


Figure 4.5.5b. Output regulator design.

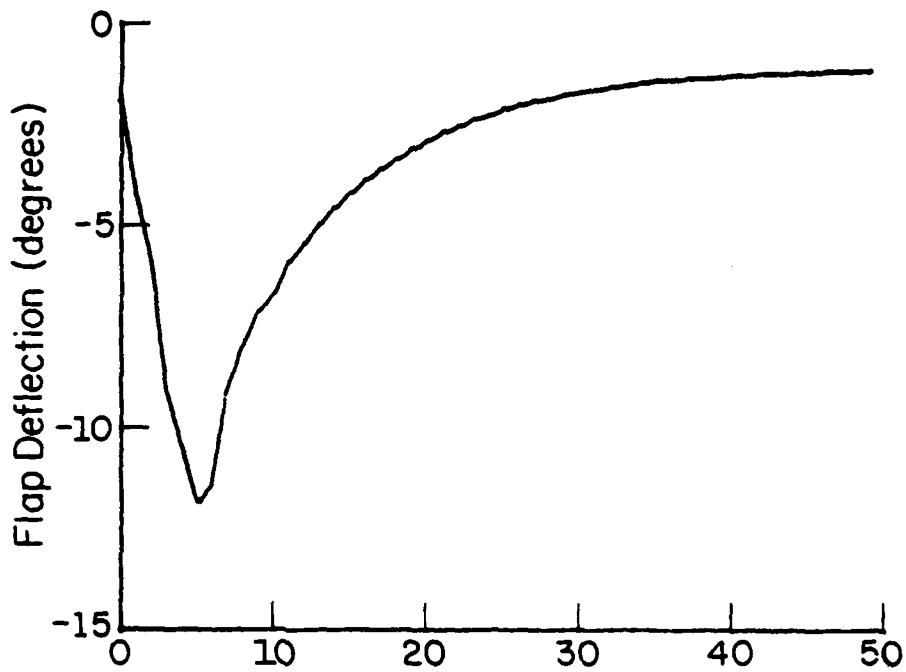


Figure 4.5.6a. Singular perturbation design.

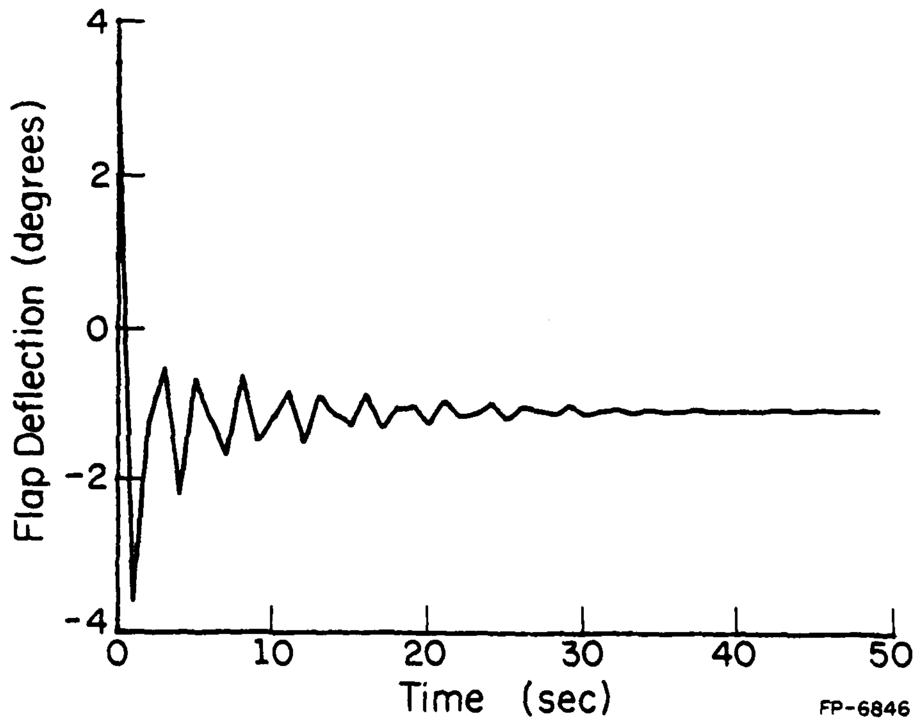


Figure 4.5.6b. Output regulator design.

Figures 4.3-4.5 show the system responses with the dynamic PI controllers. A quick examination of these curves indicates that at least in terms of the state responses the two controllers perform equally well. At the nominal operating point (Figure 4.3), the stability regions with the two controllers are almost identical. It was found that the stability region is enclosed within the boundaries 150-250 ft/sec, $\pm 3.5^{\circ}$, 1700-2400 ft. There were larger overshoots in velocity and altitude responses with controller A than with controller B; whereas the overshoot in the pitch angle was larger with controller B than with controller A. Controller B required a much larger control effort than controller A, which may prove to be an undesirable feature in real time applications.

At a trajectory which forms an 'upper envelope' to the nominal trajectory (Figure 4.4), the performance of the two controllers, in terms of state and control responses, is identical to their performance at the nominal trajectory. The stability regions in this case were 170-300 ft/sec, $\pm 3^{\circ}$, 2000-2500 ft.

At a trajectory which forms a 'lower envelope' to the nominal trajectory (Figure 4.5), controller B is seen to perform significantly better than controller A in terms of overshoot and settling time of the state responses. The control effort required is also smaller in magnitude with controller B than with controller A, although the control responses are not quite 'smooth.' The stability regions with the two controllers were almost identical and were found to be 130-210 ft/sec, $\pm 1.8^{\circ}$, 1650-2000 ft.

From the real-time testing of the controller designs, it is seen that when dealing with systems possessing a two-time-scale property, output regulator theory may not provide a satisfactory solution. If the problem is ill-conditioned, in the sense that it is not possible to retain all the 'small' eigenvalues in the output regulator, the resulting controller will give a performance poorer than that obtained by singular perturbation theory. But, if the problem is not ill-conditioned, then the two techniques may given comparable results. In such a case, which design to use would depend on the specific problem, and the priority of the performance criteria (like the state response, control effort or the stability region).

In dealing with problems such as the one treated in this thesis, singular perturbation theory would be the better technique for the controller design, as it is computationally more efficient than output regulator theory. Output regulator theory involves the solution of the full state regulator problem as a part of the design procedure, which is altogether bypassed in singular perturbation theory. Also, singular perturbation theory is guaranteed to give a satisfactory solution. Output regulator theory, which is based on a sufficient condition of output stabilizability, may not be applicable in many cases.

It is to be pointed out here that the above comments should not lead one to the conclusion that output regulator theory is in any way inferior to singular perturbation theory. The output regulator theory is applicable to a much wider class of problems; and the contention here is that, when dealing with systems possessing a two-time-scale property, singular perturbation theory which specifically handles such problems, would give a better solution than output regulator theory.

A final comment on the small angle of attack approximation made while arriving at the aircraft model. This assumption was shown to be justified by the real-time responses, where it was seen that the angle of attack never exceeded $\pm 1.5^{\circ}$.

5. CONCLUSION

In this thesis, the applicability of two optimal control theories--singular perturbation theory and output regulator theory--have been examined. The performance of these two design methodologies has been judged in terms of the speed of regulation from initial conditions close to the equilibrium trajectory, the control effort required during regulation, the magnitude of the stability region around the equilibrium trajectory, and the system behavior while tracking trajectories other than the nominal one for which the controller has been designed. It was shown that, when dealing with systems possessing a two-time-scale property, singular perturbation theory provides an elegant solution to the control problem. If the 'fast' subsystem is stable, then a partial state feedback controller can be designed based on a reduced order model. When dealing with such systems, output regulator theory will not give a satisfactory solution if the problem is ill-conditioned in the sense discussed before.

In dealing with a more general class of problem (not tried here), where states that are accessible for feedback are a combination of both 'fast' and 'slow,' a combination of the two techniques may be applied. The original system may be decomposed into two lower order subsystems--the 'fast' and the 'slow,' and to each subsystem the output regulator technique may be applied. The resulting controller will be near optimal, provided each of the two subsystem problems are well-conditioned in the sense discussed before.

Also, in this thesis, the versatility of a microcomputer system as a digital controller has been demonstrated. Almost any complex controller structure can be implemented using a microcomputer just by a minor variation in the software.

Since the response at flight conditions away from the nominal degrades rapidly, it is not feasible to use the same feedback matrix over a wide range of flight conditions. A simple thing to do in such a case would be to have a set of precalculated feedback matrices to be used under different flight conditions. But, perhaps a more elegant solution would be to do an on-line estimation of the model parameters, and then to continuously update the feedback matrix as the flight conditions vary. This idea would probably lead one to think in terms of an adaptive control scheme. Any implementation of such a scheme would require a much more sophisticated microcomputer system than the one used in this work (for e.g., it must have a hardware multiplier unit to speed up the on-line computations). The adaptive control technique when applied to nonlinear systems, like an aircraft, has had only a limited success so far, but is quite possibly the method for the future.

REFERENCES

1. D. W. Daly, "A Digital Control System for Reduced Order Decoupled Control of an Aircraft Simulator," M.S. thesis, University of Illlinois, Urbana, 1975.

2. R. L. Jackson, "Feedback Controlled Aircraft Sensitivity to Parameter Variations," M.S. thesis, University of Illinois, Urbana, 1977.

3. J. Medanic, "On Stabilization and Optimization by Output Feedback," *12th Annual Asilomar Conf. on Circuits and Systems*, Pacific Grove, Calif.. November, 1978.

4. J. Medanic, "Design of Low Order Optimal Dynamic Regulators for Linear Time-Invariant Systems," *1979 Conf. on Information Science and Systems*, Baltimore, Md., March 1979.

5. J. H. Chow, "Separation of Time Scales in Linear Time-Invariant Systems," M.S. thesis, University of Illinois, Dept. of Electrical Engineering, Urbana, IL, 1975.

6. P. V. Kokotovic, R. E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory--An Overview," *Automatica*, Vol. 12, 1976, pp. 123-132.

7. P. V. Kokotovic, and R. A. Yackel, "Singular Perturbation of Linear Regulators: Basic Theorems," *IEEE Trans. on Automatic Control*, Vol. AC-17, February 1972, pp. 29-37.

8. R. R. Wilde and P. V. Kokotovic, "Optimal Open- and Closed-Loop Control of Singularly Perturbed Linear Systems," *IEEE Trans. on Automatic Control*, Vol. AC-18, December 1973, pp. 616-625.

9. B. Etkin, *Dynamics of Atmospheric Flight*, Wiley, New York, 1972.

10. S. Bingulac, "LINSYS, Conversational Software for Analysis and Design of Linear Systems," Report T-17, Coordinated Science Lab., University of Illinois, Urbana, June 1975.

APPENDIX A

The purpose of this appendix is to provide adequate information on the existing Z-80 microprocessor. Here an effort has been made to collect the important information pertaining to the chips' hardware and software and present it with some comments on its functional aspect.

A.1. Z-80 CPU Architecture

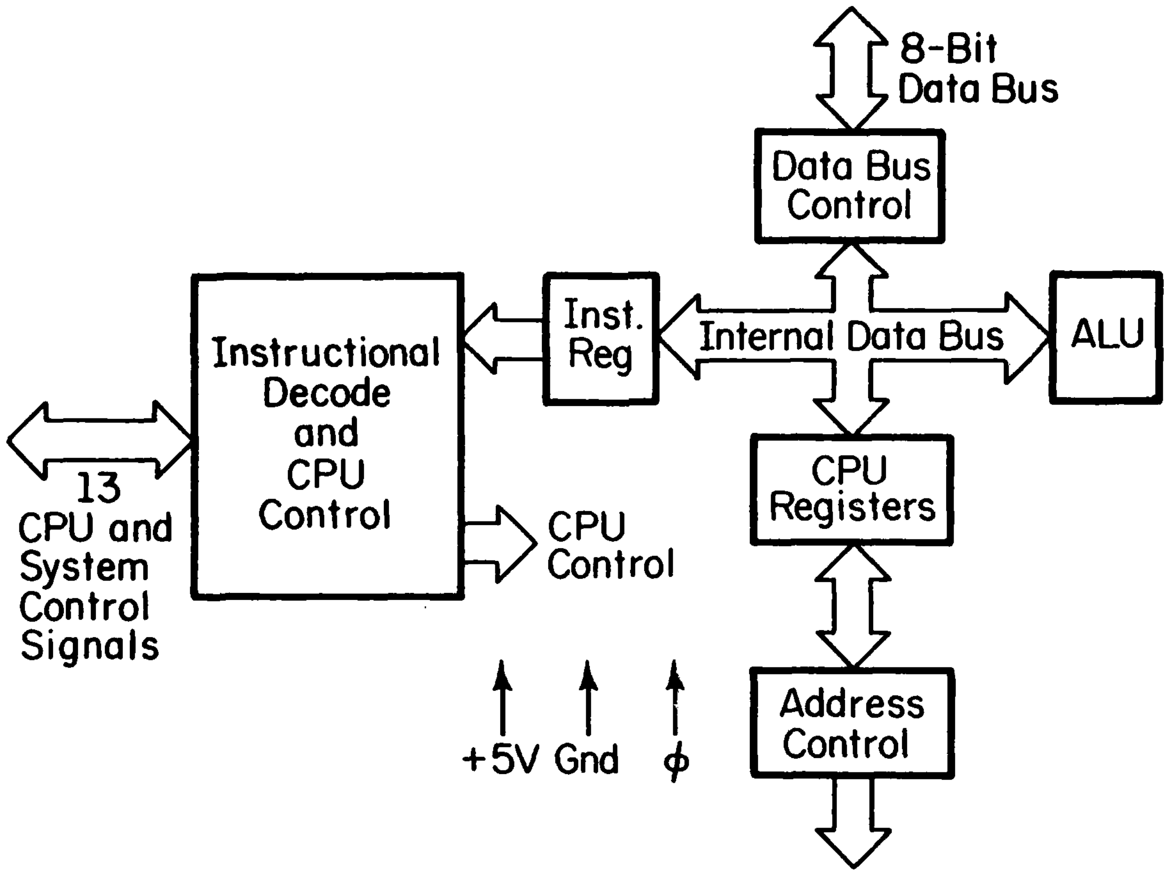
A block diagram of the internal architecture of the Z-80 CPU is shown in Figure A.1. The diagram shows all of the major elements in the CPU.

A.2. CPU Registers

The Z-80 CPU contains 208 bits of R/W memory that are accessible to the programmer. Figure A.2 illustrates how this memory is configured into eighteen 8-bit registers and four 16-bit registers. All Z-80 registers are implemented using static RAM. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers, or in pairs as 16-bit registers. There are also two sets of accumulator and flag registers.

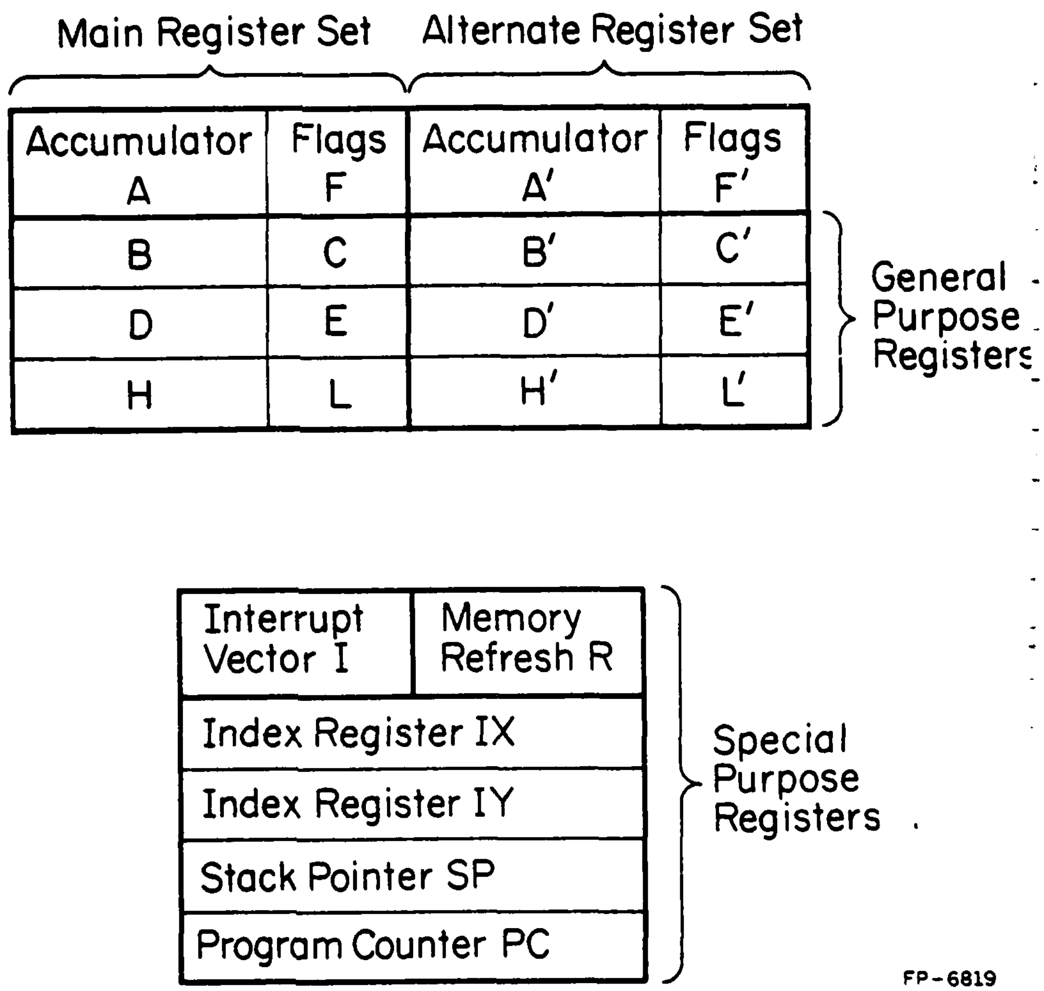
A.2.1. Special purpose registers

i) Program Counter (PC): The program counter holds the 16-bit address of the current instruction being fetched from memory. The PC is automatically incremented after its contents have been transferred to the



FP-6818

Figure A.1. Z80-CPU block diagram.



FP-6819

Figure A.2. Z80-CPU register configuration.

address lines. When a program jump occurs the new value is automatically placed in the PC, overriding the incrementer.

ii) Stack Pointer (SP): The stack pointer holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in first-out (LIFO) file. The stack allows simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data manipulation.

iii) Two Index Registers (IX and IY): The two independent index registers hold a 16-bit base address that is used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's complement signed integer.

iv) Interrupt Page Address Register (I): The Z-80 CPU can be operated in a mode where an indirect call to any memory location can be achieved in response to an interrupt. The I register is used for this purpose to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. This feature allows interrupt routines to be dynamically located anywhere in memory with absolute minimal access time to the routine.

v) Memory Refresh Register (R): The Z-80 CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. This 7-bit register is automatically incremented after each instruction fetch. The data in the refresh counter is set out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer.

A.2.2. Accumulator and flag registers

The CPU includes two independent 8-bit accumulators and associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operations while the flag register indicates specific conditions for 8- or 16-bit operations. The programmer selects the accumulator and flag pair that he wishes to work with with a single exchange instruction so that he may easily work with either pair.

A.2.3. General purpose registers

There are two matched sets of general purpose registers, each set containing six 8-bit registers that may be used individually as 8-bit registers or 16-bit register pairs by the programmer. One set is called BC, DE, and HL while the complementary set is called BD', DE', and HL'. At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set. In systems where fast interrupt response is required, one set of general purpose registers and an accumulator/flag register may be reserved for handling this very fast routine. Only a simple exchange command need be executed to go between the routines. This greatly reduces interrupt service time by eliminating the requirement for saving and retrieving register contents in the external stack during interrupt or subroutine processing. These general purpose registers are used for a wide range of applications by the programmer. They also simplify programming, especially in ROM based systems where little external read/write memory is available.

A.3. Arithmetic and Logic Unit (ALU)

The 8-bit arithmetic and logical instructions of the CPU are executed in the ALU. Internally the ALU communicates with the registers and the external data bus on the internal data bus. The type of functions performed by the ALU include

Add	Left or right shifts (arithmetic and logical)
Subtract	Increment
Logical AND	Decrement
Logical OR	Set bit
Logical EX-OR	Reset bit
Compare	Test bit

A.4. Instruction Registers and CPU Control

As each instruction is fetched from memory, it is placed in the instruction register and decoded. The control section performs this function and then generates and supplies all of the control signals necessary to read or write data from or to the registers, controls the ALU and provides all required external control signals.

A.5. Z-80 CPU Pin Description

The Z-80 CPUis packaged in a standard 40-pin dual in-line package. The I/O pins are shown in Figure A.3 and the function of each is described below.

A_0-A_{15} (Address Bus)
: Tri-state output, active high. A_0-A_{15} constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges. I/O addressing uses the 8 lower address bits to allow the user to directly select up to 256 input or output parts. During refresh time, the lower 7-bits contain a valid refresh address.

D_0-D_7 (Data Bus)
: Tri-state input/output, active high. D_0-D_7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

$\bar{M}_1$ (Machine Cycle One)
: Output, active low. $\bar{M}_1$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution. Note that during execution of 2-byte OP-codes, $\bar{M}_1$ is generated as each OP code byte is fetched. These two byte OP-codes always begin with CBH, DDH, EDH, or FDH. $\bar{M}_1$ also occurs with $\overline{\text{IORQ}}$ to indicate an interrupt acknowledge cycle.

$\overline{\text{MREQ}}$ (Memory Request)
: Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

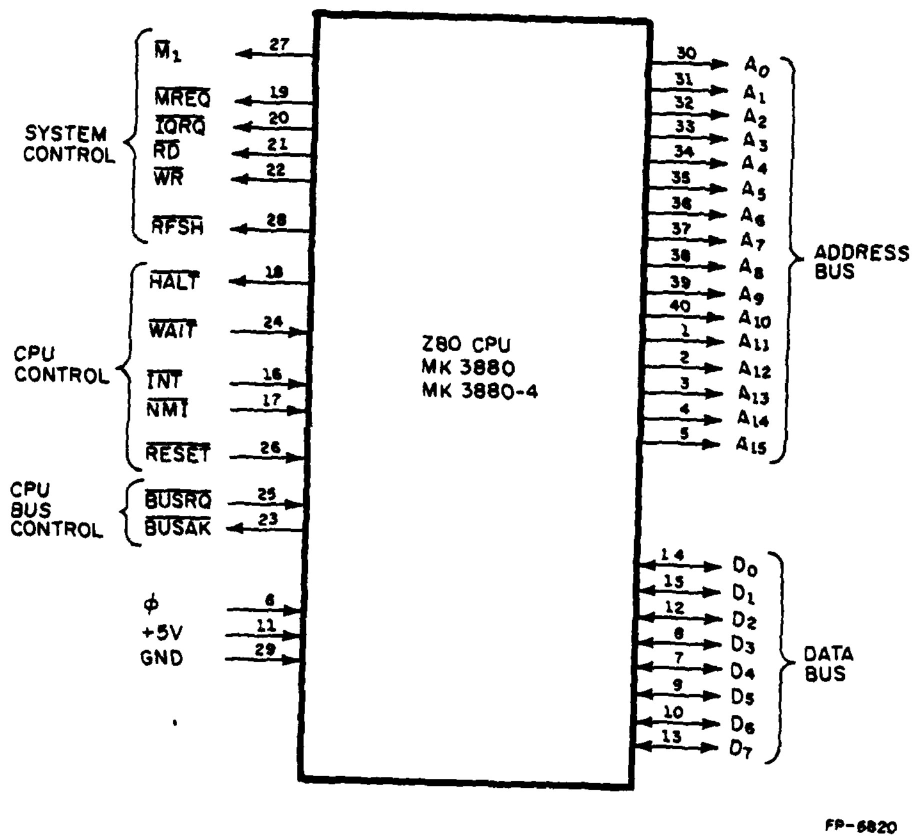


Figure A 3. Z80 pin configuration.

$\overline{\text{IORQ}}$
(Input/Output Request)

Tri-state output, active low. The $\overline{\text{IORQ}}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{\text{IORQ}}$ signal is also generated with an $\overline{M}_1$ signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus.

$\overline{\text{RD}}$
(Memory Read)

Tri-state output, active low. $\overline{\text{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data into the CPU data bus.

$\overline{\text{WR}}$
(Memory Write)

Tri-state output, active low. $\overline{\text{WR}}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.

$\overline{\text{RFSH}}$
(Refresh)

Output, active low. $\overline{\text{RFSH}}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and current $\overline{\text{MREQ}}$ signal should be used to do a refresh read to all dynamic memories. A_7 is a logic zero and the upper 8 bits of the address bus contains the I register.

$\overline{\text{HALT}}$
(Halt State)

Output, active low. $\overline{\text{HALT}}$ indicates that the CPU has executed a HALT instruction and is awaiting an interrupt before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.

$\overline{\text{WAIT}}$ (Wait)

Input, active low. $\overline{\text{WAIT}}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU.

$\overline{\text{INT}}$ (Interrupt Request)

Input, active low. The interrupt request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop is enabled and if the $\overline{\text{BUSRQ}}$ signal is not active. When the CPU accepts an interrupt, an acknowledge signal is sent out at the beginning of the next instruction cycle.

$\overline{\text{NMI}}$ (Nonmaskable Interrupt)

Input, negative edge triggered. The $\overline{\text{NMI}}$ request line has a higher priority than $\overline{\text{INT}}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{\text{NMI}}$ automatically forces the CPU to restart to location 0066H. The PC is automatically saved in the external stack so that the user can return to the program that was interrupted.

$\overline{\text{RESET}}$ (Reset)

Input, active low. $\overline{\text{RESET}}$ forces the PC to zero and initializes the CPU. This includes

1) Disable the interrupt enable flip-flop

2) Set register I = 00H

3) Set register R = 00H

4) Set interrupt mode 0

During reset time, the address bus and the data bus go to a high impedance state and all control output signals go to the inactive state. No refresh occurs.

$\overline{\text{BUSRQ}}$ (Bus Request) — Input, active low. The bus request signal is used to request the CPU address bus, data bus, and tri-state output control signals to go to a high impedance state so that other devices can control these buses. When $\overline{\text{BUSRQ}}$ is activated the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated.

$\overline{\text{BUSAK}}$ (Bus Acknowledge) — Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus, and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

Φ — Single phase system clock.

A.6. CPU Timing

The Z-80 CPU executes instructions by stepping through a very precise set of a few basic operations. These include

Memory read or write

I/O device read or write

Interrupt acknowledge.

All instructions are merely a series of these basic operations. Each of these basic operations can take from three to six clock periods to complete or they can be lengthened to synchronize the CPU to the speed of external devices. The basic clock periods are referred to as T states and the basic operations are referred to as M cycles. Figure A.4 illustrates how a typical instruction will be merely a series of specific M and T cycles. The first machine cycle of any instruction is a fetch cycle which is four, five, or six T stages long (unless lengthened by the wait signal). The fetch cycle (M1) is used to fetch the OP code of the next instruction to be executed. Subsequent machine cycles move data between the CPU and memory or I/O devices and they may have anywhere from three to five T cycles (again they may be lengthened by wait states to synchronize the external devices to the CPU).

A.7. Z-80 CPU Instruction Set

The Z-80 CPU can execute 158 different instruction types including all 78 of the 8080A CPU. The instructions can be broken down into the following major groups

- Load and exchange
- Block transfer and search
- Arithmetic and logical
- Rotate and shift
- Bit manipulation (set, reset, test)
- Jump, call, and return
- Input/output
- Basic CPU control.

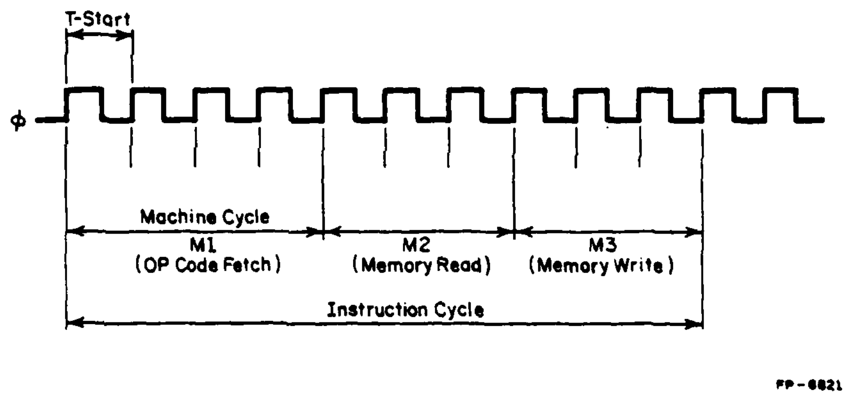


Figure A.4. Basic CPU timing example.

A.7.1. Introduction to instruction types

The load instructions move data internally between CPU registers or between CPU registers and external memory. The source location is not altered by a load instruction. This group also includes load immediate to any CPU register or to any external memory location. The exchange instructions can trade the contents of two registers.

A unique set of block transfer instructions is provided in the Z-80. With a single instruction a block of memory of any size can be moved to any other location in memory. With a single Z-80 block search instruction, a block of external memory of any desired length can be searched for any 8-bit character. Once the character is found the instruction automatically terminates. Both the block transfer and the block search instructions can be interrupted during their execution so as to not occupy the CPU for long periods of time.

The arithmetic and logical instructions operate on data stored in the accumulator and other general purpose CPU registers or external memory locations. The results of the operations are placed in the accumulator and the appropriate flags are set according to the result of the operation. This group also includes 16-bit addition and subtraction between 16-bit CPU registers.

The bit manipulation instructions allow any bit in the accumulator, any general purpose register or any external memory location to be set, reset, or tested with a single instruction. This group is especially useful in control applications and for controlling software flags in general purpose programming.

The jump, call, and return instructions are used to transfer control between various locations in the user's program. This group uses several different techniques for obtaining the new PC address from specific external memory locations. A unique type of jump is the restart instruction. Program jumps may also be achieved by loading register HL, IX, or IY directly into the PC, thus allowing the jump address to be a complex function of the routine being executed.

The input/output group of instructions in the Z-80 allows for a wide range of transfers between external memory locations or the general purpose CPU registers, and the external I/O devices. In each case, the port number is provided on the lower 8 bits of the address bus during any I/O transaction. One instruction allows this port number to be specified by the second byte of the instruction while other Z-80 instructions allow it to be specified as the content of the C register. One major advantage of using the C register as a pointer to the I/O device is that it allows difficult I/O ports to share common software driver routines. This is not possible when the address is part of the OP code if the routines are stored in ROM. Another feature of these input instructions is that they set the flag register automatically so that additional operations are not required to determine the state of the data. The CPU includes single instructions that can move blocks of data (up to 256 bytes) automatically to or from any I/O port directly to any memory location. In conjunction with the dual set of general purpose registers, these instructions provide for fast I/O block transfer rates. The value of this I/O instruction set is demonstrated by the fact that the CPU can provide all required floppy disk formatting on double density floppy disk drives on an interrupt driven basis.

Finally, the basic CPU control instructions allow various options and modes. This group includes instructions such as setting or resetting the interrupt enable flip flop or setting the mode of interrupt response.

APPENDIX B

In this appendix, the software in Z-80 assembly language for implementing the PI-controller is given. The first part of the program computes the control signals at each sampling instant. The second part of the program consists of the various subroutines which were used to perform all the floating point computations (addition, multiplication, vector multiplication, and conversion from floating point to fixed point). The program has been properly documented with appropriate comments to facilitate easier understanding of the algorithms involved.

Z-80 Assembler V1.1

```
  2314                          DIM EQU 2314H
  2315                          AD1 EQU 2315H
  2317                          AD2 EQU 2317H
  231D                          TEMP EQU 231DH
  231F                          CNT EQU 231FH
                                     ORG 1500H

                                ;FEEDBACK GAINS FOR U1
  1500   0000                   K11 DW 0000H
  1502   0000                   K12 DW 0000H
  1504   0000                   K13 DW 0000H
  1506   0000                   K14 DW 0000H
  1508   0000                   K15 DW 0000H
  150A   0000                   K16 DW 0000H
                                ;FEEDBACK GAINS FOR U2
  150C   0000                   K21 DW 0000H
  150E   0000                   K22 DW 0000H
  1510   0000                   K23 DW 0000H
  1512   0000                   K24 DW 0000H
  1514   0000                   K25 DW 0000H
  1516   0000                   K26 DW 0000H
                                ;FEEDBACK GAINS FOR U3
  1518   0000                   K31 DW 0000H
  151A   0000                   K32 DW 0000H
  151C   0000                   K33 DW 0000H
  151E   0000                   K34 DW 0000H
  1520   0000                   K35 DW 0000H
  1522   0000                   K36 DW 0000H
                                ;CURRENT STATE ERRORS
  1524   0000                   X2 DW 0000H
  1526   0000                   X3 DW 0000H
  1528   0000                   X5 DW 0000H
                                ;CURRENT INTEGRATOR VALUES
  152A   0000                   X6 DW 0000H
  152C   0000                   X7 DW 0000H
  152E   0000                   X8 DW 0000H
                                ;NEGATIVE OF STATE SET POINTS
  1530   0186                   X2S DW 8601H
  1532   0000                   X3S DW 0000H
  1534   02C0                   X5S DW 0C002H
                                ;EQUILIBRIUM CONTROLS
  1536   0248                   U1S DW 4802H
  1538   FF7B                   U2S DW 7BFFH
  153A   FE5D                   U3S DW 5DFEH
  153C   FB                             EI                ;ENABLE INTERRUPTS
  153D   21 0010                        LD HL,1000H       ;INITIALIZE STACK POINTER
  1540   F9                             LD SP,HL
  1541   DB19                   LOOP    IN A,(19H)        ;INPUT CURRENT VELOCITY
  1543   47                             LD B,A            ;CONVERT TO FLOATING POINT
  1544   0E00                           LD C,0
  1546   2A 3015                        LD HL,(X2S)       ;COMPUTE ERROR(X2-X2S)
```

Z-80 Assembler V1.1

```
1549  CD C515     CALL ERROR
154C  22 2415     LD (X2),HL      ;STORE ERROR
154F  DB1A        IN A,(1AH)      ;INPUT CURRENT PITCH
1551  47          LD B,A          ;CONVERT TO FLOATING POINT
1552  0E00        LD C,0
1554  2A 3215     LD HL,(X3S)     ;COMPUTE ERROR(X3-X3S)
1557  CD C515     CALL ERROR
155A  22 2615     LD (X3),HL      ;STORE ERROR
155D  DB1B        IN A,(1BH)      ;INPUT CURRENT ALTITUDE
155F  47          LD B,A          ;CONVERT TO FLOATING POINT
1560  0E00        LD C,0
1562  2A 3415     LD HL,(X5S)     ;COMPUTE ERROR(X5-X5S)
1565  CD C515     CALL ERROR
1568  22 2815     LD (X5),HL      ;STORE ERROR
156B  21 0015     LD HL,K11       ;COMPUTE FEEDBACK CONTROL U1
156E  CD CC15     CALL COMPU
1571  2A 3615     LD HL,(U1S)     ;COMPUTE OVERALL CONTROL U1
1574  CD DE15     CALL CTROUT     ;-IN FIXED POINT
1577  D319        OUT (19H),A     ;OUTPUT U1
1579  21 0C15     LD HL,K21       ;COMPUTE FEEDBACK CONTROL U2
157C  CD CC15     CALL COMPU
157F  2A 3815     LD HL,(U2S)     ;COMPUTE OVERALL CONTROL U2-
1582  CD DE15     CALL CTROUT     ;-IN FIXED POINT
1585  D31A        OUT (1AH),A     ;OUTPUT U2
1587  21 1815     LD HL,K31       ;COMPUTE FEEDBACK CONTROL U3
158A  CD CC15     CALL COMPU
158D  2A 3A15     LD HL,(U3S)     ;COMPUTE OVERALL CONTROL U3-
1590  CD DE15     CALL CTROUT     ;-IN FIXED POINT
1593  D31B        OUT (1BH),A     ;OUTPUT U3
1595  2A 2415     LD HL,(X2)      ;UPDATE INTEGRATOR FOR X2-
1598  EB          EX DE,HL        ;X6=X6+(X2-X2S)
1599  2A 2A15     LD HL,(X6)
159C  44          LD B,H
159D  4D          LD C,L
159E  CD E715     CALL ACCUM
15A1  22 2A15     LD (X6),HL
15A4  2A 2615     LD HL,(X3)      ;UPDATE INTEGRATOR FOR X3-
15A7  EB          EX DE,HL        ;X7=X7+(X3-X3S)
15A8  2A 2C15     LD HL,(X7)
15AB  44          LD B,H
15AC  4D          LD C,L
15AD  CD E715     CALL ACCUM
15B0  22 2C15     LD (X7),HL
15B3  2A 2815     LD HL,(X5)      ;UPDATE INTEGRATOR FOR X5-
15B6  EB          EX DE,HL        ;X8=X8+(X5-X5S)
15B7  2A 2E15     LD HL,(X8)
15BA  44          LD B,H
15BB  4D          LD C,L
15BC  CD E715     CALL ACCUM
15BF  22 2E15     LD (X8),HL
15C2  C3 4115     JP LOOP
```

Z-80 Assembler V1.1

```
                         ;SUBROUTINE CALCULATES STATE ERROR
15C5  EB                 ERROR   EX DE,HL
15C6  CD ED15                    CALL FADD
15C9  60                         LD H,B
15CA  69                         LD L,C
15CB  C9                         RET
                         ;SUBROUTINE COMPUTES FEEDBACK CONTROL
15CC  22 1523            COMPU   LD (AD1),HL
15CF  21 2415                    LD HL,X2
15D2  22 1723                    LD (AD2),HL
15D5  3E06                       LD A,06H
15D7  32 1423                    LD (DIM),A
15DA  CD FC16                    CALL VCMLT
15DD  C9                         RET
                         ;SUBROUTINE CALCULATES OVERALL CONTROL
                         ;IN FIXED POINT
15DE  EB                 CTROUT  EX DE,HL
15DF  CD ED15                    CALL FADD
15E2  CD 3317                    CALL CNRT
15E5  78                         LD A,B
15E6  C9                         RET
                         ;SUBROUTINE UPDATES INTEGRATOR STATES
15E7  CD ED15            ACCUM   CALL FADD
15EA  60                         LD H,B
15EB  69                         LD L,C
15EC  C9                         RET
                         ;SUBROUTINE PERFORMS FLOATING POINT ADDITION
                         ;(BC)+(DE)=(BC)
15ED  78                 FADD    LD A,B
15EE  A7                       AND A
15EF  CA 5B16                    JP Z,RSLTD
15F2  7A                       LD A,D
15F3  A7                       AND A
15F4  C8                       RET Z
15F5  79                       LD A,C
15F6  93                       SUB E
15F7  67                       LD H,A
15F8  CA 2816                    JP Z,AD
15FB  F2 1816                   JP P,SFTS
15FE  3E00               SFTB    LD A,00H
1600  94                       SUB H
1601  67                       LD H,A
1602  4B                       LD C,E
1603  FE08                     CP 08H
1605  F2 5B16                   JP P,RSLTD
1608  78                 SFTLP   LD A,B
1609  E6FE                      AND OFEH
160B  F2 0F16                   JP P,SFTRP
160E  3F                        CCF
160F  1F                 SFTRP    RRA
1610  47                        LD B,A
```

Z-80 Assembler V1.1

```
1611  25                  DEC H
1612  C2 0816             JP NZ,SFTLP
1615  C3 2816             JP AD
1618  FE08          SFTS  CP 08H
161A  F0                  RET P
161B  7A            SFTL  LD A,D
161C  E6FE                AND 0FEH
161E  F2 2216             JP P,SFTR
1621  3F                  CCF
1622  1F            SFTR  RRA
1623  57                  LD D,A
1624  25                  DEC H
1625  C2 1B16             JP NZ,SFTL
1628  78            AD    LD A,B
1629  AA                  XOR D
162A  FA 4516             JP M,ADZ
162D  78                  LD A,B
162E  A7                  AND A
162F  FA 3E16             JP M,LZRO
1632  82                  ADD A,D
1633  F2 5E16             JP P,POSS
1636  1F            NRM   RRA
1637  D2 3B16             JP NC,NNCR
163A  3C                  INC A
163B  0C            NNCR  INC C
163C  47            DON   LD B,A
163D  C9                  RET
163E  82            LZRO  ADD A,D
163F  FA 6716             JP M,NEGG
1642  C3 3616             JP NRM
1645  78            ADZ   LD A,B
1646  82                  ADD A,D
1647  CA 5616             JP Z,ZER
164A  FA 6716             JP M,NEGG
164D  0D            LL    DEC C
164E  87                  ADD A,A
164F  F2 4D16             JP P,LL
1652  1F                  RRA
1653  0C                  INC C
1654  47                  LD B,A
1655  C9                  RET
1656  0600          ZER   LD B,00H
1658  0E00                LD C,00H
165A  C9                  RET
165B  42            RSLTD LD B,D
165C  4B                  LD C,E
165D  C9                  RET
165E  0D            POSS  DEC C
165F  87                  ADD A,A
1660  F2 5E16             JP P,POSS
1663  1F                  RRA
```

Z-80 Assembler V1.1

```
1664  0C                     INC C
1665  47                     LD B,A
1666  C9                     RET
1667  0D               NEGG  DEC C
1668  87                     ADD A,A
1669  FA 6716                JP M,NEGG
166C  1F                     RRA
166D  0C                     INC C
166E  47                     LD B,A
166F  C9                     RET
                        ;SUBROUTINES PERFORMS FLOATING POINT MULTIPLICATION
                        ;(BC)*(DE)=(BC)
1670  79               FMULT  LD A,C
1671  83                      ADD A,E
1672  6F                      LD L,A
1673  78                      LD A,B
1674  AA                      XOR D
1675  FA A016                 JP M,NG
1678  78                      LD A,B
1679  A7                      AND A
167A  F2 8416                 JP P,BPOS
167D  2F                      CPL
167E  3C                      INC A
167F  47                      LD B,A
1680  7A                      LD A,D
1681  2F                      CPL
1682  3C                      INC A
1683  57                      LD D,A
1684  48               BPOS   LD C,B
1685  CD D516                 CALL MLT
1688  78               LO     LD A,B
1689  A7                      AND A
168A  FA 9716                 JP M,L101
168D  79                      LD A,C
168E  87                      ADD A,A
168F  4F                      LD C,A
1690  78                      LD A,B
1691  17                      RLA
1692  47                      LD B,A
1693  2D                      DEC L
1694  C3 8816                 JP LO
1697  1F               L101   RRA
1698  47                      LD B,A
1699  D2 9D16                 JP NC,NAD
169C  04                      INC B
169D  2C               NAD    INC L
169E  4D                      LD C,L
169F  C9                      RET
16A0  78               NG     LD A,B
16A1  A7                      AND A
16A2  F2 AB16                 JP P,DNEG
```

Z-80 Assembler V1.1

```
16A5  2F                    CPL
16A6  3C                    INC A
16A7  47                    LD B,A
16A8  C3 AF16               JP L202
16AB  7A          DNEG      LD A,D
16AC  2F                    CPL
16AD  3C                    INC A
16AE  57                    LD D,A
16AF  48          L202      LD C,B
16B0  CD D516               CALL MLT
16B3  78          L3        LD A,B
16B4  A7                    AND A
16B5  FA C216               JP M,L4
16B8  79                    LD A,C
16B9  87                    ADD A,A
16BA  4F                    LD C,A
16BB  78                    LD A,B
16BC  17                    RLA
16BD  47                    LD B,A
16BE  [illegible]           DEC L
16BF  C3 B316               JP L3
16C2  1F          L4        RRA
16C3  47                    LD B,A
16C4  D2 C816               JP NC,NNAD
16C7  04                    INC B
16C8  2C          NNAD      INC L
16C9  4D                    LD C,L
16CA  78                    LD A,B
16CB  2F                    CPL
16CC  3C                    INC A
16CD  47                    LD B,A
16CE  C9                    RET
16CF  0600        ZRO       LD B,00H
16D1  0E00                  LD C,00H
16D3  E1                    POP HL
16D4  C9                    RET
16D5  79          MLT       LD A,C
16D6  A7                    AND A
16D7  CA CF16               JP Z,ZRO
16DA  4F                    LD C,A
16DB  7A                    LD A,D
16DC  A7                    AND A
16DD  CA CF16               JP Z,ZRO
16E0  0600                  LD B,00H
16E2  1E09                  LD E,09H
16E4  79          MULTO     LD A,C
16E5  1F                    RRA
16E6  4F                    LD C,A
16E7  1D                    DEC E
16E8  CA F516               JP Z,DONE
16EB  78                    LD A,B
```

Z-80 Assembler V1.1

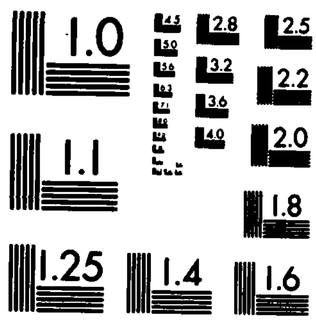
```
16EC  D2 F016                JP NC,MULT1
16EF  82                     ADD A,D
16F0  1F            MULT1    RRA
16F1  47                     LD B,A
16F2  C3 E416                JP MULTO
16F5  79            DONE     LD A,C
16F6  87                     ADD A,A
16F7  4F                     LD C,A
16F8  78                     LD A,B
16F9  17                     RLA
16FA  47                     LD B,A
16FB  C9                     RET
                    ;SUBROUTINE PERFORMS FLOATING POINT VECTOR MULTIPLICATION
                    ;DIIMENSION OF VECTORS : DIM
                    ;POINTER TO FIRST VECTOR : AD1
                    ;POINTER TO SECOND VECTOR : AD2
                    ;TEMPORARY STORAGE : TEMP
                    ;COUNTER : CNT
                    ;RESULT : BC-PAIR
16FC  3A 1423       VCMLT    LD A,(DIM)
16FF  32 1F23                LD (CNT),A
1702  01 0000                LD BC,0000H
1705  ED43 1D23              LD (TEMP),BC
1709  2A 1523       L10      LD HL,(AD1)
170C  4E                     LD C,(HL)
170D  23                     INC HL
170E  46                     LD B,(HL)
170F  23                     INC HL
1710  22 1523                LD (AD1),HL
1713  2A 1723                LD HL,(AD2)
1716  5E                     LD E,(HL)
1717  23                     INC HL
1718  56                     LD D,(HL)
1719  23                     INC HL
171A  22 1723                LD (AD2),HL
171D  CD 7016                CALL FMULT
1720  ED5B 1D23              LD DE,(TEMP)
1724  CD ED15                CALL FADD
1727  ED43 1D23              LD (TEMP),BC
172B  21 1F23                LD HL,CNT
172E  35                     DEC (HL)
172F  C2 0917                JP NZ,L10
1732  C8                     RET Z
                    ;SUBROUTINE PERFORMS FLOATING POINT TO
                    ;FIXED POINT CONVERSION
                    ;(BC)=(B)
1733  79            CNRT     LD A,C
1734  FEF8                   CP 0F8H
1736  F2 3C17                JP P,NTZ
1739  0600                   LD B,00H
173B  C9                     RET
```

AD-A123 945 A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM (U) ILLINOIS UNIV AT URBANA DECISION AND CONTROL LAB V R SAKSENA APR 80 DC-37 N00014-79-C-0424

2/2

UNCLASSIFIED F/G 1/3 NL

END
DATE FILMED
83
DTIC



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

Z-80 Assembler V1.1

```
   173C  A7                NTZ    AND A
   173D  C8                       RET Z
   173E  FE01                     CP 01H
   1740  F2 5117                  JP P,SAT
   1743  78                       LD A,B
   1744  E6FE              LPP    AND OFEH
   1746  F2 4A17                  JP P,STRP
   1749  3F                       CCF
   174A  1F                STRP   RRA
   174B  0C                       INC C
   174C  C2 4417                  JP NZ,LPP
   174F  47                       LD B,A
   1750  C9                       RET
   1751  78                SAT    LD A,B
   1752  A7                       AND A
   1753  067F                     LD B,7FH
   1755  F0                       RET P
   1756  04                       INC B
   1757  C9                       RET
                                 END
```

No program errors.