**A MICROCOMPUTER BASED AIRCRAFT FLIGHT CONTROL SYSTEM**

**VIKRAM RAJ SAKSENA**

**Performing Organization Name and Address**
Joint Services Electronics Program

**Report Date**
April 1980

**Number of Pages**
89

**SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)**
UNCLASSIFIED

**DISTRIBUTION STATEMENT**
Approved for public release; distribution unlimited.

**KEY WORDS**
Microprocessor control  
Singular perturbation  
Optimal output regulator

**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.
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.
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.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

Approved for public release. Distribution unlimited.
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

Director of Thesis Research
Head of Department

Committee on Final Examination†

Chairman

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. AIRCRAFT MODELING</td>
<td>4</td>
</tr>
<tr>
<td>2.1. Dynamical Equations</td>
<td>4</td>
</tr>
<tr>
<td>2.2. Linearization</td>
<td>11</td>
</tr>
<tr>
<td>3. CONTROLLER DESIGN</td>
<td>15</td>
</tr>
<tr>
<td>3.1. Singular Perturbation Theory</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1. General problem</td>
<td>15</td>
</tr>
<tr>
<td>3.1.2. Aircraft controller design</td>
<td>17</td>
</tr>
<tr>
<td>3.2. Output Regulator Theory</td>
<td>22</td>
</tr>
<tr>
<td>3.2.1. General problem</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2. Aircraft controller design</td>
<td>24</td>
</tr>
<tr>
<td>4. REAL TIME IMPLEMENTATION</td>
<td>31</td>
</tr>
<tr>
<td>4.1. Simulation</td>
<td>31</td>
</tr>
<tr>
<td>4.2. Implementation</td>
<td>32</td>
</tr>
<tr>
<td>4.3. Results and Discussion</td>
<td>35</td>
</tr>
<tr>
<td>5. CONCLUSION</td>
<td>63</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>65</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td></td>
</tr>
<tr>
<td>A.1. Z-80 CPU Architecture</td>
<td>66</td>
</tr>
<tr>
<td>A.2. CPU Registers</td>
<td>66</td>
</tr>
<tr>
<td>A.2.1. Special purpose registers</td>
<td>66</td>
</tr>
<tr>
<td>A.2.2. Accumulator and flag registers</td>
<td>70</td>
</tr>
<tr>
<td>A.2.3. General purpose registers</td>
<td>70</td>
</tr>
<tr>
<td>A.3. Arithmetic and Logic Unit (ALU)</td>
<td>71</td>
</tr>
<tr>
<td>A.4. Instruction Registers and CPU Control</td>
<td>71</td>
</tr>
<tr>
<td>A.5. Z-80 CPU Pin Description</td>
<td>72</td>
</tr>
<tr>
<td>A.6. CPU Timing</td>
<td>76</td>
</tr>
<tr>
<td>A.7. Z-80 CPU Instruction Set</td>
<td>77</td>
</tr>
<tr>
<td>A.7.1 Introduction to instruction types</td>
<td>79</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td></td>
</tr>
<tr>
<td>B.1. Controller Software</td>
<td>81</td>
</tr>
</tbody>
</table>
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 microcomputer 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 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

\[
\sin \alpha \approx \alpha \\
\cos \alpha \approx 1 \\
u = v \cos \alpha \approx v \\
\dot{u} = \dot{v} \cos \alpha - \nu \sin \alpha \approx \dot{v} - \nu \dot{\alpha} \approx \dot{v} \\
u = v \sin \alpha \approx \nu \alpha \\
\dot{w} = \dot{v} \sin \alpha + \nu \cos \alpha = \dot{\nu} + \frac{\nu}{v} \approx \nu \\
\sin \theta = \sin(\nu + \alpha) = \sin \nu \cos \alpha + \sin \alpha \cos \nu \\
\approx \sin \nu + \alpha \cos \nu \\
\approx \sin \nu
\]
Table 2.1. Definition of symbols used

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>angle of attach</td>
</tr>
<tr>
<td>θ</td>
<td>pitch angle</td>
</tr>
<tr>
<td>ψ</td>
<td>flight path angle</td>
</tr>
<tr>
<td>M</td>
<td>mass of the aircraft</td>
</tr>
<tr>
<td>V</td>
<td>velocity</td>
</tr>
<tr>
<td>H</td>
<td>altitude</td>
</tr>
<tr>
<td>W</td>
<td>weight of the aircraft</td>
</tr>
<tr>
<td>I_{yy}</td>
<td>moment of inertia</td>
</tr>
<tr>
<td>X_{CG}</td>
<td>center of gravity</td>
</tr>
<tr>
<td>S</td>
<td>wing surface area</td>
</tr>
<tr>
<td>ρ</td>
<td>air density</td>
</tr>
<tr>
<td>C</td>
<td>chord length</td>
</tr>
<tr>
<td>X</td>
<td>body axis</td>
</tr>
<tr>
<td>Z</td>
<td>verticle normal to body axis</td>
</tr>
<tr>
<td>L</td>
<td>lift force</td>
</tr>
<tr>
<td>D</td>
<td>drag force</td>
</tr>
<tr>
<td>T</td>
<td>thrust</td>
</tr>
</tbody>
</table>
\[
\cos \theta = \cos(v + \alpha) = \cos v \cos \alpha - \sin v \sin \alpha \\
= \cos v - a \sin v \\
= \cos v.
\]

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

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

Again, using the above approximations

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

or,

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

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

\[L = \frac{1}{2} \rho V^2 Sc \]

\[D = \frac{1}{2} \rho V^2 Sc_d \]

\[M_y = \frac{1}{2} \rho V^2 Sc m \]
where the coefficients $C_\alpha$, $C_m$, 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

\[ C_\alpha = C_{\alpha 0} + C_{\alpha \alpha} \alpha + C_{\alpha \delta_f} \delta_f \]
\[ C_d = C_{d 0} + C_d^2 + C_{d \delta_f} \delta_f \]
\[ C_m = C_{m 0} + C_{m \alpha} \alpha + C_{m \delta e} \delta e + C_{m \delta_f} \delta_f - \frac{C}{2V} (\dot{\alpha} + \dot{\delta}) \]

where
\[ \delta_f: \text{ flap deflection} \]
\[ \delta_e: \text{ aileron deflection} \]
\[ t: \text{ throttle position}. \]

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

\[ \text{Map} = C_{p0} + C_{p H} + C_{p N} + C_{p t} N \delta t \]
\[ \text{Bhp} = C_{b0} + C_{b N} + C_{b \text{Map}} + C_{b H} \]
\[ T = \text{Ne} \times \text{Bhp}(C_{t0} + C_{t v} V + C_{t h} H + C_{t v h} VH) \]
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

\[ V_o = 190.66 \text{ ft/sec.} \]
\[ H_o = 2000 \text{ ft} \]
\[ \theta_o = \dot{\theta}_o = \alpha_o = 0. \]

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

\[
\begin{align*}
    x_1 &= \alpha & u_1 &= \delta_e \\
    x_2 &= V & u_2 &= \delta_f \\
    x_3 &= \theta & u_3 &= \delta_t \\
    x_4 &= \dot{x} \\
    x_5 &= H.
\end{align*}
\]

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

\[
\begin{align*}
    \dot{x}_1 = x_4 - \frac{1}{Mx_2} \left[ \frac{1}{2} \rho x_2^2 S (C_{k0} + C_{ka} x_1 + C_{ka} u_2) - W \cos (x_3-x_1) \right] \\
    \dot{x}_2 = \frac{1}{M} \left[ -W \sin (x_3-x_1) - \frac{1}{2} \rho x_2^2 S (C_{do} + C_{dcl} (C_{ko} + C_{k0} x_1 + C_{k0} u_2)^2 + C_{df} u_2) \\
    \right. \\
    & \left. \quad + Ne (C_{to} + C_{tv} x_2 + C_{th} x^5 + C_{tvh} x_2 x_5) (C_{bo} + C_{bn} x_5 + C_{bhn} x_5) \\
    & \quad \quad \quad + C_{bp} (C_{po} + C_{ph} x_5 + C_{phn} x_5 + C_{phnt} u_3)) \right]
\end{align*}
\]
Table 2.2. Aircraft parameters

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

$N = 2500 \text{ rpm}$

$Ne = 2$

$p = 0.004842 \text{ slugs/ft}^3$

$S = 180 \text{ ft}^2$

$x_{cg} = 0.2 \text{ ft}$

$c = 5 \text{ ft}$

$W = 4000 \text{ lbs}$

$I_{YY} = 2050 \text{ slugs ft}^2$
\[ \dot{x}_3 = x_4 \]
\[ \dot{x}_4 = \frac{1}{I_{YY}} \left( \frac{1}{2} \rho x_2^2 \sigma \right) \left[ C_{mo} + C_{mcx} (C_{ko} + C_{kax} x_1 + C_{kfu} u_2) + C_{me} \right] \]
\[ + \frac{C_{mf}}{k} \left( x_{2} \right) (x_1 + x_4) \]
\[ \dot{x}_5 = x_2 \sin (x_3 - x_1). \] (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
\[ u_{2e} = \frac{1}{C_{mf}} \left[ \frac{W}{Q_s} - C_{ko} - \frac{C_{a}}{a_0} \right]. \] (2.10)

From \( \dot{x}_4 = 0 \) we obtain
\[ u_{1e} = -\frac{1}{C_{me}} \left[ C_{mo} + C_{mcx} x_1 + C_{mf} u_{2e} \right]. \] (2.11)

From \( \dot{x}_2 = 0 \) we obtain
\[ u_{3e} = \frac{-1}{C_{bp} \ C_{pnt}} \left[ C_{bo} + C_{bp} x_1 + N(C_{bh} + C_{bp} \ C_{pnt}) + (C_{bh} + C_{bp} \ C_{pnt}) \right] \]
\[ - \frac{Q_s}{T_{bhp}} C_d \] (2.12)

where
\[ Q = \frac{1}{2} \rho x_2^2 \]
\[ C_x = C_{ko} + C_{kax} x_1 + C_{kfu} u_2 \]
\[ C_d = C_{do} + C_{dcx} x_1 + C_{dfu} u_2 \]
\[ T_{bhp} = N \left( C_{to} + C_{tv} x_2 + C_{th} x_5 + C_{tvh} x_2 x_5 \right). \]

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
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

\[
\begin{pmatrix}
-3.1 & -0.18 & 0 & 1 & 0 \\
0.14 & -0.07 & -0.32 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
-0.74 & 0.09 & 0 & -1.02 & 0 \\
-1.91 & 0 & 1.91 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
x \\
x \\
x \\
x \\
x
\end{pmatrix}
+
\begin{pmatrix}
0 & -0.25 & 0 \\
0 & -0.04 & -0.16 \\
0 & 0 & 0 & 0 & 0 \\
-1.37 & -1.49 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
0 \\
0 \\
0 \\
u
\end{pmatrix}
\]

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

\[
\begin{align*}
\text{a}_{11} &= -\frac{Q \sigma}{M X_{2e}} C_{la} \\
\text{a}_{12} &= -\frac{p \sigma}{2M} C_{l} - \frac{W}{M X_{2e}} \\
\text{a}_{14} &= 1 \\
\text{c}_{21} &= g - 2Q \frac{S}{M} C_{dc} \frac{C_{la} C_{l}}{C_{l}^2} \\
\text{a}_{22} &= -x_{2e} C_{d} \frac{S \sigma}{M} + N_{e} (C_{tv} + C_{tvh} x_{2e}) \frac{Bhp}{M} \\
\text{a}_{23} &= -g \\
\text{a}_{25} &= \frac{N_{e}}{M} \left[(C_{th} + C_{tvh} x_{2e}) Bhp + Tbh p (C_{bh} + C_{bp} ph)\right] \\
\text{a}_{41} &= \frac{Q \sigma}{4 I_{YY}} \left(C_{mc} C_{la} - \frac{c}{2x_{2e}} a_{11}\right) \\
\text{a}_{42} &= \frac{Q \sigma}{4 I_{YY}} \left[C_{mo} + C_{mc} C_{lo} + C_{mc} C_{la} x_{2e} + C_{me} u_{1e} + (C_{mc} C_{lf} + C_{mf}) u_{2e}\right] + \frac{S \sigma^{2}}{4 I_{YY}} (-2x_{4e} + \frac{Q \sigma}{M} S C_{l}) \\
\text{a}_{43} &= -\frac{Q \sigma^{2}}{2 I_{YY} x_{2e}} a_{13} \\
\text{a}_{44} &= -\frac{Q \sigma^{2}}{I_{YY} x_{2e}} \\
\text{a}_{51} &= -x_{2e} \\
\text{a}_{53} &= x_{2e} \\
\text{a}_{13} &= a_{15} = a_{24} = a_{31} = a_{32} = a_{33} = a_{35} = a_{45} = a_{52} = a_{54} = a_{55} = 0
\end{align*}
\]
Table 2.4. Linearized 'B' coefficients

\[ b_{12} = -\frac{QS}{MX_{2e}} C_{zf} \]

\[ b_{22} = -\frac{SQ}{M} (2C_{dcl} C_{lf} C_{lf} + C_{df}) \]

\[ b_{23} = Thgp C_{bp} C_{pnt} \cdot N/M \]

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

\[ b_{42} = \frac{QSc}{I_{YY}} (C_{mcz} C_{lf} + C_{mf} - \frac{c}{2x_{2e}} b_{12}) \]

\[ b_{11} = b_{13} = b_{21} = b_{31} = b_{32} = b_{33} = b_{43} = b_{51} = b_{52} = b_{53} = 0 \]
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

\[
\begin{align*}
\dot{z}_1 &= A_{11}z_1 + A_{12}z_2 + B_1u; \\
\dot{z}_2 &= A_{21}z_1 + A_{22}z_2 + B_2u; \\
\end{align*}
\]

\[z_1(0) = z_{10}, z_2(0) = z_{20}\]  

where \(z_1 \in \mathbb{R}^{n_1}, z_2 \in \mathbb{R}^{n_2}, u \in \mathbb{R}^m,\) and \(0 < u << 1\)

and the performance index

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

(3.2)
where

\[ Q_1 = Q' > 0 \quad \text{and} \quad R = R' > 0. \]

It is desired to obtain a feedback control \( u = F_z \), 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 \( w = 0 \)

\[
\dot{Z}_s = A_0 Z_s + B_0 u_s; \quad Z_s(0) = Z_{10}
\]

\[
Z_2 = -A_{22}^{-1}(A_{21} Z_s + B_2 u_s)
\]

where,

\[
A_0 = A_{11} - A_{12} A_{22}^{-1} A_{21}
\]

\[
B_0 = B_1 - A_{12} A_{22}^{-1} B_2
\]

\[
J_s = \frac{1}{2} \int_{0}^{\infty} (z'Qz + u'Ru_s) \, dt.
\]

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

\[
u_s = -R^{-1} B' K Z_s
\]

where \( K_s \) is the positive definite solution of the algebraic Riccati equation

\[
A_{0s}' K_s + K_s A_{0s} + Q - K_s B_{0s} R^{-1} B_{0s}' K_s = 0.
\]

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

\[
u = -R^{-1} B_{0s}' K Z_1 = F Z_1
\]
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

$$J_s(\text{opt}) = J(\text{opt}) + O(\mu). \quad (3.8)$$

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

$$Z_1(t) = \exp[(A_0 + B_0 F)t]Z_s(0) + O(\mu)$$

$$Z_2(t) = A_{22}^{-1}(A_{22} + B_2 F)\exp[(A_0 + B_0 F)t]Z_s(0) + \exp[A_{22} t/\mu]Z_\mu(0) + O(\mu)$$

where,

$$Z_s(0) = Z_{10}$$
$$Z_\mu(0) = Z_{20} - Z_2(0). \quad (3.9)$$

3.1.2. Aircraft controller design

The linearized plane equations as given by (2.14) are

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

The eigenvalues of the open loop system are

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

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:

\[
\begin{align*}
\dot{z}_1 &= \begin{bmatrix} -0.07 & -0.32 & 0 \\ 0 & 0 & 0 \\ 0 & 1.91 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 0.14 & 0 \\ 0 & 1 \\ -1.91 & 0 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.04 & -0.16 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u \\
\mu \dot{z}_2 &= \begin{bmatrix} -0.009 & 0 & 0 \\ 0.0045 & 0 & 0 \end{bmatrix} z_1 + \begin{bmatrix} -0.155 & 0.05 \\ -0.037 & -0.051 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.125 & 0 \\ -0.0685 & -0.0745 & 0 \end{bmatrix} u
\end{align*}
\]

where,

\[
\begin{align*}
z_1 &= [x_2 \ x_3 \ x_5]^t \\
z_2 &= [x_1 \ x_4]^t.
\end{align*}
\]

The performance index is chosen to be

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

\[
Q = R = I_{3 \times 3}.
\]

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

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

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

\[
K_s = \begin{bmatrix} 4.29 & 0.27 & 0.71 \\ 0.27 & 2.75 & 1.6 \\ 0.71 & 1.6 & 1.49 \end{bmatrix}
\]
Hence, from (3.7) we obtain
\[
\mathbf{u} = \begin{bmatrix} 0.03 & 1.93 & 0.78 \\ 0.14 & 1.79 & 0.62 \\ 0.69 & 0.04 & 0.11 \end{bmatrix} \mathbf{z}_1
\]  \quad (3.15)

Therefore, the partial state feedback to be applied to the original nonlinear plane (2.9) is given by
\[
\begin{align*}
U_1 &= 2.2344 + 0.03(x_2-x_{2s}) + 1.93(x_3-x_{3s}) + 0.78(x_5-x_{5s}) \\
U_2 &= 0.4798 + 0.14(x_2-x_{2s}) + 1.79(x_3-x_{3s}) + 0.62(x_5-x_{5s}) \\
U_3 &= 0.1816 + 0.69(x_2-x_{2s}) + 0.04(x_3-x_{3s}) + 0.11(x_5-x_{5s}).
\end{align*}
\]  \quad (3.16)

The closed loop eigenvalues of the linearized system (3.10) with the control (3.15) are
\[-0.17, -0.28 \pm j1.98, -1.34, -2.23.\]

For \(x'_0=[1 \ 0 \ 1 \ 1 \ 0]\), 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,
\[J(\text{opt}) = 6.27.\]

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

\[
\begin{align*}
\dot{x}_6 &= x_2 - v_{\text{ref}} \\
\dot{x}_7 &= x_3 - \theta_{\text{ref}} \\
\dot{x}_8 &= x_5 - H_{\text{ref}}.
\end{align*}
\]  

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

\[
\begin{bmatrix}
\dot{z}_1 \\
\dot{z}_2
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -0.07 & -0.32 & 0 & 0.14 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.91 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} u
\]

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

where,

\[
\begin{align*}
z_1 &= [x_6 \ x_7 \ x_8 \ x_2 \ x_3 \ x_5]' \\
z_2 &= [x_1 \ x_4]'.
\end{align*}
\]  

The performance index is chosen to be

\[
J = \frac{1}{2} \int_0^\infty \left( z_1'Qz_1 + u'Ru \right) dt
\]
where
\[
Q = I^{6 \times 6} \quad R = \begin{bmatrix}
0.1 & 0 & 0 \\
0 & 0.5 & 0 \\
0 & 0 & 0.1
\end{bmatrix}.
\] (3.19)

Letting \( u \to 0 \), we obtain the slow subsystem as
\[
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & -0.07 & -0.32 & 0 \\
0 & 0 & 0.11 & 0 & 0 \\
0 & 0 & 0.05 & 1.91 & 0
\end{bmatrix}
\begin{bmatrix}
z_s \\
\dot{z}_s + u_s
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
-0.05 & -0.1 & -0.16 \\
-1.09 & -1.14 & 0 \\
0.67 & 0.85 & 0
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0 \\
z_1
\end{bmatrix}.
\] (3.20)

Based on this reduced order model, the near-optimal control is obtained as
\[
u = \begin{bmatrix}
-0.47 & -0.28 & 3.11 & -0.01 & 9.44 & 6.64 \\
0.11 & 1.4 & 0.14 & 0.16 & 0.86 & -0.12 \\
3.12 & -1.28 & 0.44 & 6.67 & 0.49 & 1.29
\end{bmatrix} z_1.
\] (3.21)

The eigenvalues of the linearized closed loop system with the control of (3.21) are
\[-0.11, \ -0.23 \pm j3.3, \ -0.56 \pm j0.43, \ -0.89, \ -1.34 \pm j1.04.\]

For \( x'_0 = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0] \), 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
\[ U_1 = 2.2344 - 0.01(x_2 - x_{2s}) + 9.44(x_3 - x_{3s}) + 6.64(x_5 - x_{5s}) - 0.46x_6 - 0.28x_7 + 3.11x_8 \]
\[ U_2 = 0.4798 + 0.16(x_2 - x_{2s}) + 0.86(x_3 - x_{3s}) - 0.12(x_5 - x_{5s}) + 0.11x_6 + 1.4x_7 + 0.14x_8 \]
\[ U_3 = 0.1816 + 6.67(x_2 - x_{2s}) + 0.49(x_3 - x_{3s}) + 1.29(x_5 - x_{5s}) + 3.12x_6 - 1.28x_7 + 0.44x_8 \]  

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

\[ \dot{z}_1 = A_{11}z_1 + A_{12}z_2 + B_1u; \quad z_1(0) = z_{10} \]
\[ \dot{z}_2 = A_{21}z_1 + A_{22}z_2 + B_2u; \quad z_2(0) = z_{20} \]
\[ y = z_1 \]

where

\[ z_1 \in \mathbb{R}^n, \quad z_2 \in \mathbb{R}^r, \quad u \in \mathbb{R}^m \]  

(3.23)

and the performance index,

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

where

\[ Q = Q' > 0 \quad \text{and} \quad R = R' > 0. \]  

(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

\[ A'M_c + M_c A + Q - M_c B R^{-1} B'M_c = 0 \]  

(3.28)

and

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

Let

\[ x = \begin{bmatrix} Y \\ Z \end{bmatrix} , \quad Y \in \mathbb{R}^{r \times r} \]

consist of the subset of \( r \) eigenvectors of \( F \) associated with a particular subspectrum \( \Lambda_r \) that we wish to retain in the output regulator.

It has been shown in [3] that, if, for some \( \Lambda_r \), the matrix \( A_r = A_{22} - N A_{12} \), where \( N = Z Y^{-1} \), 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

\[ u = -R^{-1} B'M_c P y \]  

(3.26)

where

\[ P = \begin{bmatrix} I \\ N \end{bmatrix} \]

The cost matrix associated with the control (3.26) is

\[ M_o = M_c + V'D_o V \]  

(3.27)

where

\[ V = \begin{bmatrix} -N & I \end{bmatrix} \]

and \( D_o \) is the unique positive definite solution of the Lyapunov equation
\[ A'\delta + D\delta + G = 0 \]  \hspace{1cm} (3.28)

where
\[ G = [0 \quad I]M_{\text{SM}}[0 \quad I]' \].

### 3.2.2. Aircraft controller design

The linearized plant equations given by (2.14) are put in the form (3.23),

\[
\begin{align*}
\dot{z}_1 &= \begin{bmatrix} -0.07 & -0.32 & 0 \\ 0 & 0 & 0 \\ 0 & 1.91 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 0.14 & 0 \\ 0 & 1 \\ -1.91 & 0 \end{bmatrix} z_2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u \\
\dot{z}_2 &= \begin{bmatrix} -0.18 & 0 & 0 \\ 0.09 & 0 & 0 \end{bmatrix} z_1 + \begin{bmatrix} -3.1 & 1 \\ -0.74 & -1.02 \end{bmatrix} z_2 + \begin{bmatrix} 0 & -0.25 & 0 \\ -1.37 & -1.49 & 0 \end{bmatrix} u
\end{align*}
\]

\[ y = z_1 \]

where
\[ z_1 = [x_2 \quad x_3 \quad x_5]' \]
\[ z_2 = [x_1 \quad x_4]' \]. \hspace{1cm} (3.29)

The performance index is chosen to be

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

\[ Q = R = I_{3x3} \]. \hspace{1cm} (3.30)

Solving (3.25) we obtain the cost for the full state regulator problem as
The eigenvalues of $F$ are

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

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

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

The components of the corresponding eigenvectors are

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

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

$$A_r = A_{22}^{-1}N_{12} = \begin{bmatrix} 0.11 & -0.3 \\ 1.87 & -0.77 \end{bmatrix}$$

$$\Lambda(A_r) = -0.33 \pm j0.6.$$
Hence, $A_r$ being stable the sufficient condition is satisfied

$$
P = \begin{bmatrix} I \\ N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
\begin{bmatrix} 14.25 & 1.3 & 2.73 \\ 17.54 & -0.25 & 2.65 \end{bmatrix}
$$

Hence, the output feedback gain matrix is

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

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

The solution of (3.28) is obtained as

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

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

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
The eigenvalues of the linearized closed loop system are

\[-0.33 + j0.6, -1.63 + j1.22, -1.81.\]

For \(x' = [1 \ 0 \ 1 \ 1 \ 0]\), the optimal cost with full state feedback is

\[J(\text{opt}) = 6.27.\]

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

\[U_1 = 2.2344 + 6.67(x_2-x_{2s}) + 1.5(x_3-x_{3s}) + 1.43(x_5-x_{5s})\]
\[U_2 = 0.4798 + 7.47(x_2-x_{2s}) + 1.39(x_3-x_{3s}) + 1.43(x_5-x_{5s})\]
\[U_3 = 0.1816 + 0.2(x_2-x_{2s}) - 0.01(x_3-x_{3s}) + 0.01(x_5-x_{5s}).\]  
(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
\[
\dot{z}_1 = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & -0.07 & -0.32 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.91 & 0 & 0
\end{bmatrix}z_1 + \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0.14 & 0 \\
0 & 1 \\
0 & 0 \\
-1.91 & 0
\end{bmatrix}z_2 + \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix}u
\]

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

\[y = z_1\]

where

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

\[z_2 = [x_1 \ x_4]'\] (3.33)

The performance index is chosen to be

\[J = \frac{1}{2} \int_0^\infty (z_1'Qz_1 + u'Ru)dt\] (3.34)

where

\[R = \begin{bmatrix}
0.1 & 0 & 0 \\
0 & 0.5 & 0 \\
0 & 0 & 0.1
\end{bmatrix}\]

\[Q = I_{6 \times 6}\]

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

-0.1, -0.56 ± j0.43, -0.99, -1.43 ± j1.82, -2.32 ± j0.23.

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

\[N = ZY^{-1} = \begin{bmatrix}
-0.75 & 1.05 & 5.83 & -0.24 & 8.1 & 8.54 \\
-1.4 & 1.87 & 10.7 & -0.41 & 12.43 & 14.88
\end{bmatrix}\]
The eigenvalues of the linearized closed loop system are

\[-0.1, \ -0.1 \pm j4.35, \ -0.56 \pm j0.43, \ -0.99, \ -1.43 \pm j1.82.\]

For

\[x_0' = [1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0],\]

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

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

(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. REALTIME 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 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 inputting 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.

\[
\dot{x} = f(x,u) \\
Y = Cx \\
u = F_1(Y-Y_{set}) + F_2\int_0^t(Y-Y_{set})dt \\
\]

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 B 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.
Figure 4.2.4a. Singular perturbation design.

Figure 4.2.4b. Output regulator design.
Figure 4.2.5. Singular perturbation 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.

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.

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, \(+3.5^\circ\), 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, \(+3^\circ\), 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, \(+1.8^\circ\), 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 give 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


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

1) 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
Figure A.1. 280-CPU block diagram.
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 incremerter.

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
- Subtract
- Logical AND
- Logical OR
- Logical EX-OR
- Compare

- Left or right shifts (arithmetic and logical)
- Increment
- Decrement
- Set bit
- Reset bit
- 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 CPU is 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<sub>0</sub>-A<sub>15</sub> (Address Bus)**: Tri-state output, active high. A<sub>0</sub>-A<sub>15</sub> 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<sub>0</sub>-D<sub>7</sub> (Data Bus)**: Tri-state input/output, active high. D<sub>0</sub>-D<sub>7</sub> constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

- **M<sub>_1</sub> (Machine Cycle One)**: Output, active low. M<sub>_1</sub> 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, M<sub>_1</sub> is generated as each OP code byte is fetched. These two byte OP-codes always begin with CBH, DDH, EDH, or FDH. M<sub>_1</sub> also occurs with IORQ to indicate an interrupt acknowledge cycle.

- **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.
Tri-state output, active low. The 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 IORQ signal is also generated with an M signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus.

Tri-state output, active low. 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.

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

Output, active low. RFSH indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and current MREQ signal should be used to do a refresh read to all dynamic memories. A7 is a logic zero and the upper 8 bits of the address bus contains the I register.

Output, active low. 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.
WAIT (Wait) Input, active low. 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.

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 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.

NMI (Nonmaskable Interrupt) Input, negative edge triggered. The NMI request line has a higher priority than INT and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. 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.

RESET (Reset) Input, active low. 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 = OOH
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.

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 BUSRQ is activated the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated.

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.

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 inter-
rupt 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  A00 EQU 2315H
2317  A02 EQU 2317H
2319  TEMP EQU 2319H
231F  CNT EQU 231FH
ORG 1500H

1500 0000
1502 0000
1504 0000
1506 0000
1508 0000
150A 0000
150C 0000
150E 0000
1510 0000
1512 0000
1514 0000
1516 0000
1518 0000
151A 0000
151C 0000
151E 0000
1520 0000
1522 0000
1524 0000
1526 0000
1528 0000
152A 0000
152C 0000
152E 0000
1530 0184
1532 0000
1534 02C0
1536 0248
1538 FF78
153A FE5D
153C FB
153D 21 0010
1540 19
1541 0019
1543 47
1544 0E00
1546 :A 3015

82
```

Feedback Gains for U1

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>0000H</td>
</tr>
<tr>
<td>K2</td>
<td>0000H</td>
</tr>
<tr>
<td>K3</td>
<td>0000H</td>
</tr>
<tr>
<td>K4</td>
<td>0000H</td>
</tr>
<tr>
<td>K5</td>
<td>0000H</td>
</tr>
<tr>
<td>K6</td>
<td>0000H</td>
</tr>
<tr>
<td>K7</td>
<td>0000H</td>
</tr>
<tr>
<td>K8</td>
<td>0000H</td>
</tr>
</tbody>
</table>

Feedback Gains for U2

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>0000H</td>
</tr>
<tr>
<td>K2</td>
<td>0000H</td>
</tr>
<tr>
<td>K3</td>
<td>0000H</td>
</tr>
<tr>
<td>K4</td>
<td>0000H</td>
</tr>
<tr>
<td>K5</td>
<td>0000H</td>
</tr>
<tr>
<td>K6</td>
<td>0000H</td>
</tr>
<tr>
<td>K7</td>
<td>0000H</td>
</tr>
<tr>
<td>K8</td>
<td>0000H</td>
</tr>
</tbody>
</table>

Feedback Gains for U3

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>0000H</td>
</tr>
<tr>
<td>K2</td>
<td>0000H</td>
</tr>
<tr>
<td>K3</td>
<td>0000H</td>
</tr>
<tr>
<td>K4</td>
<td>0000H</td>
</tr>
<tr>
<td>K5</td>
<td>0000H</td>
</tr>
<tr>
<td>K6</td>
<td>0000H</td>
</tr>
<tr>
<td>K7</td>
<td>0000H</td>
</tr>
<tr>
<td>K8</td>
<td>0000H</td>
</tr>
</tbody>
</table>

Current State Errors

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>0000H</td>
</tr>
<tr>
<td>X1</td>
<td>0000H</td>
</tr>
<tr>
<td>X2</td>
<td>0000H</td>
</tr>
<tr>
<td>X3</td>
<td>0000H</td>
</tr>
<tr>
<td>X4</td>
<td>0000H</td>
</tr>
<tr>
<td>X5</td>
<td>0000H</td>
</tr>
<tr>
<td>X6</td>
<td>0000H</td>
</tr>
<tr>
<td>X7</td>
<td>0000H</td>
</tr>
<tr>
<td>X8</td>
<td>0000H</td>
</tr>
</tbody>
</table>

Current Integrator Values

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X25</td>
<td>0000H</td>
</tr>
<tr>
<td>X26</td>
<td>0000H</td>
</tr>
<tr>
<td>X27</td>
<td>0000H</td>
</tr>
<tr>
<td>X28</td>
<td>0000H</td>
</tr>
</tbody>
</table>

Equilibrium Controls

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>15U</td>
<td>0402H</td>
</tr>
<tr>
<td>U25</td>
<td>7H8F</td>
</tr>
<tr>
<td>U35</td>
<td>5F7F</td>
</tr>
</tbody>
</table>

Initialize Interrupts

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>E90H</td>
</tr>
<tr>
<td>LD MH,1000H</td>
<td>INITIALIZE STACK POINTER</td>
</tr>
<tr>
<td>LD SP,HL</td>
<td>INPUT CURRENT VOLTAGE</td>
</tr>
<tr>
<td>IN A,(YH)</td>
<td>CONVERT TO FLOATING POINT</td>
</tr>
<tr>
<td>LD B+X</td>
<td>COMPUTE EKKUR(X2-X2S)</td>
</tr>
</tbody>
</table>
ISUBROUTINE CALCULATES STATE ERROR
ERROR \( E \) FROM \( \Delta E \)
CALL FADD
LD H,B
LD L,C
RET

ISUBROUTINE COMPUTES FEEDBACK CONTROL
COMP \( L \) \((A\theta_1) + H\)
LD ML,X2
LD (A\theta_2) + H
LD A,0AH
LD (DI) + A
CALL VCMLT
HST

ISUBROUTINE CALCULATES OVERALL CONTROL
IN FIXED POINT
CTROUT LD H,HL
CALL FADD
LD A,B
RET

ISUBROUTINE UPDATES INTEGRATOR STATES
ACCUM CALL FADD
LD H,B
LD L,C
RET

ISUBROUTINE PERFORMS FLOATING POINT ADDITION
(BC)+(DE)=(EH)
FADD LD A,B
AND A
JP Z,KSLTU
LD A,D
AND A
RET Z
LD A,C
SUH E
LD H,A
JP Z,AD
JP P,BFTS
CA 2816
CA 1816
SFTB LD A,0DH
SFTB LD A,0DH
SUM H
LD H,A
LD C,E
CP OHH
JP P,RSSTD
LD H,B
JP P,SHRBP
JP P,SHF
JP P,SHTRP
JP P,SHF
JP P,SHTRP
RET A
RET A
LD H,A
Z-80 Assembler V1.1

1611 25  DEC H
1612 C2 0B1A  JP NZ, SFTLP
1613 C3 2816  JP AD
1614 FF0B  SFTS CP 0HH
1615 F0  RET F
1616 7A  SFTL LD A+D
1617 FA  AND OFEH
1618 F2 2216  JP P+SFTR
1619 3F  CIR
1620 1F  SFTR KRA
1621 57  LD R+A
1622 29  DEC H
1623 C2 1B16  JP NZ, SFTL
1624 7B  AD LD A+D
1625 AA  XOR D
1626 78  JP HADZ
1627 FA 3E16  LD A+H
1628 A7  AND A
1629 46  JP H:ZERO
1630 82  AMD A+D
1631 F2 5E16  JP F+POSS
1632 1F  NRM KRA
1633 D2 3B16  JP NC; NNCR
1634 3C  INC A;
1635 0C  NNCR INC A;
1636 47  DGN LD B+1
1637 C9  RET
1638 82  LZRO ADD A+D
1639 FA 6716  JP HNEG
1640 C3 3E16  JP NRM
1641 7B  ADZ LD A+D
1642 82  AND A+D
1643 CA 5E16  JP Z; ZER
1644 FA 4716  JP K; ZER
1645 0D  LI DEC C
1646 87  ADD A+D
1647 F2 4D16  JP P+LL
1648 1F  RRA
1649 CC  INC C
1650 47  LD H+1
1651 C9  RET
1652 0A  ZER LD B+00H
1653 00  LD C+00H
1654 0E00  RET
1655 C9  RSLTP LD B+D
1656 42  LD C+L
1657 4B  RET
1658 C9  POSS DEC C
1659 0D  ADD A+H
165A 87  JP P+POSS
165B F2 5E16  KRA
165C 1F

85
Z-80 Assembler V1.1

1464 0C INC C
1465 47 LD B+A
1464 C9 RET
1467 0D NEGG DEC C
1468 87 ADD A+A
1469 FA 6716 JP M+NEGG
146C 1F RRA
146D 0C INC C
146E 47 LD B+A
146F C9 RET

SUBROUTINES PERFORMS FLOATING POINT MULTIPLICATION
((BC)*(DE)+(HC)

1470 79 FMULT LD A+C
1471 83 ADD A+E
1472 4F LD L+A
1473 7B LD A+B
1474 AA XOR D
1475 FA A016 JP M+NG
1476 7B LD A+B
1477 87 AND A
1478 47 JP P+BPOS
147A F2 8416 CPL
147D 3C INC A
147E 47 LD B+A
1480 7A LD A+D
1481 2F CPL
1482 3C INC A
1483 57 LD D+A
1484 40 BPOS LD D+B
1485 CD 0516 CALL MLT
1488 7B LO LD A+B
1489 A7 AND A
148A FA 9716 JP M+L101
148D 79 LD A+C
148E 87 ADD A+A
148F 4F LD C+A
1490 7B LU A+H
1491 17 MLA
1492 47 LD B+A
1493 2D DEC L
1494 C3 8816 JP LO
1497 1F L101 RRA
1498 47 LD B+A
1499 D2 9B16 JP NC+HAV
149C 24 INC B
149D 2C NAD INC L
149E 4D LD C+L
149F C9 RET
14A0 7B NG LD A+B
14A1 47 AND A
14A2 F2 AR16 JP P+OHG
<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16A5</td>
<td>2F</td>
<td>CPL</td>
</tr>
<tr>
<td>16A6</td>
<td>3C</td>
<td>INC A</td>
</tr>
<tr>
<td>16A7</td>
<td>47</td>
<td>LD B+ A</td>
</tr>
<tr>
<td>16A8</td>
<td>C3 AF16</td>
<td>JP L202</td>
</tr>
<tr>
<td>16A9</td>
<td>7A</td>
<td>DNEG</td>
</tr>
<tr>
<td>16AC</td>
<td>2F</td>
<td>CRP</td>
</tr>
<tr>
<td>16AD</td>
<td>3C</td>
<td>INC A</td>
</tr>
<tr>
<td>16AE</td>
<td>48</td>
<td>L202</td>
</tr>
<tr>
<td>16B0</td>
<td>CD 0516</td>
<td>CALL ML1</td>
</tr>
<tr>
<td>16B3</td>
<td>7B</td>
<td>L3</td>
</tr>
<tr>
<td>16B4</td>
<td>A7</td>
<td>ANB A</td>
</tr>
<tr>
<td>16B5</td>
<td>FA C216</td>
<td>JP ML4</td>
</tr>
<tr>
<td>16BB</td>
<td>79</td>
<td>ADD A+C</td>
</tr>
<tr>
<td>16BC</td>
<td></td>
<td>AND A+ A</td>
</tr>
<tr>
<td>16BD</td>
<td></td>
<td>LD C+ A</td>
</tr>
<tr>
<td>16BE</td>
<td></td>
<td>LD A+ B</td>
</tr>
<tr>
<td>16BF</td>
<td></td>
<td>RLA</td>
</tr>
<tr>
<td>16C2</td>
<td>1F</td>
<td>DEC L</td>
</tr>
<tr>
<td>16C3</td>
<td>47</td>
<td>JP L3</td>
</tr>
<tr>
<td>16C4</td>
<td>D2 CB16</td>
<td>JP ML4,NNAD</td>
</tr>
<tr>
<td>16C7</td>
<td>04</td>
<td>INC L</td>
</tr>
<tr>
<td>16C8</td>
<td>2C</td>
<td>NNAD</td>
</tr>
<tr>
<td>16CA</td>
<td>4D</td>
<td>INC B</td>
</tr>
<tr>
<td>16CB</td>
<td>78</td>
<td>LD C+L</td>
</tr>
<tr>
<td>16CC</td>
<td>2F</td>
<td>LD A+R</td>
</tr>
<tr>
<td>16CD</td>
<td>3C</td>
<td>CPL</td>
</tr>
<tr>
<td>16CE</td>
<td>47</td>
<td>INC A</td>
</tr>
<tr>
<td>16CF</td>
<td>0600</td>
<td>RET</td>
</tr>
<tr>
<td>16D1</td>
<td>0E00</td>
<td>LD B+0OH</td>
</tr>
<tr>
<td>16D3</td>
<td>E1</td>
<td>LD C+0OH</td>
</tr>
<tr>
<td>16D4</td>
<td>C9</td>
<td>POP HI</td>
</tr>
<tr>
<td>16D5</td>
<td>79</td>
<td>RET</td>
</tr>
<tr>
<td>16D6</td>
<td>A7</td>
<td>LD A+C</td>
</tr>
<tr>
<td>16D7</td>
<td>CA CF16</td>
<td>ANB A</td>
</tr>
<tr>
<td>16DA</td>
<td>4F</td>
<td>JP Z+Z&lt;0</td>
</tr>
<tr>
<td>16DH</td>
<td>7A</td>
<td>LD C+ A</td>
</tr>
<tr>
<td>16DC</td>
<td>A7</td>
<td>LD A+ D</td>
</tr>
<tr>
<td>16DD</td>
<td>CA CF16</td>
<td>JP Z+Z&lt;0</td>
</tr>
<tr>
<td>16E0</td>
<td>0E00</td>
<td>LD B+0OH</td>
</tr>
<tr>
<td>16E2</td>
<td>1E09</td>
<td>LD E+09H</td>
</tr>
<tr>
<td>16EA</td>
<td>79</td>
<td>MULTO</td>
</tr>
<tr>
<td>16E3</td>
<td>1F</td>
<td>LD A+C</td>
</tr>
<tr>
<td>16E4</td>
<td>4F</td>
<td>RRA</td>
</tr>
<tr>
<td>16E7</td>
<td>1D</td>
<td>LD C+ A</td>
</tr>
<tr>
<td>16FB</td>
<td>CA F516</td>
<td>JP Z+DONE</td>
</tr>
<tr>
<td>16FB</td>
<td>7B</td>
<td>LD A+ B</td>
</tr>
</tbody>
</table>
Z-80 Assembler V1.1

16EC D2 F016 JP HC,MULTI
16EF R2 AND A,B
16FO 1F MULTI KNA
16F1 47 LD B,A
16F2 C3 E416 JP MULTO
16F5 79 DONE LH A,C
16F6 87 ADD A,A
16F7 4F LD C,A
16F8 7B LD A,B
16F9 17 MLA
16FA 47 LD B,A
16FB C9 RT

ISUBROUTINE F-FROM-FLOATING-POINT VECTOR MULTIPLICATION
IDIMENSION OF VECTORS : DIM
IPINTER TO FIRST VECTOR : A+1
IPINTER TO SECOND VECTOR : A+2
ITEMPORARY STORAGE : IEMP
ICOUNTER :CNT
IRRESULT :B1-PAIR

UCNT LD A+(DIM)
16FC 3A 1423 LD (CNT)+A
16FF 32 1F23 LD B,C,0000H
1702 01 0000 LD (TEMP)+HC
1705 ED43 1023 LD (TEMP)+HC
1709 2A 1523 L10 LD HL,(ANI)
170C 4E LD C,(HL)
170F 23 INC HL
170E 46 LD B,(HL)
1710 23 INC HL
1711 3A 1723 LD (ANI)+HL
1713 2A 1723 (LD HL+)HL2)
1716 5E LD E,(HL)
1717 23 INC HL
1716 56 LD D,(HL)
1719 23 INC HL
171A 22 1723 LD (ANI)+HL
171D LD 7016 CALL FAULT
1720 FDSB 1U23 LD D,B,(I/FMP)
1724 CD ED15 CALL FAH
1727 ED43 1023 LD (TEMP)+BC
172B 21 1F23 LD HL,CNT
172E 35 DEC (HL)
172F C2 0917 JP MZ,L10
1732 CB RET Z

ISUBROUTINE F-FROM-FLOATING-POINT TO FIXED-POINT CONVERSION
I(MC)+B

1733 79 CNRT LD A+6
1734 FEF8 CP OFH
1736 F2 3C17 JP P,N1Z
1739 0600 LD B,000H
173C 09 RET
### Z-80 Assembler v1.1

<table>
<thead>
<tr>
<th>Hex</th>
<th>Instruction</th>
<th>Hex</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>173C</td>
<td>A7</td>
<td>173B</td>
<td>CB</td>
</tr>
<tr>
<td>173E</td>
<td>FE01</td>
<td>1740</td>
<td>F2 5117</td>
</tr>
<tr>
<td>1743</td>
<td>7B</td>
<td>1744</td>
<td>E6FE</td>
</tr>
<tr>
<td>1746</td>
<td>F2 4A17</td>
<td>1749</td>
<td>3F</td>
</tr>
<tr>
<td>174A</td>
<td>1F</td>
<td>174B</td>
<td>0C</td>
</tr>
<tr>
<td>174C</td>
<td>C2 4417</td>
<td>174F</td>
<td>47</td>
</tr>
<tr>
<td>1750</td>
<td>C9</td>
<td>1751</td>
<td>7B</td>
</tr>
<tr>
<td>1752</td>
<td>A7</td>
<td>1753</td>
<td>067F</td>
</tr>
<tr>
<td>1755</td>
<td>F0</td>
<td>1756</td>
<td>04</td>
</tr>
<tr>
<td>1757</td>
<td>C9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- **NTZ**: AND A
- **RET Z**: R encontrar
- **CP 01H**: JP P + SAT
- **LPP**: LD A + R
- **AND 0FEH**: JP P + STRP
- **CCF**: CCF
- **STPF**: STRP
- **INC**: ING C
- **JP M7 + LPP**: JP M7 + LPP
- **LD B + A**: LD B + A
- **RET**: RET
- **LD A + B**: LD A + B
- **ANU A**: ANU A
- **LD B + 7F8H**: LD B + 7F8H
- **RET P**: RET P
- **INC B**: INC B
- **RET**: RET
- **END**: END

No program errors.