33487212

MISSILE AUTOPILOT DESIGN USING A
GAIN SCHEDULING TECHNIQUE

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
David Paul White
March, 1994

THE
LIBRARY
OF
OHIO
UNIVERSITY

OHIO UNIVERSITY
LIBRARY

Acknowledgements

Now that my goals have been attained and the work ultimately seeming worthwhile, there are several people that I wish to thank and recognize.

Foremost, I thank Dr. Douglas Lawrence for his guidance and for sharing his wisdom with me. I could not have completed this thesis without his help.

I would like to thank my wife, Juley, and my son, Joshua, for giving me the motivation and support needed to see me through the last few months. I love you both.

To my family and friends that were patient and undemanding of my very limited time, thanks so much for your understanding.

Finally, thanks to the whole Electrical and Computer Engineering Department for their excellent teaching and instruction, and to Jason Wozniak for his feedback throughout this project.

Table of Contents

Acknowledgements	i
Table of Contents	ii
List of Figures	iii
List of Tables	iv
Chapter 1: Introduction	1
1.1 <u>Gain Scheduling Concept</u>	2
1.2 <u>Gain Scheduling Approach</u>	2
1.3 <u>Project Objectives</u>	3
1.4 <u>Notation and Definitions</u>	3
Chapter 2: Missile Autopilot Problem Formulation	5
2.1 <u>Pitch-Axis Missile Description</u>	5
2.2 <u>Design Objectives</u>	7
Chapter 3: Gain Scheduling Theory	10
3.1 <u>Constant Operating Point Family</u>	10
3.2 <u>Family of Plant Linearizations</u>	14
3.3 <u>Parameterized Family of Linear Controllers</u>	15
3.4 <u>Nonlinear Dynamic Controller</u>	16
Chapter 4: Missile Autopilot Design	20
4.1 <u>Computation of Family of Constant Operating Points</u>	20
4.2 <u>Design of the Family of Linear Controllers</u>	27
4.3 <u>A Gain Scheduled Missile Autopilot</u>	39
4.4 <u>Evaluation of the Gain Scheduled Missile Autopilot</u>	43
Chapter 5: Conclusions	56
References	58

List of Figures

Figure 2.1	Pitch-axis missile.	6
Figure 3.1	Plant and controller diagram.	11
Figure 4.1	Constant operating point angle of attack.	25
Figure 4.2	Constant angle of attack contours.	25
Figure 4.3	Constant operating point tail deflection angle.	26
Figure 4.4	Simulink s-function of missile plant.	26
Figure 4.5	Linear controller structure.	29
Figure 4.6	Step response at design point $(\eta_c^\circ, M^\circ) = (0, 3)$	38
Figure 4.7	Bode diagram with the loop open at the actuator input.	38
Figure 4.8	Mesh plot of maximum real part eigenvalues.	38
Figure 4.9	Mesh plot of minimum return difference.	38
Figure 4.10	Step response at operating point with lowest MRD.	39
Figure 4.11	Mesh plot of rise time.	39
Figure 4.12	Simulink block diagram of missile plant and autopilot.	50
Figure 4.13	Mesh plot of K_M values.	50
Figure 4.14	Mach profile for Simulation # 2, $(\eta_c(0) = 0)$	50
Figure 4.15	Simulation # 1, $(\eta_c(0) = 0, M(0) = 3)$	51
Figure 4.16	Simulation # 2, $(\eta_c(0) = 0, M(0) = 3)$	51
Figure 4.17	Simulation # 3, $(\eta_c(0) = 0, M(0) = 3)$	52
Figure 4.18	Simulation # 4, $(\eta_c(0) = 0, M(0) = 3)$	52
Figure 4.19	Plot of $\eta_z(t)$ vs. $M(t)$, Simulation #2, $(\eta_c(0) = 0, M(0) = 3)$	53
Figure 4.20	Step responses demonstrating undershoot, Simulation #2, $(\eta_c(0) = 0, M(0) = 3)$	53
Figure 4.21	Improved performance by parameterizing <i>controllermex</i> controller on $\eta_z(t)$, $(\eta_c(0) = 0, M(0) = 2.6)$	53
Figure 4.22	Step responses at $(\eta_c^\circ, M^\circ) = (10.75, 2.25)$ comparing linear performance.	54
Figure 4.23	Dc-gains for incorrect interpolation.	54
Figure 4.24	Robustness analysis for Simulation # 2, $(\eta_c(0) = 0, M(0) = 3)$	54
Figure 4.25	Simulation results comparing incorrect interpolation to Simulation # 2, $(\eta_c(0) = 0, M(0) = 2.6)$	55
Figure 4.26	Simulation results for $K_M(0) \neq 0$, $(\eta_c(0) = 0, M(0) = 3)$	55

List of Tables

Table 2.1	Constants for missile problem equations.	9
Table 4.1	Design point controller coefficients.	37
Table 4.2	Simulation parameterizations.	46

Chapter 1: Introduction

There are several methods used to design nonlinear control systems. One method is to linearize a nonlinear plant about a nominal equilibrium point and then use linear design techniques. Since the controller is designed using an approximation of the plant which is only accurate near the nominal equilibrium point, the performance of the controller will likely be acceptable only near the equilibrium point. This technique is only successful in cases where the desired operating region of the system is contained within this region of acceptable performance. Robust linear control is another class of methods used in designing controllers for nonlinear systems. This method uses, in addition to the approximation of the plant at the nominal point, characterizations of the linearized plant's operating point variations as uncertainties. Methods in this class of design techniques are H^∞ and μ -synthesis. A problem with these methods is that controller order is typically too high. Another method for dealing with nonlinearities is adaptive control. In this method the plant model and the controller parameters are varied due to measurements during operation to adjust to the nonlinearities. This computation online causes the cost of the system to be high in the controls sense [6].

This thesis explores an alternative to the above methods, called gain scheduling. Gain scheduling is another method for designing controllers for nonlinear systems which like the other methods, has advantages and disadvantages. Section 1.1 describes the concept of gain scheduling and some of the advantages and disadvantages. The gain scheduling approach used in this project is detailed in Section 1.2. The objectives of this

thesis are given in Section 1.3. In Section 1.4 the notation and definitions are presented.

1.1 Gain Scheduling Concept

The basic idea behind gain scheduling is to apply linear design techniques to a nonlinear plant. This involves designing linear controllers at a number of constant operating points, called design points, in the plant's operating region and interpolating or "scheduling" the linear controllers to obtain the nonlinear or gain scheduled controller used at intermediate operating points. An advantage of this method is that well known linear design methods can be utilized in the linear design process. If done correctly, any linear design methodology can be applied to the gain scheduling method.

In designing the linear controllers, the plant is linearized about each design point, therefore the performance of the nonlinear controller in a region close to the design point is similar to the performance of the linear controller. However, for operating points away from the design points, the changes in the controller may not be proportional to the changes in the plant linearizations. This causes degraded performance of the nonlinear controller in all or some of the operating region.

1.2 Gain Scheduling Approach

This section deals with a particular method for designing a gain scheduled controller. This method is described in [2] and is listed here:

- Compute the plant's family of constant operating points, parameterized by constant values of a selected set of scheduling signals.
- From the family of constant operating points, construct the family of linearized plants.

For this family of linearized plants design a family of linear controllers to meet the design specifications throughout the operating region.

- Compute a gain scheduled controller that linearizes to the correct linear controller at each constant operating point.
- Evaluate the performance of the gain scheduled controller by simulation.

1.3 Project Objectives

There are several objectives of this project. The main objective is to apply the above gain scheduling method to a pitch-channel autopilot design for a hypothetical missile. This will be done using classical design techniques to realize the linear controllers at the design points. The aim of the linear design is to create controllers that are of low order to emphasize that through gain scheduling, a nonlinear plant can be controlled by scheduling low order linear controllers, as opposed to designing a single high order linear controller. Also, in the linear designs, exogenous inputs will be treated as disturbance inputs to the plant. This is done so that steady state errors will be decreased. The final objective of the project is to be able to use plant data obtained "experimentally" and stored in tabular form. This enables controllers to be designed for plants whose dynamics are so complicated as to preclude modeling by closed-form analytical expressions.

1.4 Notation and Definitions

This section summarizes the notation and definitions used in this paper.

Notation and definitions:

- Matrices and sub-matrices will be denoted by capital letters, e.g. A denotes a matrix, while A_{II} denotes the upper left sub-matrix of A .
- Vectors will be denoted by lower case letters, e.g. x , while x_i denotes the i -th element of x .
- The cartesian product space is defined as in [1] as the set of all pairs (x,y) where $x \in X$ and $y \in Y$. The cartesian product is denoted by $X \times Y$.
- Operating point parameterization variables are denoted by w^ρ .
- Subscript notation will be used when a transfer function's coefficients are a function of the parameterization variables w^ρ , e.g. $G_{w^\rho}(s)$.
- A matrix that is a function of the parameterization variables w^ρ will be denoted with w^ρ in parenthesis, e.g. $A(w^\rho)$.
- The Jacobian matrix is defined as follows: If $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$, then $\frac{\partial f}{\partial x}$ denotes the $p \times n$ Jacobian matrix whose (i,j) -entry is the partial derivative $\frac{\partial f_i}{\partial x_j}$.
- Matlab and Simulink commands will be denoted in upper case letters, e.g. LINMOD.

Chapter 2: Missile Autopilot Problem Formulation

This thesis applies the gain scheduling method outlined in Section 1.2 to the pitch-axis missile problem described in [5]. Section 2.1 details the missile problem and the describing equations. Section 2.2 lists the design objectives.

2.1 Pitch-Axis Missile Description

The problem treated in this project involves a three degree of freedom model of a missile. Figure 2.1 illustrates the pitch-axis missile model and the essential variables. The control problem involves determining a tail fin deflection that will generate the proper normal acceleration vector $\eta_z(t)$, or lift vector, as shown in the figure. This lift is generated by the body of the missile acting as an airfoil when there is a nonzero angle between the velocity vector and the center line of the missile. This angle is defined as the angle of attack $\alpha(t)$. The tail fin deflection creates a lift vector that rotates the missile about the center of mass. Since the velocity vector passes through the center of mass, the angle between the velocity vector and the center line of the missile changes due to this rotation of the missile about the center of mass. This change in angle of attack yields a change in normal acceleration. The resultant normal acceleration acts on one point which is defined as the center of pressure. When the center of pressure is forward of the center of mass, the missile is statically unstable, and when the center of pressure is behind the center of mass the missile is statically stable.

Figure 2.1 Pitch-axis missile.

The following equations describe the pitch-axis dynamics for the missile. [5] The relevant variables are:

- $\alpha(t)$, angle of attack in degrees,
- $q(t)$, pitch rate in degrees per second,
- $M(t)$, mach number,
- $\delta_c(t)$, commanded tail deflection angle in degrees,
- $\delta(t)$, actual tail deflection angle in degrees,
- $\eta_c(t)$, commanded normal acceleration in g's,
- $\eta_z(t)$, actual normal acceleration in g's.

The equations for angle of attack and pitch rate are given by

$$\begin{aligned}\dot{\alpha}(t) &= K_\alpha M(t) C_n[\alpha(t), \delta(t), M(t)] \cos(\alpha(t)) + q(t), \\ \dot{q}(t) &= K_q M^2(t) C_m[\alpha(t), \delta(t), M(t)].\end{aligned}\tag{2.1}$$

The aerodynamic coefficients in the above equations are modeled as

$$\begin{aligned} C_n[\alpha, \delta, M] &= \text{sgn}(\alpha)[a_n|\alpha|^3 + b_n|\alpha|^2 + c_n(2-M/3)|\alpha|] + d_n\delta, \\ C_m[\alpha, \delta, M] &= \text{sgn}(\alpha)[a_m|\alpha|^3 + b_m|\alpha|^2 + c_m(-7+8M/3)|\alpha|] + d_m\delta. \end{aligned} \quad (2.2)$$

The tail fin actuator is described by the following state equation

$$\frac{d}{dt} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_a^2 & -2\zeta\omega_a \end{bmatrix} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_a^2 \end{bmatrix} \delta_c(t). \quad (2.3)$$

The output equation is

$$\eta_z(t) = K_z M^2(t) C_n[\alpha(t), \delta(t), M(t)]. \quad (2.4)$$

The coefficients that were not listed as variables above are given in Table 2.1 along with their values. There are two inputs to the plant, the control input $\delta_c(t)$, and the exogenous input $M(t)$, which will be treated as a disturbance input in the controller design. The output variables available for feedback are $\eta_z(t)$, $M(t)$, and $q(t)$.

2.2 Design Objectives

The performance goals defined in [5] are used in this project and are quoted below. The objectives will be modified slightly in Chapter 4 because the parameterization variables that will be used are $\eta_c(t)$ and $M(t)$, whereas the performance goals are stated here in terms of $\alpha(t)$ and $M(t)$.

- Maintain robust stability over the operating range specified by $(\alpha(t), M(t))$ such that $-20^\circ \leq \alpha(t) \leq 20^\circ$ and $1.5 \leq M(t) \leq 3$. Robustness refers to uncertainty in pitching moment, which will be interpreted as uncertainty in the aerodynamic coefficient

$C_m[\alpha, \delta, M]$. The closed loop system should remain stable while the coefficients of the angle of attack and tail-deflection portions of $C_m[\alpha, \delta, M]$ vary independently by $\pm 25\%$.

- Track step commands in $\eta_c(t)$ with time constant no greater than 0.35 seconds, maximum overshoot no greater than 10%, and steady-state error no greater than 1%.
- Maintain at least 30dB attenuation at 300 rad/sec for the open-loop transfer function of the linearized system, with the loop opened at the input to the actuator, that is, at $\delta_c(t)$. This requirement insures that the autopilot avoids exciting unmodeled structural dynamics.

Verification that the nonlinear gain scheduled controller meets these criterion will be given in Section 4.4.

K_α	$= (0.7)P_o S/mv_s$	
K_q	$= (0.7)P_o Sd/I_y$	
K_z	$= (0.7)P_o S/m$	
A_x	$= (0.7)P_o SC_\alpha/m$	
P_o	$= 973.3 \text{ lbs/ft}^2$	static pressure at 20,000 ft
S	$= 0.44 \text{ ft}^2$	surface area
m	$= 13.98 \text{ slugs}$	mass
v_s	$= 1036.4 \text{ ft/sec}$	speed of sound at 20,000 ft
d	$= 0.75 \text{ ft}$	diameter
I_y	$= 182.5 \text{ slug} \cdot \text{ft}^2$	pitch moment inertia
C_α	$= -0.3$	drag coefficient
ω_a	$= 150 \text{ rad/sec}$	actuator undamped natural frequency
ζ	$= 0.7$	actuator damping ratio
a_n	$= 0.000103 \text{ deg}^{-3}$	
b_n	$= -0.00945 \text{ deg}^{-2}$	
c_n	$= -0.1696 \text{ deg}^{-1}$	
d_n	$= -0.034 \text{ deg}^{-1}$	
a_m	$= 0.000215 \text{ deg}^{-3}$	
b_m	$= -0.0195 \text{ deg}^{-2}$	
c_m	$= 0.051 \text{ deg}^{-1}$	
d_m	$= -0.206 \text{ deg}^{-1}$	

Table 2.1 Constants for missile problem equations.

Chapter 3: Gain Scheduling Theory

In this chapter, background material on gain scheduling is presented. Section 3.1 deals with the theory needed to perform Step 1. of the gain scheduling method discussed in Chapter 1, the computation of the family of constant operating points of the plant. The existence of constant operating point families is discussed. Next, in Section 3.2, the first part of Step 2. of the method is investigated, namely, finding the family of linearized plants corresponding to constant operating points in the operating range. Then in Section 3.3 the family of linear controllers is presented. Finally, in Section 3.4 the existence of the dynamic nonlinear controller is examined. Existence conditions and their effects on the structure of the nonlinear controller are determined.

3.1 Constant Operating Point Family

In this section the existence of a parameterized family of constant operating points is established by use of the implicit function theorem. To begin, the control problem is put in the basic form given in [2]. The block diagram of this form is shown in Figure 3.1. The signals shown in the figure and their dimensions are:

- $w(t)$, vector of exogenous signals, $w(t) \in \mathbb{R}^l$,
- $z(t)$, error vector, $z(t) \in \mathbb{R}^q$,
- $u(t)$, control input, $u(t) \in \mathbb{R}^m$,
- $y(t)$, vector of measurable outputs, $y(t) \in \mathbb{R}^p$,
- $r(t)$, reference input to be tracked, $r(t) \in \mathbb{R}^h$,
- $x(t)$, plant state, $x(t) \in \mathbb{R}^n$,
- $w_M(t)$, measured exogenous scheduling signal, $w_M(t) \in \mathbb{R}^g$,

- $y_R(t)$, tracking output signal, $y_R(t) \in \mathbb{R}^h$,
- $y_M(t)$, additional measured output signals, $y_M(t) \in \mathbb{R}^k$.

Figure 3.1 Plant and controller diagram.

From the plant model, the nonlinear state space equations for the plant dynamics can be expressed as

$$\begin{aligned}
 \dot{x}(t) &= f(x(t), w(t), u(t)), \\
 z(t) &= h_1(x(t), w(t)), \\
 y(t) &= h_2(x(t), w(t)) \\
 &= \begin{bmatrix} z(t) \\ w(t) \\ y_M(t) \end{bmatrix}.
 \end{aligned} \tag{3.1}$$

The first step in gain scheduling involves computing the constant operating point family for the plant. Constant operating points and constant operating point functions are defined as in [8]. For the m -input, p -output, n -dimensional system with l exogenous

variables in equation (3.1) , the set of constant operating points corresponding to zero error is given by

$$E = \{(x^o, w^o, u^o, y^o) \in \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^p \mid f(x^o, w^o, u^o) = 0, \\ h_1(x^o, w^o) = 0, y^o = h_2(x^o, w^o)\}.$$

In order to show the existence of a parameterized family of constant operating points we will begin by stating the implicit function theorem [7].

Theorem 1. (Implicit Function Theorem)

Suppose $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ is continuously differentiable on an open set S of $\mathbb{R}^n \times \mathbb{R}^m$ containing (x_1^o, x_2^o) . Let $F(x_1^o, x_2^o) = 0$ and suppose $\frac{\partial f}{\partial x_2}(x_1^o, x_2^o)$ is nonsingular. Then there is a neighborhood M of (x_1^o, x_2^o) , contained in S , and a neighborhood O of x_1^o in $\mathbb{R}^n$ on which a unique continuously differentiable transformation $G : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is defined, such that

$$(x_1, G(x_1)) \in M, \quad \forall x_1 \in O,$$

$$\text{where } G(x_1^o) = x_2^o, \text{ and}$$

$$F(x_1, G(x_1)) = 0, \quad \forall x_1 \in O.$$

Proof: See [7], pp. 439-41.

Now, a theorem for the existence of a family of parameterized constant operating points for the system in equation (3.1) is presented.

Theorem 2. (Local Existence of Constant Operating Point Families)

Assume $x_o = 0 \in \mathbb{R}^n$, $w_o = 0 \in \mathbb{R}^l$, and $u_o = 0 \in \mathbb{R}^m$. For the system in equation (3.1) suppose $f : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ is continuously differentiable on an open neighborhood $S \subset \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m$ containing (x_o, w_o, u_o) , and $h_1 : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}^q$ is continuously differentiable on

an open neighborhood $T \subset \mathbb{R}^n \times \mathbb{R}^l$ containing (x_o, w_o) , and suppose $f(x_o, w_o, u_o) = 0$, $h_1(x_o, w_o) = 0$. Consider the case where $m = q$. Suppose,

$$\text{rank} \begin{bmatrix} \frac{\partial f}{\partial x}(x_o, w_o, u_o) & \frac{\partial f}{\partial u}(x_o, w_o, u_o) \\ \frac{\partial h_1}{\partial x}(x_o, w_o) & 0 \end{bmatrix} = n+q.$$

Then there exists an open set $0 \in O \subset \mathbb{R}^l$ and continuously differentiable functions $x^o : O \rightarrow \mathbb{R}^n$ and $u^o : O \rightarrow \mathbb{R}^m$, with $x^o(0) = 0$, $u^o(0) = 0$, such that

$$\begin{aligned} 0 &= f(x^o(w^o), w^o, u^o(w^o)), \\ 0 &= h_1(x^o(w^o), w^o), \end{aligned} \quad \text{for } w^o \in O. \quad (3.2)$$

For the function $h_2 : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}^p$ define

$$y^o(w^o) = h_2(x^o(w^o), w^o).$$

which is continuously differentiable.

Proof: Direct application of the implicit function theorem by identifying

$$\begin{aligned} x_1 &\iff w, \\ x_2 &\iff (x, u), \\ F(x_1, x_2) &\iff \begin{bmatrix} f(x, w, u) \\ h_1(x, w) \end{bmatrix}. \quad \blacksquare \end{aligned}$$

3.2 Family of Plant Linearizations

If there exists a locally parameterized family of constant operating points for (3.1), the parameterized family of plant linearizations can be written by taking partial derivatives of equation (3.1). First, the deviation variables are defined as the difference between the actual value of the variable and the value of the variable at a constant operating point. Deviation variables will be denoted with a δ subscript, e.g.

$$x_{\delta}(t) = x(t) - x^o(w^o),$$

$$u_{\delta}(t) = u(t) - u^o(w^o),$$

$$z_{\delta}(t) = z(t),$$

$$y_{\delta}(t) = y(t) - y^o(w^o),$$

$$w_{\delta}(t) = w(t) - w^o,$$

The matrices for the parameterized linear state space equations are computed by taking partial derivatives as follows

$$F(w^o) = \frac{\partial f}{\partial x}(x^o(w^o), w^o, u^o(w^o)),$$

$$G_1(w^o) = \frac{\partial f}{\partial w}(x^o(w^o), w^o, u^o(w^o)),$$

$$G_2(w^o) = \frac{\partial f}{\partial u}(x^o(w^o), w^o, u^o(w^o)),$$

$$H_1(w^o) = \frac{\partial h_1}{\partial x}(x^o(w^o), w^o),$$

$$J_1(w^o) = \frac{\partial h_1}{\partial w}(x^o(w^o), w^o),$$

$$H_2(w^o) = \frac{\partial h_2}{\partial x}(x^o(w^o), w^o),$$

$$J_2(w^o) = \frac{\partial h_2}{\partial w}(x^o(w^o), w^o).$$

Now the plant linearization family is given by

$$\begin{aligned}\dot{x}_\delta(t) &= F(w^o)x_\delta(t) + G_1(w^o)w_\delta(t) + G_2(w^o)u_\delta(t), \\ z_\delta(t) &= H_1(w^o)x_\delta(t) + J_1(w^o)w_\delta(t), \\ y_\delta(t) &= H_2(w^o)x_\delta(t) + J_2(w^o)w_\delta(t).\end{aligned}\tag{3.3}$$

It is possible to transform the linearized state space family to a family of parameterized transfer functions. The Laplace transforms of the deviation variables are

$$\begin{aligned}X_\delta(s) &= \mathcal{L}[x_\delta(t)], \\ U_\delta(s) &= \mathcal{L}[u_\delta(t)], \\ Z_\delta(s) &= \mathcal{L}[z_\delta(t)], \\ Y_\delta(s) &= \mathcal{L}[y_\delta(t)], \\ W_\delta(s) &= \mathcal{L}[w_\delta(t)].\end{aligned}$$

Now the transfer functions are computed by

$$\begin{aligned}G_{w^o}^{ij}(s) &= H_i(w^o)[sI - F(w^o)]^{-1}G_j(w^o) + J_{ij}(w^o), \\ J_{ij}(w^o) &= \begin{cases} J_{ij}(w^o) & \text{for } j=1, \\ 0 & \text{for } j=2. \end{cases}\end{aligned}$$

Then the family of parameterized transfer functions is written as

$$\begin{bmatrix} Z_\delta(s) \\ Y_\delta(s) \end{bmatrix} = \begin{bmatrix} G_{w^o}^{11}(s) & G_{w^o}^{12}(s) \\ G_{w^o}^{21}(s) & G_{w^o}^{22}(s) \end{bmatrix} \begin{bmatrix} W_\delta(s) \\ U_\delta(s) \end{bmatrix}.\tag{3.4}$$

3.3 Parameterized Family of Linear Controllers

For the plant linearization family it is assumed that a family of parameterized linear controllers can be constructed to meet the design specifications throughout the

operating region. To achieve zero steady-state error it will be necessary to have an integrator in the controller. Since the system will be type one, the controller will be written with an integrator component, $x_I(t)$, separate from the other controller states, $x_C(t)$. The dimension of $x_I(t)$ is n_I , and the dimension of $x_C(t)$ is n_C . Then the linear controller family can be written as a parameterized state equation

$$\begin{aligned}\dot{x}_{C\delta}(t) &= A_{11}(w^o)x_{C\delta}(t) + A_{12}(w^o)x_{I\delta}(t) + B_1(w^o)y_\delta(t), \\ \dot{x}_{I\delta}(t) &= A_{21}(w^o)x_{C\delta}(t) + B_2(w^o)y_\delta(t), \\ u_\delta(t) &= C_1(w^o)x_{C\delta}(t) + C_2(w^o)x_{I\delta}(t) + D(w^o)y_\delta(t),\end{aligned}$$

or as a parameterized transfer function

$$U_\delta(s) = K_w(s)Y_\delta(s)$$

3.4 Nonlinear Dynamic Controller

In this section the general form of the "gain scheduled" or nonlinear dynamic controller is presented. The controller will be written in a similar way as the linear family of controllers, with an integrator component, $x_I(t)$, separate from the other controller states, $x_C(t)$. The general form for the nonlinear dynamic controller is

$$\begin{aligned}\dot{x}_C(t) &= a_1(x_C(t), x_I(t), y(t)), \\ \dot{x}_I(t) &= a_2(x_C(t), y(t)), \\ u(t) &= c(x_C(t), x_I(t), y(t)).\end{aligned}\tag{3.5}$$

A requirement of the nonlinear dynamic controller is to linearize to the family of linear controllers at each operating point. In order for this requirement to be met there

are existence conditions that must be addressed. The existence conditions are the necessary and sufficient conditions for functions $x_c^o(w^o)$, and $x_l^o(w^o)$ to exist such that at a constant operating point, the controller applies a constant control signal to the plant, which yields zero error. If the existence conditions are met then the solutions to the resulting system of total differential equations are the functions $x_c^o(w^o)$, and $x_l^o(w^o)$. It is required that the existence conditions are met in order for the nonlinear dynamic controller to function properly.

In order to discuss the existence conditions for the nonlinear dynamic controller the issue of the placement of the integrator must be addressed. First, write the parameterized transfer function for the linear controller as

$$\begin{aligned}
 U_\delta(s) &= K_{w^o}(s)Y_\delta(s) \\
 &= \begin{bmatrix} K_{w^o}^{11}(s) + \frac{K_{w^o}^{12}(s)}{s} & K_{w^o}^{21}(s) & K_{w^o}^{31}(s) \end{bmatrix} \begin{bmatrix} Z_\delta(s) \\ W_\delta(s) \\ Y_{M\delta}(s) \end{bmatrix}. \tag{3.6}
 \end{aligned}$$

As in [2] the error signal $Z_\delta(s)$ can be filtered, and then integrated or integrated and then filtered. The "integral filtered error" is given by $\frac{1}{s} \cdot [K_{w^o}^{12}(s)Z_\delta(s)]$, where the "filtered integral error" is described by $K_{w^o}^{12}(s) \left[\frac{Z_\delta(s)}{s} \right]$. It is shown in [2] that the integral filtered error setup leads to existence conditions that are independent of $K_{w^o}^{12}(s)$, while the existence conditions of the "filtered integral error" setup are not. Therefore, in this project the "integral filtered error" setup will be used.

Now, a theorem and the corresponding assumptions from [2] are presented which

will establish conditions for there to exist a controller of the form (3.5) that linearizes to the correct linear controller family.

Assumption 1. For the system in equation (3.1) there exists an open set $0 \in \mathcal{O} \subset \mathbb{R}^l$ and functions $x^o : \mathcal{O} \rightarrow \mathbb{R}^n$ and $u^o : \mathcal{O} \rightarrow \mathbb{R}^m$, with $x^o(0)=0$, $u^o(0)=0$, such that equation (3.2) holds.

Assumption 2. Each entry of the linear controller transfer function family $K_{w^o}(s)$ in equation (3.6) is a fixed degree proper rational function of s with coefficients that are functions of w^o for $w^o \in \mathcal{O}$. Also, assume for $w^o \in \mathcal{O}$ the transfer functions $K_{w^o}^{11}(s)$ and $K_{w^o}^{22}(s)$ have no poles at $s = 0$, and $K_{w^o}^{12}(s)$ has neither poles nor transmission zeros at $s = 0$.

Requirement 1. There exist constant operating point functions $x_c^o : \mathcal{O} \rightarrow \mathbb{R}^{n_c}$, $x_I^o : \mathcal{O} \rightarrow \mathbb{R}^{n_I}$ such that the controller (3.5) satisfies

$$\begin{aligned} 0 &= a_1(x_c^o(w^o), x_I^o(w^o), y^o(w^o)), \\ 0 &= a_2(x_c^o(w^o), y^o(w^o)), \\ u^o(w^o) &= c(x_c^o(w^o), x_I^o(w^o), y^o(w^o)). \end{aligned} \tag{3.7}$$

Furthermore, the controller (3.5), when linearized at any constant operating point, is required to have the transfer function given by (3.6).

Theorem 3. (Existence Conditions)

With $n_I=m$ and under Assumptions 1 and 2

a) If there exists a function $x_I^o : \mathcal{O} \rightarrow \mathbb{R}^{n_I}$ satisfying

$$\frac{\partial u^o(w^o)}{\partial w^o} = \frac{\partial x_I^o(w^o)}{\partial w^o} + K_{w^o}^2(0) + K_{w^o}^3(0) \frac{\partial y_M^o(w^o)}{\partial w^o}, \quad w^o \in \mathcal{O}, \quad (3.8)$$

then there exists a controller (3.5) satisfying Requirement 1.

b) If there exists a controller (3.5) satisfying Requirement 1, and if $A_{11}(w^o)$ is invertible, $A_{12}(w^o)=0$, and $C_2(w^o)=I$, $w^o \in \mathcal{O}$, then the function $x_I^o(w^o)$ satisfies (3.8).

Proof: See [2].

The nonlinear controller construction is omitted in the general setting here (see [2] for details). A construction of the missile autopilot will be given in Chapter 4.

Chapter 4: Missile Autopilot Design

This chapter deals with the design of the missile autopilot for the hypothetical missile introduced in Chapter 2. The nonlinear dynamic autopilot must be constructed such that design objectives in Section 2.2 are met. To design the autopilot the four steps in Section 1.2 are followed. In Section 4.1, Step 1. is performed, which is the computation of the family of constant operating points. In Section 4.2 the family of linear controllers is designed, as in Step 2., to meet the existence conditions investigated in Chapter 3. Once the existence conditions are met, Step 3., which is the computation of the gain scheduled controller, can be performed in Section 4.3.

An important objective of this project is to investigate how the gain scheduling approach outlined in Section 1.2 leads to improved results compared to gain scheduling methods that deviate from this method. These issues will be addressed in Section 4.4 in which the gain scheduled missile autopilot is evaluated.

4.1 Computation of Family of Constant Operating Points

The first step in the design procedure is to compute the family of parameterized constant operating points. In order to perform the linear design and evaluation, closed form expressions for the constant operating point functions are not necessary. It is sufficient to obtain a table of parameterized operating points on a grid of the operating region. The operating region is parameterized by (η_c°, M°) . To be consistent with [5] the operating region is defined as having a trapezoid shape with the corners of the

trapezoid given to be $(\eta_c^\circ, M^\circ) = (10, 1.5), (33, 3), (-10, 1.5), (-33, 3)$. This is approximately equivalent to the operating region defined by parameterizing on (α°, M°) such that $-20^\circ \leq \alpha^\circ \leq 20^\circ$ and $1.5 \leq M^\circ \leq 3$. Figure 4.1 and Figure 4.2 illustrate the relationship between the two operating regions. Figure 4.1 shows a mesh plot of the constant operating point values of α° for the range of $-33 \leq \eta_c^\circ \leq 33$, and $1.5 \leq M^\circ \leq 3$. Figure 4.2 gives the plot of constant angle of attack contours corresponding to the plot in Figure 4.1. The similarity between the two operating regions can easily be seen by observing these plots. To build the required tables of constant operating point data, it is necessary to first show that the parameterized family of constant operating points exist.

The signals shown in Figure 3.1, as they apply to the missile problem are

$$w(t) = \begin{bmatrix} \eta_c(t) \\ M(t) \end{bmatrix}, \quad z(t) = \eta_e(t) = \eta_z(t) - \eta_c(t),$$

$$u(t) = \delta_c(t), \quad y(t) = \begin{bmatrix} \eta_e(t) \\ \eta_c(t) \\ M(t) \\ q(t) \end{bmatrix}.$$

Then, by observing the above configuration, the states for the missile can be written as

$$x(t) = \begin{bmatrix} \alpha(t) \\ q(t) \\ \delta(t) \\ \dot{\delta}(t) \end{bmatrix}.$$

In order to show that there exists a family of constant operating points for the missile problem all that is required is to verify the assumptions for Theorem 2. The first step is to show that $(x^o, w^o, u^o) = (0, [0 \ 1.5]^T, 0)$ is an equilibrium point of the system. By substituting

$$x^o = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad u^o = \delta_c^o = 0,$$

into equations (2.1) through (2.4) at

$$w^o = \begin{bmatrix} \eta_c^o \\ M^o \end{bmatrix} = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix},$$

it is clear that

$$\dot{x}(t) = 0, \quad \text{and} \quad z(t) = 0.$$

It is clear that the functions $f(x, w, u)$, and $h_1(x, w)$ in equations (3.1) are continuously differentiable for all $(x, w, u) \in \mathbf{R}^n \times \mathbf{R}^l \times \mathbf{R}^m$. Also, for $(0, [0 \ 1.5]^T, 0) \in \mathbf{R}^n \times \mathbf{R}^l \times \mathbf{R}^m$, $f(x^o, w^o, u^o) = 0$ and $h_1(x^o, w^o, u^o) = 0$. Hence the point $(x^o, w^o, u^o) = (0, [0 \ 1.5]^T, 0) \in \mathbf{R}^n \times \mathbf{R}^l \times \mathbf{R}^m$ is an equilibrium point. Now, we verify the remaining assumptions for Theorem 2. Since (x^o, w^o, u^o) is an equilibrium point, a linearization of the plant about this point can be computed numerically by using Matlab 4.0. To begin, equations (2.1) through (2.4) are implemented in a Simulink s-function (see Figure 4.4). There are

three inputs to the s-function, the control input $\delta_c(t)$, the exogenous input $M(t)$, and the reference input signal $\eta_c(t)$. The outputs for the s-function are the variables available for feedback, $\eta_z(t)$, $M(t)$ and $q(t)$, along with the tracking error signal $\eta_e(t)$. The Matlab command LINMOD is used to obtain state equations as in (3.3) for the operating point (x^o, w^o, u^o) . The RANK command is used to show that

$$\begin{bmatrix} \frac{\partial f}{\partial x}(x^o, w^o, u^o) & \frac{\partial f}{\partial u}(x^o, w^o, u^o) \\ \frac{\partial h_1}{\partial x}(x^o, w^o, u^o) & 0 \end{bmatrix} = \begin{bmatrix} F(0, [0 \ 1.5]^T, 0) & G_2(0, [0 \ 1.5]^T, 0) \\ H_1(0, [0 \ 1.5]^T, 0) & 0 \end{bmatrix},$$

has full rank. Finally, it is clear by comparing the dimensions of $\mathbf{R}^m$ and $\mathbf{R}^q$ that $m=q=1$. Therefore, by Theorem 3, in a region local to $(0, [0 \ 1.5]^T, 0) \in \mathbf{R}^n \times \mathbf{R}^1 \times \mathbf{R}^m$ the missile problem has a family of constant operating points corresponding to parameterization on $w^o = (\eta_c^o, M^o)$.

The tables of constant operating point data are computed using the Simulink command TRIM to solve for values of the control input and states necessary for the state derivatives to be equal to zero, while keeping the tracking error signal equal to zero for specified (η_c^o, M^o) . The TRIM command is executed at each point on the grid of operating points in the operating region starting at the point $(0, [0 \ 1.5]^T, 0)$. Once a solution is obtained at a point on the grid of η_c^o and M^o , that solution is used as the starting point for TRIM to solve for the next point on the grid. In this way tables of data for $\alpha^o(\eta_c^o, M^o)$, $\delta^o(\eta_c^o, M^o)$, $q^o(\eta_c^o, M^o)$ are obtained. The values of $\delta^o(\eta_c^o, M^o)$ are plotted in Figure 4.3. Figure 4.1, which was used to demonstrate the operating region, is a plot

for $\alpha^\circ(\eta_c^\circ, M^\circ)$. The plot for $q^\circ(\eta_c^\circ, M^\circ)$ is omitted because the other two plots are representative of the constant operating point functions. Once the constant operating point data is calculated, the data can be checked by evaluating the s-function at each point and checking that the state derivatives are sufficiently small. For the constant operating point data computed the largest state derivative is on the order of $1e-8$.

Figure 4.1 Constant operating point angle of attack.

Figure 4.2 Constant angle of attack contours.

Figure 4.3 Constant operating point tail deflection angle.

Figure 4.4 Simulink s-function of missile plant.

4.2 Design of the Family of Linear Controllers

In this section the family of linear controllers is designed as in Step 2. This is done by obtaining linear controllers at chosen design points and then interpolating. There is symmetry of the missile linearizations with respect to the sign of η_c° . Therefore, linear designs at operating points characterized by positive η_c can be used for the linear design points corresponding to negative η_c .

The first step in the linear design process is to obtain plant linearizations at chosen design points. Using the s-function in Figure 4.4, the constant operating point tables, and the Simulink command LINMOD, plant linearizations at four design points were obtained. These design points are $(\eta_c^\circ, M^\circ) = (0, 1.5)$, $(0, 3)$, $(10, 1.5)$, $(33, 3)$. These design points were chosen at the boundary of the operating region characterized by positive η_c so that six design points cover the entire operating region. As mentioned before, the linear designs for the points $(\eta_c^\circ, M^\circ) = (-10, 1.5)$ and $(\eta_c^\circ, M^\circ) = (-33, 3)$ are easily obtained from the linear designs at $(\eta_c^\circ, M^\circ) = (10, 1.5)$ and $(\eta_c^\circ, M^\circ) = (33, 3)$, respectively. Therefore, only four linear designs need to be obtained. Another reason for this choice of design points is that this missile problem was gain scheduled in [5] with a similar set of design points, therefore direct comparison between results is possible.

Once the linearized state equations are obtained, multi-input, multi-output transfer functions can be determined for the linearized airframe using the Matlab command SS2TF on appropriately partitioned matrices. For design purposes, only a two input, two output version of equation (3.4) is needed. The two inputs for this transfer function are $M(s)$ and $\delta_c(s)$, and the outputs are $\eta_z(s)$ and $q(s)$. To distinguish the transfer function

from the full three-input, five-output transfer function corresponding to (3.4), the transfer function matrix elements are denoted by g as opposed to G . The transfer function has the form

$$\begin{bmatrix} \eta_{z,\delta}(s) \\ q_\delta(s) \end{bmatrix} = \begin{bmatrix} g_{w^\circ}^{11}(s) & g_{w^\circ}^{12}(s) \\ g_{w^\circ}^{21}(s) & g_{w^\circ}^{22}(s) \end{bmatrix} \begin{bmatrix} M_\delta(s) \\ \delta_{c,\delta}(s) \end{bmatrix}. \quad (4.1)$$

The transfer function matrix elements are scalar transfer functions. For a given (η_c°, M°) the transfer function matrix elements in equation (4.1) corresponding to the input δ_c are the same for (η_c°, M°) as for $(-\eta_c^\circ, M^\circ)$. The transfer function matrix elements in equation (4.1) corresponding to the input M differ only by a sign between the points (η_c°, M°) and $(-\eta_c^\circ, M^\circ)$. For example

$$\begin{aligned} g_{[\eta_c^\circ, M^\circ]}^{11}(s) &= -g_{[-\eta_c^\circ, M^\circ]}^{11}(s), \\ g_{[\eta_c^\circ, M^\circ]}^{12}(s) &= g_{[-\eta_c^\circ, M^\circ]}^{12}(s). \end{aligned}$$

These transfer functions can be computed at any point on the grid in the operating region where tabular data for the constant operating point exists and therefore describe a family of linearized plants. These parameterized plant linearizations can be written in the form of a state equation as in equation (3.3) or as transfer functions as in equation (3.4).

The next step is to design the linear controller family. Figure 4.4 gives the structure of the linear controllers. From the figure it is clear that

$$\delta_{c,\delta}(s) = \begin{bmatrix} \frac{K_w^1(s)}{s} & K_w^2(s) & K_w^3(s) \end{bmatrix} \begin{bmatrix} \eta_{c,\delta}(s) - \eta_{z,\delta}(s) \\ M_\delta(s) \\ q_\delta(s) \end{bmatrix}. \quad (4.2)$$

The family of linear controllers must conform to the existence conditions that were

Figure 4.5 Linear controller structure.

developed in Chapter 3. In Section 3.4 the "integral filtered error" setup was presented because this choice leads to less complicated existence conditions. For the missile problem, with the "integral filtered error" setup equation (3.8) becomes

$$\begin{aligned} \delta_{c,\delta}(s) = & \frac{1}{s} \left[K_w^1(s) (\eta_{c,\delta}(s) - \eta_{z,\delta}(s)) \right] + K_w^2(s) M_\delta(s) \\ & + K_w^3(s) q_\delta(s). \end{aligned}$$

For the design in this project, the scalar compensator transfer functions in (4.2) are

$$K_{w^o}^1(s) = \frac{a_1(w^o)s + a_0(w^o)}{b_1(w^o)s + 1},$$

$$K_{w^o}^2(s) = K_M(w^o) \frac{s}{s+p},$$

$$K_{w^o}^3(s) = K \frac{c_1(w^o)s + 1}{d_1(w^o)s + 1}.$$

With the above choice, the family of linear controllers will be such that Assumption 2 holds. Assumption 1 was shown to hold in Section 4.1. By applying Theorem 3 we get

$$\begin{bmatrix} \frac{\partial \delta_c^o}{\partial \eta_c} & \frac{\partial \delta_c^o}{\partial M} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_I^o}{\partial \eta_c} & \frac{\partial x_I^o}{\partial M} \end{bmatrix} + K_{w^o}^2(0) \begin{bmatrix} 0 & 1 \end{bmatrix} + K_{w^o}^3(0) \begin{bmatrix} \frac{\partial q^o}{\partial \eta_c} & \frac{\partial q^o}{\partial M} \end{bmatrix},$$

or equivalently,

$$\begin{aligned} \frac{\partial \delta_c^o}{\partial \eta_c} &= \frac{\partial x_I^o}{\partial \eta_c} + K_{w^o}^3(0) \frac{\partial q^o}{\partial \eta_c}, \\ \frac{\partial \delta_c^o}{\partial M} &= \frac{\partial x_I^o}{\partial M} + K_{w^o}^2(0) + K_{w^o}^3(0) \frac{\partial q^o}{\partial M}. \end{aligned}$$

Since

$$K_{w^o}^2(0) = 0,$$

and

$$K_{w^o}^3(0) = K,$$

(4.3)

where K is a constant, the constant operating function

$$x_I^o(w^o) = \delta_c^o(w^o) - Kq^o(w^o), \quad (4.4)$$

satisfies equation (3.8). From Theorem 3, the existence of the function in (4.4) guarantees the existence of $x_c^o(w^o)$. This constant operating function will be determined when the missile autopilot is constructed in the next section.

With the above conditions in mind, controllers at the design points can be obtained. The process of designing the linear controller family is iterative. Once controllers have been designed that meet the performance criterion at the design points, the family of linear controllers is constructed by interpolating the design point controllers. Then the family of linear controllers is evaluated and adjustments are made if the family of linear controllers does not meet the performance criterion over the operating region. The controllers include an inner pitch rate compensator $K_{w^o}^3(s)$, which is implemented as a lag filter to increase the open loop attenuation for the loop broken at the input to the actuator, and to increase the damping of the dominant complex conjugate poles. To conform with the existence conditions the dc-gain of this filter, K , is chosen to be the same at each design point. By choosing the same dc-gain at the design points the interpolated value of K will be constant over the operating region. The lead filter, $K_{w^o}^1(s)$ is used to obtain the desired open loop gain crossover frequency and a phase margin of 70 degrees at each design point. This filter is designed using the analytical lead filter design technique in [4] pp. 366-69. In the design of the $(\eta_c^o, M^o) = (33, 3)$ design point it was found that the lead filter was not necessary to obtain the necessary phase margin at the desired crossover frequency. To abide by Assumption

2 the design point controllers must have the same degree. Therefore the coefficients of the lead filter at this design point are chosen so that the pole and zero cancelled. An integrator is included in the controllers to have zero steady state error to step inputs of the tracking variable η_c .

Once linear designs have been computed that satisfy (4.3), the family of linear controllers can be computed. Computation of the family of linear controllers is done by interpolating the dc-gains and either the poles and zeros of the controllers, or the polynomial coefficients of the controllers. The only requirement of the interpolation scheme is that the existence conditions need to be met at the design points and all intermediate operating points.

In order to obtain an equation for the linear interpolation, the operating region, which has the shape of a trapezoid, is mapped into a square shaped region. The mapping that will accomplish this is

$$s^o = \frac{M^o - 1.5}{1.5}$$

$$t^o = \frac{1.5 \eta_c^o}{15 + 23(M^o - 1.5)}.$$

After this mapping, the operating region for the parameterization variables (s^o, t^o) is defined as $0 \leq s^o \leq 1$ and $0 \leq t^o \leq 1$. For the (s^o, t^o) region, coefficients can be linearly interpolated in each variable. The result is then mapped back to the (η_c^o, M^o) region. Following this procedure, an equation for the linear interpolation can be obtained. For the coefficient a_1 , where $a_1(\eta_c^o, M^o)$ denotes the coefficient a_1 at the design point (η_c^o, M^o) ,

the equation used for interpolation is

$$\begin{aligned}
 a_1(\eta_c^o, M^o) = & a_1(0, 1.5) + \frac{1.5(a_1(10, 1.5) - a_1(0, 1.5)) \eta_c^o}{15 + 23(M^o - 1.5)} \\
 & + \frac{(a_1(0, 3) - a_1(0, 1.5))(M^o - 1.5)}{1.5} \\
 & + \frac{(a_1(33, 3) - a_1(10, 1.5) - a_1(0, 3) + a_1(0, 1.5))(M^o - 1.5) \eta_c^o}{15 + 23(M^o - 1.5)}.
 \end{aligned}$$

Using this equation the families of linear controllers $K_{w^o}^1(s)$ and $K_{w^o}^3(s)$ are constructed by interpolation.

The objective of the remaining filter $K_{w^o}^2(s)$ is to reduce the unwanted effects of mach variations. From Figure 4.14 we can see that the mach profile generated by the nonlinear simulation is ramp-like. Therefore, in order to improve the steady state error, we choose $K_M(w^o)$ so that the closed loop transfer function from $M(s)$ to $\eta_z(s)$ has two zeros at $s=0$ to reject ramp mach disturbance inputs in steady state. To write the closed loop transfer function from $M(s)$ to $\eta_z(s)$ the transfer function in equation (4.1) is expanded as

$$\begin{aligned}
 \eta_{z,\delta}(s) &= g_{w^o}^{11}(s) M_\delta(s) + g_{w^o}^{12}(s) \delta_{c,\delta}(s), \\
 q_\delta(s) &= g_{w^o}^{21}(s) M_\delta(s) + g_{w^o}^{22}(s) \delta_{c,\delta}(s).
 \end{aligned}$$

From equation (4.2), for $\eta_c(s)=0$

$$\delta_{c,\delta}(s) = K_{w^o}^2(s) M_\delta(s) + K_{w^o}^3(s) q_\delta(s) - \frac{K_{w^o}^1(s)}{s} \eta_{z,\delta}(s).$$

The three equations above can be solved to obtain the closed loop transfer function from $M_\delta(s)$ to $\eta_{z,\delta}(s)$ to get

$$\frac{\eta_{z,\delta}(s)}{M_\delta(s)} = \frac{s[g_w^{11}(s)(1 - g_w^{22}(s)K_w^3(s)) + g_w^{12}(s)(K_w^2(s) + g_w^{21}(s)K_w^3(s))]}{(1 - g_w^{22}(s)K_w^3(s))s + g_w^{12}(s)K_w^1(s)}. \quad (4.5)$$

It is required that the above transfer function have two zeros at $s=0$. This can be obtained by setting the dc-gain of the mach filter equal to

$$\frac{-1}{g_w^{12}(0)} [(g_w^{12}(0)g_w^{21}(0)K - g_w^{11}(0)g_w^{22}(0)K + g_w^{11}(0))]. \quad (4.6)$$

A table of $K_M(w^\circ)$ values is computed by obtaining a linearized missile transfer function at every constant operating point in the grid. This table will be implemented as a lookup table in the gain scheduled missile autopilot.

The family of linear controllers must have the dc-gains given in equation (4.3) hold throughout the operating region. This requirement for the dc-gains impacts the freedom allowed in the design of $K_w^2(s)$. To satisfy the first equation in (4.3), the mach filter, which consisted of only the gain determined in equation (4.6), is implemented as

$$K_w^2(s) = \frac{K_M(w^\circ)s}{s+p}, \quad (4.7)$$

where for small values of p the frequency response is approximately equal to the gain $K_M(w^\circ)$. One problem with this setup is that there are no longer two zeros at $s=0$ for the closed loop transfer function from $M(s)$ to $\eta_z(s)$. This means that there will be a steady state error caused by the ramp like input $M(t)$. From the final value theorem of

the Laplace transform and equations (4.5) and (4.7) this error is given to be

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \frac{\eta_z(s)}{M(s)} \cdot \frac{1}{s^2} \\ &= g_{w^o}^{11}(0) - g_{w^o}^{11}(0)g_{w^o}^{22}(0)K_{w^o}^3(0) + g_{w^o}^{12}(0)g_{w^o}^{21}(0)K_{w^o}^3(0). \end{aligned}$$

The linear designs whose coefficients are given in Table 4.1 were obtained by the procedure mentioned above. These coefficients are interpolated to obtain the family of linear controllers that will be gain scheduled in the next section. For each design point the performance is verified by plotting a closed loop step response, and a bode plot for the loop opened at the actuator input. Plots for the design point $(\eta_c^o, M^o) = (0, 3)$ are given in Figure 4.6 and Figure 4.7. Since the performance criteria are met at the design points, the family of linear controllers can be evaluated. The first step in evaluating this family of linear controllers is to show that the controller stabilizes the plant at each operating point in the operating region. Figure 4.8 is a plot of the maximum real part eigenvalue for the closed loop system. For each operating point, where constant operating point data exists, a plant linearization and an interpolated controller is obtained. The closed loop transfer function is then formed. The largest real part closed loop eigenvalues are shown in the plot. The plot proves that the linear controller family stabilizes the plant over the operating region.

The next step in evaluating the family of linear controllers is to form a mesh plot of the minimum return difference. The minimum return difference is the minimum value of $|1 + G(j\omega)|$ where $G(j\omega)$ denotes the open loop transfer function with the pitch rate loop closed. This transfer function is obtained by first setting the inputs of the system

equal to zero. Then the outer feedback loop shown in Figure 4.4 is broken. The transfer function $G(j\omega)$ is the transfer function from the open back to the open. The minimum return difference is a rough measure of the transient response of the system. We can expect the worst transient response at an operating point to correspond to the smallest minimum return difference. The mesh plot for the family of linear controllers is given in Figure 4.9. The smallest minimum return difference shown in this graph is at the operating points $(\eta_c^\circ, M^\circ) = (\pm 27.525, 2.55)$. The step response for this operating point is given in Figure 4.10. This operating point also has one of the worst percentage overshoots for a step response.

At the same time the plot of minimum return difference is obtained, mesh plots of maximum percentage overshoot, rise time, settling time, and attenuation for the system opened at the actuator input are also obtained. The plot of rise time is given in Figure 4.11. These plots are used to verify that the design objectives listed in Section 2.2 are met for the family of linear controllers.

Finally, step responses are obtained at various points in the operating region in order to qualitatively judge the performance of the family of linear autopilots. We conclude that the family of linear autopilots adequately meets all of the performance objectives, therefore the nonlinear controller can be constructed.

Design Point $(\eta_c^\circ, M^\circ) = (0, 1.5)$	
$a_0 = 1.3871e1$	$c_1 = 4.6406e-1$
$a_1 = 2.1569e0$	$d_1 = 2.3203e0$
$b_1 = 7.5919e-3$	$K = 2.0$
Design Point $(\eta_c^\circ, M^\circ) = (0, 3)$	
$a_0 = 3.8741e0$	$c_1 = 1.2964e-1$
$a_1 = 3.2156e-1$	$d_1 = 1.579e0$
$b_1 = 8.4815e-3$	$K = 2.0$
Design Point $(\eta_c^\circ, M^\circ) = (10, 1.5)$	
$a_0 = 2.2734e1$	$c_1 = 5.5638e-1$
$a_1 = 1.5585e0$	$d_1 = 1.6860e0$
$b_1 = 2.7577e-2$	$K = 2.0$
Design Point $(\eta_c^\circ, M^\circ) = (33, 3)$	
$a_0 = 1.0377e1$	$c_1 = 1.4031e-1$
$a_1 = 4.3363e-1$	$d_1 = 9.6100e-1$
$b_1 = 4.1789e-2$	$K = 2.0$

Table 4.1 Design point controller coefficients.

Figure 4.6 Step response at design point $(\eta_c^\circ, M^\circ) = (0, 3)$.

Figure 4.7 Bode diagram with the loop open at the actuator input.

Figure 4.8 Mesh plot of maximum real part eigenvalues.

Figure 4.9 Mesh plot of minimum return difference.

Figure 4.10 Step response at operating point with lowest MRD.

Figure 4.11 Mesh plot of rise time.

4.3 A Gain Scheduled Missile Autopilot

The third design step in Section 1.2 is to compute a gain scheduled missile autopilot that linearizes to the correct linear controller at each constant operating point. In order for this to be accomplished, the existence conditions given in Theorem 3 must be met, in which case there exists a gain scheduled autopilot that linearizes to the correct linear autopilot at each operating point.

In Section 4.2 the existence conditions were investigated and the required form for the constant operating function $x_i^o(w^o)$ was found to be

$$x_i^o(w^o) = \delta_c^o(w^o) - Kq^o(w^o),$$

To find the function $x_c^o(w^o)$, the parameterized state space equation is formed as specified in Theorem 3b). The form needed for the parameterized state space equation is $A_{11}(w^o)$ invertible, $A_{12}(w^o)=0$, and $C_2(w^o)=1$. A parameterized state space equation satisfying

these requirements is

$$\begin{aligned}
 \dot{x}_{c\delta}(t) &= \begin{bmatrix} -\frac{1}{b_1(w^o)} & 0 & 0 \\ 0 & -\frac{1}{d_1(w^o)} & 0 \\ 0 & 0 & -p \end{bmatrix} x_{c\delta}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} x_{I\delta}(t) \\
 &+ \begin{bmatrix} \frac{1}{b_1(w^o)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{K}{d_1(w^o)} \\ 0 & 0 & p & 0 \end{bmatrix} y_{\delta}(t), \\
 \dot{x}_{I\delta}(t) &= \left[\left(a_0(w^o) - \frac{a_1(w^o)}{b_1(w^o)} \right) \ 0 \ 0 \right] x_{c\delta} + \begin{bmatrix} \frac{a_1(w^o)}{b_1(w^o)} & 0 & 0 & 0 \end{bmatrix} y_{\delta}(t), \\
 \delta_{c\delta}(t) &= \begin{bmatrix} 0 & \left(1 - \frac{c^1(w^o)}{d_1(w^o)} \right) & -K_M(w^o)p \end{bmatrix} x_{c\delta}(t) + x_{I\delta}(t) \\
 &+ \begin{bmatrix} 0 & 0 & K_M(w^o)p & \frac{Kc_1(w^o)}{d_1(w^o)} \end{bmatrix} y_{\delta}(t).
 \end{aligned}$$

By taking partial derivatives of equation (3.7) with respect to w^o , the following equations can be obtained

$$\begin{aligned}
 A_{11}(w^o) \frac{\partial x_c^o(w^o)}{\partial w^o} &= -B_1(w^o) \frac{\partial y^o(w^o)}{\partial w^o}, \\
 A_{12}(w^o) \frac{\partial x_c^o(w^o)}{\partial w^o} &= -B_2(w^o) \frac{\partial y^o(w^o)}{\partial w^o}, \\
 C_1(w^o) \frac{\partial x_c^o(w^o)}{\partial w^o} + \frac{\partial x_I^o(w^o)}{\partial w^o} &= \frac{\partial u^o(w^o)}{\partial w^o} - D(w^o) \frac{\partial y^o(w^o)}{\partial w^o}
 \end{aligned} \tag{4.8}$$

By substituting (4.4), the equations in (4.8) yield

$$\frac{\partial x_c^o(w^o)}{\partial w^o} = \begin{bmatrix} 0 & 0 \\ K \frac{\partial q^o(w^o)}{\partial \eta_c^o} & K \frac{\partial q^o(w^o)}{\partial M^o} \\ 0 & 1 \end{bmatrix},$$

for which the obvious solution for $x_c^o(w^o)$ is

$$x_c^o(w^o) = \begin{bmatrix} 0 \\ K q^o(w^o) \\ M^o(w^o) \end{bmatrix}.$$

The functions $x_c^o(w^o)$ and $x_I^o(w^o)$ can be substituted into

$$\begin{aligned} a_1(x_c, x_P, w, y) &= A_{11}(w)[x_c - x_c^o(w)] + A_{12}(w)[x_I - x_I^o(w)] \\ &\quad + B_1(w)[y - y^o(w)] \\ a_2(x_c, w, y) &= A_{21}(w)[x_c - x_c^o(w)] + B_2(w)[y - y^o(w)] \\ c(x_c, x_P, w, y) &= C_1(w)[x_c - x_c^o(w)] + C_2(w)[x_I - x_I^o(w)] \\ &\quad + D_1(w)[y - y^o(w)] + u^o(w). \end{aligned}$$

which reduces to

$$\begin{aligned}
 \dot{x}_c &= \begin{bmatrix} -\frac{1}{b_1(\eta, M)} & 0 & 0 \\ 0 & -\frac{1}{d_1(\eta, M)} & 0 \\ 0 & 0 & -p \end{bmatrix} x_c + \begin{bmatrix} \frac{1}{b_1(\eta, M)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{K}{d_1(\eta, M)} \\ 0 & 0 & p & 0 \end{bmatrix} y, \\
 \dot{x}_I &= \left[\left(a_0(\eta, M) - \frac{a_1(\eta, M)}{b_1(\eta, M)} \right) \ 0 \ 0 \right] x_c + \left[\frac{a_1(\eta, M)}{b_1(\eta, M)} \ 0 \ 0 \ 0 \right] y, \\
 \delta_c &= \left[0 \ \left(1 - \frac{c_1(\eta, M)}{d_1(\eta, M)} \right) \ -K_M(\eta, M)p \right] x_c + x_I \\
 &\quad + \left[0 \ 0 \ K_M(\eta, M)p \ \frac{Kc_1(\eta, M)}{d_1(\eta, M)} \right] y,
 \end{aligned} \tag{4.9}$$

where $\eta = \eta_c$ or η_z .

In the next section the nonlinear autopilot in this equation is implemented and evaluated.

4.4 Evaluation of the Gain Scheduled Missile Autopilot

In this section the nonlinear controller obtained in Section 4.3 is implemented and then evaluated by simulation. Through simulation, the nonlinear system results are shown to be predicted by the linear system results seen in Section 4.2, provided that the existence conditions are met. Finally, two simulations are presented whose linear controllers do not meet the existence conditions necessary for the nonlinear autopilot to linearize to the correct linear autopilot.

We expect the nonlinear simulations to be stable for slowly varying inputs because of the following theorem developed in [3].

Theorem 4. If

- (i) The closed-loop system of the missile and autopilot is asymptotically stable at each operating point,
- (ii) The exogenous signals $w(t)=(\eta_c(t),M(t))$ are sufficiently slowly-varying,

Then if the closed-loop system trajectory starts near the surface of constant operating points it will remain near the surface of constant operating points.

Further details and the proof of this theorem are given in [3].

To implement the nonlinear controller a Simulink block diagram of the controller and missile airframe is constructed. The block diagram is shown in Figure 4.12, with the controller parameterized on $\eta_c(t)$. Since there is an integrator in the controller, at a constant operating point $\eta_c^\circ = \eta_z^\circ$. Therefore, the system can be parameterized on $\eta_z(t)$, which is a measurable output as well. In Figure 4.12 there is a block labeled Airframe. This is a Simulink subsystem block which is the same as the s-function given in

Figure 4.4, except that an additional output, $\alpha(t)$ is needed in order for the mach profile to be generated. In this simulation the mach profile is generated by,

$$\dot{M}(t) = \frac{32.2}{v_s} [-|\eta_z(t)| \sin |\alpha(t)| + A_x M^2(t) \cos \alpha(t)],$$

$$M(0) = M_0.$$

So, strictly speaking, $M(t)$ is an endogenous variable. These mach dynamics are written in a Simulink mex-file. A mex-file is a Matlab m-function that is written in C. This mex-file is shown as the mmex block in the figure.

The nonlinear controller is implemented in two separate parts. The mex-file *controllermex* contains the portions of the nonlinear controller corresponding to $K_{w^o}^1(s)$ and $K_{w^o}^3(s)$ of the family of linear controllers. The inputs to this controller are the output of the airframe s-function in Figure 4.4, which are all measurable outputs. This function realizes the equations in (4.9). The second part of the nonlinear controller corresponds to $K_{w^o}^2(s)$ of the family of linear controllers. This controller is separate from the *controllermex* controller because it incorporates a lookup table for the gain $K_M(w^o)$. A mesh plot of the $K_M(w^o)$ values at the operating points is given in Figure 4.13. The absolute value of K_M is the same for points $(\pm \eta_c(t), M(t))$. Therefore, values of K_M for the constant operating points with positive η_c are given in the lookup table and the sign of K_M is determined by the Matlab function block. The state space equation block is used to realize the pole and zero of equation (4.7). The signals leaving

these three blocks are multiplied together resulting in, with $h_p(t) = \mathcal{L}^{-1}[s/(s+p)]$,

$$\delta_2(t) = \text{sgn}(\eta(t)) K_M(|\eta(t)|, M(t)) [h_p(t) * M(t)],$$

where η can be either η_c or η_z .

Using the block diagram in Figure 4.12 and Simulink's simulation routine RK45, the simulation in [5] is reproduced using the nonlinear controller designed in this thesis. As mentioned before, the controller can be parameterized on $\eta_c(t)$ or $\eta_z(t)$. In fact, a convex combination of $\eta_c(t)$ and $\eta_z(t)$ can be used to schedule the controller. Since there are two distinct sections to the controller in this simulation another possible parameterization method is to schedule on a different signal for each section of the controller. In order to investigate the effect of different choices in scheduling variables, four cases of the simulation are presented. These cases correspond to the four combinations possible by parameterizing the mach controller on $\eta_c(t)$ or $\eta_z(t)$ and the *controllermex* portion of the controller by $\eta_c(t)$ or $\eta_z(t)$. These combinations are given in Table 4.2, which lists the simulation number and corresponding parameterization. The simulation results are given in Figure 4.15 through Figure 4.18. The mach profile is shown in Figure 4.14 for Simulation # 2. Figure 4.19 is a plot of $\eta_z(t)$ vs. $M(t)$. This plot shows where in the operating region the system is located.

These simulations show that the undershoot is consistently larger at $t=1$ and $t=3$ compared to $t=0$ and $t=2$. This can be explained by looking at step responses of plant linearizations at the operating points corresponding to the "frozen" values of the scheduling variables at this time. Figure 4.20 is a plot of the step response for Simulation # 2 for linearizations at $(\eta_c(t), M(t)) = (-14.7, 2.6)$ and $(\eta_c(t), M(t)) = (9.98,$

2.5). The plot shows step responses for plant linearizations and linearly interpolated controllers corresponding to the times $t=2$ and $t=3$. The linear performance corresponds to the performance shown in the nonlinear simulation. The same results can be seen at linearizations at $t=0$ and $t=1$.

Simulation #	Mach controller parameterization	<i>Controllermex</i> parameterization	Comments
1	$\eta_c(t)$	$\eta_c(t)$	
2	$\eta_c(t)$	$\eta_z(t)$	
3	$\eta_z(t)$	$\eta_c(t)$	
4	$\eta_z(t)$	$\eta_z(t)$	
5	$\eta_c(t)$	$\eta_z(t)$	Existence condition for K^3 violated.
6	$\eta_c(t)$	$\eta_z(t)$	Existence condition for K^2 violated.

Table 4.2 Simulation parameterizations.

The simulations also show parameterizing the mach controller on $\eta_z(t)$ yields less undershoot than parameterizing on $\eta_c(t)$. However, parameterizing on $\eta_z(t)$ yields unacceptable overshoot which can be seen in Simulation # 3 and Simulation # 4. The system also seems to take longer to reach steady state. Therefore, parameterizing the mach controller on $\eta_c(t)$ is used for the remainder of this project.

One might conclude from Simulation # 1 and Simulation # 4 that parameterizing the *controllermex* controller on $\eta_c(t)$ yields a faster rise time of the system. This conclusion comes from the first step in each of these simulations. This first step has such a fast rise time because the controller that is used throughout this step is at points close to the $(\eta_c^\circ, M^\circ) = (33, 3)$ design point. For step commands in $\eta_c(t)$ which are not in

regions close to the design points, parameterizing on $\eta_z(t)$ actually gives improved performance (see Figure 4.21). The improvements shown in this simulation include less overshoot, less undershoot, and faster rise times.

Using Simulation # 2 as the case to verify the performance objective outlined in Section 2.2 we can see that all of the objectives have been met. The time constants are less than .35 seconds, there is almost no overshoot, and the steady state error is within 1%. A plot of the robustness analysis for the parameterization of Simulation # 2 is shown in Figure 4.24. The plot shows the simulation results for four cases where the alpha and delta coefficients are each varied by $\pm 25\%$. The system is clearly stable for each case in this analysis.

Now, two simulations are presented which demonstrate the importance of meeting the existence conditions given in (4.3). The first example investigates the effect of interpolating the poles, zeros, and high frequency gains of $K_w^3(s)$. Because of this interpolation method the dc-gain $K_w^3(0)$ will not be constant at intermediate operating points. This violates the existence condition which specifies the dc-gains for this controller must be constant through the operating region. If we ignore this requirement a family of linearized controllers can be obtained by interpolating poles, zeros, and the high frequency gains. By using this interpolation scheme the dc-gain of $K_w^3(s)$ is no longer constant. A mesh plot of this dc-gain is given in Figure 4.23. By comparing the performance of the family of linear controllers with this interpolation to the family of linear controllers obtained from the interpolation used in Simulation # 2 it is found that interpolating poles, zeros, and high frequency gains leads to improved transient

responses. There is improvement in rise time, settling time, and percentage overshoot for most of the operating region. Figure 4.22 shows the step response at the operating point $(\eta_c^\circ, M^\circ) = (10.75, 2.25)$ which is the middle of the operating region, compared to the step response at the same point for Simulation # 2. One might expect nonlinear simulations with better transient responses than that of Simulation # 2. Figure 4.25 shows the performance of the incorrectly interpolated nonlinear controller plotted against the performance of the nonlinear controller in Simulation # 2. Even though the linear performance of Simulation # 2 is worse the nonlinear simulation shows better performance. There are several effects caused by the failure to meet the existence conditions. First notice the shape of the step response for the nonlinear simulation with incorrect interpolation is not the same as the step response for the corresponding family of linear controllers, while the shape of the family of linear controllers that is correctly interpolated is reflected in the nonlinear simulation. Another effect of the failure to meet the existence conditions is shown when the system is in the region corresponding to the highest dc-gain shown in Figure 4.23. From this figure the point $(\eta_c^\circ, M^\circ) = (10, 2.2)$ is found to have the highest deviation of the dc-gain from the designed value. The nonlinear simulation shows that the performance is the worst in this region. It is interesting to note that the steady state performance of this interpolation scheme is not affected because at steady state the deviation variables approach zero and therefore are not affected by the incorrect gain.

The second example of a failure to meet existence conditions is shown by removing the pole and zero in the state space to transfer function block of the mach

filter. This causes the dc-gain $K_{w.o}^2(0)$ to be nonzero which also violates the existence conditions. One would expect this to have little impact on the simulation because this pole and zero are very close to each other. Figure 4.26 shows that this is not the case. The effect of this failure to meet the existence conditions is more drastic than the above example. In this case a cross-coupling term is introduced that is driven by the deviation variable $M_\delta(t)$. We conclude that nonlinear simulation performance can be predicted by linear simulation results only if the existence conditions are met.

Figure 4.15 Simulation # 1, ($\eta_c(0) = 0$, $M(0) = 3$).

Figure 4.16 Simulation # 2, ($\eta_c(0) = 0$, $M(0) = 3$).

Figure 4.17 Simulation # 3, ($\eta_c(0) = 0$, $M(0) = 3$).

Figure 4.18 Simulation # 4, ($\eta_c(0) = 0$, $M(0) = 3$).

Figure 4.19 Plot of $\eta_z(t)$ vs. $M(t)$, Simulation #2, ($\eta_c(0) = 0$, $M(0) = 3$).

Figure 4.20 Step responses demonstrating undershoot, Simulation #2, ($\eta_c(0) = 0$, $M(0) = 3$).

Figure 4.21 Improved performance by parameterizing *controllermex* controller on $\eta_z(t)$, ($\eta_c(0) = 0$, $M(0) = 2.6$).

Figure 4.22 Step responses at $(\eta_c^o, M^o) = (10.75, 2.25)$ comparing linear performance.

Figure 4.23 Dc-gains for incorrect interpolation.

Figure 4.24 Robustness analysis for Simulation # 2, $(\eta_c(0) = 0, M(0) = 3)$.

Figure 4.25 Simulation results comparing incorrect interpolation to Simulation # 2, ($\eta_c(0) = 0$, $M(0) = 2.6$).

Figure 4.26 Simulation results for $K_M(0) \neq 0$, ($\eta_c(0) = 0$, $M(0) = 3$).

Chapter 5: Conclusions

In this thesis a missile autopilot was designed for a hypothetical pitch-axis missile model. Chapter 2 gave an explanation of the missile's dynamics as well as the equations governing the dynamics. In Chapter 3, the theory needed to correctly compute a gain scheduled controller was presented. In Chapter 4, the steps outlined in Section 1.2 were followed in order to design the autopilot. First the family of constant operating points was computed by using the Simulink TRIM command. Once this was done the family of plant linearizations was determined. The next step was to design controllers for the four chosen design points. These controllers were designed such that the existence conditions of the resulting family of linear controllers were met. At this point the performance of the family of linear controllers was evaluated. After the family of linear controllers was shown to meet the performance objectives the nonlinear controller was obtained. Simulations of the nonlinear controller showed that the performance of the family of linear controllers was reflected in the nonlinear controller performance as long as the existence conditions were met. Examples of nonlinear controllers that did not meet the existence conditions were obtained by incorrect interpolation, or incorrect choice of linear controller structure. These controllers showed poor performance, which was not predicted by the performance of the corresponding family of linear controllers.

The nonlinear controller of Simulation # 2 gave performance that met all of the design objectives. The only problem with this simulation was shown to be excessive undershoot.

There are several areas where further research is necessary. This method of gain scheduling could be applied to other problems, for example a twin-jet transport aircraft model that is available in the department. Another area where further study is needed is the method used to obtain the family of linear controllers. A more systematic method of designing the family of linear controllers as opposed to the ad-hoc method of interpolating controllers at distinct operating points is of interest. Finally, further research is required to apply this gain scheduling method to sampled-data systems. This involves the computation of constant operating point families for hybrid continuous/discrete-time systems. Also formulation of existence conditions for gain scheduled sampled-data controllers is necessary. Finally the implementation of the nonlinear controller for these systems needs to be addressed. This will involve stability analysis of gain scheduled controllers for sampled-data systems with slowly varying inputs.

References

- [1] F. M. Calier and C. A. Desoer. *Linear System Theory*. Springer-Verlag, New York, 1991.
- [2] D. A. Lawrence and W. J. Rugh. "Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant". *Proceedings on the 32nd IEEE Conference on Decision and Control*, pp.1024-1029, San Antonio, Texas, 1993.
- [3] D. A. Lawrence and W. J. Rugh. "On a Stability Theorem for Nonlinear Systems with Slowly Varying Inputs". *IEEE Transactions on Automatic Control*, 35(7):860-864, 1990.
- [4] C. L. Phillips and R. D. Harbor. *Feedback Control Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [5] R. T. Reichert, R. A. Nichols, and W. J. Rugh. "Gain Scheduling for H-Infinity Controllers: A Flight Control Example". *IEEE Transactions on Control System Technology*, 1(2): 69-79, 1993.
- [6] J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [7] W. F. Trench. *Advanced Calculus*. Harper and Row, New York, 1978.
- [8] J. Wang and W. J. Rugh. "Parameterized Linear Systems and Linearization Families for Nonlinear Systems". *IEEE Transactions on Circuits and Systems*, 34(6): 650-657, 1987.