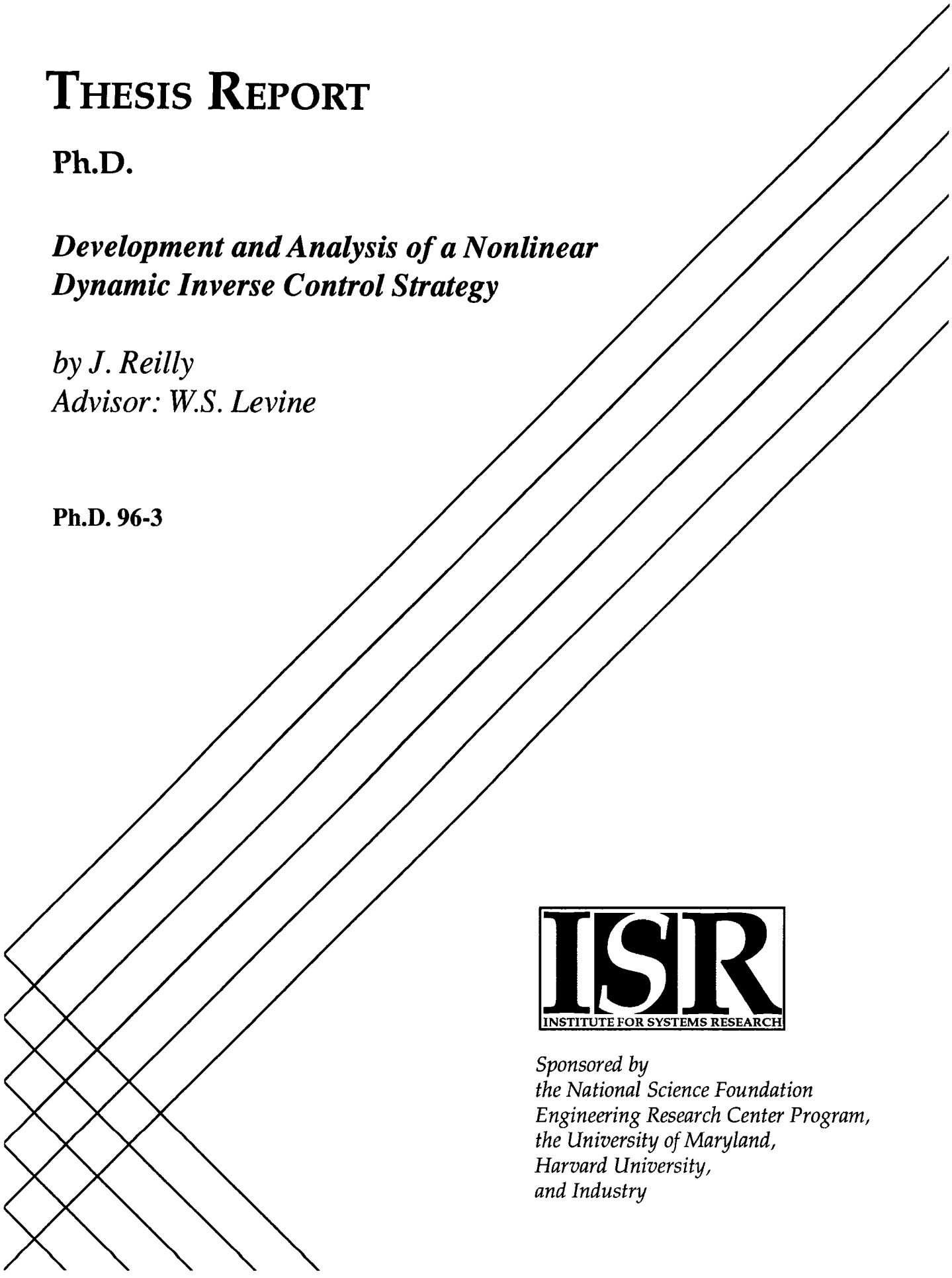
THESIS REPORT

Ph.D.

Development and Analysis of a Nonlinear Dynamic Inverse Control Strategy

by J. Reilly
Advisor: W.S. Levine

Ph.D. 96-3



Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Dedication

This dissertation is dedicated to my father, who sadly is not with us to share this happy occasion.

Acknowledgements

First,I would like to sincerely thank my advisor, Dr. W. S. Levine for his guidance and patience throughout this research. His insight in a broad array of topics has been invaluable.

I would also like to thank Dr. W. P. Dayawansa for taking the time to help me better understand nonlinear control theory.

I would like to thank Dr. Chien Huang and Dr. Gareth Knowles for the opportunity to begin this research while spending three summers at Grumman Aerospace. I look forward to collaboration with them in the future.

I would also like to thank the other committee members, Dr. G. Blankenship, Dr. E. Abed, and Dr. D. K. Schmidt, for their time and helpful comments.

I would like to thank all my friends and fellow students for helping to make graduate school fun and for their help and support.

Last, but not least, I would like to thank my loving wife Cindy, for her encouragement, understanding, and patience while this dissertation was completed.

This work was supported in part by the Institute for Systems Research.

Table of Contents

Section	Page
List of Tables	vi
List of Figures	vii
1 Introduction	1
2 Background	5
2.1 Aircraft Dynamics	5
2.1.1 Definition of Axes and Angles	7
2.1.2 Definition of Forces and Moments	11
2.1.3 The Nonlinear Aircraft Model	14
2.2 Feedback Linearization	19
2.2.1 Mathematical Preliminaries	20
2.2.2 Single-Input Single-Output Systems	23
2.2.3 Multi-Input Multi-Output Systems	26
2.3 Dynamic Inversion Applied to Aircraft	34
2.4 Robust Feedback Linearization	59

3 Dynamic Inversion Algorithms **65**

3.1 Implementation 83

3.2 Sensitivity 87

4 Application of Algorithm to X-29 Aircraft example **97**

4.1 X-29 Model 97

4.2 Simulation 104

4.2.1 Integral Control 117

4.3 Robust Control Law 118

4.3.1 Integral Robust Control 128

4.3.2 Implementation for limited control effectors 134

4.3.3 Complete implementation 143

5 Conclusions and Future Directions **148**

List of Tables

Number		Page
3.1	Sensitivity function using J_2.	96
3.2	Sensitivity function using J_3.	96

List of Figures

Number		Page
2.1	Diagram depicting axis definitions, moments and angular rates of the aircraft.	6
2.2	Simplified arrangement of the controller proposed by Meyer and Cicolani.	35
2.3	Total aircraft flight-control system proposed by Meyer and Cicolani.	36
2.4	Block diagram showing computational steps used by Kato and Sugiura to solve the airplane general motion inverse problem.	43
2.5	Dynamic inversion control law of Snell, Enns, and Garrard.	47
2.6	Dynamic Inverse controller used on the HARV.	55
3.1	Performance of linear X-29 simulation with LQR model-following controller.	67
3.2	Performance of X-29 with LQR model-following controller.	68
3.3	Comparison of LQR model-following error.	69
3.4	Comparison of states with LQR model-following controller.	71
3.5	Performance of X-29 with nonlinear model-following controller.	72
3.6	States for the nonlinear model-following controller.	73

3.7 Nonlinear dynamic inverse model-following controller 75
3.8 Pitch command following using Newton-Raphson inversion algorithm. 78
3.9 Roll command following using Newton-Raphson inversion algorithm. 79
3.10 Comparison of the error between the three control methods. . . . 86
3.11 Comparison of the error when uncertainty is present. 87
3.12 Sensitivity function using calculated z and Runga-Kutta integration. 92
3.13 Sensitivity function using calculated z and Euler integration. . . . 93
3.14 Sensitivity function using $z = \dot{x}_2$ and Runga-Kutta integration. . . 93

4.1 A top view of the X-29 aircraft displaying the flight control surfaces. 98
4.2 Angle-of-attack and sideslip envelope of the X-29 aircraft. 99
4.3 Inverse dynamics controller. 108
4.4 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft. 114
4.5 Canard-strake interference factor for the pitch up maneuver. . . . 115
4.6 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.6 and 15,000 ft. 116
4.7 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft using the robust controller. 127
4.8 Model-following error for a pitch up maneuver at Mach 0.9 and 30,000 ft using the robust controller. 128
4.9 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft using the robust controller with integral action. . 133
4.10 Comparison of the model-following error for the robust controller with integral action. 134

4.11 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft using the rate-limited robust controller with integral action. 139
4.12 Comparison of the model-following error for the robust controller. 139
4.13 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.6 and 15,000 ft using the rate-limited robust controller with integral action. 140
4.14 Model-following error for the robust controller at Mach 0.6 at 15,000 ft. 140
4.15 Simulation of the X-29 aircraft in a pitch up maneuver at Mach 1.2 and 15,000 ft using the rate-limited robust controller with integral action. 141
4.16 Model-following error for the robust controller at Mach 1.2 at 15,000 ft. 142
4.17 Normal acceleration of the X-29 at Mach 1.2 at 15,000 ft. 142
4.18 Plot of the high angle-of-attack contributions to D_2. 146
4.19 Plot of the incremental control contributions to D_2. 146
4.20 Plot of the high angle-of-attack pitch contributions to D_2. 147

Development and Analysis of a Nonlinear Dynamic Inverse Control Strategy

John Reilly

1996

Chapter 1

Introduction

Future aircraft are expected to meet higher performance and combat superiority requirements characterized as super-agility or supermaneuverability. Supermaneuverability calls for an aircraft to perform maneuvers in post-stall or high-angle-of-attack (or high-α) flight regimes. These maneuvers would enable an aircraft to gain advantage over its opponents by quick deceleration and attitude pointing [1]. However, problems, such as loss of control effectiveness and nonlinear dynamics [2], are associated with high-α maneuvers. They are primarily caused by unsteady aerodynamics, which can lead to wing rock, which is a coupling between longitudinal and lateral motions, aerodynamic jump, where (roll-axis) aerodynamic coefficients undergo sudden changes in value, and stall-spin, which is induced by being near the stall region. To address the problems at such extreme flight conditions, innovative controls, such as thrust vectoring, or novel control strategies, such as nonlinear control, may be required.

Aircraft control traditionally has relied on gain-scheduling of linear control laws (which essentially becomes a nonlinear control), an approach that has been

very successful. One can characterize nonlinear dynamics in high-alpha maneuvers by using a large number of linear models, but this method may not be adequate, especially since some of the chosen operating points may not be equilibrium points. One can build robustness into the control strategy (e.g., by requiring more gain and phase margins), but this approach translates to a more conservative control, which means possible performance degradation. An alternative method is to take explicitly into account the nonlinearities in the control design, thus better utilizing the existing dynamics and control power. We chose to investigate the dynamic inverse control technique because of its ease of implementation, and the simple way that maneuvers can enter into the control. The objective of nonlinear dynamic inversion is to invert the dynamic equations of the plant directly in order to find the control necessary to yield the given output.

We began work on this project because of a desire by Grumman Aerospace Corp. to investigate alternative methods of control for the X-29 forward-swept-wing research aircraft. The current method of control was to use gain-scheduled control laws. This required choosing many operating points about the flight envelope, designing linear controllers for each operating point, and blending them together smoothly such that stability was achieved. This was a long and sometimes difficult process (decisions needed to be made about what variable to schedule with, and how). Using a nonlinear control method we could (hopefully) design one global controller. Stability would also be achieved across a larger (hopefully global) region. We want the controller design process to be an algorithmic one, similar to the process used for linear systems. Since we are dealing with aerodynamics derived from wind tunnel testing and hence somewhat prone to error, we want the design to be robust. We first designed a dynamic

inverse controller based on the Newton-Raphson method since an analytic form for the plant was not available, only aerodynamic lookup tables. Performance was not acceptable because low gains had to be used in order to achieve stability. Stability guarantees could not be given. We then formulated a reduced analytic model in order to use feedback linearization techniques to design the controller. The controller was then modified so that stability robustness to modeling errors could be achieved.

Our contribution is the elaboration of the dynamic inverse methods first described by Meyer and Cicolani [3]. We expand the method to a more complex aerodynamics and airframe description, that of the full nonlinear simulation (wind-tunnel and flight tested model) of the X-29. We perform an engineering application of recent robust feedback linearization results to a high fidelity simulation of the X-29 at high-angles-of-attack. We reduced the model of the X-29 to a level sufficient to formulate the control. However, we have included actuator redundancy and actuator limits. In addition, we modified the robust control method to add integral action to enable the controller to reduce steady state errors.

Chapter 2 outlines the background material for nonlinear aircraft control. We first describe the terminology used in the aircraft dynamic equations. Next, we describe the feedback linearization control method. We then outline dynamic inverse control methods in the literature applied to aircraft. Robust feedback linearization methods are then discussed. Chapter 3 describes our first dynamic inverse control design for the X-29. We use a two time scale approach to separate the control design into two steps to enable the inversion to be performed easily. We then describe a more analytical control method which gives better

performance and stability guarantees. Chapter 4 discusses the X-29 model in more detail and describes the controller design and simulation. We then discuss the robust control additions needed to guarantee stability in the face of modeling errors. New simulation results are then shown. Finally, Chapter 5 gives conclusions and discussions for further research.

Chapter 2

Background

2.1 Aircraft Dynamics

Since most of the work we will describe in this chapter has been applied to aircraft models, some familiarity with aircraft dynamics is needed. Therefore, we begin this chapter with a brief overview of rigid (jet) aircraft dynamics and the associated terminology. For a complete description of aircraft dynamics and control, the reader is referred to McRuer et al. [4] or Stevens and Lewis [5].

The equations of motion of a rigid body can be decoupled into independent rotational and translational equations if the origin of the coordinate system is chosen to be at the center of gravity (cg). The rotational motion of the body will then be equivalent to pitching, yawing, and rolling motions (to be defined later) about the center of gravity as if the cg were a fixed point in space. The remaining components of the motion will be three components of translation of the cg. Therefore, we will describe the state model for a six-degrees-of-freedom (6-DOF) model.

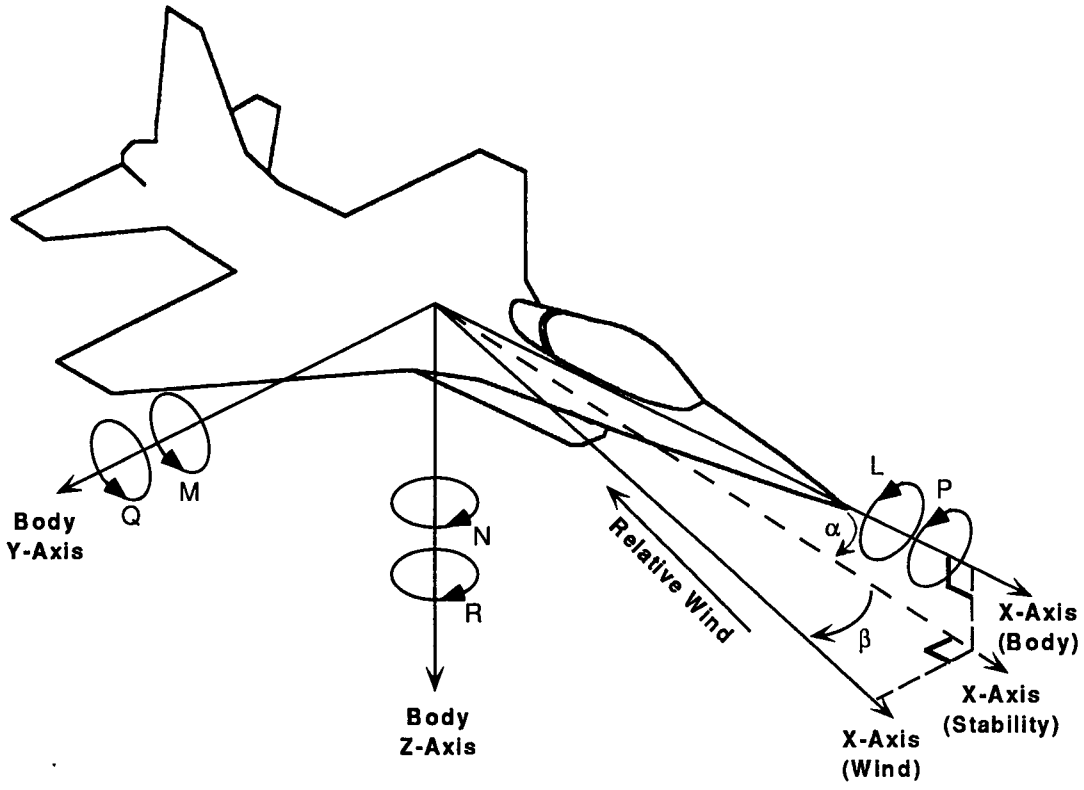


Figure 2.1: Diagram depicting axis definitions, moments and angular rates of the aircraft.

The choice of state variables follows from application of Newton's second law of motion. Three components of position are needed to specify potential energy in the gravitational field. Three components of velocity are required to specify translational kinetic energy, and three components of angular velocity to specify rotational kinetic energy. Three additional state variables are needed for attitude states. The attitude variables are used to specify orientation relative to the gravity vector. Therefore, the state vector of the basic rigid body model will contain 12 components.

2.1.1 Definition of Axes and Angles

The aerodynamic forces and moments on an aircraft are produced by the relative motion with respect to the air and depend on the orientation of the aircraft with respect to the airflow. In a uniform airflow these forces and moments are unchanged after a rotation around the free-stream velocity vector. Therefore, only two orientation angles (with respect to the relative wind) are needed to specify the aerodynamic forces and moments. The angles that are used are the angle of attack (α) and the sideslip angle (β). They are known as the aerodynamic angles and will now be defined by means of coordinate rotations in three dimensions. Note that the aerodynamic forces and moments are also dependent on angular rates, but for the moment we are concerned only with their dependence on orientation.

Figure 2.1 shows an aircraft with the relative wind on its side (i.e. sideslipping) and with the conventional right-handed (forward, right, and down) set of body-fixed axes illustrated. The angles of attack and sideslip are defined by performing a plane rotation about the body y-axis, followed by a plane rotation

about the new z-axis, such that the final x-axis is aligned directly into the relative wind. The first rotation defines the *stability axes*, and the angle of attack is the angle between the body-fixed x-axis and the stability x-axis. Alpha is positive if the rotation about the body-fixed y-axis was negative; thus a positive alpha is shown in the figure. The second rotation leads to a set of *wind axes*, and the sideslip angle is the angle between the stability x-axis and the wind x-axis. Beta is positive if the rotation about the stability z-axis is positive, and a positive beta is shown in the figure.

An aircraft of conventional shape must fly more or less directly into the apparent wind in order to have low drag. Therefore, beta is usually very small in steady flight. Alpha must be large enough to generate the required lift but is also usually quite small (in steady flight). Therefore, although the stability axes and wind axes have a variable orientation depending on the flight condition, they essentially point forward, right, and down, the same as the body-fixed axes. Note that technically all three sets of axes are "body axes" but only one is body-fixed. We shall drop the term "fixed" and simply refer to them as body, stability and wind axes.

A left-handed wind axes system, aligned backwards, left, and up relative to the aircraft, has often been used in the past for wind tunnel data. *Lift* L, *drag* D, and *cross-wind force* C were defined naturally in these axes as the aerodynamic force components along the respective positive axes. In the right-handed axes that we have defined, we will use $-L$ and $-D$ for the z and x force components, and define a component Y for the aerodynamic sideforce measured along the positive y-axis.

The transformation from the body to stability axes is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{STAB} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{BODY}, \tag{2.1}$$

and the rotation from stability axes to wind axes is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{WIND} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{STAB}. \tag{2.2}$$

We shall denote these rotations by B_α and B_β, and the complete rotation from body to wind axes by B_w. Therefore, if v_{BODY} is an arbitrary vector in the body axes, and v_{WIND} is the same vector expressed in wind axes, then

$$v_{WIND} = B_\beta B_\alpha v_{BODY} = B_w v_{BODY}, \tag{2.3}$$

where

$$B_w = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}. \tag{2.4}$$

For the remainder of the derivation, we will shorten the subscripts to b for body, s for stability, and w for wind axes.

In order to describe the position state variables, we need to define a new reference frame. We shall call this reference frame the north-east-down (NED) frame because the x-axis points north, y-axis east, and z-axis points down. The NED frame is considered fixed in space with its x-y plane tangent to the Earth's surface. The vehicle position (x and y) and altitude (h) are measured from the origin of this reference system.

We need the ability to relate the instantaneous attitude of the aircraft (in the body axes) to the NED reference frame. The sequence of rotations conventionally used in the aircraft industry to describe the attitude with respect to this (NED) reference frame is as follows. Starting from the reference frame (aircraft straight and level):

1. Rotate about the z-axis, nose right (positive yaw ψ).

2. Rotate about the new y-axis, nose up (positive pitch θ).

3. Rotate about the new x-axis, right wing down (positive roll ϕ).

The yaw (or heading), pitch, and roll (or bank) angles ψ, θ, ϕ are commonly referred to as Euler angles. To transform a vector in the NED frame to the same vector in the body-axes we use the following transformation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{BODY} = B_\phi B_\theta B_\psi \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED} . \tag{2.5}$$

The individual rotation matrices are

$$B_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$B_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$B_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

If we let $B_b = B_\phi B_\theta B_\psi$ denote the complete transformation from the NED frame to the body frame, we get

$$B_b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}. \tag{2.6}$$

This transformation will be needed later to transform the body-axes translational velocities to the NED frame to define the position state variable trajectories.

2.1.2 Definition of Forces and Moments

The forces and moments at the aircraft cg have components due to aerodynamic effects and to engine thrust; these components will be denoted, respectively, by the subscripts A and T. The velocity vector of the cg, expressed in wind axes, has (by definition) an x-component equal to the true airspeed V_T, and no other components. We will now define the various forces and moments and their relationships to the body-axes quantities. The total force vectors in the body and wind axes respectively, are given by

$$F_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} F_{x_A} \\ F_{y_A} \\ F_{z_A} \end{bmatrix} + \begin{bmatrix} F_{x_T} \\ F_{y_T} \\ F_{z_T} \end{bmatrix} \equiv F_{b_A} + F_{b_T} \tag{2.7}$$

$$B_w F_b = F_w = F_{w_A} + F_{w_T} = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} + B_w F_{b_T}. \tag{2.8}$$

The moment vectors in the body and wind axes, respectively are given by

$$T_b = \begin{bmatrix} \bar{L} \\ M \\ N \end{bmatrix} = \begin{bmatrix} \bar{L}_A \\ M_A \\ N_A \end{bmatrix} + \begin{bmatrix} \bar{L}_T \\ M_T \\ N_T \end{bmatrix} \equiv T_{b_A} + T_{b_T} \tag{2.9}$$

$$T_w = \begin{bmatrix} \bar{L}_{w_A} \\ M_{w_A} \\ N_{w_A} \end{bmatrix} + B_w T_{b_T} \equiv T_{w_A} + T_{w_T}. \tag{2.10}$$

The velocity vector in the body axes is given as

$$v_b = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = B_w^T v_w = B_w^T \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V_T \cos\alpha \cos\beta \\ V_T \sin\beta \\ V_T \sin\alpha \cos\beta \end{bmatrix}. \tag{2.11}$$

The thrust component F_{y_T} can be produced by unbalanced engine power because in a multiengined aircraft, the (inboard) engines may be toed-in to align them with the airflow from the forebody. The component F_{z_T} may result from a design in which the thrust vector is aligned to pass through the cg, when an engine is above or below the cg. From eq. (2.11) we get the following expressions for α, β, and true airspeed:

$$\begin{aligned} \tan\alpha &= \frac{W}{U} \\ \sin\beta &= \frac{V}{V_T} \\ V_T &= (U^2 + V^2 + W^2)^{1/2} \end{aligned} \tag{2.12}$$

α, β, and V_T are the three most important variables determining aerodynamic forces and moments, so these equations will prove very useful in conjunction with the body-axes equations of motion.

The forces and moments acting on the complete aircraft are defined in terms of dimensionless aerodynamic coefficients. We have

$$\begin{aligned} \text{drag}, D &= \bar{q}SC_D \\ \text{lift}, L &= \bar{q}SC_L \\ \text{sideforce}, Y &= \bar{q}SC_Y \\ \text{rolling moment}, \bar{L} &= \bar{q}SbC_l \\ \text{pitching moment}, M &= \bar{q}S\bar{c}C_M \\ \text{yawing moment}, N &= \bar{q}SbC_N, \end{aligned} \tag{2.13}$$

where

$$\begin{aligned} \bar{q} &= \frac{1}{2}\rho V_T^2 = \text{free-stream dynamic pressure} \\ \rho &= \text{air density} \\ S &= \text{wing reference area} \\ b &= \text{wing span} \\ \bar{c} &= \text{wing mean geometric chord.} \end{aligned}$$

The dynamic pressure equation models the way in which forces and moments depend on airspeed and density. Note that the dimensionless force coefficients have been specified for wind axes. The moment coefficients may be specified in wind, stability, or body axes, but the same symbols are used in each case.

The various dimensionless coefficients C_D, C_L, ..., C_N, are primarily dependent on the aerodynamic angles α and β and less dependent on a number of other variables. The coefficients are also dependent on control surface deflections; otherwise, the vehicle would not be controllable. Furthermore, although the effect of airspeed is accounted for through the dynamic pressure, the aerodynamic coefficients are still airspeed dependent at other than low subsonic Mach numbers.

This is because the flexibility of the airframe causes it to change shape when large aerodynamic forces are generated at higher Mach numbers, and because the nature of the flow field changes at high Mach numbers. The aerodynamic coefficients are also dependent on other factors, such as engine power level, configuration effects (e.g. landing gear, external tanks, etc.) and ground proximity effects.

Because of the complicated functional dependence of the aerodynamic coefficients, each "total" coefficient in eq. (2.13) is modeled as a sum of components that are, individually, functions of fewer variables. The coefficients are usually expressed as a baseline component, plus incremental or correction terms. The baseline component is primarily a function of α, β, and Mach number. The incremental terms that must be added will depend on flight conditions. Data for these equations are derived from wind-tunnel tests on models, flight tests, and aerodynamic-prediction computer programs, and are compiled in the aerodynamic database in tabular or graphical form. The extent to which individual force and moment equations can be broken down into summations of simpler functions of fewer variables depends on the complexity of the airframe, the range of validity required, and the skill of the stability and control engineer.

2.1.3 The Nonlinear Aircraft Model

The standard 6-DOF equations used for conventional aircraft assume a flat Earth, since the accelerations associated with the Earth's rotation are negligible compared to the accelerations that can be produced by a maneuvering aircraft. Furthermore, when designing control systems for maneuvering aircraft, there is no need to perform accurate navigation over the surface of the Earth.

The elements of the state vector will comprise, respectively, the components of the velocity vector v_B (eq, 2.11), the Euler angles, the angular velocities, and the position vector. Therefore, we have the state vector

$$X = [U \;\; V \;\; W \;\; \phi \;\; \theta \;\; \psi \;\; p \;\; q \;\; r \;\; x \;\; y \;\; h]^T . \tag{2.14}$$

We now define the standard set of body-axes state equations. The body-axes force equations are:

$$\begin{aligned} \dot{U} &= rV - qW - g\sin\theta + \tfrac{F_x}{m} \\ \dot{V} &= -rU + pW + g\sin\phi\cos\theta + \tfrac{F_y}{m} \\ \dot{W} &= qU - pV + g\cos\phi\cos\theta + \tfrac{F_z}{m} \end{aligned} \tag{2.15}$$

The kinematic equations are:

$$\begin{aligned} \dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi) \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \sec\theta(q\sin\phi + r\cos\phi) \end{aligned} \tag{2.16}$$

The moment equations are:

$$\begin{aligned} \dot{p} &= (c_1 r + c_2 p)q + c_3\bar{L} + c_4 N \\ \dot{q} &= c_5 pr - c_6(p^2 - r^2) + c_7 M \\ \dot{r} &= (c_8 p - c_2 r)q + c_4\bar{L} + c_9 N \end{aligned} \tag{2.17}$$

where the constants c_i are defined by

$$\begin{aligned} c_1 &= \tfrac{(I_y - I_z)I_z - I_{xz}^2}{\Gamma} & c_2 &= \tfrac{(I_x - I_y + I_z)I_{xz}}{\Gamma} \\ c_3 &= \tfrac{I_z}{\Gamma} & c_4 &= \tfrac{I_{xz}}{\Gamma} \\ c_5 &= \tfrac{I_z - I_x}{I_y} & c_6 &= \tfrac{I_{xz}}{I_y} \\ c_7 &= \tfrac{1}{I_y} & c_8 &= \tfrac{I_x(I_x - I_y) + I_{xz}^2}{\Gamma} \\ c_9 &= \tfrac{I_x}{\Gamma} \end{aligned} \tag{2.18}$$

with

$$\Gamma = I_x I_z - I_{xz}^2$$

and the inertia matrix (under the assumption that x-z is a plane of symmetry) is given by

$$J = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}.$$

The navigation equations represent, respectively, the north, east, and vertical components of the aircraft velocity in the locally level geographic (NED) frame on the surface of the Earth and are given by

$$\begin{aligned} \dot{x} &= U\cos\theta\cos\psi + V(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi) \\ &\quad +W(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \\ \dot{y} &= U\cos\theta\sin\psi + V(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) \\ &\quad +W(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) \\ \dot{h} &= U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta. \end{aligned} \tag{2.19}$$

The force and moment components (F_x, F_y, F_z, $\bar{L}$, M, N) in the 6-DOF equations must be broken into aerodynamic and thrust contributions as in eqs. (2.7) through (2.10). Then the aerodynamic contributions can be obtained from eq. (2.13). The aerodynamic force and moment components in eqs. (2.15) and (2.17) depend on the aerodynamic angles and true airspeed. Therefore, we must use eq. (2.12) to calculate these quantities. It is usually convenient to go one step further and replace the state variables U, V, and W in the 6-DOF model by V_T, β, and α. Then U, V, and W can be calculated from eq. (2.11) for the force equations, and from eq. (2.12) we can derive the following expressions for

the new state derivatives:

$$\begin{aligned} \dot{V}_T &= \frac{U\dot{U}+V\dot{V}+W\dot{W}}{V_T} \\ \dot{\beta} &= \frac{\dot{V}V_T-V\dot{V}_T}{V_T^2\cos\beta} \\ \dot{\alpha} &= \frac{U\dot{W}-W\dot{U}}{U^2+W^2}. \end{aligned} \tag{2.20}$$

The new state vector becomes

$$X = [V_T \ \beta \ \alpha \ \phi \ \theta \ \psi \ p \ q \ r \ x \ y \ h]^T. \tag{2.21}$$

At this point a difficulty may arise because the aerodynamic force coefficients can depend on $\dot{\alpha}$ or $\dot{\beta}$. However, the $\dot{\alpha}$ and $\dot{\beta}$ derivatives are not available until after the force equations have been evaluated. In terms of a state-variable formulation, the body-axes state equations cannot be arranged in a form that allows an explicit solution for the derivatives. This problem can be avoided by a change of axes, provided that the aerodynamic forces are linearly dependent on $\dot{\alpha}$ and $\dot{\beta}$. Note that if $\dot{\alpha}$ and $\dot{\beta}$ dependence is present only in the aerodynamic moment coefficients, implicit state equations can be avoided simply by evaluating the force equations before the moment equations. This is a common situation. The rate derivative $C_{L_{\dot{\alpha}}}$ is frequently neglected as are all the $\dot{\beta}$ derivatives, while the $C_{m_{\dot{\alpha}}}$ is significant.

In addition to solving the implicit solution problems of eq. (2.20), the wind axes are natural axes for aerodynamic forces. Furthermore, the wind-axes equations will easily lead to linear, small perturbation equations whose coefficients contain the partial derivatives of the aerodynamic forces and moments with respect to the wind-axes variables α, β, and V_T. These are the aerodynamic derivatives (or "stability derivatives"). Methods exist to estimate their values for a complete aircraft, and the values are commonly specified in aerodynamic

databases. For simplicity, the engine thrust vector will be assumed to be parallel to the body x-axis. The body-axes thrust force component will be denoted by F_T. We can express the aerodynamic angles and true airspeed derivatives in the wind-axes as:

$$\begin{aligned} \dot{V}_T &= \tfrac{1}{m}\left[F_T \cos\alpha \cos\beta - D + mg_1\right] \\ \dot{\beta} &= r_w + p\sin\alpha - r\cos\alpha \\ \dot{\alpha} &= q - q_w \sec\beta - \tan\beta(p\cos\alpha + r\sin\alpha) \end{aligned} \tag{2.22}$$

with

$$\begin{aligned} q_w &= \frac{1}{mV_T}\left[F_T \sin\alpha + L + mg_3\right] \\ r_w &= \frac{1}{mV_T}\left[-F_T \cos\alpha \sin\beta + Y + mg_2\right] \end{aligned}$$

where q_w is the pitch rate in the wind-axes, r_w is the roll rate in the wind-axes, and the components of the gravity vector are given by

$$\begin{aligned} g_1 &= g(-\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\phi\cos\theta + \sin\alpha\cos\beta\cos\phi\cos\theta) \\ g_2 &= g(\cos\alpha\sin\beta\sin\theta + \cos\beta\sin\phi\cos\theta - \sin\alpha\sin\beta\cos\phi\cos\theta) \\ g_3 &= g(\sin\alpha\sin\theta + \cos\alpha\cos\phi\cos\theta). \end{aligned} \tag{2.23}$$

It is now apparent from the wind-axes force equations that if linear dependence on $\dot{\alpha}$ is present in the lift coefficient, the $\dot{\alpha}$ terms can be collected on the left-hand side of the lift equations. Similarly, the side-force equation can be made explicit in $\dot{\beta}$ if $\dot{\beta}$ dependence is present in the side-force coefficient. Therefore, explicit state equations can be derived by using the wind-axes force equations when this is difficult or impossible with the body-axes state equations.

2.2 Feedback Linearization

We consider a nonlinear system modeled as

$$\dot{x} = f(x,u) \tag{2.24}$$

$$y = h(x). \tag{2.25}$$

with internal state $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, measured output $y \in \mathbb{R}^p$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. We will investigate when the dynamics of eqs. (2.24) - (2.25) can be transformed by a (local) state space transformation $z = \Phi(x)$ and a static state feedback

$$u = \alpha(x,v) \tag{2.26}$$

into a linear system

$$\dot{z} = Az + Bv \tag{2.27}$$

$$y = Cz. \tag{2.28}$$

This is called the feedback linearization problem (see [6], [7]). If the controlled dynamics (2.24)–(2.25) are feedback linearizable, then the control of the system can be split into two parts: a nonlinear feedback loop which renders the system (in suitable local coordinates) linear, and a super imposed linear control strategy for the obtained linear system. The material in this section is taken largely from Isidori [6]. We will first give some necessary mathematical background for solving this problem. Then we will give necessary and sufficient conditions such that a nonlinear single-input single-output system is feedback linearizable. The solution to the SISO case has a simple structure which will give insight into the more complicated multivariable problem. We will then give the solution to the multivariable system case.

2.2.1 Mathematical Preliminaries

A nonlinear change of coordinates can be described in the form

$$z = \Phi(x) \tag{2.29}$$

where $z \in \mathbb{R}^n$ is the new state vector, $x \in \mathbb{R}^n$ is the original state vector and $\Phi(x)$ represents a $\mathbb{R}^n$-valued function of n variables,

$$\Phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_n(x) \end{bmatrix}$$

with the following properties

1. $\Phi(x)$ is invertible, i.e. there exists a function $\Phi^{-1}(z)$ such that

$$\Phi^{-1}(\Phi(x)) = x$$
$$\Phi(\Phi^{-1}(z)) = z$$

   for all $x \in \mathbb{R}^n$.

2. $\Phi(x)$ and $\Phi^{-1}(z)$ are both smooth mappings, i.e. have continuous partial derivatives of any order.

A transformation of this type is called a global diffeomorphism on $\mathbb{R}^n$. Sometimes a transformation possessing both these properties and defined for all x is difficult to find and the properties in question are difficult to verify. Thus, in most cases one rather looks at transformations defined only in a neighborhood of a given point. A transformation of this type is called a local diffeomorphism.

Proposition 2.1 ([6]) *Suppose $\Phi(x)$ is a smooth function defined on some subset U of $\mathbb{R}^n$. Suppose the Jacobian matrix*

$$\frac{\partial \Phi}{\partial x} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} & \cdots & \frac{\partial \phi_1}{\partial x_n} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} & \cdots & \frac{\partial \phi_2}{\partial x_n} \\ . & . & \cdots & . \\ \frac{\partial \phi_n}{\partial x_1} & \frac{\partial \phi_n}{\partial x_2} & \cdots & \frac{\partial \phi_n}{\partial x_n} \end{bmatrix}$$

is nonsingular at a point $x = x^0$. Then, for some suitable open subset U^0 of U, containing x^0, $\Phi(x)$ defines a local diffeomorphism between U^0 and its image $\Phi(U^0)$.

Given a real valued function of $x = (x_1, x_2, \cdots, x_n)$, $\lambda(x)$, and a (n vector)-valued function of x

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix},$$

and letting

$$\frac{\partial \lambda}{\partial x} = \left[\frac{\partial \lambda}{\partial x_1} \cdots \frac{\partial \lambda}{\partial x_n} \right],$$

we define a new real-valued function of x, denoted $L_f\lambda(x)$, in the following way

$$L_f\lambda(x) = \frac{\partial \lambda}{\partial x} f(x) = \sum_{i=1}^{n} \frac{\partial \lambda}{\partial x_i} f_i(x).$$

This function is sometimes called the derivative of $\lambda(x)$ along f. Occasionally, for compactness we may use a more condensed notation and write

$$d\lambda(x) = \frac{\partial \lambda}{\partial x}.$$

Repeated use of the derivative of $\lambda(x)$ along a function is possible. Thus, for instance, by differentiating $\lambda(x)$ first along $f(x)$ and then along a (n vector)-

valued function $g(x)$, we may construct the new function

$$L_g L_f \lambda(x) = \frac{\partial L_f \lambda}{\partial x} g(x).$$

If λ is being differentiated k times along f, we may construct a function recursively defined as

$$L_f^k \lambda(x) = \frac{\partial L_f^{k-1} \lambda}{\partial x} f(x)$$

with $L_f^0 \lambda(x) = \lambda(x)$.

Another type of differential operation involves two (n vector)-valued functions of $x = (x_1, x_2, \cdots, x_n)$, $f(x)$ and $g(x)$. We define a new (n vector)-valued function of x, denoted $[f, g](x)$ as

$$[f, g](x) = \frac{\partial g}{\partial x} f(x) - \frac{\partial f}{\partial x} g(x)$$

where $\frac{\partial g}{\partial x}$ and $\frac{\partial f}{\partial x}$ are the Jacobian matrices of $g(x)$ and $f(x)$, respectively. This new function is called the Lie product or Lie bracket of $f(x)$ and $g(x)$. Repeated bracketing of a function $g(x)$ with the same function $f(x)$ is possible. Whenever this is needed, in order to avoid notation of the form $[f, [f, \cdots, [f, g]]]$ that could generate confusion, it is preferable to define such an operation recursively, as

$$ad_f^k g(x) = [f, ad_f^{k-1} g](x)$$

for any $k \geq 1$, setting $ad_f^0 g(x) = g(x)$.

Definition 2.1 ([6]) *A set of k (n vector)-valued functions $\{X_1(x), \cdots, X_k(x)\}$, such that the matrix*

$$(X_1(x) \;\; X_2(x) \;\cdots\; X_k(x))$$

has rank k at the point $x = x^0$, is said to be involutive near x^0, if for each pair of integers $(i, j), 1 \leq i, j \leq k$, the matrix

$$(X_1(x) \;\; X_2(x) \;\cdots\; X_k(x) \;\; [X_i, X_j](x))$$

still has rank k for all x in a neighborhood of x^0.

2.2.2 Single-Input Single-Output Systems

We begin by looking at a single-input single-output plant modeled by nonlinear differential equations of the form

$$\dot{x} = f(x) + g(x)u \tag{2.30}$$

$$y = h(x). \tag{2.31}$$

with internal state $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, control input $u \in \mathbb{R}$, and measured output $y \in \mathbb{R}$. The $\mathbb{R}^n$-valued functions $f(x)$, $g(x)$ and the real-valued function $h(x)$ are nonlinear functions of their arguments, and are assumed to be differentiable a sufficient number of times.

Definition 2.2 ([6]) *The nonlinear system described by eqs. (2.30) - (2.31) has relative degree r at* x^0 *if:*

1. $L_g L_f^k h(x) = 0$ *for all x in a neighborhood of* x^0 *and all* $k < r - 1$;

2. $L_g L_f^{r-1} h(x^0) \neq 0$.

It can be shown that the relative degree, r, is exactly the number of times $y(t)$ has to be differentiated at $t = t^0$ for $u(t^0)$ to appear explicitly.

Consider a nonlinear system (2.30) - (2.31) having at some point $x = x^0$ relative degree equal to the dimension of the state space, i.e. $r = n$. Construct

a change of coordinates by setting

$$\Phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_n(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix}. \tag{2.32}$$

In the new coordinates

$$z_i = \phi_i(x) = L_f^{i-1} h(x) \quad 1 \leq i \leq n$$

we get

$$\frac{dz_1}{dt} = \frac{\partial \phi_1}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial h}{\partial x}\frac{\partial x}{\partial t} = L_f h(x(t)) = \phi_2(x(t)) = z_2(t)$$

$$\vdots$$

$$\frac{dz_{n-1}}{dt} = \frac{\partial \phi_{n-1}}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial L_f^{r-2} h}{\partial x}\frac{\partial x}{\partial t} = L_f^{n-1} h(x(t)) = \phi_n(x(t)) = z_n(t).$$

For z_n we get

$$\frac{dz_n}{dt} = L_f^n h(x(t)) + L_g L_f^{n-1} h(x(t)) u(t).$$

Replacing $x(t)$ on the right hand side of the equation by $x(t) = \Phi^{-1}(z(t))$ we get

$$\frac{dz_n}{dt} = b(z(t)) + a(z(t)) u(t)$$

with

$$\begin{aligned} a(z) &= L_g L_f^{n-1} h(\Phi^{-1}(z)) \\ b(z) &= L_f^n h(\Phi^{-1}(z)). \end{aligned}$$

Summarizing, the transformed system equations become:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_n &= b(z) + a(z) u. \end{aligned}$$

Recall that at the point $z^0 = \Phi(x^0)$, and hence at all z in a neighborhood of z^0, the function $a(z)$ is nonzero by definition of the relative degree.

Choose now the following state feedback control law

$$u = \frac{1}{a(z)}[-b(z) + v]. \tag{2.33}$$

The resulting closed-loop system described by the equations

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_n &= v, \end{aligned} \tag{2.34}$$

is linear and controllable. Thus we conclude that any nonlinear system of the form (2.30) – (2.31) with relative degree n at some point x^0 can be transformed into a system which is linear and controllable by means of (i) a local change of coordinates, and (ii) a local static state feedback. We can also express the control law in the original x coordinates as

$$u = \frac{1}{a(\Phi(x))}[-b(\Phi(x)) + v] = \frac{1}{L_g L_f^{n-1} h(x)}[-L_f^n h(x) + v]. \tag{2.35}$$

Note that while eq. (2.35) appears to contain only the output function $h(x)$, when the derivatives are taken the final form for u will in general be a function of the entire state vector.

Up to this point, the existence of an "output" function $h(x)$ relative to which system (2.30) – (2.31) has relative degree exactly equal to n (at x^0) has been key in making it possible to transform the system into a linear and controllable one. If such a function $h(x)$ is not available beforehand, either because the actual output of the system does not satisfy the conditions required to have relative

degree n or simply because no specific output is defined for the given system, the question arises whether it is possible to find an appropriate $h(x)$ that allows output linearization. The following theorem gives conditions under which an output exists which linearizes the system.

Theorem 2.1 ([6]) *Suppose a system of the form (2.30) is given. There exists an "output" function $h(x)$ for which the system has relative degree n at a point x^0 if and only if the following conditions are satisfied:*

1. *the matrix $[g(x^0)\ ad_f g(x^0)\ \cdots ad_f^{n-2}g(x^0)\ ad_f^{n-1}g(x^0)]$ has rank n,*
2. *the set $\{g(x^0), ad_f g(x^0), \cdots ad_f^{n-2}g(x^0)\}$ is involutive near x^0.*

2.2.3 Multi-Input Multi-Output Systems

In this section, we shall see how the theory developed for single-input single-output systems can be extended to nonlinear systems having many inputs and many outputs. In order to avoid unnecessary complications, we shall restrict our analysis to the consideration of systems having the same number m of inputs and outputs. We will state how the conditions for the main theorem (Theorem 2.2) should be changed to handle a system with a different number of inputs and outputs. The multivariable nonlinear systems we consider are described in state space form as

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \tag{2.36}$$

$$\begin{aligned} y_1 &= h_1(x) \\ &\vdots \\ y_m &= h_m(x) \end{aligned} \tag{2.37}$$

with state $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, control inputs $u_i \in \mathbb{R}$, and measured outputs $y_i \in \mathbb{R}$. We can write these equations in a more condensed form as

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

with

$$\begin{aligned} u &= [u_1 \ \cdots \ u_m]' \\ y &= [y_1 \ \cdots \ y_m]' \end{aligned}$$

and where

$$\begin{aligned} g(x) &= [g_1(x) \ g_2(x) \ \cdots \ g_m(x)], \\ h(x) &= [h_1(x) \ h_2(x) \ \cdots \ h_m(x)]' \end{aligned}$$

are respectively an $n \times m$ matrix and an m-vector.

Definition 2.3 ([6]) *The multivariable nonlinear system described by eqs. (2.36) – (2.38) has a (vector) relative degree $\{r_1, \cdots, r_m\}$ at x^0 if:*

1.

$$L_{g_j} L_f^k h_i(x) = 0$$

for all $1 \leq j \leq m$, for all $1 \leq i \leq m$, for all $k \leq r_i - 1$, and for all x in a neighborhood of x^0,

2. the $m \times m$ matrix

$$A(x) = \begin{bmatrix} L_{g_1} L_f^{r_1 - 1} h_1(x) & \cdots & L_{g_m} L_f^{r_1 - 1} h_1(x) \\ L_{g_1} L_f^{r_2 - 1} h_2(x) & \cdots & L_{g_m} L_f^{r_2 - 1} h_2(x) \\ \cdots & \cdots & \cdots \\ L_{g_1} L_f^{r_m - 1} h_m(x) & \cdots & L_{g_m} L_f^{r_m - 1} h_m(x) \end{bmatrix} \tag{2.38}$$

is nonsingular at $x = x^0$.

It is seen that this definition incorporates the one given for the single-input single-output case in the previous section. Note that each integer r_i is associated with the i-th output of the system. By definition, for all $k \leq r_i - 1$, the row vector

$$[L_{g_1} L_f^k h_i(x) \;\; L_{g_2} L_f^k h_i(x) \;\; \cdots \;\; L_{g_m} L_f^k h_i(x)]$$

is zero for all x in a neighborhood of x^0 and, for $k = r_i - 1$, this row vector is nonzero (i.e. has at least a nonzero element at x^0), because the matrix $A(x^0)$ is assumed to be nonsingular. As a consequence, in view also of the condition (1), we see that for each i there is at least one choice of j such that the (single-input single-output) system having output y_i and input u_j has exactly relative degree r_i at x^0 and, for any other possible choice of j (i.e. of input channel), the corresponding relative degree at x^0 (if any) is necessarily higher than or equal to r_i. Finally, it can be shown that r_i is exactly the number of times the i-th output $y_i(t)$ has to be differentiated at $t = t^0$ to have at least one component of the input vector $u(t^0)$ appear explicitly.

We note that the matrix $A(x^0)$ can be written as

$$A(x^0) = \begin{bmatrix} L_f^{r_1 - 1} h_1(x) \\ L_f^{r_2 - 1} h_2(x) \\ \vdots \\ L_f^{r_m - 1} h_m(x) \end{bmatrix} \begin{bmatrix} g_1(x^0) & g_2(x^0) & \cdots & g_m(x^0) \end{bmatrix}.$$

From the nonsingularity of the matrix $A(x^0)$ we deduce that the matrix

$$[g_1(x) \;\; g_2(x) \;\; \cdots \;\; g_m(x)]$$

has rank m near x^0.

We will now discuss the multivariable extension of the feedback linearization problem. Consider the nonlinear system (2.36)–(2.38) having a (vector) relative

degree $\{r_1, \cdots, r_m\}$ at x^0 and suppose the sum $r = r_1 + r_2 + \cdots + r_m$ is exactly equal to the dimension n of the state space. Then the set of functions

$$\phi_k^i(x) = L_f^{k-1} h_i(x) \quad \text{for } 1 \leq k \leq r_i, 1 \leq i \leq m$$

with

$$\Phi(x) = \text{col}[\phi_1^1(x), \cdots, \phi_{r_1}^1(x), \cdots, \phi_1^m(x), \cdots, \phi_{r_m}^m(x)]$$

defines completely a local coordinate transformation at x^0. In the new coordinates

$$z_k^i = \phi_k^i(x)$$

the system is described by m sets of equations of the form

$$\begin{aligned} \dot{z}_1^i &= z_2^i \\ &\vdots \\ \dot{z}_{r_i-1}^i &= z_{r_i}^i \\ \dot{z}_{r_i}^i &= b_i(z) + \sum_{j=1}^{m} a_{ij}(z) u_j(t) \end{aligned}$$

for $1 \leq i \leq m$ with

$$\begin{aligned} a_{ij}(z) &= L_{g_i} L_f^{r_i - 1} h_i(\Phi^{-1}(z)) \quad \text{for} 1 \leq i, j \leq m \\ b_i(z) &= L_f^{r_i} h_i(\Phi^{-1}(z)) \qquad\quad \text{for} 1 \leq i \leq m \end{aligned}$$

Note that in the original coordinates a_{ij} is exactly equal to the (i, j) entry of the matrix $A(x)$ in (2.38). Note also, that if we look at a particular

$$z^i = \begin{bmatrix} z_1^i \\ z_2^i \\ \vdots \\ z_{r_i}^i \end{bmatrix}$$

the notation is consistent with the single-input single-output case discussed in the previous section.

Define

$$b(x) = \begin{bmatrix} L_f^{r_1} h_1(x) \\ L_f^{r_2} h_2(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{bmatrix}. \tag{2.39}$$

Recall that in a neighborhood of the point $z^0 = \Phi^{-1}(x^0)$ the matrix $A(z)$ is nonsingular and therefore the equations

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = b(z) + A(z)u \tag{2.40}$$

can be solved for u. The input u solving these equations has the form of a state feedback

$$u = A^{-1}(z)[-b(z) + v].$$

Imposing this feedback yields a system characterized by the m sets of equations

$$\begin{aligned} \dot{z}_1^i &= z_2^i \\ &\vdots \\ \dot{z}_{r_i-1}^i &= z_{r_i}^i \\ \dot{z}_{r_i}^i &= v_i \end{aligned} \tag{2.41}$$

for $1 \leq i \leq m$, which is clearly linear and controllable. In terms of the original description of the system, the linearizing feedback has the form

$$u = \alpha(x) + \beta(x)v \tag{2.42}$$

where

$$\alpha(x) = -A^{-1}(x)b(x) \quad \beta(x) = A^{-1}(x)$$

with $A(x)$ and $b(x)$ defined in (2.38) and (2.39), respectively. We can conclude by looking at (2.41) that the new system description can be written as

$$\dot{z} = Az + Bv \tag{2.43}$$

$$y = Cz \tag{2.44}$$

with

$$A = \text{diag}(A_1, \cdots, A_m)$$

$$B = \text{diag}(b_1, \cdots, b_m)$$

$$C = \text{diag}(c_i, \cdots, c_m)$$

where A_i is the $r_i \times r_i$ matrix

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

b_i is the $r_i \times 1$ vector

$$b_i = \text{col}(0, \cdots, 0, 1)$$

and c_i is the $1 \times r_i$ vector

$$c_i = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}.$$

It can be shown that the conditions used above to construct the linearizing feedback are also necessary conditions.

Lemma 2.1 ([6]) *Given a nonlinear system of the form (2.36)–(2.38), suppose the matrix $g(x^0)$ has rank m. There exists a static state feedback and a change of coordinates, defined locally around x^0, so that the system is transformed into the form (2.44) if and only if the nonlinear system (2.36)–(2.38) has some (vector) relative degree $\{r_1, \cdots, r_m\}$ at x^0 and $r_1 + r_2 + \cdots + r_m = n$.*

As in the SISO case, if the actual output of the system $h(x)$ does not satisfy the conditions of Lemma 2.1 or no specific output is defined for the given system, we wish to know whether it is possible to find an appropriate $h(x)$ that allows output linearization. Let us first define

$$\begin{aligned} G_0 &= span\{g_1, \cdots, g_m\} \\ G_1 &= span\{g_1, \cdots, g_m, ad_f g_1, \cdots, ad_f g_m\} \\ &\vdots \\ G_i &= span\{ad_f^k g_j : 0 \leq k \leq i, \ \ 1 \leq j \leq m\} \end{aligned}$$

for $i = 0, 1, \cdots, n-1$. Note that by definition, $G_{i-1} \subset G_i$ for any $i \geq 1$. The following theorem gives conditions under which an output exists which linearizes the system.

Theorem 2.2 ([6]) *Suppose the matrix $g(x^0)$ has rank m. Then there exist m real-valued functions $\lambda_1(x), \lambda_2(x), \cdots, \lambda_m(x)$ such that the system*

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= \lambda(x) \end{aligned}$$

has some (vector) relative degree $\{r_1, \cdots, r_m\}$ at x^0 with

$$r_1 + r_2 + \cdots + r_m = n$$

if and only if:

1. *for each* $0 \leq i \leq n-1$, *the distribution* G_i *has constant dimension near* x^0;

2. *the distribution* G_{n-1} *has dimension* n;

3. *for each* $0 \leq i \leq n-2$, *the distribution* G_i *is involutive.*

Observe that the conditions stated in Theorem 2.2, in the case of a single-input system, reduce exactly to those described in Theorem 2.1. When $m = 1$ the distribution G_i reduces to

$$G_i = span\{g, ad_f g, \cdots, ad_f^i g\}.$$

The condition (2) above, i.e. $\dim(G_{n-1}) = n$, implies that $\dim(G_i) = i + 1$, i.e. the condition (1). This being the case, the involutivity of G_{n-2} implies also that of $G_0, \cdots, G_{n-3}$.

Theorem 2.2 can be extended for a system with p outputs and m inputs, with $p \neq m$, provided condition (2) of Definition 2.38 is replaced by the assumption that the $p \times m$ matrix $A(x)$ has full row rank (i.e. rank equal to the number of output channels). This implies that the number of inputs must be larger than or equal to the number of outputs. Under this new assumption, the same coordinate transformation can be used. If $A(x)$ has rank p, then equation (2.40) can be solved for u as

$$u = A_R^{-1}(x)[-b(x) + v]$$

where $A_R^{-1}(x)$ is the right inverse of $A(x)$.

We will now extend the results of Theorem 2.2 to general nonlinear systems of the form

$$\dot{x} = f(x, u) \tag{2.45}$$

with state $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, and control input $u \in \mathbb{R}^m$. The conditions for feedback linearizability rely on the notion of extended system of a general nonlinear system.

Definition 2.4 ([8]) *The extended system of (2.45) is the affine nonlinear system*

$$\begin{aligned} \dot{x} &= f(x,u) \\ \dot{u} &= w \end{aligned} \tag{2.46}$$

with state $(x,u) \in \mathbb{R}^n \times \mathbb{R}^m$ *and new input* $w \in \mathbb{R}^m$.

The extended system is of the form

$$\dot{x}_e = f(x_e) + g(x_e)w$$

where $x_e = (x,u)$ and

$$g(x_e) = \begin{bmatrix} 0_{n\times m} \\ I_{m\times m} \end{bmatrix}$$

Theorem 2.3 ([8, 9]) *Consider the nonlinear system (2.45) with* $f(x^0,u^0) = 0$. *The nonlinear system (2.45) is feedback linearizable around* (x^0,u^0) *if and only if the extended system (2.46) is feedback linearizable around* (x^0,u^0), *i.e. satisfies the conditions of Theorem 2.2.*

2.3 Dynamic Inversion Applied to Aircraft

The dynamic inversion technique was first proposed by Meyer and Cicolani [3] as part of a total automatic flight control system for short take-off and landing (STOL) and vertical take-off and landing (VTOL) aircraft. A control system was needed which could follow rough trajectory commands given by either the

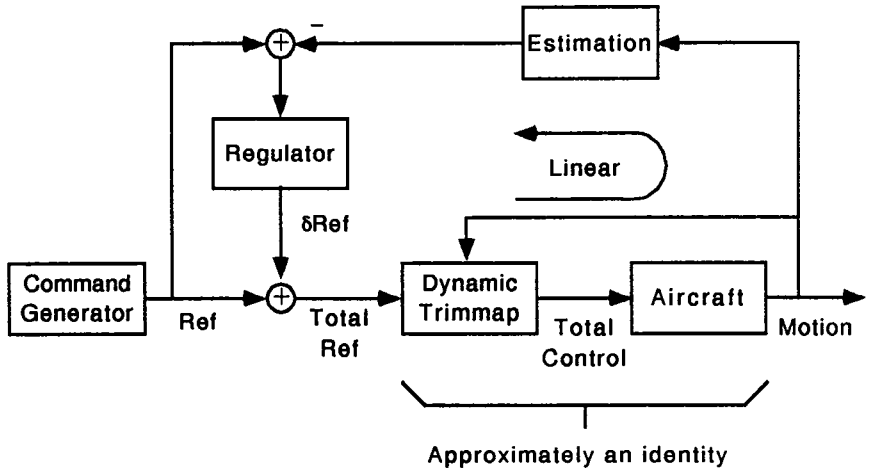


Figure 2.2: Simplified arrangement of the controller proposed by Meyer and Cicolani.

pilot or an air traffic controller despite strong nonlinearities in the description of the plant. A simplified description of the control loop proposed can be seen in Figure 2.2. The combination of the trimmap and the aircraft is approximately an identity in the following sense. If a trajectory is defined for the aircraft to follow, the acceleration, and hence the required force, are known. The trimmap represents an inverse model of the aircraft and given a commanded force solves for the control settings that will generate that force. If the trimmap contains an accurate representation of the vehicle dynamics, the controls generated from the trimmap when applied to the aircraft will result in a trajectory that matches the one commanded. From the standpoint of the control loop operation, the application of the trimmap concept makes the combined trimmap and aircraft blocks approximately an identity and results in operational characteristics that are approximately linear. From the linearization theory in the previous section, and the fact that the aircraft dynamics take the form of second-order differential equations for each channel (pitch, roll, yaw) the combined operational

Development and Analysis of a Nonlinear Dynamic Inverse Control Strategy

by

John Reilly

Dissertation submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1996

Advisory Committee:

Professor William S. Levine, Chairman/Advisor
Associate Professor Wijesuriya P. Dayawansa
Professor Gilmer L. Blankenship
Professor Eyad Abed
Professor David K. Schmidt

Abstract

Title of Dissertation: Development and Analysis of a Nonlinear Dynamic Inverse Control Strategy

John Reilly, Doctor of Philosophy, 1996

Dissertation directed by: Professor William S. Levine
Department of Electrical Engineering

Aircraft control normally relies on gain-scheduling of linear control laws, an approach that has been very successful. One can characterize nonlinear dynamics in high-alpha maneuvers by using a large number of linear models, but this method may not be adequate, especially since some of the chosen operating points may not be equilibrium points. One can build robustness into the control strategy (e.g., by requiring more gain and phase margins), but this approach translates to a more conservative control, which means possible performance degradation. An alternative method is to take explicitly into account the nonlinearities in the control design, thus better utilizing the existing dynamics and control power. We choose to investigate the dynamic inverse control technique because of its ease of implementation, and the simple way that maneuvers can enter into the control. The objective of nonlinear dynamic inversion is to invert the dynamic equations of the plant directly in order to find the control necessary to yield the given output.

We elaborate the dynamic inverse methods first described by Meyer and Cicolani. We expand the method to a more complex aerodynamics and airframe

description, that of the full nonlinear simulation (wind-tunnel and flight tested model) of the X-29.To achieve additional realism, the simulation contains actuator redundancy and actuator limits. We first formulate a reduced analytic model in order to use feedback linearization techniques to design the controller. Since we neglect some of the aerodynamic terms, the controller is then modified so that stability robustness to modeling errors can be achieved. In addition, we modify the robust control method to add integral action to enable the controller to reduce steady state errors and to lower the control rates.

characteristics take the form of double integrators for each channel.

The major parts of the controller can be broken into two subsystems: one for translation and one for rotation, see Figure 2.3. This exploits the natural division of an aircraft's dynamics as discussed in Section 2.1. Each subsystem will then be in the control affine form as required for feedback linearization, if viewed properly. The aircraft rotation state equations take the form of equation (2.17). The inputs are the control deflections (appearing linearly in the N, $\bar{L}$, and M terms) and the outputs are the angular rates. This subsystem satisfies the form required in equations (2.36)–(2.38). The (vector) relative degree for this subsystem can easily be found to be {1,1,1}. The linearizing feedback is calculated through the moment trimmap. The combination of the linearizing feedback with the rotation subsystem of the aircraft (the inner loop) results in a set of integrators (one for each channel) where the inputs are the commanded angular accelerations and the outputs are the angular rates. Similarly, the translation subsystem with the attitude angles and thrust as inputs and the translation velocities as outputs takes the form of equations (2.36)–(2.38). The third-order system has three inputs (δ_T, β,α) and three outputs (U, V, W) (the state equations are given by equation (2.15) with the inputs appearing linearly in the F_x, F_y, and F_z terms respectively) and can trivially be shown to have (vector) relative degree {1,1,1}. The linearizing feedback for this subsystem is the force trimmap. The combination of the linearizing feedback with the translation subsystem of the aircraft (the outer loop) results in a set of integrators (one for each channel) where the inputs are the commanded forces and the outputs are the attitude angles and thrust command. We will now describe these transformations in more detail.

The function of the translational block is to convert rough commanded trajectories into a throttle command and attitude command angles to be sent to the rotational block. The trajectory command generator smoothes the pilot inputs (R_i, V_i, and $\dot{V}_i$) to create the trajectory commands (R_c, V_c, and $\dot{V}_c$), and generates the open loop forces

$$f_{oc} = m\dot{V}_c$$

to be added to the correctional forces, f_{cc} from the perturbation regulator. The total inertial (relative to the runway) force, f_{tc}, is then converted into the corresponding force coefficients relative to the velocity vector by means of eq. (2.13). A commanded roll angle, ϕ_c is also generated by

$$\phi_c = \tan^{-1}(\frac{C_{Y_C}}{-C_{L_C}}).$$

The commanded force coefficients, C_{L_C}, C_{D_C} and C_{Y_C}, are then input to a force trim map to find the required attitude angles and thrust command. A trim map is a table which maps a given output (or operating condition) into the inputs required to "trim" the aircraft at the desired output. To trim an aircraft means that one needs to find the appropriate control deflections (inputs) to maintain the aircraft at the desired operating condition (outputs). Usually the resulting aircraft motion has zero acceleration. However, in the context of the following description, trim has been generalized to include nonzero accelerations. In this case, the maneuvers sent to the force trim map are static accelerations which assume no angular motion. The trim map "inverts" any admissible acceleration into the required attitude and thrust commands. The entire force trim equations which need to be solved are:

$$C_L(\alpha, \beta, u) - C_{L_C} = 0$$

$$C_D(\alpha, \beta, u) - C_{D_C} = 0$$
$$C_Y(\alpha, \beta, u) - C_{Y_C} = 0$$

where u is the control vector. The force trim map consists of a table lookup where the predominant independent variables are the lift coefficient, C_L, drag coefficient, C_D, and sideforce coefficient, C_Y, and the predominant outputs are the angle of attack, α, yaw angle β, and the throttle, δT. The lookup tables may also be functions of control surfaces such as flap settings, dynamic variables such as α and β (which can be calculated from body axes velocities, see equation (2.12)) and other variables such as air temperature and density. Dynamics such as $\dot{\alpha}$, $\dot{\beta}$, etc. were neglected at this point and taken care of in the perturbation loop.

The function of the rotational block is to convert rough commanded attitude and flight-path angles into control surface deflections. The attitude command generator smoothes the attitude and flight-path inputs to create the rotational commands (A_c, ω_c, and $\dot{\omega}_c$), The open loop angular acceleration command, $\dot{\omega}_{oc}$, is then added to the correctional angular acceleration, $\dot{\omega}_{cc}$ from the perturbation regulator. The dynamic equation for rotation is given by

$$\dot{\omega} = J^{-1}\left[M + S(\omega)J\omega\right]$$

where J is the aircraft moment of inertia in body axes, M is the total aerodynamic and propulsive moment, and for any column vector $x = (x_1, x_2, x_3)^T$,

$$S(x) = \begin{pmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{pmatrix}.$$

Therefore, the total moment can be found by rearranging the above equation. The commanded moment coefficients can then be calculated from

$$M = \bar{q}S \begin{bmatrix} bC_{l_c} \\ \bar{c}C_{m_c} \\ bC_{n_c} \end{bmatrix}$$

where C_{l_c} is the rolling moment coefficient, C_{m_c} is the pitching moment coefficient, and C_{n_c} is the yawing moment coefficient, $\bar{q}$ is the dynamic pressure, S is the wing area, b is the reference span of the wing, and $\bar{c}$ is the reference aerodynamic chord of the wing. The moment coefficients are input to the moment trim map which attempts to find the control surface deflections by "inverting" the equations:

$$\begin{aligned} C_l(\alpha, \beta, u) - C_{l_C} &= 0 \\ C_m(\alpha, \beta, u) - C_{m_C} &= 0 \\ C_n(\alpha, \beta, u) - C_{n_C} &= 0 \end{aligned}$$

where u is the control vector.

A perturbation controller closes each loop from the output of the plant to the trim maps to regulate any inaccuracies or intentionally neglected terms in the trim map calculations as well as to account for any disturbances. As described earlier, the trimmaps perform the function of calculating the linearizing feedback, hence the combination of the controller and the aircraft results in a series of integrators. Therefore the controller regulates a basically linear function over the entire operating range of the aircraft. The first flight test of this control concept was presented by Wehrend and Meyer [10] on a DHC-6 Twin Otter aircraft using a modified racetrack course with satisfactory performance. Simplified analytical

models using only the major aerodynamic terms were used for the trimmaps because of the simplicity of the aerodynamics of the Twin Otter aircraft. For example, no state information (α, β) was used in the force trimmap. This concept was next used by Smith and Meyer [11] for a simulation study of automatic landing of an A-7E aircraft on a carrier. This application, more than others, placed limits on the aircraft motion to ensure that the commands were within the flight envelope of the aircraft. Because of the algorithmic nature of the controller, this was easily accomplished by placing limits on the commanded values output from the trim map. For example, the commanded angle of attack α_c calculated from the force trim map was limited to between $-2°$ and $+16°$. Calculated values outside that range were simply forced to the boundary. The study showed that this concept improved the performance of carrier landing compared to conventional approaches.

Up to this point, the trim map inversion was performed using exhaustive searches of the (simplified) aerodynamic tables to find the proper output (attitude angles, thrust, control deflections) for the given input (forces and moments). This technique was satisfactory for aircraft for which the configuration allowed the force and moment equations to be treated separately. For more complex aircraft, the inverse cannot be solved directly because analytical expressions for the trim variables do not exist. Smith and Meyer [12] instead chose to use an iterative Newton-Raphson procedure in a simulation study of a proposed vertical attitude take-off and landing (VATOL, tail sitter) aircraft to trim at the given forces and moments. A six-degree-of-freedom (three forces and three moments) trim was first used to trim the translational maneuvering assuming zero angular velocity and acceleration. The commanded attitudes were then used to gen-

erate angular accelerations as input to a four-degree-of-freedom (forward force and three moments) trim to find the four controls necessary to produce the commanded accelerations. Vertical attitude maneuvering about hover and transition from forward flight to hover were simulated using this control concept. Simulation results were satisfactory. However, during the transition from forward flight to hover, the Newton-Raphson algorithm became unstable. The trim procedure was trying to maintain the desired lift force. However, the lift curve slope, C_{L_α}, changes sign (from positive to negative) near an α of 32 deg. Therefore, near an α of 32 deg the algorithm will hunt for the proper α to maintain the trim force. This causes the procedure to oscillate between successive values for α. To enable trim near this position, the predicted corrections to control, i.e. the outputs from the Newton-Raphson algorithm, were reduced to 80%.

Kato and Sugiura [13] attempt to add insight into the inverse problem by performing a theoretical study of airplane general motion. By expressing the aircraft state variables as functions only of the triple (α, β, φ) and the given flight trajectory (x, y, z), they have formally justified the separation simplifications made by Meyer, et. al. in calculating the linearizing feedback. The block diagram shown in Figure 2.4 is a representation of an example given in the paper. For conventional aircraft, the main controls are the elevator, rudder, and aileron which control the angle of attack α, the sideslip angle β, and the bank angle φ, respectively. Therefore, they use these angles (α, β, φ) as the independent variables in their approach. The "motion" they use as the inputs are the flight path angles (ψ_W, θ_W) and the velocity (V_T) along the flight path. If the flight-path angles are known functions of time, the attitude angles, heading ψ and pitch θ, can be written in terms of the path angles and the triplet (α, β, φ),

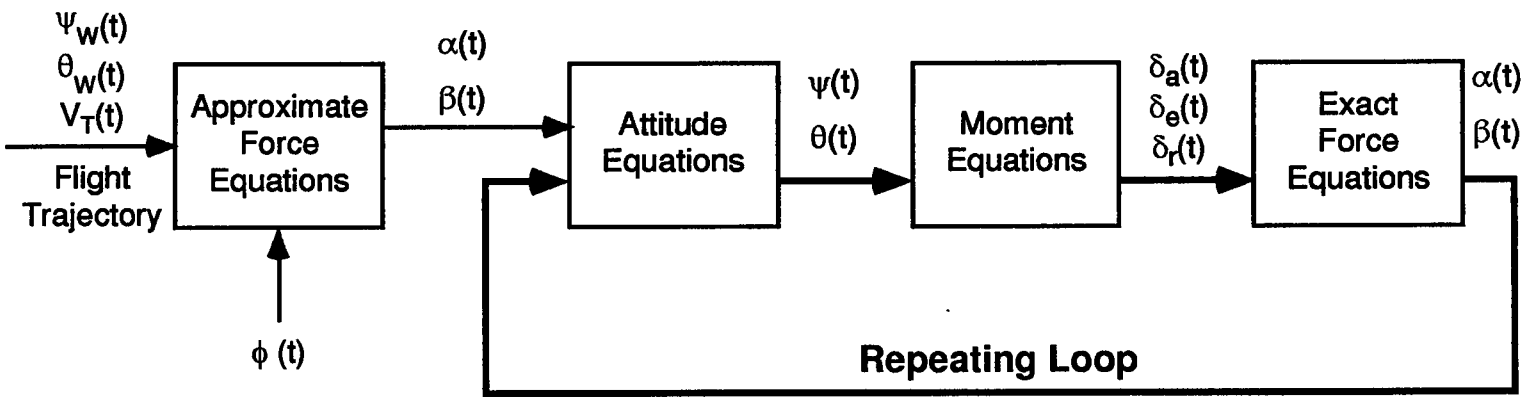


Figure 2.4: Block diagram showing computational steps used by Kato and Sugiura to solve the airplane general motion inverse problem.

i.e.

$$\begin{aligned} \psi &= \psi(\psi_W, \theta_W, \alpha, \beta, \varphi) \\ \theta &= \theta(\psi_W, \theta_W, \alpha, \beta, \varphi) \end{aligned}$$

The path angles (ψ_W, θ_W) should be distinguished from the attitude angles (ψ, θ). Note that the path angles are two of the Euler angles for the wind-axis, while the attitude angles are two of the Euler angles for the body-axis. (Alternatively, they mention how to use their approach using attitude angles as inputs.) Similarly, the angular velocities (p, q, r) and linear velocities (u, v, w) can be written as functions of the path angles, tangent velocity (V_T) and the triplet (α, β, φ). Therefore, the forces acting on the aircraft can be written as functions of the unknown variables (α, β, φ), and the unknown controls (δa,δe,δr,δT). Since the control surfaces (δa,δe,δr) only give a small contribution to the forces, they are neglected at this time. The remaining approximate force equations are then seen to be functions of four unknowns, the angles (α, β, φ) and the thrust δT. A fourth equation is needed in order to simultaneously solve the equations for (α, β, φ). For the numerical example given, a prescribed bank angle φ was chosen. Alternative constraints could be used. One example could be that the flight be coordinated, i.e. that $\beta = 0$. Therefore, approximate values for $(\alpha, \beta, \delta_T)$ can be found by solving the approximate force equations

$$\begin{aligned} F_X(t; \alpha, \beta, \delta_T) &= 0 \\ F_Y(t; \alpha, \beta) &= 0 \\ F_Z(t; \alpha, \beta) &= 0 \end{aligned}$$

Since the force equations are nonlinear, even though the example uses linear aerodynamic forces, a successive iteration method was used to find α and β. The

aircraft moment equations were then used to solve for the control deflections. Since the aerodynamic forces due to the control surfaces were neglected, the solution found is only an approximate one. The control deflections are then substituted into the exact force equations to again solve for the angles (α, β, φ). The entire process of solving the force and moment equations is then repeated until the solutions found for the control deflections converge. Obviously, this method of inversion to find the necessary control can not be used in a real-time implementation because of the computational power required for both the iterative search for the angles (α, β, φ) and the repetitive nature of the algorithm.

Lane and Stengel [14] apply decoupling theory methods developed by Singh and Rugh [15] and Freund [16] to compute the inverse dynamics for a Navion General Aviation aircraft. This amounts to constructing the linearizing feedback u as defined in eq. (2.42). This method requires that the plant dynamics be in the form:

$$\begin{aligned} \dot{x} &= F(x) + G(x)u \\ y &= h(x) \end{aligned} \tag{2.47}$$

where $F(x)$ is an (n $\times$ 1) vector, $G(x)$ is an (n $\times$ m) matrix, and $h(x)$ is an (m $\times$ 1) vector. Since the aircraft dynamics usually take the more general nonlinear form (see the description of the aerodynamic coefficients in equation (2.13)),

$$\begin{aligned} \dot{x}' &= f(x', u') \\ y &= h(x') \end{aligned}$$

a transformation into the form in (2.47) is required. This can be accomplished by augmenting the original system with derivatives of the appropriate control inputs, i.e. modifying the original system into it's extended form (eq. (2.46)). To compute the inverse dynamics of (2.47), the solution to the exact state space

feedback linearization problem outlined in Section 2.2.3 is used, namely that

$$u = A^{-1}(x)[-b(x) + v]. \tag{2.48}$$

Application of this control leaves the original system in the integrator-decoupled form

$$y^* = v$$

with

$$y^* = (y_1^{r_1} \cdots y_m^{r_m})^T.$$

With an appropriate choice for the input v, the original system can be made to perform as desired. Lane and Stengel chose an input of the form

$$v = -\sum_{k=0}^{r-1} P_k y^k + P_0 w \tag{2.49}$$

where P_k are (m $\times$ m) constant matrices chosen so that

$$r = \sum_{i=1}^{m} r_i$$

poles can be placed arbitrarily and w is the new external input. The control law consisting of (2.48) and (2.49) was then simulated for the Navion aircraft with control saturation logic and stall prevention logic installed to evaluate the performance of this control method. The performance was adequate. However, as seen from eq. (2.49), derivatives of the outputs and possibly derivatives of the inputs (to convert the system into extended form) are required for the controller, making this scheme impractical for any actual implementation on a real aircraft. Lane and Stengel [17] present a discrete-time version of the above control laws.

Snell, Enns, and Garrard [18] designed a nonlinear dynamic inverse controller for a highly maneuverable aircraft (representative of the X-31 research aircraft).

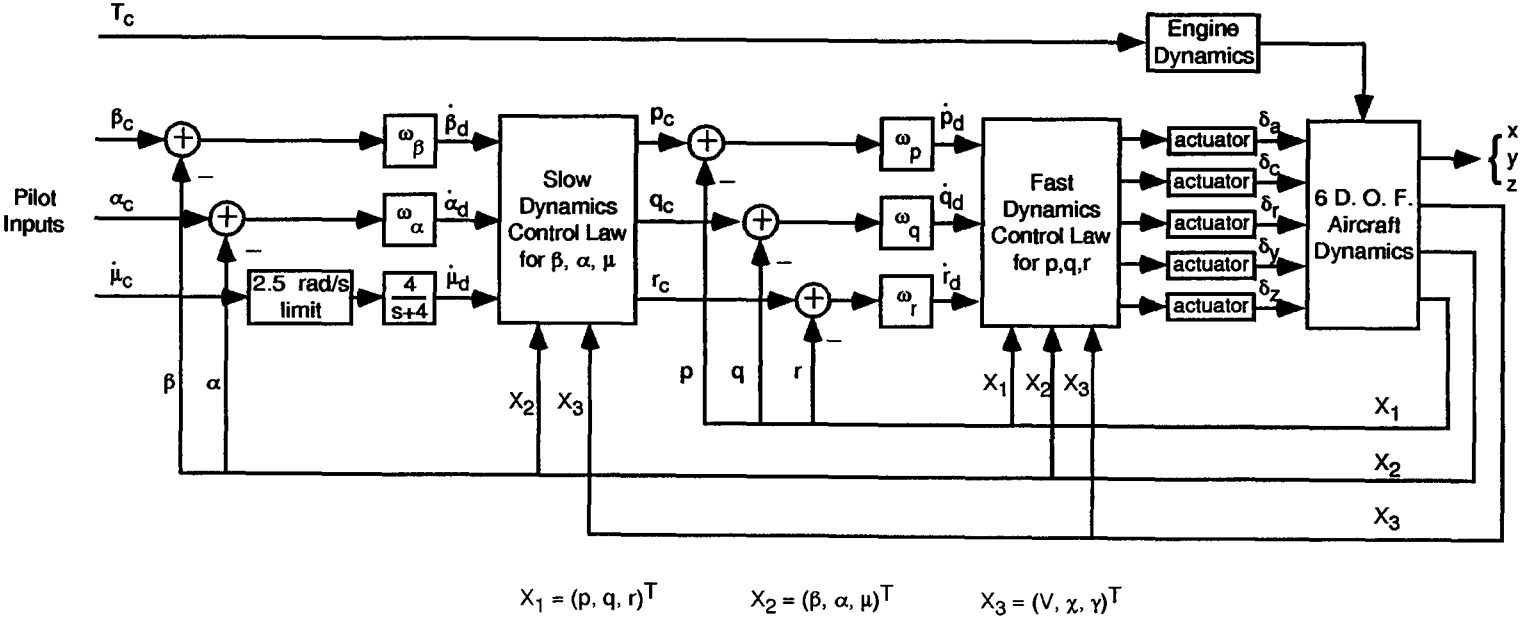


Figure 2.5: Dynamic Inversion Control law of Snell, Enns and Garrard.

This work was motivated by a previous control design using a gain-scheduled proportional plus integral control law [19]. Performance of the controller for a small maneuver was adequate. However, when used with a more aggressive maneuver (reversing the velocity vector in the minimum amount of time), the controller generated excessive amounts of β and lateral acceleration. Their inverse controller design [18] followed the approach of Kokotovic et. al. [20, 21] as applied to aircraft by Menon et. al. [22] to formulate the problem as a two time-scale problem. Separation of the aircraft dynamics into fast and slow systems has two advantages. First, from an implementation point of view, the flight control computer can update certain control loops at a slower rate than others. Second, and more important, the inversion transformations have been greatly simplified. For example, look at the (fast) moment equations

$$\dot{x}_1 = F(x_1, x_2, x_3) + G(x_1, x_2, x_3)u \tag{2.50}$$

where $x_1 = \text{col}(q, p, r)$. Note that $x_2 = \text{col}(\beta, \alpha, \mu)$, $x_3 = \text{col}(V, \chi, \gamma)$ are slower states that can be treated as constants in equation (2.50). G is full rank, and according to Definition 2.3 the system trivially has vector relative degree $\{1,1,1\}$. Since $h(x) = x_1$,

$$\frac{\partial h}{\partial x_1} = I_{3\times 3}$$

and

$$A(x) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} \\ \frac{\partial h_2}{\partial x_1} \\ \frac{\partial h_3}{\partial x_1} \end{bmatrix} G(x) = G(x).$$

Then according to equation (2.42) the linearizing control is

$$u = A^{-1}(x)[-b(x) + v]$$

with

$$b(x) = \begin{bmatrix} L_f h_1(x) \\ L_f h_2(x) \\ L_f h_3(x) \end{bmatrix} = \frac{\partial h}{\partial x_1} F(x) = F(x).$$

Substituting we then get

$$u = G^{-1}[-F(x) + v].$$

If we then let $v = \dot{x}_{1_d} \triangleq \dot{y}_{des}$, we have inverted the system defined by equation (2.50). A similar transformation can be performed on the slower states using the fast states, x_1 as inputs, i.e.

$$\dot{x}_2 = F(x_2, x_3) + G(x_2, x_3)x_1 \tag{2.51}$$

However, the time scale separation must be valid for the maneuvers under consideration and the airframe dynamics given. This is a valid assumption for most currently operational fighter aircraft since the control surface deflections contribute relatively small forces on the airframe.

The fast dynamics, referred to in Figure 2.5 by x_1, were defined by the attitude rates p, q, and r, which were controlled by the five inputs: aileron, canard, rudder, and lateral and normal thrust vectoring (TVC) (δa,δc,δr,δy,δz). The slow dynamics, referred to by x_2, were defined by the angle of attack, α, sideslip angle, β, and bank angle about the velocity vector μ using the rates p, q, and r, as inputs. Simplifications were made to the dynamical equations which allowed the inversion to be done algebraically, namely, that the inputs in each case appeared in an affine manner. The pilot commands (α, β, $\dot{\mu}$) were used as the inputs to the slow dynamic inversion. By neglecting the small terms due to

the control surfaces, the slow dynamics could be written as

$$\begin{bmatrix} \dot{\beta} \\ \dot{\alpha} \\ \dot{\mu} \end{bmatrix} = \begin{bmatrix} f_\beta(\bar{x}_s) \\ f_\alpha(\bar{x}_s) \\ f_\mu(\bar{x}_s) \end{bmatrix} + g(\bar{x}_s) \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

where $\bar{x}_s$ consists of the slowly varying states

$$\bar{x}_s = [V, \beta, \alpha, \mu, \gamma]$$

with γ representing the glide-slope angle. The form of $g(\bar{x}_s)$ is defined by kinematics (see equation (2.22) for $\dot{\alpha}$ and $\dot{\beta}$, $\dot{\mu}$ is similar) and is identical for any aircraft. The matrix is full rank except when $\cos\beta = 0$. Therefore the angular rate commands (p_c, q_c, r_c) can be found by solving

$$\begin{bmatrix} p_c \\ q_c \\ r_c \end{bmatrix} = g^{-1}(\bar{x}_s) \left[\begin{bmatrix} \dot{\beta}_d \\ \dot{\alpha}_d \\ \dot{\mu}_d \end{bmatrix} - \begin{bmatrix} f_\beta(\bar{x}_s) \\ f_\alpha(\bar{x}_s) \\ f_\mu(\bar{x}_s) \end{bmatrix} \right]. \tag{2.52}$$

These commands were then used as inputs to the fast dynamic inversion. The fast dynamics could be written as

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} f_p(\bar{x}) \\ f_p(\bar{x}) \\ f_p(\bar{x}) \end{bmatrix} + g(\bar{x}) \begin{bmatrix} \delta_a \\ \delta_c \\ \delta_r \\ \delta_y \\ \delta_z \end{bmatrix}$$

where $\bar{x}$ is the eight-vector of system states

$$\bar{x} = [V, \beta, \alpha, p, q, r, \mu, \gamma].$$

The 3×5 matrix $g(\bar{x})$ is generically of rank 3 and, therefore the right inverse $g_R^{-1}(\bar{x})$ exists. The commanded control inputs were found from

$$\begin{bmatrix} \delta_a \\ \delta_c \\ \delta_r \\ \delta_y \\ \delta_z \end{bmatrix} = g_R^{-1}(\bar{x}) \left[\begin{bmatrix} \dot{p}_d \\ \dot{q}_d \\ \dot{r}_d \end{bmatrix} - \begin{bmatrix} f_p(\bar{x}) \\ f_p(\bar{x}) \\ f_p(\bar{x}) \end{bmatrix} \right]. \tag{2.53}$$

A specific right inverse was chosen to minimize the norm of the weighted input vector $\hat{u}$ defined by

$$\hat{u} = \begin{bmatrix} \delta_a/\delta_{amax} \\ \delta_c/\delta_{cmax} \\ \delta_r/\delta_{rmax} \\ \delta_y/\delta_{ymax} \\ \delta_z/\delta_{zmax} \end{bmatrix}.$$

As shown in equations (2.52) and (2.53), the "control" could be calculated via a matrix equation because both the slow and fast dynamics assumed a "control" affine form. Bugajski, Enns, and Elgersma [23] took this approach one step further by using a Newton-Raphson inversion algorithm on a nonlinear model of NASA's High Angle-of-Attack Research Vehicle (HARV) to eliminate any restrictions on the manner in which the control inputs enter in the equations of motion. The nonlinear model takes the general form of equation (2.45) and is constructed from aerodynamic table lookups. The Newton-Raphson algorithm enables the linearization to be performed just as in the affine MIMO case (Section 2.2.3) by essentially finding the $A(x)$ matrix at each new desired point. Linearization is simplified as before by separating into fast and slow states. Control surface limit-

ing was accommodated by a real time reduction of the bandwidth of the attitude loop. Again, stability was not studied in the dynamic inversion controllers.

Enns [24] performed a robustness analysis of two control laws applied to a linearized version of the lateral (roll and yaw directions) equations of motion of the same aircraft: one developed using the dynamic inverse method and the other using a μ-synthesis approach. The μ-synthesis was performed for robust performance and model uncertainty. Stability margins, handling qualities, and robustness analysis of the two designs were compared. The μ-synthesis provided better turn coordination, but because no uncertainty was modeled at the sensors, poor stability margins were the result for the yaw rate sensor. Also, the dynamic inverse control had a faster roll rate response to lateral stick inputs. In general, the μ-synthesis design was more robust. However, no robustness properties were designed into the dynamic inverse method. Because of the ability of the dynamic inverse method to handle nonlinearities, some robustness properties should be included in the design method.

Work on controlling the HARV using dynamic inversion based control continued in [25]. Since the control laws would eventually be implemented in the actual aircraft for flight testing, a large portion of the work dealt with defining the proper variables to control and the desirable pilot handling qualities. The pilot would manually close the three velocity loops (magnitude, heading, and flight path angle) and the bank angle loop with throttle, lateral and longitudinal stick, and rudder pedals. The angle-of-attack and pitch-rate loops were combined and the yaw-rate and sideslip loops were combined. Therefore, the loops to control automatically corresponded to roll, pitch, and yaw. The control variables were predominantly defined by roll rate, a blend of pitch rate and angle-of-attack, and

a blend of yaw rate and sideslip angle. The control variables, CV, were modeled with differential equations

$$\dot{CV} = F(x, u) \tag{2.54}$$

which depend on aircraft state variables (x) and control effectors (u). The differential equations are just the rigid body equations of motion. The state variables were assumed to be measured, and the control effectors take the form of aerodynamic surfaces, and thrust vectoring. A desired rate of change of the control variable was selected to achieve satisfactory handling qualities in response to pilot commands. This was modeled with a first order differential equation

$$\dot{CV}^{desired} = \frac{1}{\tau}(CV^{cmd} - CV^{meas}) \tag{2.55}$$

with a specified time constant for each control variable. The two differential equations above (eqs. (2.54) and (2.55)) were then equated, leaving an expression

$$F(x^{meas}, u^{cmd}) = \frac{1}{\tau}(CV^{cmd} - CV^{meas}) \tag{2.56}$$

to be solved for the control effectors (u^{cmd}). To facilitate solving equation (2.56), a reduced aerodynamic data base was obtained with a least squares fit to HARV data for a limited flight envelope (Mach $<$ 0.7, 15,000 ft $<$ h $<$ 45,000 ft). Fixed values of Mach = 0.4, h = 25,000 ft and $\dot{\alpha} = 0$ were used to produce the reduced database for the limited flight envelope. Thus for the reduced database, dependence on Mach, altitude, and dynamic angle-of-attack effects were neglected. Furthermore, it was assumed that the aerodynamic coefficients are linear in everything except angle-of-attack with angle-of-attack dependent multipliers. More precisely, the aerodynamic coefficients were written as

$$C_k = C_k(\alpha) + C_{k_\beta}(\alpha)\beta + C_{k_p}(\alpha)p + C_{k_q}(\alpha)q$$

$$+C_{k_r}(\alpha)r + C_{k_{\delta_a}}(\alpha)\delta_a + C_{k_{\delta_e}}(\alpha)\delta_e + C_{k_{\delta_r}}(\alpha)\delta_r \qquad (2.57)$$

where k = D, Y, L, l, m, n (drag, side force, lift, roll, pitch, yaw). The reduced model was checked against the full database model (visually by comparing plots) at various points in the limited flight envelope. As plots of the aerodynamic coefficients showed, the agreement between the models was good. Using this reduced database, the left hand side of equation (2.56) reduces to

$$F(x^{meas}, u^{cmd}) = f(x^{meas}) + g(x^{meas})u^{cmd}. \qquad (2.58)$$

Eq (2.58) was then equated with eq. (2.56) to yield

$$f(x^{meas}) + g(x^{meas})u^{cmd} = \frac{1}{\tau}(v^{cmd} - v^{meas})$$

which can be solved for u^{cmd}. The dynamic inversion then amounts to simple algebra using the inverse of the matrix $g(x^{meas})$ to find u^{cmd}. If the minimum singular value of g falls below a certain number (usually at high α conditions where surfaces become ineffective), it is replaced by a predefined constant matrix to ensure that the matrix is invertible. Since the ability to achieve a perfect inversion is limited by modeling errors, actuator dynamics which were unaccounted for, delays, etc., an integrator was placed on the control variable error signal. This decreases the sensitivity of the closed loop response to modeling errors and low frequency atmospheric disturbances and implies that the control variable will have zero steady state error in response to step commands. Desirable pilot handling qualities were achieved by precompensation of stick and pedal commands and trim inputs. The pilot inputs were scaled with flight condition dependent gains and a first order shaping filter to achieve time constants other than those resulting from feedback objectives. Figure 2.6 depicts the dynamic inversion control law used on the HARV.

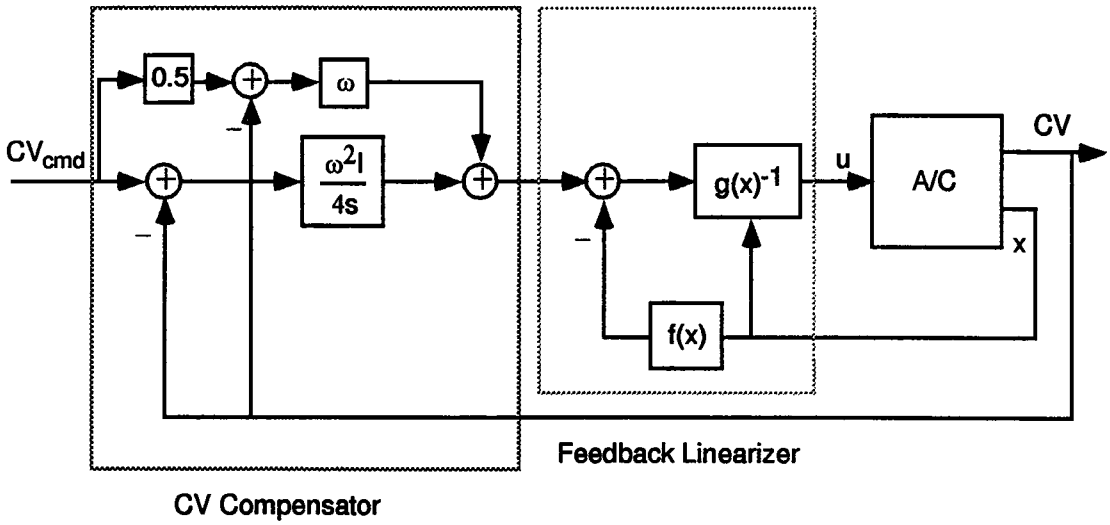


Figure 2.6: Dynamic Inverse controller used on the HARV.

Stability and robustness analysis of the closed-loop system was performed on linearized versions of the system at 11 flight conditions throughout the flight envelope. The longitudinal and lateral/directional axes were analyzed separately. The linear analysis consisted of closed-loop poles, frequency responses, singular value, and structured singular value tests. The phugoid mode was unstable for five of the 11 flight conditions. To test robust stability, a multiplicative unstructured uncertainty was added at the interface between the actuators and the aircraft. By looking at the maximum singular value of the closed-loop transfer function from actuator to aircraft, it was shown that modeling errors up to 50% could be tolerated in the longitudinal case and up to 33% for the lateral/directional case. Multiplicative uncertainties, in this case uncoupled - i.e. diagonal, were also added to the sensed quantities to check for robust stability. The structured singular value (μ) was used because of the diagonal nature of the uncertainties. It was shown that large values for μ occurred at low frequencies in the longitudinal axis (phugoid mode) and hence not much uncertainty could

be tolerated. However, other than this frequency range, it was shown that about the same level of uncertainty could be tolerated at the sensors as was shown for the actuators. The lateral/directional axes were shown to tolerate about 33% uncertainty at the sensor locations. The nonlinear analysis primarily consisted of nonlinear simulation. Because a directional instability was observed during real-time simulation, caused by real-time simulator delays coupled with actuator rate limiting, some describing function analysis was also done. This analysis revealed that the bandwidth of the yaw axis should be reduced.

Modifications to the HARV control laws were made in [26]. The modifications mostly attempted to fix the deficiencies found from testing in [25] and increase stability. For example, a velocity feedback term was added to the pitch control variable (MCV) to increase stability of the phugoid mode. Also, an α-limiting loop was added to the commanded pitch control variable (MCV^{cmd}) to impose "protective" limits on the aircraft such as those derived from structural g-limits. In addition, the longitudinal and lateral/directional axes were split up in the dynamic inversion routine. This allowed the longitudinal axis to be inverted first so as to give pitch commands first priority. The longitudinal control effector, δ_e was found from inversion of the pitch control variable equation. The lateral/directional control effectors (δ_a, δ_r, and δ_e') were found from inversion of the roll and yaw control variable equations. Linear stability and robustness analysis was performed similar to [25]. For the 21 flight conditions tested, the longitudinal axis was unstable for one flight condition and the lateral/directional axes were unstable for six flight conditions. The robust stability results were similar to those found in [26]. More work must be performed to resolve the stability issues. Disagreements were also found between the linear

and nonlinear simulations for the lateral/directional axes. Since the nonlinear simulation shows the desired behavior, the control law definitions in the linear simulation may be incorrect. Overall, the stability and robustness analysis performed in [25] and [26] amount to "standard" linear methods. Coupling between the longitudinal and lateral/directional axes, along with other nonlinear effects, cannot be accounted for with these methods. In addition, as was stated in [25], it was assumed that all the states were available for feedback. However, β was not measured and therefore modifications to the control laws must be made.

Enns et. al. [27] give an overview of the dynamic inversion concept as an alternative design method for flight controls. This work summarizes earlier reports [25, 26] and again emphasizes that the selection of proper controlled variables is central to the performance and robustness of the dynamic inverse method. Robustness was shown about linearizations of the aircraft model. It was stated that this design method will benefit substantially from additional research efforts, particularly in the areas of nonlinear zero dynamics and nonlinear robustness. There remains one other problem using this method - all the states must be measured, since full-state information is needed for the dynamic inverse controller. However, as commented in [27], most modern aircraft now carry a full compliment of sensors. The only remaining sensing issues revolve around airstream-relative measurements, i.e. true airspeed, angle-of-attack, and sideslip angle. The current solution to this problem is to use blends of inertial measurements and air-data for these signals. The paper concluded with an example of the dynamic inversion method applied to the F-18 HARV.

In summary, Meyer and Cicolani first proposed the dynamic inversion technique because a control system was needed which could follow rough trajectory

commands despite strong nonlinearities in the description of the plant. Simple aircraft models were used and inversion was performed using pure table lookups. Smith and Meyer used this method in a simulation study of automatic landing of an A7-E aircraft on a carrier. Flight envelope limiting was added for this study. They next applied the method to a VATOL aircraft simulation. In this case, inversion was performed using an iterative Newton-Raphson procedure instead of table lookups. Kato and Suguira performed a theoretical study of airplane general motion to show that separation simplifications between translational and rotational motion made by Meyer, et. al. were valid. Lane and Stengel then applied the more general method of feedback linearization to a general aviation aircraft. After the linearization, they chose a specific form for the new input so that some of the closed-loop poles could be arbitrarily placed. Snell, Enns, and Garrard designed a dynamic inverse controller for a representation of the X-31 research aircraft. They used a simple description of the aircraft dynamics which allowed the inversion to be performed algebraically. Bugajski, Enns, and Elgersma took this approach one step further by using a Newton-Raphson inversion algorithm on a nonlinear model of NASA's High Angle-of-Attack Research Vehicle (HARV) to eliminate any restrictions on the manner in which the control inputs enter in the equations of motion. Work on the HARV continued by choosing the proper variables to control in order to satisfy pilot handling qualities. A reduced aerodynamic model was constructed for a limited flight envelope by neglecting the dependence on Mach, altitude, and dynamic angle-of-attack effects. The reduced model was affine in the controls so the inversion could again be performed algebraically. The next logical step would be to describe an inverse method that can address the more general non-affine input case for more

complex systems. Since the aircraft dynamics will not be known exactly, the method must work well despite uncertainties in the system description. We will outline such a method and show its performance with full nonlinear simulations in the remaining chapters.

2.4 Robust Feedback Linearization

In the previous sections, we outlined methods for computing the control needed for a nonlinear system to follow a commanded trajectory. As was shown, explicit knowledge of the system to be controlled was needed to calculate the "inverting" control. In order to make the control calculations robust to modeling uncertainties and unmodeled dynamics, additional terms must be added to the nominal feedback linearizing control. This section outlines an approach by Slotine and Hedrick [28] to solve the robust feedback linearization problem. The method is based on sliding control and the reader is referred to Slotine and Li [29] for a complete description.

We consider a single-input nonlinear system modeled as

$$\dot{x} = f(x) + g(x)u \tag{2.59}$$

$$y = h(x). \tag{2.60}$$

where the state vector $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ and is available for measurement, $y \in \mathbb{R}^p$. Assume that the system's relative degree is equal to the system order n, and that the system's model relative degree is also equal to n. Assume that f is not exactly known, but approximated as $\hat{f}$, with

$$\hat{f} = f + \tilde{f} \tag{2.61}$$

We assume that the functions $h(x)$ and $g(x)$ are exactly known. The control problem is to have the output y track a desired output trajectory y_d (we assume that y_d and its time derivatives up to order n are given and bounded) while maintaining all system states bounded.

Based on the available model (eqs. (2.59)-(2.60)) we now compute the normal states as

$$\begin{aligned} \mu_1 &= y \\ \mu_2 &= L_{\hat{f}} h \\ \mu_3 &= L_{\hat{f}}^2 h \\ &\vdots \\ \mu_n &= L_{\hat{f}}^{n-1} h \end{aligned} \tag{2.62}$$

Because of model uncertainties (eq. (2.61)), the μ_i are not simply successive derivatives of the output y but instead are

$$\begin{aligned} \dot{\mu}_1 &= \mu_2 + D_1 \\ \dot{\mu}_2 &= \mu_3 + D_2 \\ &\vdots \\ \dot{\mu}_n &= L_{\hat{f}}^n h + L_g L_{\hat{f}}^{n-1} h u + D_n \end{aligned} \tag{2.63}$$

where

$$\begin{aligned} D_1 &= -L_{\tilde{f}} h \\ D_2 &= -L_{\tilde{f}} L_{\hat{f}} h \\ &\vdots \\ D_n &= -L_{\tilde{f}} L_{\hat{f}}^{n-1} h. \end{aligned} \tag{2.64}$$

We need to compute the trajectories, z_i for $i = 1, \ldots, n$ we wish the system (2.63) to follow. We define $z_1 = y_d$ with z_2 through z_n to be defined later based on a Lyapunov-like argument. Define

$$s_i = \tilde{\mu}_i - \phi_i \text{sat}(\tilde{\mu}_i / \phi_i) \quad i = 1, \ldots, n \tag{2.65}$$

where $\tilde{\mu}_i = \mu_i - z_i$, the ϕ_i are strictly positive quantities, and sat is the saturation function

$$\text{sat}(x) = \begin{cases} \text{sgn}(x) & \text{if} \quad |x| \geq 1, \\ x & \text{otherwise.} \end{cases}$$

By definition of s_i (eq.(2.65)), when the $\text{sat}(\tilde{\mu}_i/\phi_i)$ is not in effect ($\tilde{\mu}_i < \phi_i$) on a time interval $[t_1,\ t_2]$ $s_i = 0$ and hence $\dot{s}_i = 0$ on (t_1, t_2).

Otherwise we get

$$\begin{aligned} \dot{s}_1 &= \mu_2 + D_1 - \dot{z}_1 \\ &= s_2 + \phi_2 \text{sat}(\tilde{\mu}_2/\phi_2) + z_2 - \dot{z}_1 + D_1 \\ \dot{s}_2 &= \mu_3 + D_2 - \dot{z}_2 \\ &= s_3 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3) + z_3 - \dot{z}_2 + D_2 \\ &\vdots \\ \dot{s}_n &= L_{\hat{f}}^n h + L_g L_{\hat{f}}^{n-1} h u - \dot{z}_n + D_n \end{aligned} \tag{2.66}$$

We will select the z_i as known functions of the states. Thus, their derivatives in the above equations are not known exactly, but rather are estimated as $\dot{z}_{ie}$ based on the system model.

Letting

$$\begin{aligned} \Delta_i &= D_i + \dot{z}_{ie} - \dot{z}_i + \phi_{i+1} \text{sat}(\tilde{\mu}_{i+1}/\phi_{i+1}) \quad i = 1, \ldots, n-1 \\ \Delta_n &= D_n + \dot{z}_{ne} - \dot{z}_n \end{aligned} \tag{2.67}$$

we get (for $s_i \neq 0$)

$$\begin{aligned} \dot{s}_1 &= s_2 + z_2 - \dot{z}_{1e} + \Delta_1 \\ \dot{s}_2 &= s_3 + z_3 - \dot{z}_{2e} + \Delta_2 \\ &\vdots \\ \dot{s}_n &= L_{\hat{f}}^n h + L_g L_{\hat{f}}^{n-1} h u - \dot{z}_{ne} + \Delta_n. \end{aligned} \tag{2.68}$$

Assuming that state dependent bounds on the modeling error $\tilde{f}$ are known, then one can compute state dependent functions F_i such that

$$|\Delta_i| \leq F_i \quad i = 1, \ldots, n. \tag{2.69}$$

Consider the Lyapunov function candidate

$$V = \frac{1}{2} \sum_{i=1}^{n} \sigma^{n-i} s_i^2 \tag{2.70}$$

where σ is a strictly positive constant. We then get

$$\begin{aligned} \dot{V} &= \sigma^{n-1} s_1 (s_2 + z_2 - \dot{z}_{1e} + \Delta_1) \\ &\quad + \sigma s_{n-2} (s_3 + z_3 - \dot{z}_{2e} + \Delta_2) \\ &\quad + \cdots + s_n (L_{\tilde{f}}^n h + L_g L_{\tilde{f}}^{n-1} h u - \dot{z}_{ne} + \Delta_n). \end{aligned}$$

Define

$$\begin{aligned} z_1 &= y_d \\ z_2 &= \dot{z}_{1e} - F_1 \text{sat}(\tilde{\mu}_1/\phi_1) - \lambda s_1 \\ z_3 &= \dot{z}_{2e} - F_2 \text{sat}(\tilde{\mu}_2/\phi_2) - \lambda s_2 - \sigma s_1 \\ z_4 &= \dot{z}_{3e} - F_3 \text{sat}(\tilde{\mu}_3/\phi_3) - \lambda s_3 - \sigma s_2 \\ &\cdots \end{aligned} \tag{2.71}$$

where λ is a strictly positive constant. The last term in this recursive construction, z_n is such that $\dot{z}_{ne}$ may itself contain the input u.

One can rewrite $\dot{z}_{ne}$ in the general form

$$\dot{z}_{ne} = \dot{z}_{nex} + \dot{z}_{neu} u.$$

If we choose the control input as

$$u = (L_g L_{\tilde{f}}^{n-1} h - \dot{z}_{neu})^{-1} (-L_{\tilde{f}}^n h + \dot{z}_{nex} - F_n \text{sat}(\tilde{\mu}_n/\phi_n) - \lambda s_n - \sigma s_{n-1}) \tag{2.72}$$

we then have

$$\dot{V} + 2\lambda V \leq 0 \tag{2.73}$$

which shows that the s_i all exponentially converge to zero [30]. Hence the error terms, $\tilde{\mu}_i$ are asymptotically bounded by ϕ_i. More importantly, since $\mu_1 = y$ and $z_1 = y_d$, we have that the output asymptotically approaches to within ϕ_1 of y_d.

Proof of equation 2.73

$$\begin{aligned}
\dot{V} &= \sigma^{n-1}s_1(s_2 - F_1\text{sat}(\tilde{\mu}_1/\phi_1) - \lambda s_1 + \Delta_1) \\
&\quad +\sigma^{n-2}s_2(s_3 - F_2\text{sat}(\tilde{\mu}_2/\phi_2) - \lambda s_2 - \sigma s_1 + \Delta_2) \\
&\quad +\cdots + s_n(L_{\hat{f}}^n h + (L_g L_{\hat{f}}^{n-1} h - \dot{z}_{neu})(L_g L_{\hat{f}}^{n-1} h - \dot{z}_{neu})^{-1} \\
&\quad \left\{-L_{\hat{f}}^n h + \dot{z}_{nex} - F_n\text{sat}(\tilde{\mu}_n/\phi_n) - \lambda s_n - \sigma s_{n-1}\right\} - \dot{z}_{nex} + \Delta_n) \\
&= -2\lambda V + \sigma^n s_1\left[\Delta_1 - F_1\text{sat}(\tilde{\mu}_1/\phi_1)\right] \\
&\quad +\sigma^{n-1}s_2\left[\Delta_2 - F_2\text{sat}(\tilde{\mu}_2/\phi_2)\right] \\
&\quad +\cdots + s_n\left[\Delta_n - F_n\text{sat}(\tilde{\mu}_n/\phi_n)\right]
\end{aligned}$$

If the saturation, $\text{sat}(\tilde{\mu}_i/\phi_i)$ is not in effect, then $s_i = 0$. Otherwise, when $\tilde{\mu}_i > 0$ and $|\tilde{\mu}_i| \geq \phi_i$,

$$\begin{aligned}
s_i &= \tilde{\mu}_i - \phi_i\text{sat}(\tilde{\mu}_i/\phi_i) \\
&= \tilde{\mu}_i - \phi_i \\
&\geq 0.
\end{aligned}$$

We also have that

$$\Delta_i - F_i\text{sat}(\tilde{\mu}_i/\phi_i) = \Delta_i - F_i \leq 0.$$

Therefore, $s_i\left[\Delta_i - F_i\right] \leq 0$. When $\tilde{\mu}_i < 0$ and $|\tilde{\mu}_i| \geq \phi_i$,

$$\begin{aligned}
s_i &= \tilde{\mu}_i - \phi_i\text{sat}(\tilde{\mu}_i/\phi_i) \\
&= \tilde{\mu}_i - \phi_i \\
&\leq 0.
\end{aligned}$$

We also have that

$$\Delta_i - F_i \text{sat}(\tilde{\mu}_i/\phi_i) = \Delta_i + F_i \geq 0$$

Therefore, $s_i [\Delta_i - F_i] \leq 0$.

So for all values of $\tilde{\mu}_i$, we have that

$$s_i [\Delta_i - F_i \text{sat}(\tilde{\mu}_i/\phi_i)] \leq 0$$

therefore we get

$$\dot{V} \leq -2\lambda V$$

or

$$\dot{V} + 2\lambda V \leq 0.$$

Chapter 3

Dynamic Inversion Algorithms

We have designed a nonlinear dynamic inverse model-following control law using Grumman's X-29 forward-swept-wing research aircraft as a test vehicle [31]. To generate the smooth output we wish to track, the first part of the controller consists of a zero-input model driven by an initial condition which generates the desired output. In practice, part of the model would be replaced by the pilot stick command which would be the input to the remaining part of the model. The new model would generate an output which satisfies the handling requirements for the aircraft. Before we give a detailed description of the design procedure, let us first give some motivation for using the nonlinear control method.

We will compare the performance of an LQR control law to a nonlinear dynamic inverse controller, both applied to the nonlinear simulation. The control actuators will be ignored during construction of both controllers, but will be included in the simulation. To construct the LQR control we first need to linearize the plant about some operating point. We chose to linearize about Mach 0.6 at 15,000 ft. The states for the linear system are w, q, u, θ with canards, flaps, and

strakes as the input surfaces. With this four state model we attempted to design an LQR model-following controller [32]. The model following error could not be decreased to an acceptable level without excessively high gains in addition to a nonzero steady state error. Therefore, we add an integral path to the controller. The performance index we want to minimize is

$$J = \int_0^\infty \{u'(t)Ru(t) + x'(t)Q_1x(t) + e'(t)Q_2e(t) + e_i'(t)Q_ie_i(t)\}dt \qquad (3.1)$$

where $R = R' > 0$, $Q_1 = Q_1' \geq 0$, $Q_2 = Q_2' \geq 0$, $Q_i = Q_i' \geq 0$, $e(t)$ is the error between the plant and the model, and $\dot{e}_i = e = y - y_d$. We chose a 20 deg step command in pitch to evaluate the controllers. Figure 3.1 shows the performance of the (fixed) linear simulation along with the commanded controls using the LQR controller with $Q_1 = I$, $Q_2 = 500$, $Q_i = 500$, and R =diag(1,1,50) to minimize the flaperon movement. This produces the following K matrix

$$K = \begin{bmatrix} 1.36e{-}01 & 1.52e{+}02 & 2.44e{-}01 & 1.46e{+}03 & -2.23e{+}01 & -1.47e{+}01 & 2.14e{+}01 \\ -3.94e{-}01 & -6.23e{+}01 & 7.38e{-}01 & -3.52e{+}02 & 7.35e{+}00 & 3.39e{+}00 & -6.49e{+}00 \\ -6.39e{-}02 & -3.85e{+}00 & 4.10e{-}02 & 3.54e{+}00 & 2.60e{-}01 & -5.64e{-}02 & -1.74e{-}01 \end{bmatrix}$$

The first four columns are the state feedback gains, while the fifth and sixth columns are the feedforward gains from the trajectory model and the last column is the gain on the integral of the error. Note that in actuality the control commands would be added to the control trim values. Figure 3.2 shows the performance of the nonlinear simulation using the same LQR controller. Along with the desired (commanded) θ, we show the output from a linear simulation (the design point) and the output from the nonlinear simulation. The trim values for θ, and the control surfaces are subtracted off of the nonlinear simulation signals to facilitate a direct comparison with the linear simulation case. Clearly we can

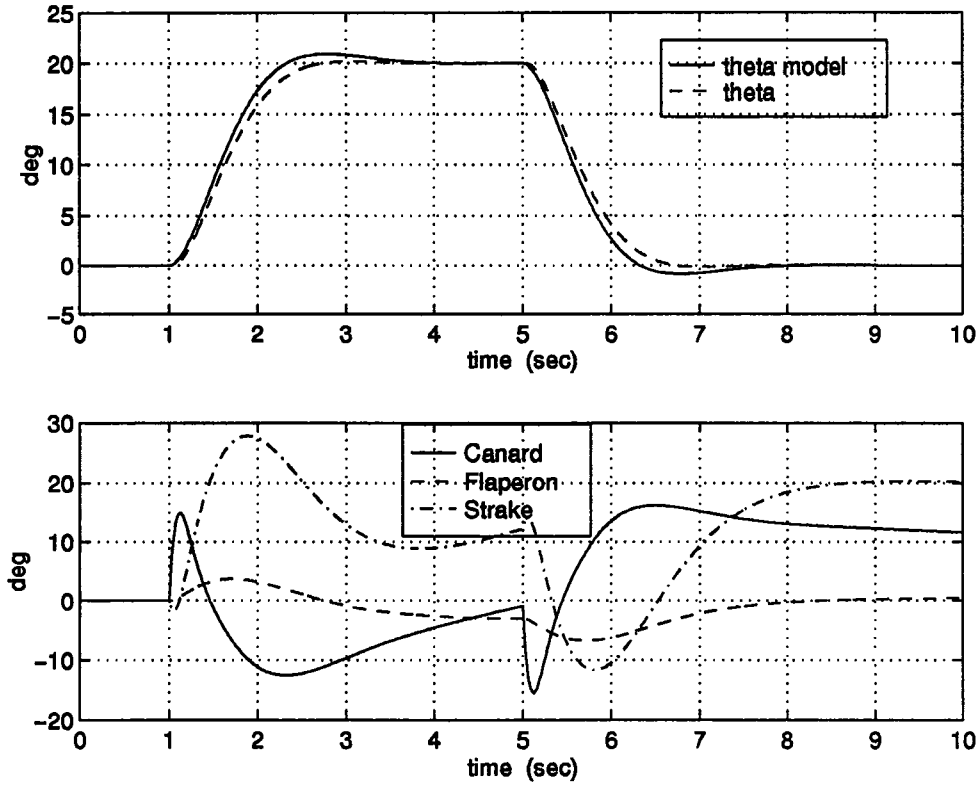


Figure 3.1: Performance of linear X-29 simulation with LQR model-following controller.

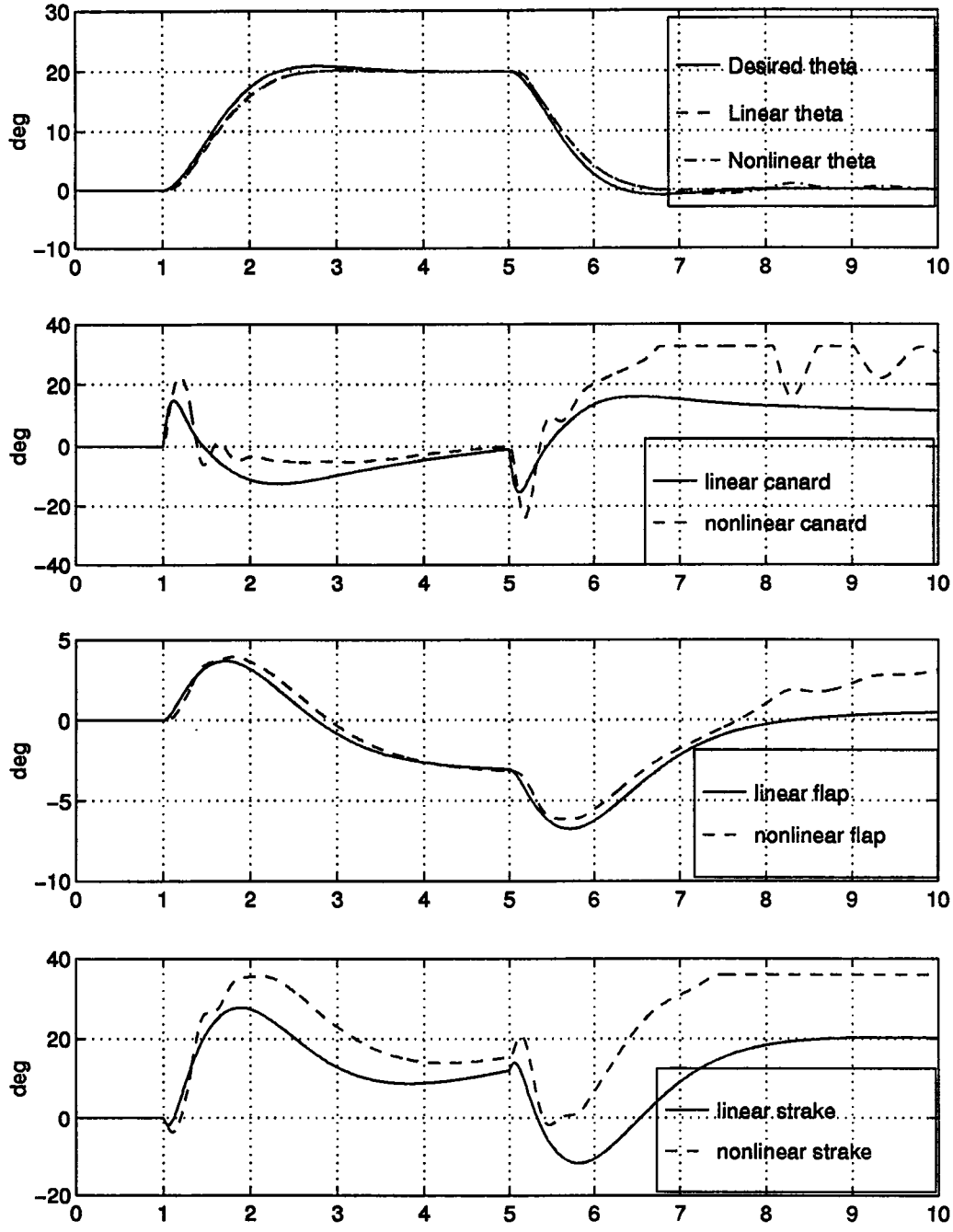


Figure 3.2: Performance of X-29 with LQR model-following controller.

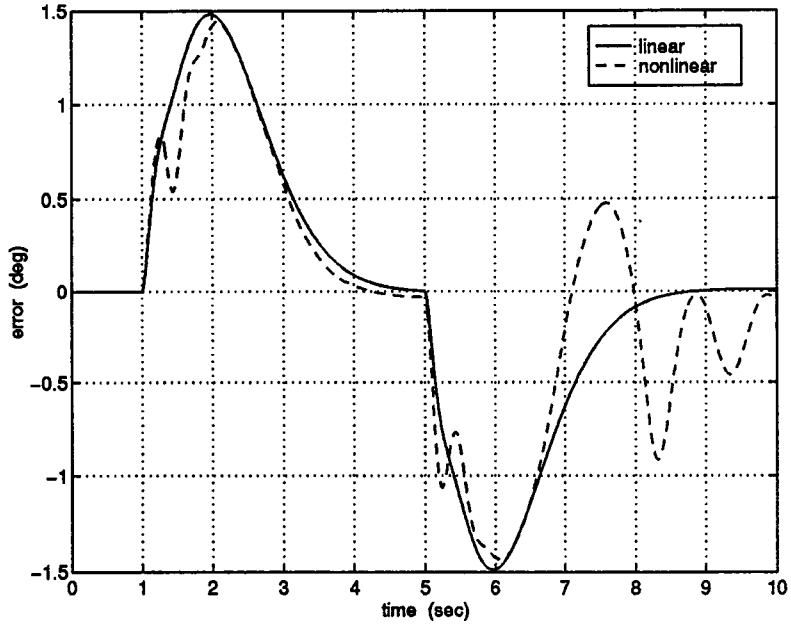


Figure 3.3: Comparison of LQR model-following error.

see the large difference between the expected performance as shown by the linear simulation and the actual performance of the full nonlinear simulation. This is partly due to the fact that the actuators are not modeled in the linear simulation. These effects can be seen in the oscillations of the canard around $t = 1.5$ sec. and $t = 8.3$ sec. The oscillations also end up at the output, θ as we see more clearly by looking at the error shown in Figure 3.3. Notice that both the canard and the strake become saturated from around 7 seconds on. This occurs because of the change in flight condition by that time. Figure 3.4 shows a comparison of the remaining states, w, q, and u for the linear simulation and the nonlinear simulation, both using the same LQR controller. Again, for the nonlinear case, only the change in velocities from the trim condition are shown. We have not shown that by the end of the simulation the height has increased by 700 ft. This change together with the decrease in forward velocity, u, is enough to change

the aerodynamic coefficients used to calculate the aircraft forces and moments (see equations 4.1–4.3). Also, the force due to thrust is neglected in the linear simulation. These changes are enough to cause the nonlinear simulation model to deviate from the (fixed) linear model.

Figure 3.5 shows the performance of the simulation using the nonlinear dynamic inverse controller outlined in this chapter. Specifically, the controller is calculated via equations (3.21) and (3.28) with gains: $K_1 = 225$, $K_2 = 75$,$K_3 = 11$. These gains place the poles of the error dynamics at

$$\begin{matrix} -5 \\ -3 \pm 6i \end{matrix} \tag{3.2}$$

Again, the trim values are subtracted off of the signals for direct comparison to the linear case. As shown in the figure, the nonlinear controller tracks the commanded θ very well. Figure 3.6 shows the remaining states, w, q, and u for the nonlinear simulation. Again, only the change in velocities from the trim condition are shown. For this simulation and the robust control construction in the next chapter, we have swapped the roles of α and θ since θ is usually the regulated variable for an autopilot and α is usually not regulated.

A brief description of the procedure for designing a nonlinear dynamic inverse model-following controller now follows. Let the aircraft dynamics be described by the general nonlinear state equation

$$\dot{x} = f(x, u) \tag{3.3}$$

and the linear observation equation

$$y = Cx, \tag{3.4}$$

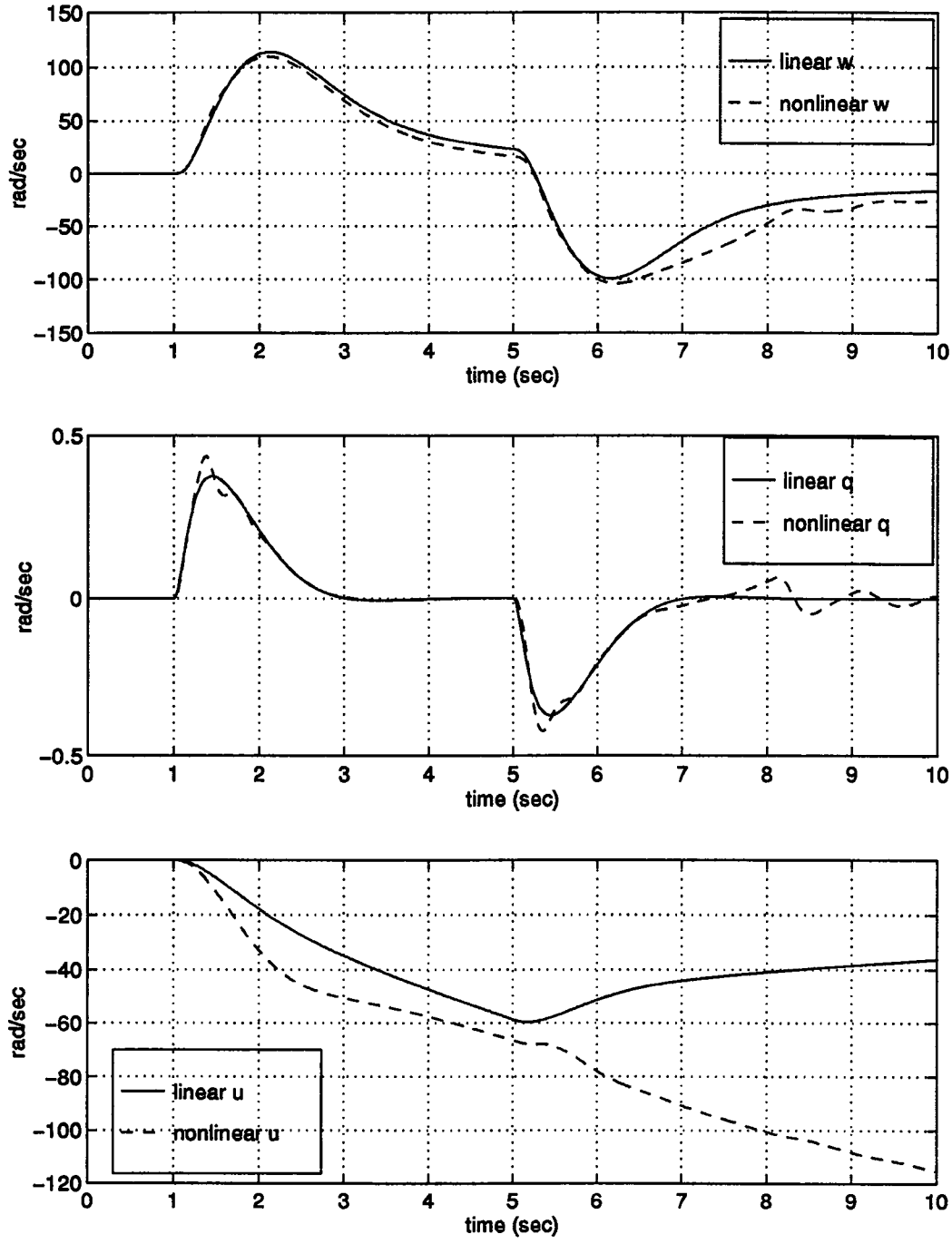


Figure 3.4: Comparison of states with LQR model-following controller.

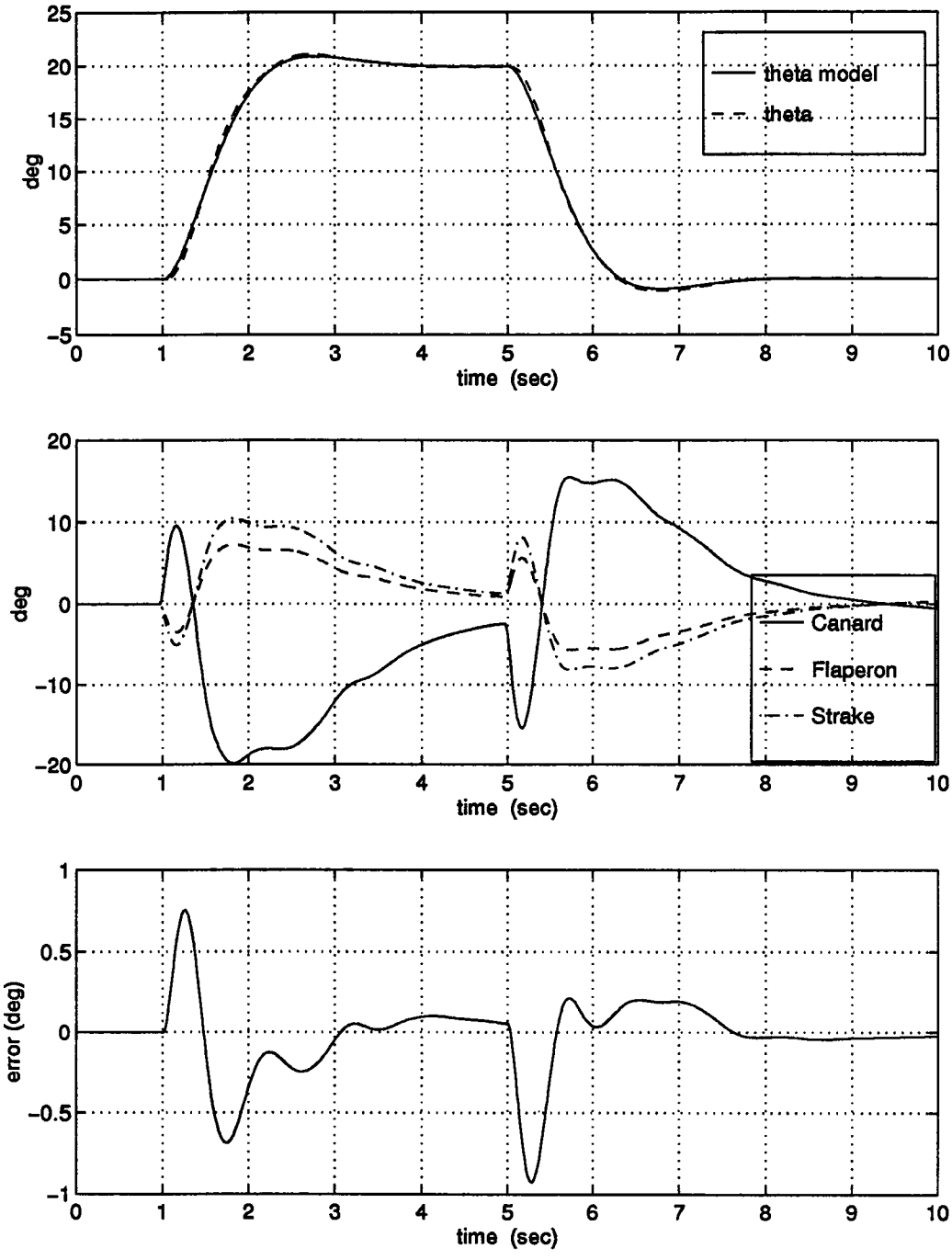


Figure 3.5: Performance of X-29 with nonlinear model-following controller.

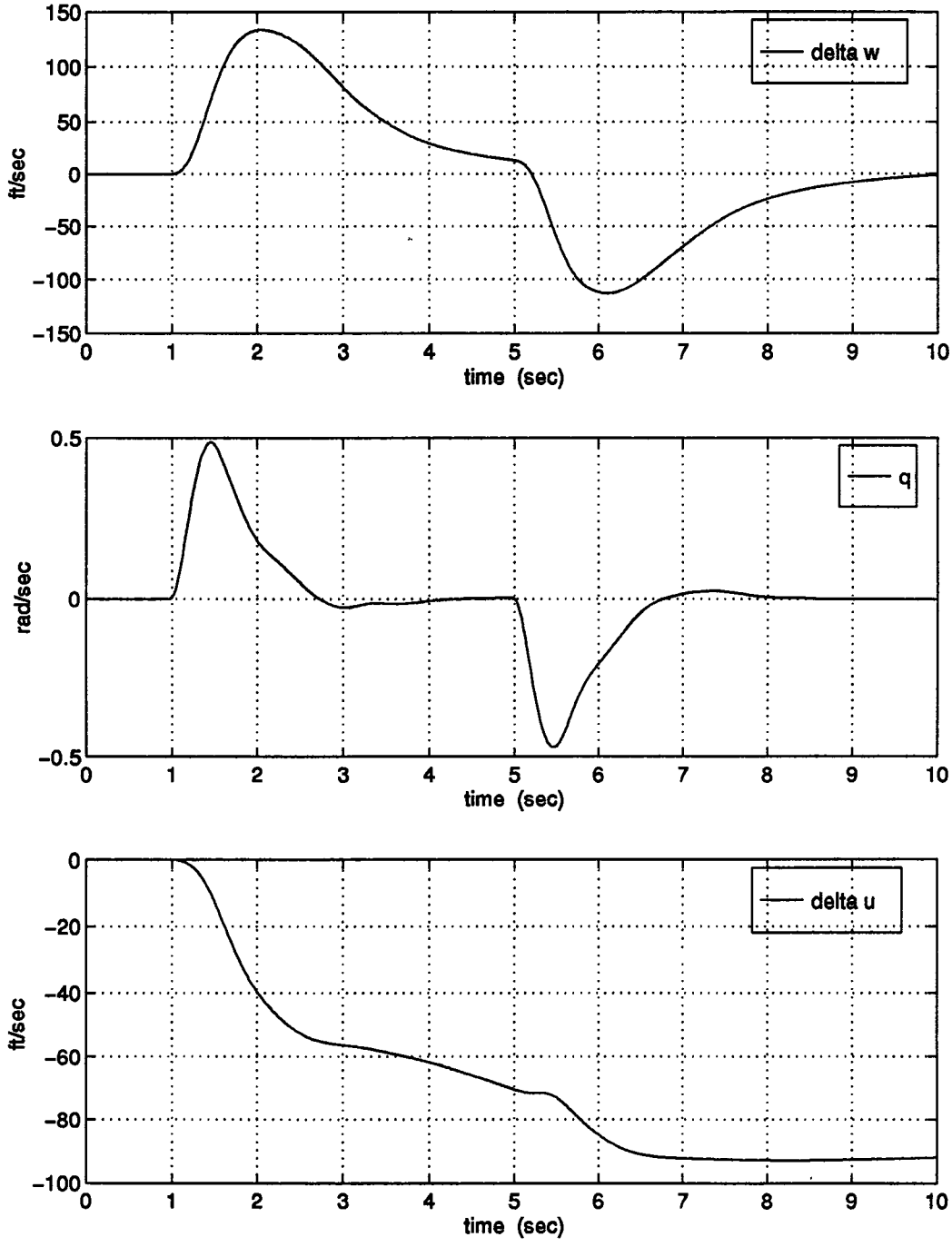


Figure 3.6: States for the nonlinear model-following controller.

For ease of discussion, we have neglected the sensor dynamics. In fact, the sensor dynamics can be neglected because they are much faster than the plant dynamics. If the bandwidth of the sensor dynamics is close to that of the controller, we can easily include the sensor dynamics in the description of the plant. Given some desired output $y_d(t)$ and that $x(t)$ is available for feedback we would like to find a control u such that $y = y_d$. Because of the general way in which the control enters into the state equation (see equation (3.3) e.g. it is not in control affine form), we use an approach similar to [23] to do the inversion. Using an elaboration of the two-time-scale approach of Kokotovic [21] as applied to aircraft by Menon et. al. [22], we partition the dynamics of the aircraft into the fast attitude rate states, x_2, the slow attitude angle states, x_1, and the "exogenous" states, x_3

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ with } x_1 = \begin{pmatrix} \alpha \\ \varphi \\ \beta \end{pmatrix}, x_2 = \begin{pmatrix} q \\ p \\ r \end{pmatrix}, x_3 = \begin{pmatrix} \theta \\ V \end{pmatrix} \text{ and } y = x_1 \quad (3.5)$$

where q is the pitch rate, p is the roll rate, r is the yaw rate, α is the angle of attack, φ is the roll angle, β is the sideslip angle, θ is the pitch angle, and V is the forward speed. Since θ is usually not a regulated state, and V has dynamics much slower than the attitude angles, θ and V have been lumped together into x_3. Typically the state equations can be simplified as

$$\dot{x}_1 = f_1(x_1, x_2, x_3) \quad (3.6)$$

$$\dot{x}_2 = f_2(x_1, x_2, x_3, u) \quad (3.7)$$

where we have assumed that the control does not affect the slow states. This is a reasonable assumption as stated in the previous chapter, because the control surfaces do not contribute large forces. We can measure x_3, and we know that the

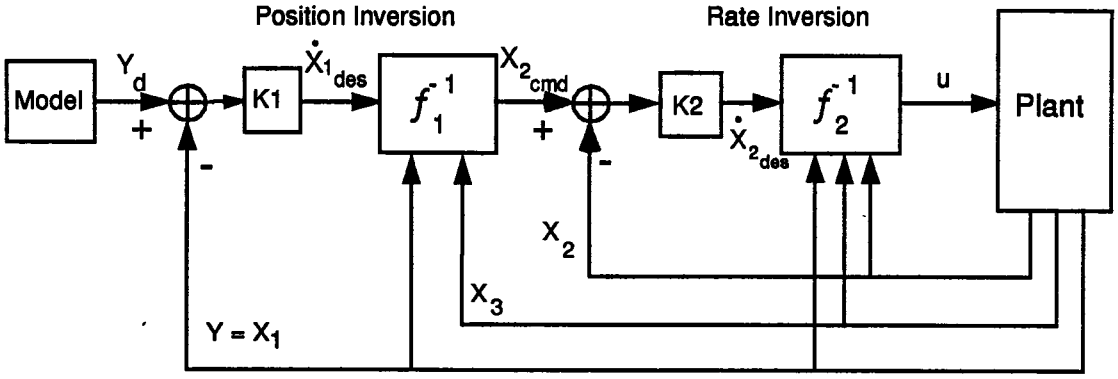


Figure 3.7: Nonlinear dynamic inverse model-following controller

rates x_2 evolve much faster than the positions x_1. Then given y_d and measured outputs y, we can approximate $\dot{x}_1$ (remember that $y(t) = x_1(t)$) as

$$\dot{x}_{1_{des}} \cong K_1(y_d(t + \Delta T) - y(t)) \tag{3.8}$$

where K_1 is chosen to ensure reasonable command rates. Because our controller tries to keep $y(t)$ close to $y_d(t)$, we expect $\dot{x}_{1_{des}}$ to be close to the correct rates, those that would drive the system toward y_d. We then "invert" equation (3.6) to find the needed x_2 (call it $x_{2_{cmd}}$). Similarly, we can approximate $\dot{x}_2$ from the calculated $x_{2_{cmd}}$ and the measured x_2 as

$$\dot{x}_{2_{des}} \cong K_2(x_{2_{cmd}}(t + \Delta T) - x_2(t)). \tag{3.9}$$

Then, "inverting" equation (3.7), we can find u. A block diagram describing this approach can be seen in figure 3.7.

Because evaluating the functions f_1 and f_2 (taken from wind tunnel data) involves table lookup, the inverses to eqs. (3.6) and (3.7) must be found numerically. We have used the iterative Newton-Raphson method. However, even if we were to obtain analytical expressions for f_1 and f_2, they are not in control

affine form. The Newton-Raphson method essentially creates a local definition for the functions f_1 and f_2. For example, locally f_2 can be expressed as

$$f_{2_{NR}}(x,u) = F(x) + G(x)u. \tag{3.10}$$

Since we want to find the control such that $\dot{x}_{2_{des}} = f_2(x,u)$, the algorithm will then iteratively compute the inverse as

$$u = G^{-1}(x)[-F(x) + \dot{x}_{2_{des}}]. \tag{3.11}$$

This is (locally) identical to the feedback linearization method described in the previous chapter, provided the approximation and hence $G^{-1}(x)$ can be found.

In order to find the necessary u so that the output y tracks the desired output y_d, we must use the Newton-Raphson algorithm twice. Given the desired output y_d, we must first use the algorithm to find the necessary x_2 such that

$$F_1(x_2) \triangleq f_1(x_1, x_2, x_3) - \dot{x}_{1_{des}} = 0 \tag{3.12}$$

where $\dot{x}_{1_{des}}$ is given by eq. (3.8) and $F_1 : \mathbb{R}^n \rightarrow \mathbb{R}^n$. To find the zero of F_1, we proceed as follows: If ξ_1 is a zero of F_1, and if F_1 is sufficiently differentiable in a neighborhood $S(\xi_1)$ of ξ_1, then we can approximate the function F_1 by a Taylor series expansion about $x_{2_0} \in S(\xi_1)$. If we ignore all but the first order terms in the expansion we get the formula for the classical Newton-Raphson method:

$$\hat{\xi}_1 = x_{2_0} - J_1^{-1}(x_{2_0})F_1(x_{2_0}) \tag{3.13}$$

where $J_1(x_{2_0})$ is the Jacobian matrix for F_1 evaluated at x_{2_0}. Using similar arguments, starting with the calculated x_2 from eqs. (3.12) and (3.13), we must use the Newton-Raphson algorithm for the second time to find the necessary u such that

$$F_2(u) \triangleq f_2(x_1, x_2, x_3, u) - \dot{x}_{2_{des}} = 0 \tag{3.14}$$

where $\dot{x}_{2_{des}}$ is given by eq. (3.9) and $F_2 : \mathbb{R}^n \rightarrow \mathbb{R}^n$. We use the Newton-Raphson calculation

$$\hat{\xi}_2 = u_0 - J_2^{-1}(u_0)F_2(u_0) \tag{3.15}$$

to find the new value for u (call this u_d), where $J_2(u_0)$ is the Jacobian matrix for F_2 evaluated at u_0.

However, when eqs. (3.13) and (3.15) were used to find the necessary control u_d there were occasions when use of the iterate $\hat{\xi}_2$ caused eq. (3.14) to diverge from zero. One way to prevent divergence is to scale the search direction as

$$\hat{\xi}_2 = u_0 - \left(\frac{1}{2}\right)^k J_2^{-1}(u_0)F_2(u_0) \tag{3.16}$$

where $k = 0, 1, \ldots$ is the smallest integer such that $\|F_2(\xi)\| < \|F_2(u_0)\|$. This procedure is called the stabilized Newton method. This has no effect on the convergence of the algorithm. It only prevents divergence of the iterate $\hat{\xi}_2$. Therefore, use of the modified Newton algorithm does not guarantee that the plant states will stably track some desired trajectories. It does provide a needed bound on the next value of the control signal. This bound is necessary if the controller is to be implemented on a real aircraft.

Through simulations we have found this control scheme to be robust enough so that a simplified version of the nonlinear equations in eqs. (3.6) and (3.7) can be used to perform the inversion. Figure 3.8 shows the performance of the X-29 given an α command to follow. The full six-degree-of-freedom nonlinear simulation was used. However, the inversion algorithm used mostly (the specifics will be described below) only the linear terms from the force and moment equations to calculate the control. (See Chapter 4 for a detailed development of the X-29 force and moment equations.) For the Newton-Raphson inversion of the

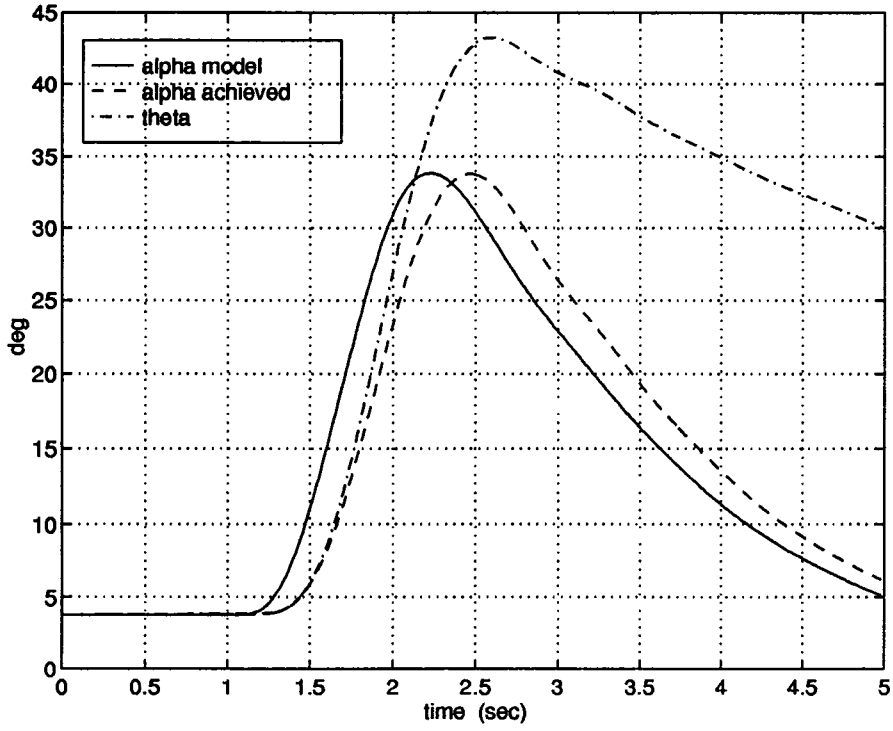


Figure 3.8: Pitch command following using Newton-Raphson inversion algorithm.

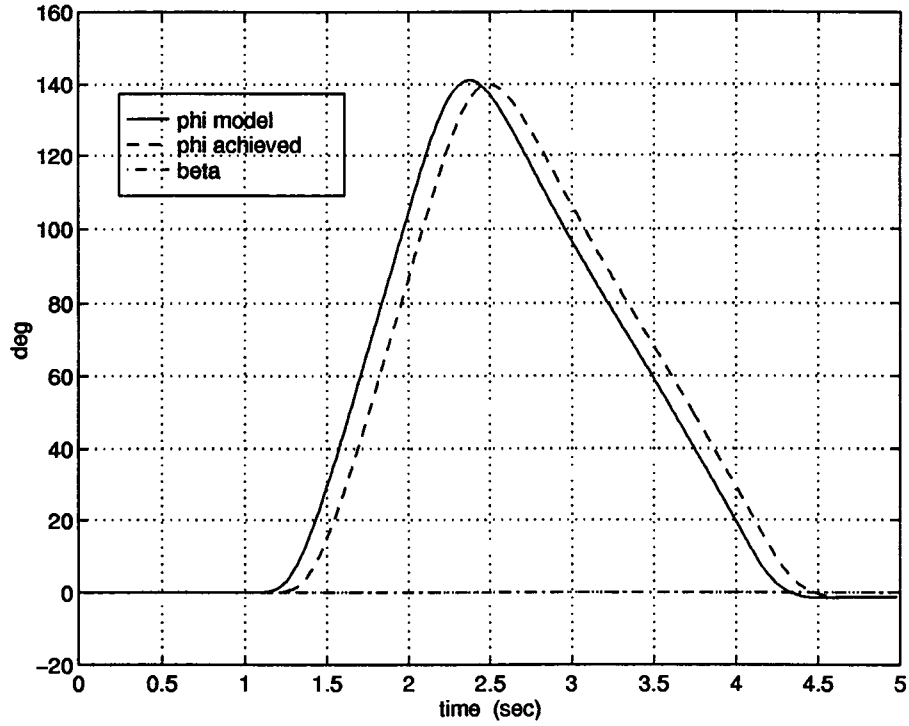


Figure 3.9: Roll command following using Newton-Raphson inversion algorithm.

longitudinal equations, we used the reduced set of force and moment equations, equations (4.16)–(4.18), but without any of the quadratic terms. The inversion from $\dot{\alpha}$ to q was then performed using equation (2.22). (Note that most of the nonlinear terms introduced come from the conversion from the body axis to the wind axis.) Inversion from $\dot{q}$ to u was performed using equation (2.17). For the lateral direction, a similar set of equations were developed for the forces and moments. Inversion from $\dot{\phi}$ to p was performed using equation (2.16), while equation (2.22) was used to invert from $\dot{\beta}$ to r. Inversion from $\dot{p}$ and $\dot{r}$ to u was performed simultaneously using equation (2.17). Similarly, figure 3.9 shows the performance of the X-29 for a roll command. Throughout both simulations we used $K_1 = [5\ \ 10\ \ 10]$ and $K_2 = [20\ \ 30\ \ 30]$. (Recall that $y_m = x_1$, hence K_1 and K_2 are 1×3 vectors.) Notice that the outputs lag the commands. This is partly

due to the fact that the actuators were not modeled in the control design but were included in the simulation. However, the major cause of the delay is that the gains were not high enough. From equations (3.8) and (3.9) we see that to closely approximate the pure derivative we need $K_1 \approx \frac{1}{\Delta T}$. Since the simulation uses $\Delta T = .0125$ sec. for the controller loop, the ideal value for K_1 should be about 80. As shown above, the gains needed for stability are much lower. Similar reasoning applies to K_2. The gains K_1 and K_2 were chosen by trial and error to ensure stability for the chosen set of maneuvers. Thus, a formal measure of stability of the closed-loop system remains unknown.

One of the keys to the success of this dynamic inverse procedure is to find a good algorithm for carrying out the inversion of $u \rightarrow f(x_1, x_2, u)$. It is easy to show that the Newton-Raphson method may fail to yield a system with stable tracking properties, assuming that the desired trajectory is within the constraints of the aircraft's control power. This is apparent from the simulations in [31] and those of Smith and Meyer [12]. In the remainder of this section, we shall develop a stable inversion algorithm for a general nonlinear system which we believe will enhance the robustness of the dynamic inverse control method.

In view of the previous discussion, we propose the following approach for an inversion algorithm for a system in the general nonlinear form. The following approach can be shown to be an alternative way of viewing the extended system (see equation (2.46)) and gives a method of constructing such a system. In fact, the method we will describe shows what the extended system looks like after a change of coordinates. For ease of discussion, we look at a simplified version of eqs.(3.6) and (3.7) (We drop dependence on x_3 and let $f_1(\cdot) = x_2$. For

simplicity, we drop the subscript 2 from $f_2(\cdot)$):

$$\dot{x}_1 = x_2 \tag{3.17}$$

$$\dot{x}_2 = f(x_1, x_2, u). \tag{3.18}$$

We would like to find a feedback control, u, which can stably track the desired trajectory $y_d = x_{1_{des}}(t)$. Let us also assume that we can calculate $x_{2_{des}}(t)$. To find such a control, let

$$z = f(x_1, x_2, u). \tag{3.19}$$

Note that if $\dot{x_2}$ is available for feedback, then the obvious choice would be to let $z = \dot{x}_2$. Otherwise we would calculate z from eq. (3.19). Taking the derivative, we get

$$\dot{z} = f_{x_1}(x_1, x_2, u) \cdot \dot{x}_1 + f_{x_2}(x_1, x_2, u) \cdot z + f_u(x_1, x_2, u) \cdot \dot{u} \tag{3.20}$$

We then let

$$\begin{aligned} \dot{u} = f_u^{-1}(x_1, x_2, u)\{\ddot{x}_{2_{des}} - f_{x_1}(x_1, x_2, u) \cdot x_2 - f_{x_2}(x_1, x_2, u) \cdot z \\ - \Phi(x_1 - x_{1_{des}}, x_2 - x_{2_{des}}, z - z_{des})\} \end{aligned} \tag{3.21}$$

where $z_{des} = \dot{x}_{2_{des}}$ and Φ is a feedback yet to be determined. This can be shown to be equivalent to the feedback linearization problem for the extended system with

$$v = \ddot{x}_{2_{des}} - \Phi(x_1 - x_{1_{des}}, x_2 - x_{2_{des}}, z - z_{des}).$$

If we substitute eq. (3.21) into eqs. (3.17) , (3.18), and (3.20) and assume that $f(x_1, x_2, u)$ is known exactly (nominal case) we get

$$\dot{x}_1 = x_2 \tag{3.22}$$

$$\dot{x}_2 = z \tag{3.23}$$

$$\dot{z} = \ddot{x}_{2_{des}} - \Phi(x_1 - x_{1_{des}}, x_2 - x_{2_{des}}, z - z_{des}) \tag{3.24}$$

If we define the errors between the actual states and the desired ones to be

$$\Delta x_1 = x_1 - x_{1_{des}}, \quad \Delta x_2 = x_2 - x_{2_{des}}, \quad \text{and} \quad \Delta z = z - z_{des}$$

we can then write equations for the error dynamics as

$$\Delta \dot{x}_1 = \Delta x_2 \tag{3.25}$$

$$\Delta \dot{x}_2 = \Delta z \tag{3.26}$$

$$\Delta \dot{z} = -\Phi(\Delta x_1, \Delta x_2, \Delta z) \tag{3.27}$$

We can now talk about stability and robustness of the closed-loop system by placing suitable constraints on f_u and Φ. For example, if we let

$$\Phi(\Delta x_1, \Delta x_2, \Delta z) = K_1 \Delta x_1 + K_2 \Delta x_2 + K_3 \Delta z \tag{3.28}$$

then eqs. (3.25)–(3.27) become

$$\Delta \dot{x}_1 = \Delta x_2 \tag{3.29}$$

$$\Delta \dot{x}_2 = \Delta z \tag{3.30}$$

$$\Delta \dot{z} = -K_1 \Delta x_1 - K_2 \Delta x_2 - K_3 \Delta z \tag{3.31}$$

Assuming that $f_u \neq 0$, stability and robustness conditions can then be studied using only linear methods, which are much more powerful than those for nonlinear systems. Also, when comparing this method to the Newton-Raphson method discussed earlier (specifically comparing eqs. (3.8) and (3.9), where K_1 and K_2 were chosen by trial and error through simulation, with eq. (3.31)) we now have a direct method for choosing the feedback gains . If we look at the system defined by eqs. (3.29)–(3.31), we can write the characteristic eq. as

$$\Delta(\lambda) = \lambda^3 + K_3 \lambda^2 + K_2 \lambda + K_1. \tag{3.32}$$

Therefore, by appropriate choice of K_1, K_2, K_3 we can determine how fast the error dynamics go to zero.

3.1 Implementation

The above discussion shows that the continuous-time closed-loop system with the control law in (3.21) has stable tracking properties. However, when this algorithm is implemented in the form of a digital controller, precautions have to be taken to ensure stability. The first step is to use a good integration routine to approximate u from $\dot{u}$ in (3.21). Since the chosen integration routine must be used with a fixed step size, i.e. one sample, this leads us to a group of methods called *one step methods* [33]. In general, such methods are given by a function

$$\eta(t, u; h, g)$$

where $g(t, x, u) = \dot{u}$. Starting with the initial values t_0, u_0, we now obtain approximate values $\hat{u}_n$ for the quantities $u_n = u(t_n)$ of the exact solution $u(t)$ by means of the algorithm:

$$\begin{aligned} &\hat{u}_0 := u_0 \\ &\text{for } n = 0, 1, 2, \ldots \\ &\hat{u}_{n+1} = \hat{u}_n + h\eta(t_n, \hat{u}_n; h, g(t_n, x(t_n), \hat{u}_n)) \\ &t_{n+1} = t_n + h. \end{aligned} \tag{3.33}$$

The simplest of these methods would be Euler's method, where

$$\eta(t, u; h, g) = g(t, x, u).$$

Then

$$\hat{u}(t_{n+1}) = \hat{u}(t_n) + h\dot{u}. \tag{3.34}$$

While this has the advantage of being quick to calculate, it is not very accurate or stable. It introduces errors of the size $O(h^2)$ per step. A good example

of the numerical accuracy and stability of some one-step integration methods can be found in Athans, et. al [34]. They compare the performance of the Euler, Heun (a second order method), and fourth-order Runga-Kutta methods for solving a second-order nonlinear differential equation known as the Van der Pol equation. By starting at an unstable equilibrium point, they show that small roundoff errors will accumulate and cause the numerical solutions to diverge. This effect varies according to step size h and the numerical integration routine used. If a small enough step size is used, none of the numerical solutions diverge. However, when the step size is increased, eventually all three of the methods diverge. The Euler method is the first to diverge, at least 20 times faster than the fourth-order RK method. In fact, at just 10 times the original step size, the Euler method solution shows significant inaccuracies while the fourth-order RK solution is virtually unchanged. This example shows that a higher-order integration method should be used for accuracy *and* stability.

A fourth-order Runge-Kutta method uses

$$\eta(t,u;h,g) = \frac{1}{6}[c_1 + 2c_2 + 2c_3 + c_4]$$

where

$$\begin{aligned} c_1 &= g(t,x,u) \\ c_2 &= g(t+\frac{1}{2}h, x(t+\frac{1}{2}h), u(t)+\frac{1}{2}hc_1) \\ c_3 &= g(t+\frac{1}{2}h, x(t+\frac{1}{2}h), u(t)+\frac{1}{2}hc_2) \\ c_4 &= g(t+h, x(t+h), u(t)+hc_3). \end{aligned}$$

This gives the following updating algorithm:

$$\hat{u}(t_{n+1}) = \hat{u}(t_n) + \frac{h}{6}[c_1 + 2c_2 + 2c_3 + c_4] \tag{3.35}$$

where

$$
\begin{aligned}
c_1 &= g(x_1(t_n), x_2(t_n), z(t_n), u(t_n)) \\
c_2 &= g(x_1(t_n + \frac{1}{2}h), x_2(t_n + \frac{1}{2}h), z(t_n + \frac{1}{2}h), u(t_n) + \frac{1}{2}hc_1) \\
c_3 &= g(x_1(t_n + \frac{1}{2}h), x_2(t_n + \frac{1}{2}h), z(t_n + \frac{1}{2}h), u(t_n) + \frac{1}{2}hc_2) \\
c_4 &= g(x_1(t_n + h), x_2(t_n + h), x_2(t_n + h), u(t_n) + hc_3).
\end{aligned}
$$

It is clear from the above equations that the control gets updated every second sample, and the sampling time $\Delta T = \frac{1}{2}h$. With this integration method, errors of size $O(h^5)$ are introduced per step.

Another concern in the digital implementation is the sampling time ΔT. Sampling must be fast enough not only to reconstruct the plant dynamics, but also to capture the closed-loop dynamics. This means that for a selected sampling time, there are limits as to how far the roots of the error characteristic equation (3.32) can be pushed into the left half plane. Stated another way, choosing a sampling time places an upper bound on the allowable bandwidth of the closed-loop system (from Nyquist's sampling theorem) which in turn places upper bounds on the feedback gains K_1, K_2, K_3.

As an example, we have applied the inversion algorithms to a simple description of an inverted pendulum with the force on the pendulum applied nonlinearly. The normalized equations for the inverted pendulum are:

$$
\begin{aligned}
\dot{\theta} &= \omega \\
\dot{\omega} &= \sin\theta + 2u + \sin(u\cos\theta).
\end{aligned}
\tag{3.36}
$$

The control portion of the above equation was constructed such that u would enter nonlinearly ($\sin(u\cos\theta)$) in a way that ensures $f_u \neq 0$, where $f = \sin\theta + 2u + \sin(u\cos\theta)$. A comparison was made between the methods we have discussed:

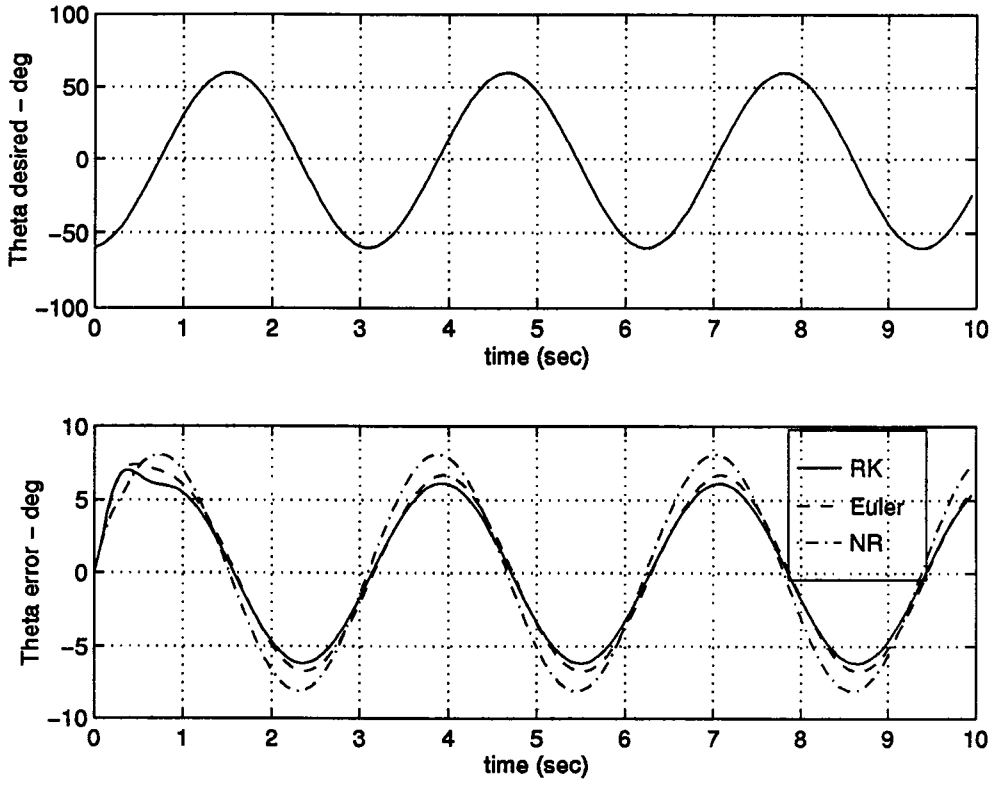


Figure 3.10: Comparison of the error between the three control methods.

stabilized Newton-Raphson (NR), the approach shown in eqs. (3.21), (3.28) using eq. (3.34) (Euler), or using eq. (3.35) (Runge-Kutta). As suspected, the Runge-Kutta method (and to a lesser extent, the Euler method) is more accurate than the NR method. Figure 3.10 shows the error between θ_{des} and the achieved θ for the different methods. The NR method uses gains: $K_1 = K_2 = 15$, while the RK and Euler methods use gains: $K_1 = 225$, $K_2 = 75$, and $K_3 = 11$. The gains in all three cases could be slightly larger before instability occurs; these gains were chosen to allow a measure of stability margin. The gains were chosen by trial and error through simulation. As stated previously, the NR gains should ideally have been 80 to approximate the derivative. The gains were scaled down until the simulation was stable and then scaled down more to allow for a measure of

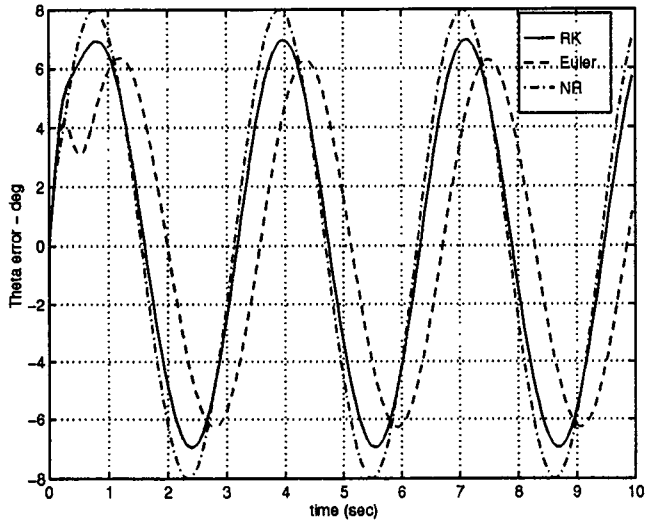


Figure 3.11: Comparison of the error when uncertainty is present.

stability margin. For the RK and Euler methods, the gains were chosen to place the roots of the characteristic equation, equation (3.32), just far enough in the left half plane to ensure good error dynamics. Performing a robustness analysis on the algorithms through simulation showed that performance barely changed (at least for the RK and NR methods) despite a $\pm 50\%$ change in the control power available (while using the same equations as before for the inversion). The same conclusions were found when changing the state contributions in the equations for the plant, e.g. changing eq. (3.18) to read $\dot{x}_2 = f(.5x_1, .5x_2, u)$. Figure 3.11 shows a representative plot of the errors for the case when eq. (3.18) is changed to read $\dot{x}_2 = f(.5x_1, .5x_2, 1.5u)$.

3.2 Sensitivity

To study robustness, we can begin by checking the sensitivity of the closed-loop system to changes in some of the parameters, p, of the plant. Suppose the actual

plant is of the form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_1, x_2, p, u). \end{aligned}$$

where p is some parameter of which we only have an estimate, e.g. some aerodynamic coefficients. As before, we want $y = x_1$ to track y_d. We can assume that

$$\begin{aligned} \dot{w} &= Aw \\ y_d &= Cw. \end{aligned}$$

Then as before, we let

$$z = f(x_1, x_2, u)$$

with some nominal value for p. Rewriting eq. (3.20)

$$\dot{z} = f_{x_1} \cdot x_2 + f_{x_2} \cdot z + f_u \cdot \dot{u}. \tag{3.37}$$

Solving the above equation for $\dot{u}$ we get

$$g(t, x, u) \triangleq \dot{u} = f_u^{-1}\{\ddot{x}_{2_{des}} - f_{x_1} \cdot x_2 - f_{x_2} \cdot z - \Phi(e)\} \tag{3.38}$$

where

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} x_1 - y_d \\ x_2 - \dot{y}_d \\ z - \ddot{y}_d \end{bmatrix}.$$

Upon digital implementation of eq. (3.38) for the controller, we get the approximate solution

$$\hat{u}(k) = u(kT) + O(T^r) \tag{3.39}$$

where $O(T^r)$ is the error introduced from an order r integration routine.

We are interested in the sensitivity of the tracking error to changes in the plant parameters, p. Therefore, let the cost function be

$$J = \|e(k)\|_2 = \left[e_1^2 + e_2^2 + e_3^2\right]^{1/2}. \tag{3.40}$$

Define the sensitivity function (sometimes referred to as the *Bode sensitivity*) to be

$$S = \frac{\partial J/J}{\partial p/p} = \frac{\partial J}{\partial p}\frac{p}{J}. \tag{3.41}$$

From the cost function above, we then have

$$\frac{\partial J}{\partial p} = \frac{\frac{1}{2}\left[\frac{\partial e_1^2}{\partial p} + \frac{\partial e_2^2}{\partial p} + \frac{\partial e_3^2}{\partial p}\right]}{\left[e_1^2 + e_2^2 + e_3^2\right]^{1/2}} \tag{3.42}$$

$$= \frac{\left[e_1\frac{\partial e_1}{\partial p} + e_2\frac{\partial e_2}{\partial p} + e_3\frac{\partial e_3}{\partial p}\right]}{\left[e_1^2 + e_2^2 + e_3^2\right]^{1/2}} \tag{3.43}$$

Substituting this into equation (3.41) we get

$$S = \frac{\left[e_1\frac{\partial e_1}{\partial p} + e_2\frac{\partial e_2}{\partial p} + e_3\frac{\partial e_3}{\partial p}\right]\cdot p}{\left[e_1^2 + e_2^2 + e_3^2\right]}. \tag{3.44}$$

Now, we examine the individual partial derivatives in the above equation. Since

$$e_1(k) = x_1(k) - y_d(k),$$

we have

$$\frac{\partial e_1(k)}{\partial p} = \frac{\partial x_1(k)}{\partial p} - \frac{\partial y_d(k)}{\partial p} \tag{3.45}$$

$$= \frac{\partial x_1(k)}{\partial p} \tag{3.46}$$

since the desired output is fixed, and not a function of p. Similarly, for the second error term

$$e_2(k) = x_2(k) - \dot{y}_d(k)$$

we get

$$\frac{\partial e_2(k)}{\partial p} = \frac{\partial x_2(k)}{\partial p} - \frac{\partial \dot{y}_d(k)}{\partial p} \tag{3.47}$$

$$= \frac{\partial x_2(k)}{\partial p}. \tag{3.48}$$

For the third error term

$$e_3(k) = z(k) - \ddot{y}_d(k)$$

we get

$$\frac{\partial e_3(k)}{\partial p} = \frac{\partial z(k)}{\partial p} - \frac{\partial y_d(k)}{\partial p} \tag{3.49}$$

$$= \frac{\partial z(k)}{\partial p} \tag{3.50}$$

$$= \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial p} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial p} + \frac{\partial f}{\partial u}\frac{\partial u}{\partial p}. \tag{3.51}$$

Note that if we have $\dot{x}_2$ available for feedback then

$$\frac{\partial e_3(k)}{\partial p} = \frac{\partial z(k)}{\partial p} \tag{3.52}$$

$$\triangleq \frac{\partial \dot{x}_2(k)}{\partial p}. \tag{3.53}$$

In order to calculate the sensitivity, we need to find the partial derivatives shown in the right hand side of eqs. (3.46), (3.48). But we don't have analytical expressions for the states x_1, x_2, only for their derivatives. However, we can solve the following set of differential equations to find the required partial derivatives:

$$\frac{\partial \dot{x}_1}{\partial p} = \frac{\partial x_2}{\partial p} \tag{3.54}$$

$$\frac{\partial \dot{x}_2}{\partial p} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial p} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial p} + \frac{\partial f}{\partial u}\frac{\partial u}{\partial p} + \frac{\partial f}{\partial p} \tag{3.55}$$

$$\frac{\partial \dot{u}}{\partial p} = \frac{\partial g}{\partial x_1}\frac{\partial x_1}{\partial p} + \frac{\partial g}{\partial x_2}\frac{\partial x_2}{\partial p} + \frac{\partial g}{\partial z}\frac{\partial z}{\partial p} + \frac{\partial g}{\partial u}\frac{\partial u}{\partial p} + \frac{\partial g}{\partial p} \tag{3.56}$$

The equations can be written as:

$$\underbrace{\begin{bmatrix} \frac{\partial \dot{x}_1}{\partial p} \\ \frac{\partial \dot{x}_2}{\partial p} \\ \frac{\partial \dot{u}}{\partial p} \end{bmatrix}}_{\dot{v}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial u} \\ \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} & \frac{\partial g}{\partial u} \end{bmatrix}}_{A(t)} \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial p} \\ \frac{\partial x_2}{\partial p} \\ \frac{\partial u}{\partial p} \end{bmatrix}}_{v} + \underbrace{\begin{bmatrix} 0 \\ \frac{\partial f}{\partial p} \\ \frac{\partial g}{\partial p} \end{bmatrix}}_{B_1(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{\partial g}{\partial z} \end{bmatrix}}_{B_2(t)} \frac{\partial z}{\partial p}. \tag{3.57}$$

It can easily be seen that equation (3.57) represents a linear system where the new states are labeled v. However, the state matrix is time-varying since the various partial derivatives are functions of the nominal values of $x_1(t)$, $x_2(t)$, and $u(t)$. Therefore, eq. (3.57) must be solved numerically. To account for the digital implementation of u (as in eq. (3.39)), we will perform the numerical integration computations exactly as computed in the controller. This solution can then be substituted into equation (3.44) to find the overall sensitivity function.

We will continue with the inverted pendulum example and calculate its sensitivity to one of the plant parameters. We replace equation (3.36) with

$$\dot{\theta} = \omega \tag{3.58}$$

$$\dot{\omega} = \sin(p\theta) + 2u + \sin(u\cos(p\theta)). \tag{3.59}$$

to show an uncertainty in the measurement of θ.

Figure 3.12 shows the sensitivity function for various values of p using a 4th-order Runga-Kutta integration routine. The simulation was performed with z calculated via equation (3.19) with f defined by equation (3.59) with $p = 1$. Figure 3.13 shows the sensitivity function when an Euler integration routine is used instead. Figure 3.14 shows the sensitivity function when we let $z = \dot{x}_2$ in the feedback calculation in equation (3.21), i.e. measured $\dot{x}_2$ instead of calculated. As expected, the sensitivity is lower since the feedback calculations include some

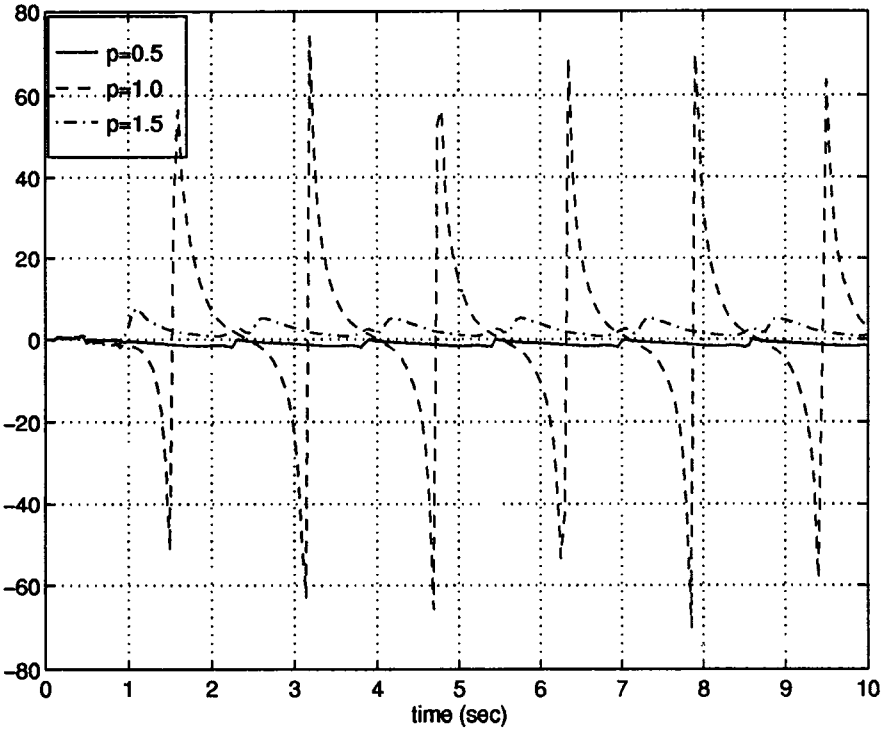


Figure 3.12: Sensitivity function using calculated z and Runga-Kutta integration.

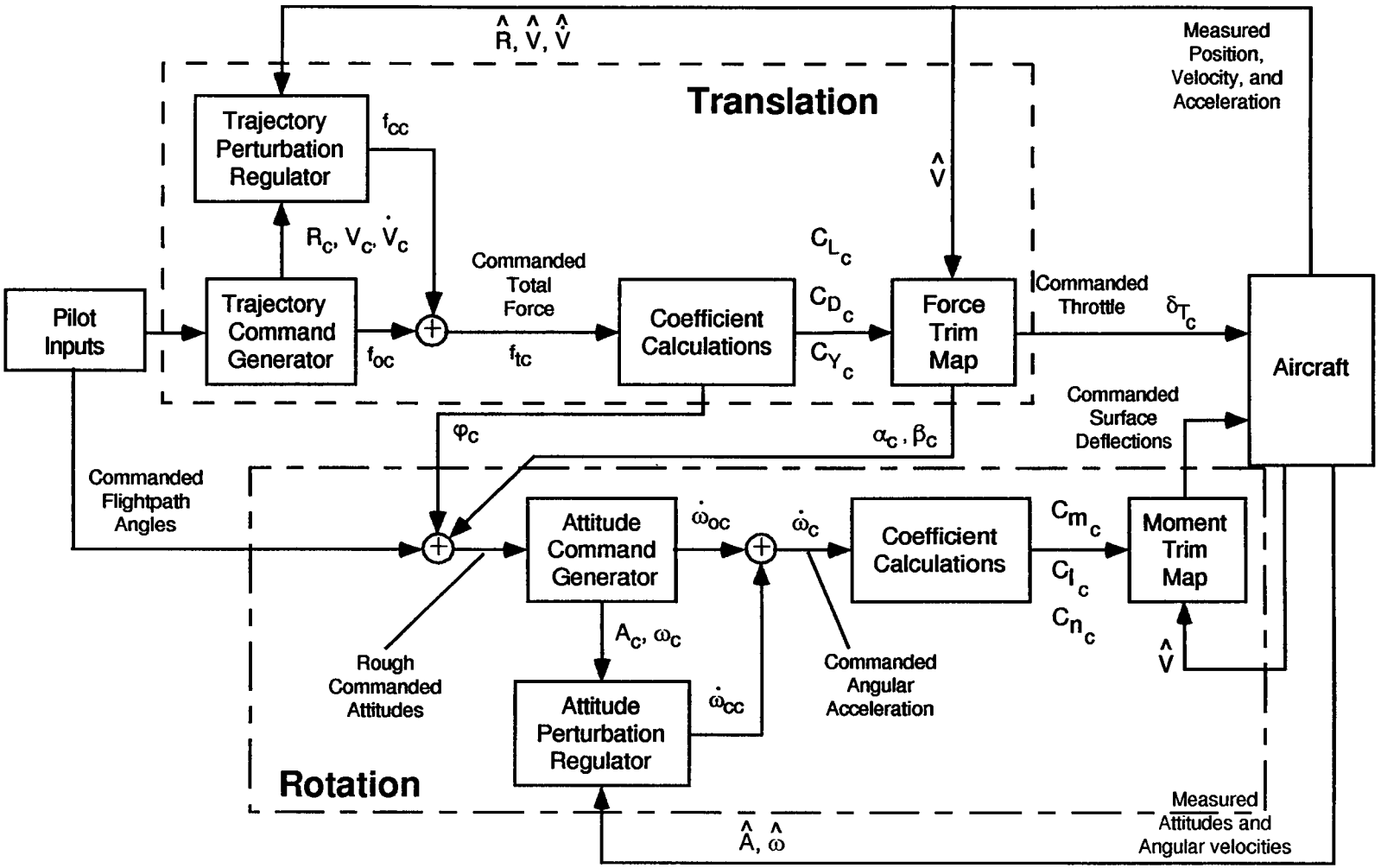


Figure 2.3: Total aircraft flight-control system proposed by Meyer and Cicolani.

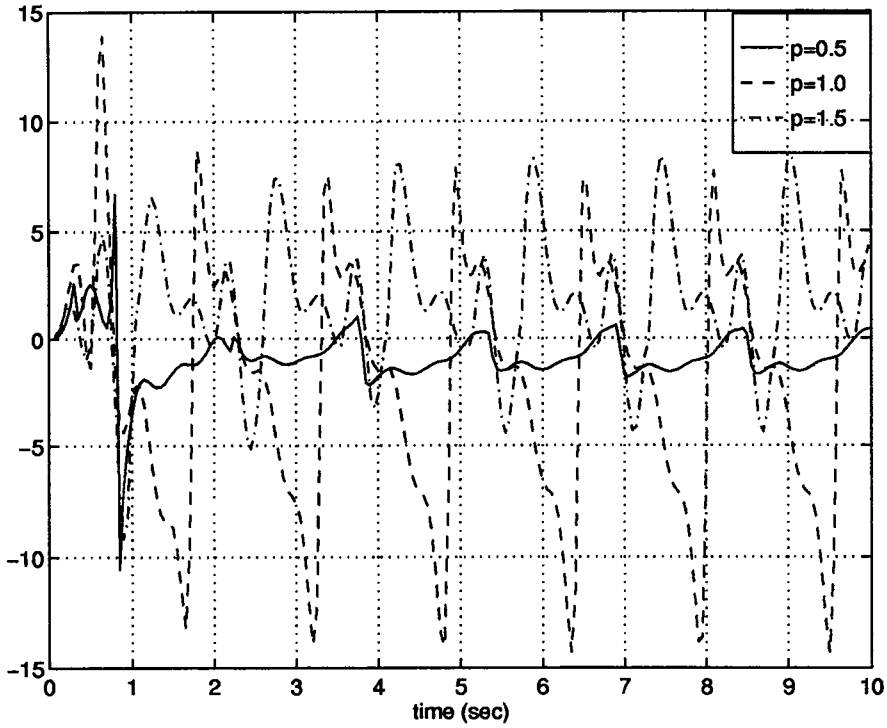


Figure 3.13: Sensitivity function using calculated z and Euler integration.

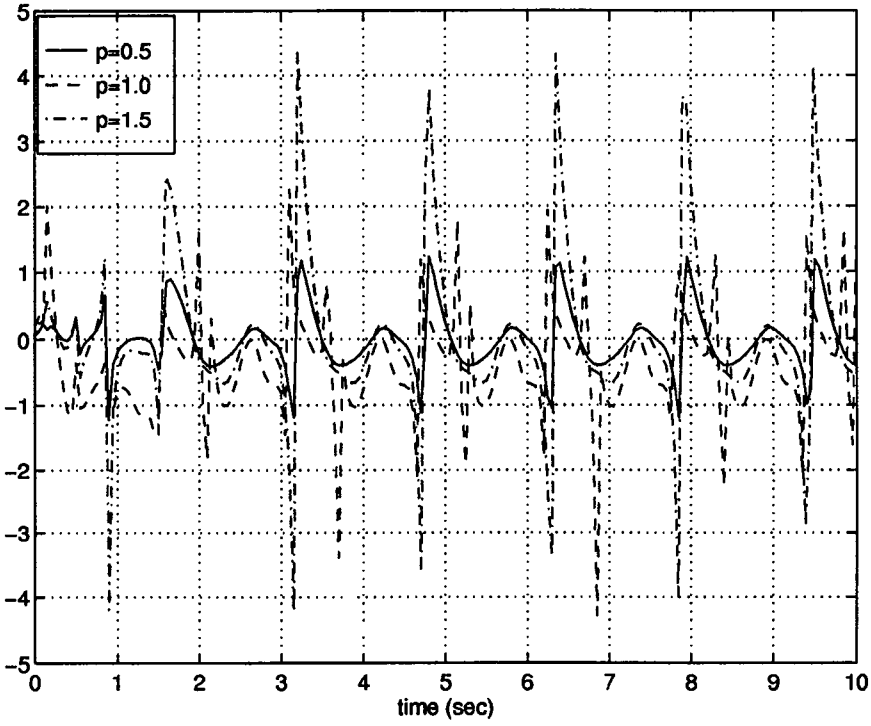


Figure 3.14: Sensitivity function using $z = \dot{x}_2$ and Runga-Kutta integration.

knowledge of the value for p. The peaks in the plots are due to the cost function J approaching zero. Since calculations using the Runga-Kutta integration routines yield the smallest errors, the peaks for the Runga-Kutta sensitivity function are the highest. Note also that because of the periodic nature of the errors, the sensitivity function is also somewhat periodic.

To avoid some of the shortcomings of the above sensitivity function, let us define a new cost function, independent of time:

$$J_2 = \|e_1(k)\|_2 + \|e_2(k)\|_2 + \|e_3(k)\|_2 \tag{3.60}$$

where $\|\cdot\|_2$ is the Euclidian norm for sequences. From the cost function above, we then have

$$\frac{\partial J_2}{\partial p} = \frac{1}{2}\frac{\frac{\partial}{\partial p}\sum_{k=1}^{n} e_1^2(k)}{\|e_1(k)\|_2} + \frac{1}{2}\frac{\frac{\partial}{\partial p}\sum_{k=1}^{n} e_2^2(k)}{\|e_2(k)\|_2} + \frac{1}{2}\frac{\frac{\partial}{\partial p}\sum_{k=1}^{n} e_3^2(k)}{\|e_3(k)\|_2} \tag{3.61}$$

$$= \frac{\sum_{k=1}^{n} e_1(k)\frac{\partial e_1(k)}{\partial p}}{\|e_1(k)\|_2} + \frac{\sum_{k=1}^{n} e_2(k)\frac{\partial e_2(k)}{\partial p}}{\|e_2(k)\|_2} + \frac{\sum_{k=1}^{n} e_3(k)\frac{\partial e_3(k)}{\partial p}}{\|e_3(k)\|_2} \tag{3.62}$$

With S as before, namely

$$S = \frac{\partial J_2}{\partial p}\frac{p}{J_2} \tag{3.63}$$

we recalculate the sensitivity. Using this new formulation, the sensitivity functions for the inverted pendulum simulation are shown in Table 3.1. The Runga-Kutta sensitivity functions above are not always the lowest. This is due to the fact that all three error signals are weighted evenly in the cost function. In all cases, the error of interest e_1 is smallest when using the Runga-Kutta integration with $z = \dot{x}_2$. However, the other two errors, e_2 and e_3, do not always have the smallest norm, largely due to higher startup transients. The cost function defined in equation (3.60) can easily be modified to weight the position error e_1 more heavily, simply by multiplying the e_1 norm by a number bigger than 1.

Taking this idea to the extreme, and only using information from the position error, i.e.

$$J_3 = \|e_1(k)\|_2 \tag{3.64}$$

Table 3.2 shows the new sensitivity function using J_3.

Chapter 4

Application of Algorithm to X-29 Aircraft example

4.1 X-29 Model

In this section we will describe the model of the X-29, Grumman's forward-swept-wing research aircraft, used in the study. Information on the primary flight control system and the results from flight tests can be found in Clarke et.al [35]. Figure 4.1 shows a picture of the X-29 outlining the control surfaces available. The two canards are slaved to move in unison, and are the primary longitudinal control surfaces. The strake flaps are also slaved together and used for longitudinal control; however, they are mostly used in trimming the aircraft since they are not as effective as the canards. As seen in the figure, there are three flaperons on each wing. While all six can be driven independently, in this study the three flaperons on each wing are controlled together for simplicity. The flaperons are used for both longitudinal and lateral motions (hence

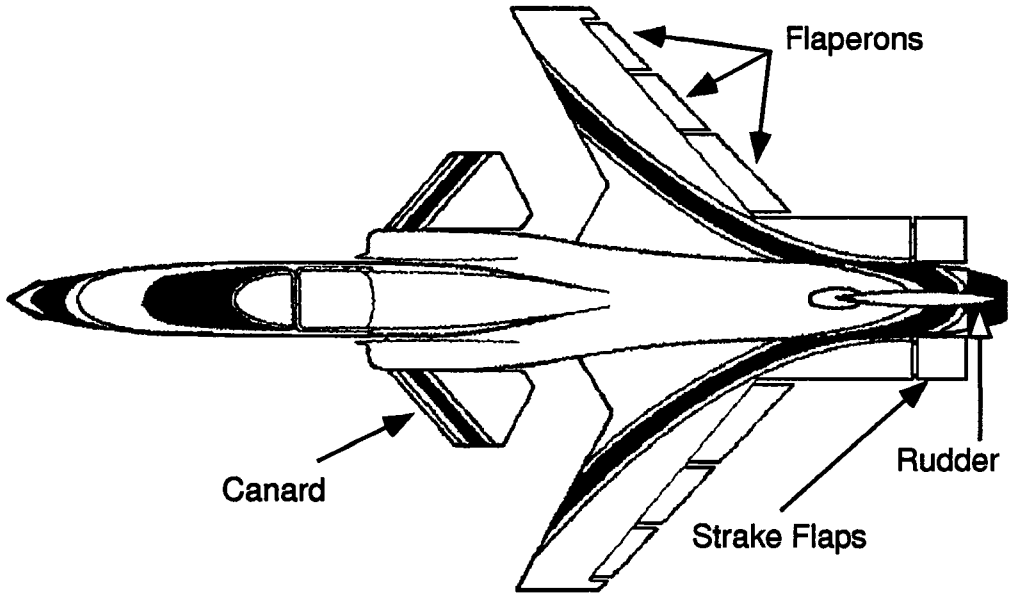


Figure 4.1: A top view of the X-29 aircraft displaying the flight control surfaces.

the name flaperon - part flaps and part aileron). The longitudinal and lateral commands for the flaperons will be divided into two fictitious control surfaces - flaps and ailerons. The longitudinal commands will be defined for the flaps, while the lateral commands will be defined for the ailerons, with priority given to the aileron commands. The remaining control surface is the rudder, driven by lateral/directional commands.

Figure 4.2 graphically describes the valid ranges of angle-of-attack and sideslip for the model. The canard is restricted to move between -60 and +30 degrees, with a positive canard deflection defined when the trailing edge of the canard moves downward. The flap has a range of -10 to +25 degrees, with a positive flap deflection defined when the flaperon is moved downward. The strake has a range of ± 30 degrees, with a positive strake deflection defined when the strake flap is moved downward. The rudder has a range of ± 30 degrees, with a positive movement defined when it is moved to the right. The aileron has a half angle

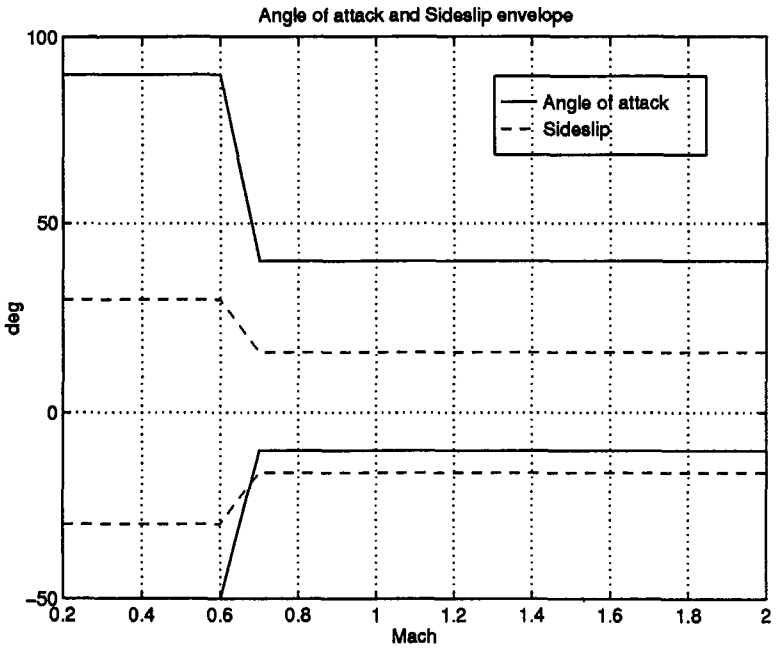


Figure 4.2: Angle-of-attack and sideslip envelope of the X-29 aircraft.

maximum of 17.5 degrees, with a positive aileron movement defined when the right flaperon is moved upward and the left flaperon is moved downward. The model is valid up to Mach 2.0 and up to 55,000 ft. However, the aircraft has only been flight tested up to Mach 1.48 and just over 50,000 ft, and up to 50° angle of attack at $1g$ and 35° angle of attack at airspeeds up to 300 knots.

A six-degrees-of-freedom simulation with table look-ups for aerodynamic coefficients, which are based on actual wind tunnel data, is used to represent the aircraft. For the longitudinal axis, the dimensionless aerodynamic coefficients

take the following form:

$$
\begin{aligned}
C_A \;=\; & C_{A_{\alpha=0}}(M) + C_{A_\alpha}(M)\cdot\alpha + C_{A_{\alpha^2}}(M)\cdot\alpha^2 \\
& + C_{A_{\delta_F}}(M)\cdot\delta_F + C_{A_{\delta_{F^2}}}(M)\cdot\delta_F^2 + \Delta C_{A_{\delta_F}}(\alpha,\delta_F,M) \\
& + C_{A_{\delta_S}}(M)\cdot\delta_S + C_{A_{\delta_{S^2}}}(M)\cdot\delta_S^2 + \Delta C_{A_{\delta_S}}(\alpha,\delta_S,M) \\
& + \left[C_{A_{\delta_C}}(M)\cdot\delta_C + C_{A_{\delta_{C^2}}}(M)\cdot\delta_C^2 + \Delta C_{A_{\delta_C}}(\alpha,\delta_C,\delta_F,M)\right] \\
& \qquad \cdot K_{A_{\delta_{C_{\delta_S}}}}(\alpha,\delta_C,\delta_S) \\
& + HA\Delta C_{A_{\delta_F}}(\alpha,M,\delta_F) \\
& + HA\Delta C_{A_{\delta_S}}(\alpha,\delta_S) + HA^2\Delta C_{A_{\delta_S}}(\alpha,M,\delta_S) \\
& + HA\Delta C_{A_{\delta_C}}(\alpha,\delta_C,\delta_F) + HA^2\Delta C_{A_{\delta_C}}(\alpha,M,\delta_C) \\
& + HA\Delta C_{A_{\delta_{C_{\delta_F}}}}(\alpha,M,\delta_C,\delta_F) + HA\Delta C_{A_{\delta_{C_{\delta_S}}}}(\alpha,M,\delta_C,\delta_S) \\
& + \Delta C_{A_{LG}}(\alpha,\delta_C,\delta_F) + \Delta C_{A_{GE}}(\alpha,\delta,h)
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
C_N \;=\; & C_{N_{\alpha=0}}(M) + C_{N_\alpha}(M)\cdot\alpha + \tfrac{\bar{c}}{2V}\left[C_{N_{\dot\alpha}}(M)\cdot\dot\alpha + C_{N_q}(M)\cdot q\right] \\
& + C_{N_{\delta_F}}(M)\cdot\delta_F + \Delta C_{N_{\delta_F}}(\alpha,\delta_F,M) \\
& + C_{N_{\delta_S}}(M)\cdot\delta_S + \Delta C_{N_{\delta_S}}(\alpha,\delta_S,M) \\
& + \left[C_{N_{\delta_C}}(M)\cdot\delta_C + \Delta C_{N_{\delta_C}}(\alpha,\delta_C,\delta_F,M)\right]\cdot K_{N_{\delta_{C_{\delta_S}}}}(\alpha,\delta_C,\delta_S) \\
& + HA\Delta C_{N_{\delta_F}}(\alpha,M,\delta_F) \\
& + HA\Delta C_{N_{\delta_S}}(\alpha,\delta_S) + HA^2\Delta C_{N_{\delta_S}}(\alpha,M,\delta_S) \\
& + HA\Delta C_{N_{\delta_C}}(\alpha,\delta_C,\delta_F) + HA^2\Delta C_{N_{\delta_C}}(\alpha,M,\delta_C) \\
& + HA\Delta C_{N_{\delta_{C_{\delta_F}}}}(\alpha,M,\delta_C,\delta_F) + HA\Delta C_{N_{\delta_{C_{\delta_S}}}}(\alpha,M,\delta_C,\delta_S) \\
& + \Delta C_{N_{LG}}(\alpha,\delta_C) + \Delta C_{N_{GE}}(\alpha,\delta,h)
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
C_m \;=\; & C_{m_{\alpha=0}}(M) + C_{m_\alpha}(M)\cdot\alpha \\
& + \tfrac{\bar{c}}{2V}\left\{C_{m_{\dot\alpha}}(M)\cdot\dot\alpha + \left[C_{m_q}(M) + HA\Delta C_{m_q}(\alpha)\right]\cdot q\right\} \\
& + C_{m_{\delta_F}}(M)\cdot\delta_F + \Delta C_{m_{\delta_F}}(\alpha,\delta_F,M) \\
& + C_{m_{\delta_S}}(M)\cdot\delta_S + \Delta C_{m_{\delta_S}}(\alpha,\delta_S,M) \\
& + \left[C_{m_{\delta_C}}(M)\cdot\delta_C + \Delta C_{m_{\delta_C}}(\alpha,\delta_C,\delta_F,M)\right]\cdot K_{m_{\delta_{C_{\delta_S}}}}(\alpha,\delta_C,\delta_S) \\
& + HA\Delta C_{m_{\delta_F}}(\alpha,M,\delta_F) \\
& + HA\Delta C_{m_{\delta_S}}(\alpha,\delta_S) + HA^2\Delta C_{m_{\delta_S}}(\alpha,M,\delta_S) \\
& + HA\Delta C_{m_{\delta_C}}(\alpha,\delta_C,\delta_F) + HA^2\Delta C_{m_{\delta_C}}(\alpha,M,\delta_C) \\
& + HA\Delta C_{m_{\delta_{C_{\delta_F}}}}(\alpha,M,\delta_C,\delta_F) + HA\Delta C_{m_{\delta_{C_{\delta_S}}}}(\alpha,M,\delta_C,\delta_S) \\
& + \Big[\Delta C_{m_{\beta_{\delta_A}}}(\alpha,\delta_A,M) + HA\Delta C_{m_{\beta_{\delta_A}}}(\alpha,\delta_A) \\
& \qquad + \Delta C_{m_{\beta_{\delta_S}}}(\alpha,\delta_S) + HA\Delta C_{m_{\beta_{\delta_S}}}(\alpha,\delta_S)\Big]\cdot|\beta| \\
& + HA\Delta C_{m_{\beta_{\delta_C}}}(\beta,\alpha,\delta_C,M) \\
& + \Delta C_{m_{LG}}(\alpha,\delta_C) + \Delta C_{m_{GE}}(\alpha,\delta,h)
\end{aligned}
\tag{4.3}
$$

where C_A is the axial force coefficient, C_N is the normal force coefficient, and C_m is the pitching moment coefficient. The coefficient C_A is the backward force along the fuselage and C_N is the upward force normal to the fuselage. Hence both coefficients are calculated in the body axes. A transformation to the wind axes must be performed to match the definitions in equation (2.13). The coefficients used in the above equations are derived from wind tunnel testing, and are functions of the variables in parenthesis. For example, $C_{m_{\delta_F}}(M)$ is the pitching moment coefficient due to the flap, and is a function of Mach number. As the Mach number changes, a new value for $C_{m_{\delta_F}}$ must be found, either directly from the data tables, or by using linear interpolation between stored values. The terms beginning with a Δ are nonlinear incremental terms that are phased in based on the values of the independent variables in parenthesis. For example,

for some nominal value of α, δ_F, and M, $\Delta C_{m_{\delta_F}}(\alpha, \delta_F, M) = 0$. Away from the nominal values, $\Delta C_{m_{\delta_F}}(\alpha, \delta_F, M)$ is nonzero. Since many of the terms are functions of multiple variables, multidimensional linear interpolation must be performed in between measured parameters. The coefficients with subscripts LG are incremental effects due to the landing gear down, while the coefficients with subscripts GE are incremental effects due to the closeness of the aircraft to the ground. Both of these terms will not come into play for the maneuvers chosen in our study. The term $K_{\cdot_{\delta_{C_{\delta_S}}}}$ is the canard-strake interference factor, and is nominally equal to one. It varies about one when the canard angle is extreme enough to change the airflow across the strake. Note that it is modelled as a reduction in effectiveness of the canard. The terms beginning with HA are high α increments and are only nonzero when $\alpha < -4°$ and $\alpha > 24°$.

The lateral-directional axis has dimensionless coefficients of the form:

$$\begin{aligned} C_Y &= C_{Y_0}(M,\alpha,\beta,\delta_C) + \Delta C_{Y_{\delta_F}}(M,\alpha,\beta) + \Delta C_{Y_{\delta_S}}(M,\alpha,\beta,\delta_C) \\ &\quad + \Delta C_{Y_{\delta_R}}(M,\alpha,\beta,\delta_C) + \Delta C_{Y_{\delta_A}}(M,\alpha,\delta_C) \\ &\quad + \Delta C_{Y_{\delta_{A_\beta}}}(M,\alpha,\beta,\delta_C) \\ &\quad + \tfrac{b}{2V}\left[C_{Y_p}(M,\alpha)\cdot p + C_{Y_r}(M,\alpha,\delta_C)\cdot r\right] \\ &\quad + \Delta C_{Y_{LG}}(M,\alpha,\beta,\delta_C,\delta_F) + \Delta C_{Y_{GE}}(M,\alpha,\beta,\delta_c,h) \end{aligned} \tag{4.4}$$

$$\begin{aligned} C_l &= C_{l_0}(M,\alpha,\beta,\delta_C) + \Delta C_{l_{\delta_F}}(M,\alpha,\beta) + \Delta C_{l_{\delta_S}}(M,\alpha,\beta,\delta_C) \\ &\quad + \Delta C_{l_{\delta_R}}(M,\alpha,\beta,\delta_C) + \Delta C_{l_{\delta_A}}(M,\alpha,\delta_C) \\ &\quad + \Delta C_{l_{\delta_{A_\beta}}}(M,\alpha,\beta,\delta_C) \\ &\quad + \tfrac{b}{2V}\left[C_{l_p}(M,\alpha,\beta)\cdot p + C_{l_r}(M,\alpha,\delta_C)\cdot r\right] \\ &\quad + \Delta C_{l_{LG}}(M,\alpha,\beta,\delta_C,\delta_F) + \Delta C_{l_{GE}}(M,\alpha,\beta,\delta_c,h) \end{aligned} \tag{4.5}$$

$$
\begin{aligned}
C_n \;=\; & C_{n_0}(M,\alpha,\beta,\delta_C) + \Delta C_{n_{\delta_F}}(M,\alpha,\beta) + \Delta C_{n_{\delta_S}}(M,\alpha,\beta,\delta_C) \\
& + \Delta C_{n_{\delta_R}}(M,\alpha,\delta_C) + \Delta C_{n_{\delta_A}}(M,\alpha,\delta_C) \\
& + \Delta C_{n_{\delta_{A_\beta}}}(M,\alpha,\beta,\delta_C) \\
& + \tfrac{b}{2V}\left[C_{n_p}(M,\alpha)\cdot p + C_{n_r}(M,\alpha,\delta_C)\cdot r\right] \\
& + \Delta C_{n_{LG}}(M,\alpha,\beta,\delta_C,\delta_F) + \Delta C_{n_{GE}}(M,\alpha,\beta,\delta_c,h)
\end{aligned}
\tag{4.6}
$$

where C_Y is the side force coefficient, C_l is the rolling moment coefficient, and C_n is the yawing moment coefficient. The lateral-directional equations (4.4)-(4.6) may look simpler than the longitudinal equations in (4.1)-(4.3), however, the lateral equation terms rely more heavily on table lookups, e.g. $C_{m_{\delta_S}}(M)$ vs. $\Delta C_{n_{\delta_S}}(M,\alpha,\beta,\delta_C)$. The lateral-directional equations also use piecewise linear (at any particular operating point) control surface coefficients vs. the simpler constant coefficient plus a nonlinear term for the longitudinal equations. For example, the relationship for the rolling moment due to the flap is described as

$$
\Delta C_{l_{\delta_F}} = \begin{cases}
0 & \delta_F = 25\,\text{deg} \\
\Delta C_{l_{\delta_{F_1}}}\left[\frac{25-\delta_F}{15}\right] & 10\,\text{deg} \le \delta_F < 25\,\text{deg} \\
\Delta C_{l_{\delta_{F_1}}} + \Delta C_{l_{\delta_{F_2}}}\left[\frac{10-\delta_F}{10}\right] & 0\,\text{deg} \le \delta_F < 10\,\text{deg} \\
\Delta C_{l_{\delta_{F_1}}} + \Delta C_{l_{\delta_{F_2}}} + \Delta C_{l_{\delta_{F_3}}}\left[\frac{-\delta_F}{10}\right] & -10\,\text{deg} \le \delta_F < 0\,\text{deg}
\end{cases}
\tag{4.7}
$$

where each coefficient $\Delta C_{l_{\delta_{F_\#}}}$ is a function of α and β. A similar term for the longitudinal pitching moment due to the flap is expressed simply as

$$
C_{m_{\delta_F}}(M)\cdot\delta_F + \Delta C_{m_{\delta_F}}(\alpha,\delta_F,M).
$$

As seen in equations (4.1) through (4.6) and the previous discussion of how they are constructed, a complete and accurate model for the X-29 is difficult and complex to describe. If the control engineer is to account for all the nonlinearities in the aircraft, the controller will be very complex and large. The design must

reduce the complexity of the aircraft model required for inversion while ensuring that the controller is robust to the inaccuracies and reductions of the inversion model. In the following sections we will show how to construct a nonlinear controller for this problem.

4.2 Simulation

The state equations representing the longitudinal equations of motion for the X-29 (or any jet aircraft, see equations (2.16), (2.17), (2.22)) can be described as follows:

$$\dot{V} = \frac{1}{m}[-D + T\cos\alpha] - g\sin(\theta - \alpha) \tag{4.8}$$

$$\dot{\alpha} = q - q_w \tag{4.9}$$

$$\dot{\theta} = q \tag{4.10}$$

$$\dot{q} = \frac{1}{I_{YY}} \cdot M \tag{4.11}$$

where V is the total velocity, α is the angle of attack, θ is the pitch angle(body axis), and q is the pitch rate(body axis), q_w is the pitch rate in the wind-axis direction, D is the total drag force, and M is the pitching moment(body axis). The equations for the additional variables are:

$$q_w = \frac{1}{mV}[L + T\sin\alpha - mg\cos(\alpha - \theta)] \tag{4.12}$$

$$M = \bar{q}S\bar{c}C_m \tag{4.13}$$

$$L = \bar{q}S(C_N\cos\alpha - C_A\sin\alpha) \tag{4.14}$$

$$D = \bar{q}S(C_A\cos\alpha + C_N\sin\alpha) \tag{4.15}$$

with inputs

$$T \quad - \quad \text{Thrust (lbs.)}$$

δ_C – Canard deflection (deg.)

δ_F – Flap deflection (deg.)

δ_S – Strake deflection (deg.)

where $\bar{q} = \frac{1}{2}\rho V^2$ represents dynamic pressure, L is the upward lift force perpendicular to the velocity vector, $\bar{c}$, S, I_{YY}, and m are constants associated with the aircraft. The axial and normal body axes forces in equations (4.14) and (4.15) are transformed into lift and drag wind axes forces to correspond to the equations of motion developed in Section 2.1.3. We will assume that we are to design an autopilot for the θ channel. Alternatively, the θ command can be generated from the pilot input. In either case, we want the θ output from the aircraft to follow a given θ command. For this simulation study we will not use the thrust input, T, as a changeable input, i.e. it will remain constant throughout the maneuvers. This is consistent with what was done in Grumman's controller. Also note that the actuators are not modelled in the following description of the aircraft used for controller synthesis. However, first order models for the actuators are used in the simulation. The strake actuators are modelled by

$$\delta_{s_{out}} = \frac{35}{s+35}\delta_{s_{in}}$$

while all the remaining actuators are modelled by

$$\delta_{out} = \frac{20}{s+20}\delta_{in}.$$

In addition, the simulation has the additional capability to place rate limits on the actuators. The rate limits are: 105 deg/sec for the canards, 73 deg/dec for the flaperons, 27 deg/sec for the strakes, and 125 deg/sec for the rudder. Therefore, the controller must be robust enough to cope with the unmodelled actuators and their limits.

A reduced (missing any high angle terms) set of force and moment equations is given below.

$$\begin{aligned} C_A &= C_{A_{\alpha=0}} + D_{RAM}\frac{\cos\alpha}{\bar{q}S} + C_{A_\alpha}\cdot\alpha + C_{A_{\alpha^2}}\cdot\alpha^2 \\ &\quad + C_{A_{\delta_F}}\cdot\delta_F + C_{A_{\delta_{F^2}}}\cdot\delta_F^2 + C_{A_{\delta_S}}\cdot\delta_S + C_{A_{\delta_{S^2}}}\cdot\delta_S^2 \\ &\quad + C_{A_{\delta_C}}\cdot\delta_C + C_{A_{\delta_{C^2}}}\cdot\delta_C^2 \end{aligned} \tag{4.16}$$

$$\begin{aligned} C_N &= C_{N_{\alpha=0}} + D_{RAM}\frac{\sin\alpha}{\bar{q}S} + C_{N_\alpha}\cdot\alpha + \frac{\bar{c}}{2V}\left(C_{N_{\dot\alpha}}\cdot\dot\alpha + C_{N_q}\cdot q\right) \\ &\quad + C_{N_{\delta_F}}\cdot\delta_F + C_{N_{\delta_S}}\cdot\delta_S + C_{N_{\delta_C}}\cdot\delta_C \end{aligned} \tag{4.17}$$

$$\begin{aligned} C_m &= C_{m_{\alpha=0}} + C_{m_\alpha}\cdot\alpha + \frac{\bar{c}}{2V}\left(C_{m_{\dot\alpha}}\cdot\dot\alpha + C_{m_q}q\right) \\ &\quad + C_{m_{\delta_F}}\cdot\delta_F + C_{m_{\delta_S}}\cdot\delta_S + C_{m_{\delta_C}}\cdot\delta_C \end{aligned} \tag{4.18}$$

where C_A is the nondimensional axial force, C_N is the nondimensional normal force, and C_m is the nondimensional pitching moment. The axial force is measured in the body axis with a positive force pointing towards the tail. The normal force is measured in the body axis with a positive force pointing upward. The pitching moment is measured in the body axis about the center of gravity with a positive moment pitching upward. We would like to rewrite eq. (4.9) so that the right hand side does not depend on $\dot\alpha$ (as seen from eqs. (4.12), (4.14), and (4.17)). Substitution of equations (4.12), (4.16), and (4.17) into (4.9) and grouping the $\dot\alpha$ terms on the left yields

$$\begin{aligned} \dot\alpha(1+\cos\alpha\tfrac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot\alpha}}) &= (1-\cos\alpha\tfrac{\bar{q}S\bar{c}}{2mV^2}C_{N_q})q \\ &\quad + \tfrac{\bar{q}S}{mV}\Big[C_A\sin\alpha - \cos\alpha\left(C_{N_{\alpha=0}} + D_{RAM}\tfrac{\sin\alpha}{\bar{q}S}\right. \\ &\quad \left. + C_{N_\alpha}\cdot\alpha + C_{N_{\delta_F}}\cdot\delta_F + C_{N_{\delta_S}}\cdot\delta_S + C_{N_{\delta_C}}\cdot\delta_C\ \right)\Big] \\ &\quad - \tfrac{T}{mV}\sin\alpha + \tfrac{g}{V}\cos(\alpha-\theta). \end{aligned} \tag{4.19}$$

If

$$1 + \cos\alpha \frac{\bar{q}S\bar{c}}{2mV^2} C_{N_{\dot{\alpha}}} \neq 0 \tag{4.20}$$

then we can calculate $\dot{\alpha}$ directly from the states and input terms. Since

$$K \triangleq \frac{\bar{q}S\bar{c}}{2mV^2} = \frac{\frac{1}{2}\rho S\bar{c}}{2m}$$

is a constant for a given altitude with $K \approx 10^{-3}$ and $0.36 < C_{N_\alpha} < 0.48$ over the entire flight envelope, then eq. (4.20) holds. If we define $\tilde{C}_N$ as the normal force equation without the dynamic effects (those due to $\dot{\alpha}$ and q), e.g.

$$\tilde{C}_N = C_{N_{\alpha=0}} + D_{RAM}\frac{\sin\alpha}{\bar{q}S} + C_{N_\alpha}\cdot\alpha + C_{N_{\delta_F}}\cdot\delta_F + C_{N_{\delta_S}}\cdot\delta_S + C_{N_{\delta_C}}\cdot\delta_C$$

then we can rewrite eq. (4.19) more compactly as

$$\begin{aligned} \dot{\alpha} \;=\; & (1 + \cos\alpha \tfrac{\bar{q}S\bar{c}}{2mV^2} C_{N_{\dot{\alpha}}})^{-1} \Big\{ (1 - \cos\alpha \tfrac{\bar{q}S\bar{c}}{2mV^2} C_{N_q}) q \\ & + \tfrac{\bar{q}S}{mV}\left[C_A \sin\alpha - \tilde{C}_N \cos\alpha\right] - \tfrac{T}{mV}\sin\alpha + \tfrac{g}{V}\cos(\alpha - \theta) \Big\}. \end{aligned} \tag{4.21}$$

We note that, similar to eq. (4.20),

$$1 - \cos\alpha \frac{\bar{q}S\bar{c}}{2mV^2} C_{N_q} \neq 0 \tag{4.22}$$

holds since $4.0 < C_{N_q} < 8.1$ over the entire flight envelope. Therefore, if we are given $\dot{\alpha}$, we can solve for q.

Now that we have a closed-form expression for $\dot{\alpha}$, and hence for $\dot{q}$ (see equations (4.11) and (4.18) we would like to find a feedback control, which can stably track the desired trajectory θ_{des}. To this end we will use the algorithm described in the previous chapter and summarized in figure 4.3, where we will use $x_1 = \theta$, $x_2 = q$ and $x = [V\ \theta\ \alpha\ q]'$. To use the algorithm, we will also need trajectories q_{des}, $\dot{q}_{des}$ and $\ddot{q}_{des}$. To find a control, as the algorithm states, we need to take the derivative of the $\dot{q}$ equation, eq. (4.11). This gives

$$\ddot{q} = \frac{1}{I_{YY}}\left[M_\alpha\dot{\alpha} + M_q\dot{q} + M_\theta\dot{\theta} + M_V\dot{V} + M_u\dot{u}\right] \tag{4.23}$$

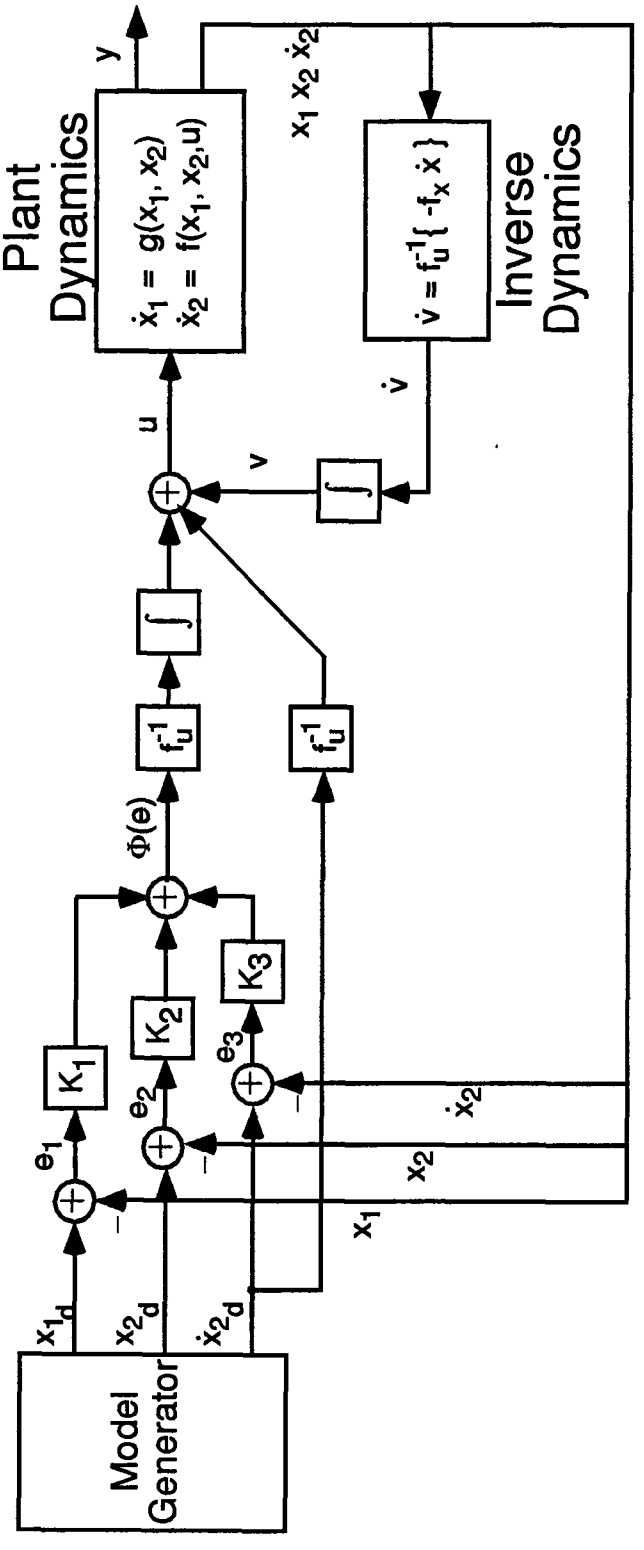


Figure 4.3: Inverse dynamics controller.

where $u = [\delta_C \ \delta_F \ \delta_S]^T$, M_α is the partial derivative of pitching moment with respect to α, M_q is the partial derivative of pitching moment with respect to q, M_θ is the partial derivative of pitching moment with respect to θ, M_V is the partial derivative of pitching moment with respect to V, and the row vector M_u is the partial derivative of pitching moment with respect to the controls u. Evaluating the partial derivatives we get

$$M_\alpha = \bar{q}S\bar{c}\left[C_{m_\alpha} + \frac{\bar{c}}{2V}C_{m_{\dot{\alpha}}}\frac{\partial\dot{\alpha}}{\partial\alpha}\right] \tag{4.24}$$

$$M_q = \frac{\bar{q}S\bar{c}^2}{2V}\left[C_{m_q} + C_{m_{\dot{\alpha}}}\frac{\partial\dot{\alpha}}{\partial q}\right] \tag{4.25}$$

$$M_\theta = \frac{\bar{q}S\bar{c}^2}{2V}C_{m_{\dot{\alpha}}}\frac{\partial\dot{\alpha}}{\partial\theta} \tag{4.26}$$

$$M_V = \bar{q}S\bar{c}\left[\frac{2C_m}{V} + \frac{\partial C_m}{\partial V}\right] \tag{4.27}$$

$$M_u = \bar{q}S\bar{c}\left[C_{m_u} + \frac{\bar{c}}{2V}C_{m_{\dot{\alpha}}}\frac{\partial\dot{\alpha}}{\partial u}\right] \tag{4.28}$$

with

$$\begin{aligned}\frac{\partial\dot{\alpha}}{\partial\alpha} = {} & (1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-2}\left\{(1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})\frac{1}{mV}\right. \\ & \left[\bar{q}S\left(\sin\alpha\frac{\bar{c}}{2V}C_{N_q}q + C_A\cos\alpha + \tilde{C}_N\sin\alpha\right.\right. \\ & +(C_{A_\alpha} - D_{RAM}\frac{\sin\alpha}{\bar{q}S} + 2C_{A^2_\alpha}\alpha)\sin\alpha \\ & \left.-(C_{N_\alpha} + D_{RAM}\frac{\cos\alpha}{\bar{q}S})\cos\alpha\right) \\ & -T\cos\alpha - mg\sin(\alpha-\theta)] \\ & +(\sin\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})\left[(1-\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_q})q\right. \\ & \left.\left.+\frac{1}{mV}\left(\bar{q}S(C_A\sin\alpha - \tilde{C}_N\cos\alpha) - T\sin\alpha + mg\cos(\alpha-\theta)\right)\right]\right\}\end{aligned} \tag{4.29}$$

$$\frac{\partial\dot{\alpha}}{\partial q} = (1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1}(1-\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_q}) \tag{4.30}$$

$$\frac{\partial\dot{\alpha}}{\partial\theta} = (1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1}\frac{g}{V}\sin(\alpha-\theta) \tag{4.31}$$

$$\frac{\partial C_m}{\partial V} = \frac{\bar{c}}{2V}\left[C_{N_{\dot{\alpha}}}\frac{\partial \dot{\alpha}}{\partial V} - (C_{m_{\dot{\alpha}}}\dot{\alpha} + C_{m_q}q)\right] \tag{4.32}$$

$$\frac{\partial \dot{\alpha}}{\partial V} = (1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1}\left\{\frac{\bar{q}S}{mV^2}\left[C_A\sin\alpha - \tilde{C}_N\cos\alpha\right]\right. \tag{4.33}$$

$$\left. +\frac{\bar{q}S}{mV}\left[\frac{\partial C_A}{\partial V}\sin\alpha - \frac{\partial \tilde{C}_N}{\partial V}\cos\alpha\right] + \frac{T}{mV^2}\sin\alpha - \frac{g}{V^2}\cos(\alpha-\theta)\right\} \tag{4.34}$$

$$\frac{\partial C_A}{\partial V} = -D_{RAM}\frac{2\cos\alpha}{\bar{q}SV} \tag{4.35}$$

$$\frac{\partial \tilde{C}_N}{\partial V} = -D_{RAM}\frac{2\sin\alpha}{\bar{q}SV} \tag{4.36}$$

$$\frac{\partial \dot{\alpha}}{\partial u} = (1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1}\frac{\bar{q}S}{mV}\left[(C_{A_\delta} + 2C_{A_{\delta 2}}u)\sin\alpha - \tilde{C}_{N_\delta}\cos\alpha\right] \tag{4.37}$$

$$C_{m_u} = [C_{m_{\delta_C}} C_{m_{\delta_F}} C_{m_{\delta_S}}] \tag{4.38}$$

$$C_{A_\delta} = [C_{A_{\delta_C}} C_{A_{\delta_F}} C_{A_{\delta_S}}] \tag{4.39}$$

$$C_{A_{\delta 2}} = [C_{A_{\delta_C^2}} C_{A_{\delta_F^2}} C_{A_{\delta_S^2}}] \tag{4.40}$$

$$C_{N_\delta} = [C_{N_{\delta_C}} C_{N_{\delta_F}} C_{N_{\delta_S}}]. \tag{4.41}$$

Note that all of the derivative terms C_x in the above equations, e.g. $C_{N_{\dot{\alpha}}}$, are constants depending on the particular Mach number as shown in equations (4.1)–(4.3). To find the control, we need to solve eq. (4.23) for $\dot{u}$. If we let

$$M_u\dot{u} = I_{YY}(\ddot{q}_{des} - \Phi(e)) - M_\alpha\dot{\alpha} - M_q\dot{q} - M_\theta\dot{\theta} - M_V\dot{V} \tag{4.42}$$

where

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \theta - \theta_{des} \\ q - q_{des} \\ \dot{q} - \dot{q}_{des} \end{bmatrix} \tag{4.43}$$

and $\Phi(e)$ is a simple feedback function, we can solve for the desired control. However, since we have three redundant control surfaces, eq. (4.42) is underdetermined. Hence there are an infinite number of solutions. Complicating this problem is the fact that each surface has different rate and position limits.

There are several different methods in the literature for solving this problem. The reader is referred to [36, 37, 38] for comparisons of the different control allocation schemes. Since no one solution was the clear choice, we decided to choose a generalized inverse solution. For clarity, we let the right hand side of equation (4.42) be represented simply as X. We then rewrite equation (4.42) as

$$M_u \dot{u} = X \tag{4.44}$$

where $M_u = [M_{\delta_c}\ M_{\delta_f}\ M_{\delta_s}]$ and since $u = [\delta_C\ \delta_F\ \delta_S]^T$, we get that $X \in \mathbb{R}$. The notion of a generalized inverse is such that if we can find a P such that $M_u P = I$, then the solution to eq. (4.44) is given by

$$\dot{u} = PX. \tag{4.45}$$

In our case, since I is 1×1 and $M_u \neq 0$, a solution P can always be found. However, there exist many solutions. The standard psuedo-inverse solution gives the P such that $\|\dot{u}\|_2$ is a minimum. In addition, the psuedo-inverse solution gives the P such that $\|M_u\dot{u} - PX\|_2$ is a minimum. However, in our case, since I is 1×1, $\|M_u\dot{u} - PX\|_2 = 0$ for all P. Since M_u is changing, a new P will constantly need to be calculated. The most reliable way to do this involves first finding the singular value decomposition [39] of M_u. Since we would like to keep the calculation as simple as possible, for implementation in the flight computer, and since we would like to take into account the different limits for the control surfaces, we will define a different way to compute the generalized inverse.

In order to find a unique solution which accounts for the different limits on the control surfaces, we will slave the control surfaces together in the following manner

$$\delta_C = -1.9\,\delta \quad \delta_F = 0.7\,\delta \quad \delta_S = 1.0\,\delta.$$

These numbers were chosen to approximately normalize the control surface deflections. Also, we want to minimize use of the flaperons for longitudinal control since they are also used for the lateral/directional control. The above scaling is equivalent to defining a term $P_1 = [-1.9\ 0.7\ 1.0]^T$, and letting $\dot{u} = P_1\dot{\delta}$. We then must solve the equation

$$M_u P_1 \dot{\delta} = X$$

for δ. Let the solution be denoted by $\bar{\delta}$. Then a solution to equation (4.44) can be computed as

$$\begin{aligned} \dot{u} &= P_1\bar{\delta} \\ &= P_1(M_u P_1)^{-1}X \\ &\triangleq PX. \end{aligned} \tag{4.46}$$

Therefore,

$$\dot{u} = P\left[I_{YY}(\ddot{q}_{des} - \Phi(e)) - M_\alpha\dot{\alpha} - M_q\dot{q} - M_\theta\dot{\theta} - M_V\dot{V}\right]. \tag{4.47}$$

If we substitute the calculated control into the state equations (4.9), and (4.11) and assuming for now that we know Z, D, M exactly (nominal case) we get

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{z} \end{bmatrix} \triangleq \begin{bmatrix} \dot{\theta} \\ \dot{q} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} q \\ z \\ \ddot{q}_{des} - \Phi(e) \end{bmatrix}. \tag{4.48}$$

The dynamics of the error equations can be found by taking the derivative of eq. (4.43). Therefore, the error dynamics can be described by

$$\begin{aligned} \dot{e}_1 &= \dot{\theta} - \dot{\theta}_{des} \\ &= q - q_{des} \\ &= e_2 \end{aligned} \tag{4.49}$$

$$\begin{aligned} \dot{e}_2 &= \dot{q} - \dot{q}_{des} \\ &= e_3 \qquad (4.50) \\ \dot{e}_3 &= \ddot{q} - \ddot{q}_{des} \\ &= -\Phi(e) \qquad (4.51) \end{aligned}$$

where the last line for $\dot{e}_3$ follows from eq. (4.48).

If we let

$$\Phi(e) = K_1 e_1 + K_2 e_2 + K_3 e_3 \qquad (4.52)$$

then we can look at the entire error dynamics equations to check stability. Combining eqs. (4.49), (4.50), and (4.51) into matrix form we get

$$\dot{e} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -K_1 & -K_2 & -K_3 \end{bmatrix}. \qquad (4.53)$$

Therefore, stability of the closed-loop system follows from a proper selection of gains for $\Phi(e)$. K_1, K_2, K_3 must be chosen such that the roots of the characteristic eq.

$$\Delta(\lambda) = \lambda^3 + K_3\lambda^2 + K_2\lambda + K_1. \qquad (4.54)$$

lie sufficiently far in the left-half of the complex plane.

We choose the roots of the characteristic eq. (4.54) to be

$$\lambda = -10,\ -5 \pm 5i$$

to ensure good transient response characteristics. This leads to a set of gains

$$\begin{aligned} K_1 &= 500 \\ K_2 &= 150 \\ K_3 &= 20. \end{aligned}$$

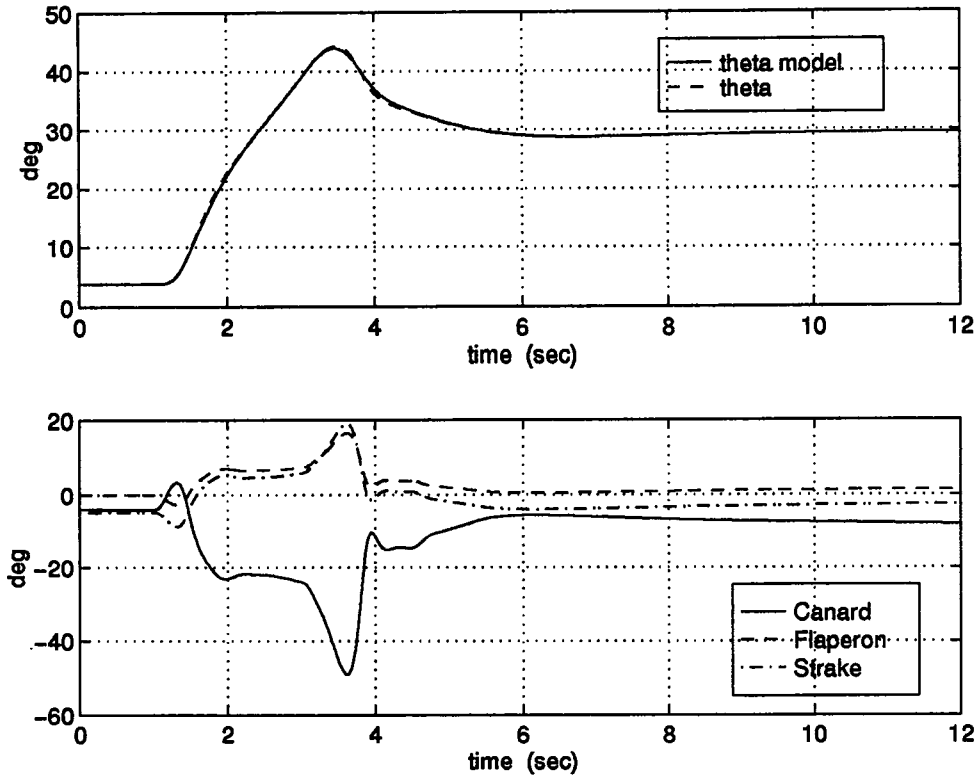


Figure 4.4: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft.

Figure 4.4 shows simulation results for a high-α pitch up maneuver at a speed of Mach 0.9 and an altitude of 30,000 ft using the control law outlined in equations (4.47) and (4.52). This is an aggressive maneuver, pulling close to 6 gs at its peak, comparable to a full stick pullup. As the figure shows, the output tracks the command closely. For this simulation, and all further ones, control displacement limits are in effect. We can see that the control surfaces stay within their displacement limits. However, rate limits were not used in the simulation. The commanded control rates were well outside their limits with a maximum rate at 3.8 sec. of 218 deg/sec for the canard, 80 deg/sec for the flaperon, and 115 deg/sec for the strake. If we turned the rate limiting on for the simulation,

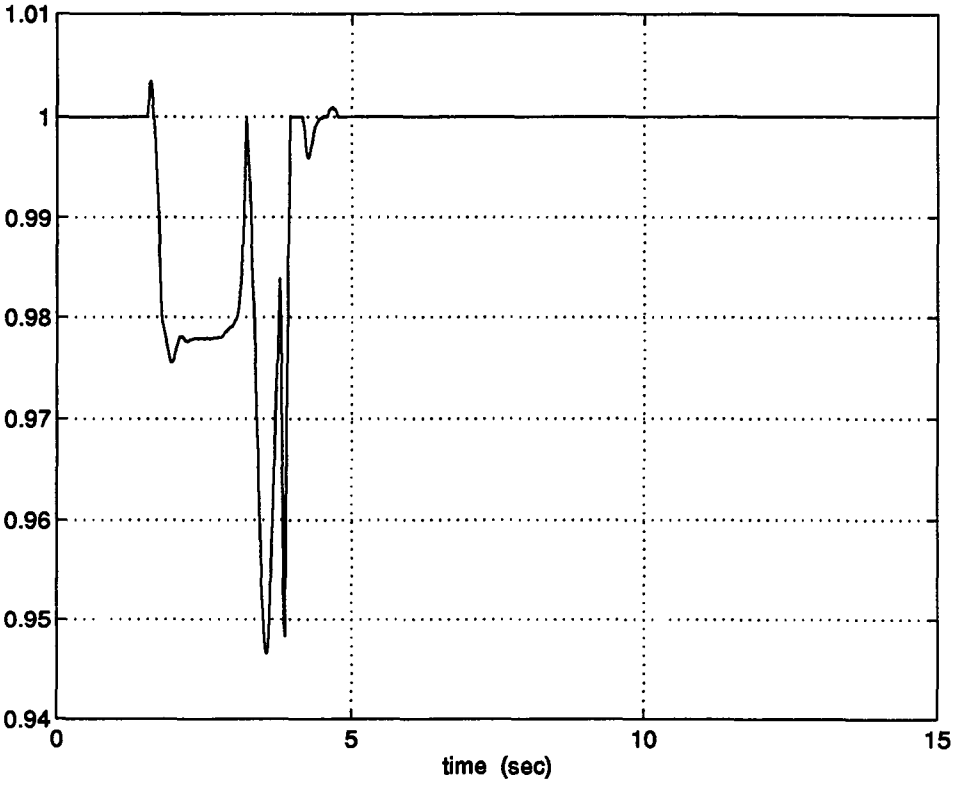


Figure 4.5: Canard-strake interference factor for the pitch up maneuver.

the control surfaces would hit their rate limits and the system would go unstable. Figure 4.5 shows the canard-strake interference factor as it changes throughout the maneuver. For the region around 3.7 sec. the canard surface is about 6.5% less effective. However, for the majority of the time the surface is close to full effectiveness.

Since the controller must work over the entire flight envelope, we used the same control law (with different aerodynamic coefficients for the new operating point) for the same pitch up maneuver as before but at a speed of Mach 0.6 and an altitude of 15,000 ft to check the performance at the new operating point. We would expect that at the lower altitude the control surfaces are more effective. This alone would make the maneuver less demanding on the aircraft.

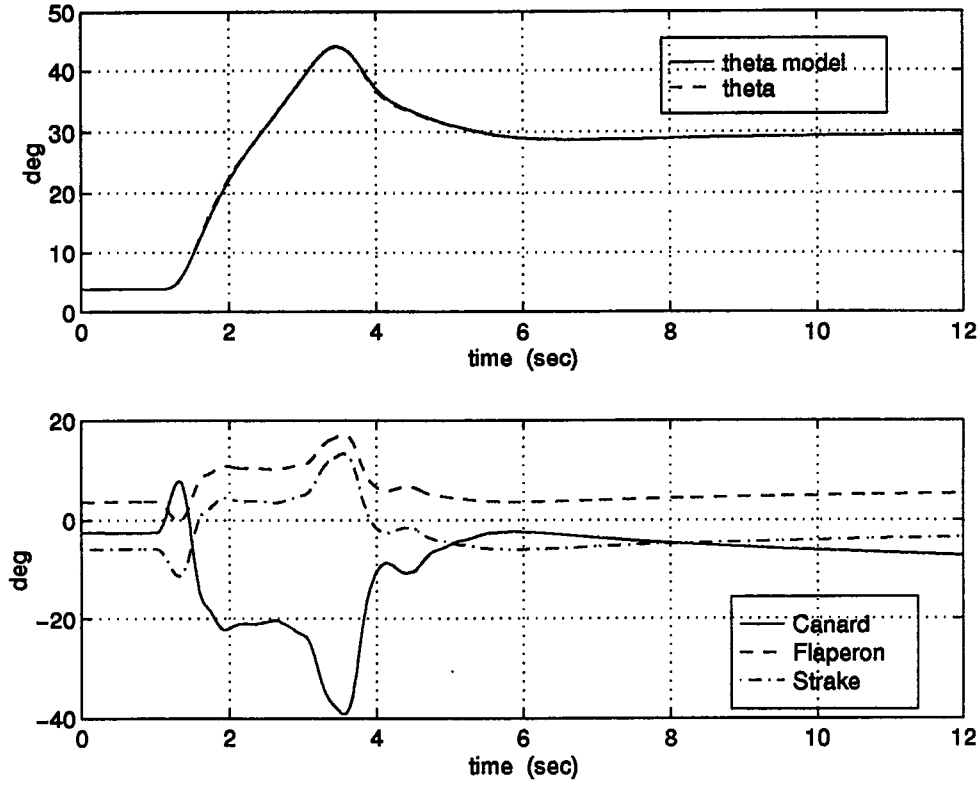


Figure 4.6: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.6 and 15,000 ft.

However, the slower speed reduces control effectiveness. Thus we get a tradeoff for effectiveness. Figure 4.6 shows the simulation results. The commanded control rates for the canard and strake were still outside their limits with a maximum rate at 1.5 sec. of 119 deg/sec for the canard, and 63 deg/sec for the strake. The flaperon rate is now within its limit at a maximum of 44 deg/sec. The canard-strake interference factor has about the same shape as before, but now only dips to a 2.5% loss in effectiveness.

4.2.1 Integral Control

If a steady state error in the θ-tracking loop exists which is larger than the designer can tolerate, we can simply add an integrator term in the error feedback signal $\Phi(e)$. We will then have

$$\Phi(e) = K_i e_i + K_1 e_1 + K_2 e_2 + K_3 e_3$$

where $e_i = \int e_1$. The dynamics of the error equations will then be defined as

$$\dot{e} = \begin{bmatrix} \dot{e}_i \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -K_i & -K_1 & -K_2 & -K_3 \end{bmatrix}. \tag{4.55}$$

Similar to the previous case, K_i, K_1, K_2, K_3 must be chosen such that the roots of the characteristic eq.

$$\Delta(\lambda) = \lambda^4 + K_3\lambda^3 + K_2\lambda^2 + K_1\lambda + K_i. \tag{4.56}$$

lie sufficiently far in the left-half of the complex plane.

4.3 Robust Control Law

The previous section described a nonlinear control law requiring exact knowledge of the plant for inversion. Since this is not realistic, the method needs to be modified to account for uncertainty in the plant description.

The state equations representing the longitudinal equations of motion for the X-29 (see eqs. (4.9)-(4.12)) are repeated here for convenience:

$$\dot{V} = \frac{1}{m}[-D + T\cos\alpha] - g\sin(\theta - \alpha) \tag{4.57}$$

$$\dot{\alpha} = q - q_w \tag{4.58}$$

$$\dot{\theta} = q \tag{4.59}$$

$$\dot{q} = \frac{1}{I_{YY}} \cdot M \tag{4.60}$$

where V is the total velocity, α is the angle of attack, θ is the pitch angle(body axis), and q is the pitch rate(body axis), q_w is the pitch rate in the wind-axis direction, D is the total drag force, and M is the pitching moment(body axis). The equations for the additional variables are:

$$q_w = \frac{1}{mV}[L + T\sin\alpha - mg\cos(\alpha - \theta)] \tag{4.61}$$

$$M = \bar{q}S\bar{c}C_m \tag{4.62}$$

$$L = \bar{q}S(C_N\cos\alpha - C_A\sin\alpha) \tag{4.63}$$

$$D = \bar{q}S(C_A\cos\alpha + C_N\sin\alpha). \tag{4.64}$$

We will restrict our attention to the variables θ and q since we wish to track a θ command. Once again we note that actuator models are not included in the aircraft equations used for controller construction, while first-order actuators are used in the simulation.

Following the notation of Slotine and Hedrick [28], and rewriting the above equations we have

$$\dot{\theta} = q \tag{4.65}$$

$$\dot{q} = \frac{1}{I_{YY}} \cdot M \triangleq C + C_\alpha(\alpha) + C_\theta(\theta) + C_q q + C_u(u). \tag{4.66}$$

Since the model is not exactly known, we approximate the terms above by:

$$\hat{C} = C + \tilde{C} \tag{4.67}$$

$$\hat{C}_\alpha(\alpha) = C_\alpha(\alpha) + \tilde{C}_\alpha(\alpha) \tag{4.68}$$

$$\hat{C}_\theta(\theta) = C_\theta(\theta) + \tilde{C}_\theta(\theta) \tag{4.69}$$

$$\hat{C}_q = C_q + \tilde{C}_q \tag{4.70}$$

$$\hat{C}_u(u) = C_u(u) + \tilde{C}_u(u). \tag{4.71}$$

The extended system can then be written as

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \\ \dot{u} \end{bmatrix} = \underbrace{\begin{bmatrix} q \\ \hat{C} + \hat{C}_\alpha(\alpha) + \hat{C}_\theta(\theta) + \hat{C}_q q + \hat{C}_u(u) \\ 0 \end{bmatrix}}_{\hat{f}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{g} \nu \tag{4.72}$$

where ν is the new control input. The system's relative degree was found to be 3, and the computed normal states are:

$$\begin{aligned} \mu_1 &= y = \theta \\ \mu_2 &= L_{\hat{f}} h = q \\ \mu_3 &= L_{\hat{f}}^2 h = \dot{q} = \hat{C} + \hat{C}_\alpha(\alpha) + \hat{C}_\theta(\theta) + \hat{C}_q q + \hat{C}_u(u). \end{aligned} \tag{4.73}$$

Because of model uncertainties, the μ_i are not simply successive derivatives of

the output y but instead are

$$\begin{aligned} \dot{\mu}_1 &= \mu_2 + D_1 \\ \dot{\mu}_2 &= \mu_3 + D_2 \\ \dot{\mu}_3 &= L_{\tilde{f}}^3 h + L_g L_{\tilde{f}}^2 h\nu + D_3 \end{aligned} \tag{4.74}$$

where

$$D_1 = -L_{\tilde{f}} h = 0 \tag{4.75}$$

$$D_2 = -L_{\tilde{f}} L_{\tilde{f}} h = -\tilde{C} - \tilde{C}_\alpha(\alpha) - \tilde{C}_\theta(\theta) - \tilde{C}_q q - \tilde{C}_u(u) \tag{4.76}$$

$$\begin{aligned} D_3 &= -L_{\tilde{f}} L_{\tilde{f}}^2 h \\ &= -\hat{C}_q \left[\tilde{C} + \tilde{C}_\alpha(\alpha) + \tilde{C}_\theta(\theta) + \tilde{C}_q q + \tilde{C}_u(u)\right] \end{aligned} \tag{4.77}$$

$$L_{\hat{f}}^3 h = \frac{\partial \hat{C}_\theta(\theta)}{\partial \theta} q + \hat{C}_q \left[\hat{C} + \hat{C}_\alpha(\alpha) + \hat{C}_\theta(\theta) + \hat{C}_q q + \hat{C}_u(u)\right] \tag{4.78}$$

$$L_g L_{\hat{f}}^2 h = \left[\frac{\partial \hat{C}_\theta(\theta)}{\partial \theta} \quad \hat{C}_q \quad \frac{\partial \hat{C}_u(u)}{\partial u}\right] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{\partial \hat{C}_u(u)}{\partial u}. \tag{4.79}$$

Following Slotine and Hedrick [28] we define $z_1 = y_d = \theta_d$ with z_2 and z_3 to be defined later based on a Lyapunov-like argument. Define

$$s_i = \tilde{\mu}_i - \phi_i \text{sat}(\tilde{\mu}_i/\phi_i) \quad i = 1, 2, 3 \tag{4.80}$$

where $\tilde{\mu}_i = \mu_i - z_i$, the ϕ_i are strictly positive quantities, and sat is the saturation function ($\text{sat}(x) = \text{sgn}(x)$ if $|x| \geq 1$, $\text{sat}(x) = x$ otherwise). When the $\text{sat}(\tilde{\mu}_i/\phi_i)$ is not in effect ($\tilde{\mu}_i < \phi_i$) $s_i = 0$ and hence $\dot{s}_i = 0$. Otherwise we get

$$\begin{aligned} \dot{s}_1 &= \mu_2 + D_1 - \dot{z}_1 \\ &= s_2 + \phi_2 \text{sat}(\tilde{\mu}_2/\phi_2) + z_2 - \dot{z}_1 + D_1 \\ \dot{s}_2 &= \mu_3 + D_2 - \dot{z}_2 \\ &= s_3 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3) + z_3 - \dot{z}_2 + D_2 \\ \dot{s}_3 &= L_{\tilde{f}}^3 h + L_g L_{\tilde{f}}^2 h\nu - \dot{z}_3 + D_3 \end{aligned} \tag{4.81}$$

We will select the z_i as known functions of the states. Thus, their derivatives in the above equations are not known exactly, but rather are estimated as $\dot{z}_{ie}$ based on the system model. Letting

$$\Delta_i = D_i + \dot{z}_{ie} - \dot{z}_i + \phi_{i+1}\text{sat}(\tilde{\mu}_{i+1}/\phi_{i+1}) \quad i = 1, 2 \tag{4.82}$$

$$\Delta_3 = D_3 + \dot{z}_{3e} - \dot{z}_3 \tag{4.83}$$

we get (for $s_i \neq 0$)

$$\begin{aligned} \dot{s}_1 &= s_2 + z_2 - \dot{z}_{1e} + \Delta_1 \\ \dot{s}_2 &= s_3 + z_3 - \dot{z}_{2e} + \Delta_2 \\ \dot{s}_3 &= L_{\hat{f}}^3 h + L_g L_{\hat{f}}^2 h\nu - \dot{z}_{3e} + \Delta_3. \end{aligned} \tag{4.84}$$

Assuming that state dependent bounds are known on the modelling error $\tilde{f}$, then one can compute state dependent functions F_i such that

$$|\Delta_i| \leq F_i \quad i = 1, 2, 3. \tag{4.85}$$

Consider the Lyapunov function candidate

$$V = \frac{1}{2}\sum_{i=1}^{3} \sigma^{3-i} s_i^2 \tag{4.86}$$

where σ is a strictly positive constant. We then get

$$\begin{aligned} \dot{V} = &\sigma^2 s_1(s_2 + z_2 - \dot{z}_{1e} + \Delta_1) \\ &+\sigma s_2(s_3 + z_3 - \dot{z}_{2e} + \Delta_2) \\ &+s_3(L_{\hat{f}}^3 h + L_g L_{\hat{f}}^2 h\nu - \dot{z}_{3e} + \Delta_3). \end{aligned}$$

Define

$$z_1 = y_d \tag{4.87}$$

$$z_2 = \dot{z}_{1e} - F_1\text{sat}(\tilde{\mu}_1/\phi_1) - \lambda s_1 \tag{4.88}$$

$$z_3 = \dot{z}_{2e} - F_2\text{sat}(\tilde{\mu}_2/\phi_2) - \lambda s_2 - \sigma s_1 \tag{4.89}$$

where λ is a strictly positive constant. Since our system has relative degree 3, the derivative of z_3 may contain the input ν. We can therefore write $\dot{z}_{3e}$ in the general form

$$\dot{z}_{3e} = \dot{z}_{3ex} + \dot{z}_{3eu}\nu.$$

If we choose the control input as

$$\nu = (L_g L_f^2 h - \dot{z}_{3eu})^{-1}(-L_f^3 h + \dot{z}_{3ex} - F_3 \text{sat}(\tilde{\mu}_3/\phi_3) - \lambda s_3 - \sigma s_2) \tag{4.90}$$

we then have

$$\dot{V} + 2\lambda V \leq 0$$

which shows that the s_i all exponentially converge to zero.

We will now follow the above approach for the longitudinal equations of motion for the X-29 as described by eqs. (4.67)–(4.72). Let $z_1 = \theta_d$. Then $\dot{z}_{1e} = \dot{z}_1 = \dot{\theta}_d$ and

$$\begin{aligned} \Delta_1 &= D_1 + \dot{z}_{1e} - \dot{z}_1 + \phi_2 \text{sat}(\tilde{\mu}_2/\phi_2) \\ &= \phi_2 \text{sat}(\tilde{\mu}_2/\phi_2) \end{aligned} \tag{4.91}$$

since $D_1 = 0$ from eq. (4.75). Therefore, we can let

$$|\Delta_1| \leq \phi_2 \triangleq F_1. \tag{4.92}$$

From eq. (4.88) we have

$$z_2 = \dot{z}_{1e} - F_1 \text{sat}(\tilde{\mu}_1/\phi_1) - \lambda s_1$$

which leads to

$$\begin{aligned} \dot{z}_{2e} &= \ddot{z}_{1e} - \dot{F}_{1e}\text{sat}(\tilde{\mu}_1/\phi_1) - F_1 \frac{d}{dt}[\text{sat}(\tilde{\mu}_1/\phi_1)] - \lambda \dot{s}_1 \\ &= \ddot{z}_{1e} - F_1 \frac{d}{dt}[\text{sat}(\tilde{\mu}_1/\phi_1)] - \lambda \dot{s}_1. \end{aligned}$$

since F_1 is a constant and hence $\dot{F}_{1e} = 0$. When the $\text{sat}(\tilde{\mu}_i/\phi_i)$ is in effect ($\tilde{\mu}_i > \phi_i$), $\frac{d}{dt}\{\text{sat}(\tilde{\mu}_1/\phi_1)\} = 0$, hence we get

$$\begin{aligned} \dot{z}_{2e} &= \ddot{z}_{1e} - \lambda(\dot{\mu}_1 - \dot{z}_1) \\ &= \dot{q}_d - \lambda[\mu_2 - q_d]. \end{aligned} \tag{4.93}$$

When the $\text{sat}(\tilde{\mu}_i/\phi_i)$ is not in effect ($\tilde{\mu}_i < \phi_i$, $s_i = 0$) and hence $\dot{s}_i = 0$ we get

$$\begin{aligned} \dot{z}_{2e} &= \ddot{z}_{1e} - F_1 \tfrac{d}{dt}[\tfrac{\mu_1 - z_1}{\phi_1}] \\ &= \dot{q}_d - \tfrac{F_1}{\phi_1}[\mu_2 - q_d] \end{aligned} \tag{4.94}$$

Combining eqs. (4.93)–(4.94) we get

$$\dot{z}_{2e} = \ddot{z}_{1e} - [\mu_2 - \dot{z}_{1e}]\left[\lambda(1 - G_1) + \frac{G_1 F_1}{\phi_1}\right] \tag{4.95}$$

where

$$G_1 = \begin{cases} 1 & |\tilde{\mu}_1| \leq \phi_1 \\ 0 & otherwise. \end{cases} \tag{4.96}$$

Since we must take some actual derivatives of the estimated derivatives, we will introduce the notation $(\dot{x}_e)'$ to mean the actual derivative of the estimated derivative of x whenever there might be some confusion. We can easily see from the above derivations that the actual derivative for z_2 is

$$\dot{z}_2 = (\dot{z}_{1e})' - [\mu_2 + D_1 - \dot{z}_1]\left[\lambda(1 - G_1) + \frac{G_1 F_1}{\phi_1}\right]. \tag{4.97}$$

From eqs. (4.95) and (4.97) we can construct

$$\begin{aligned} \dot{z}_{2e} - \dot{z}_2 &= (\ddot{z}_{1e} - (\dot{z}_{1e})') + [\dot{z}_{1e} - \dot{z}_1 + D_1]\left[\lambda(1 - G_1) + \frac{G_1 F_1}{\phi_1}\right] \\ &= 0 \end{aligned} \tag{4.98}$$

since $\dot{z}_{1e} = \dot{z}_1$, $\ddot{z}_{1e} = (\dot{z}_{1e})'$, and $D_1 = 0$. We then have

$$\begin{aligned} \Delta_2 &= D_2 + \dot{z}_{2e} - \dot{z}_2 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3) \\ &= D_2 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3). \end{aligned} \tag{4.99}$$

With substitutions from eq. (4.98), we now construct an F_2 such that $|\Delta_2| \leq F_2$ as

$$F_2 = \bar{D}_2 + \phi_3 \tag{4.100}$$

where we can define $\bar{D}_2$ for the X-29 as

$$\begin{aligned} \bar{D}_2 &\triangleq \sup |D_2| \\ &= \sup \left| \left(-\tilde{C} - \tilde{C}_\alpha(\alpha) - \tilde{C}_\theta(\theta) - \tilde{C}_q q - \tilde{C}_u(u) \right) \right|. \end{aligned} \tag{4.101}$$

From eq. (4.89) we have

$$z_3 = \dot{z}_{2e} - F_2 \text{sat}(\tilde{\mu}_2/\phi_2) - \lambda s_2 - \sigma s_1.$$

Taking the estimated derivative we get

$$\begin{aligned} \dot{z}_{3e} &= \ddot{z}_{2e} - \dot{F}_{2e}\text{sat}(\tilde{\mu}_2/\phi_2) - F_2 \tfrac{d}{dt}[\text{sat}(\tilde{\mu}_2/\phi_2)] - \lambda \dot{s}_2 - \sigma \dot{s}_1 \\ &= \ddot{z}_{2e} - \dot{F}_{2e}\text{sat}(\tilde{\mu}_2/\phi_2) - (\dot{\mu}_2 - \dot{z}_{2e}) \left[\lambda(1 - G_2) + \tfrac{G_2 F_2}{\phi_2} \right] \\ &\quad -\sigma(1 - G_1)(\dot{\mu}_1 - \dot{z}_{1e}) \\ &= \ddot{z}_{2e} - \dot{F}_{2e}\text{sat}(\tilde{\mu}_2/\phi_2) - (\mu_3 - \dot{z}_{2e}) \left[\lambda(1 - G_2) + \tfrac{G_2 F_2}{\phi_2} \right] \\ &\quad -\sigma(1 - G_1)(\mu_2 - \dot{z}_{1e}) \end{aligned} \tag{4.102}$$

with

$$G_2 = \begin{cases} 1 & |\tilde{\mu}_2| \leq \phi_2 \\ 0 & otherwise. \end{cases} \tag{4.103}$$

$$\begin{aligned} \ddot{z}_{2e} &= z_{1e}^{(3)} - [\mu_3 - \ddot{z}_{1e}] \left[\lambda(1 - G_1) + \tfrac{G_1 F_1}{\phi_1} \right] \\ &= \ddot{q}_d - [\mu_3 - \dot{q}_d] \left[\lambda(1 - G_1) + \tfrac{G_1 F_1}{\phi_1} \right] \end{aligned} \tag{4.104}$$

Now we also have

$$\begin{aligned} (\dot{z}_{2e})' &= (\ddot{z}_{1e})' - [\mu_3 + D_2 - (\dot{z}_{1e})'] \left[\lambda(1 - G_1) + \tfrac{G_1 F_1}{\phi_1} \right] \\ &= \ddot{q}_d - [\mu_3 + D_2 - \dot{q}_d] \left[\lambda(1 - G_1) + \tfrac{G_1 F_1}{\phi_1} \right] \end{aligned} \tag{4.105}$$

To construct Δ_3 we need to find the derivative of z_3 with respect to f instead of $\hat{f}$, or $\dot{z}_3$. Similar to eq. (4.102) we then get

$$\begin{aligned}
\dot{z}_3 &= (\dot{z}_{2e})' - F_2' \text{sat}(\tilde{\mu}_2/\phi_2) - F_2 \tfrac{d}{dt}[\text{sat}(\tilde{\mu}_2/\phi_2)] - \lambda s_2' - \sigma s_1' \\
&= (\dot{z}_{2e})' - F_2' \text{sat}(\tilde{\mu}_2/\phi_2) - (\mu_2' - \dot{z}_2)\left[\lambda(1-G_2) + \tfrac{G_2 F_2}{\phi_2}\right] \\
&\quad -\sigma(1-G_1)(\mu_1' - z_1') \\
&= (\dot{z}_{2e})' - F_2' \text{sat}(\tilde{\mu}_2/\phi_2) - (\mu_3 + D_2 - \dot{z}_2)\left[\lambda(1-G_2) + \tfrac{G_2 F_2}{\phi_2}\right] \\
&\quad -\sigma(1-G_1)(\mu_2 + D_1 - \dot{z}_1)
\end{aligned} \tag{4.106}$$

Substituting this in to find Δ_3 we have

$$\begin{aligned}
\Delta_3 &= D_3 + \dot{z}_{3e} - \dot{z}_3 \\
&= D_3 + [\ddot{z}_{2e} - (\dot{z}_{2e})'] - \left[\dot{F}_{2e} - \dot{F}_2\right] \text{sat}(\tilde{\mu}_2/\phi_2) \\
&\quad + [\dot{z}_{2e} - (\dot{z}_2 - D_2)]\left[\lambda(1-G_2) + \tfrac{G_2 F_2}{\phi_2}\right] + \sigma(1-G_1)\left[\dot{z}_{1e} - \dot{z}_1 + D_1\right] \\
&= D_3 + [\ddot{z}_{2e} - (\dot{z}_{2e})'] - \left[\dot{F}_{2e} - \dot{F}_2\right] \text{sat}(\tilde{\mu}_2/\phi_2) \\
&\quad + D_2\left[\lambda(1-G_2) + \tfrac{G_2 F_2}{\phi_2}\right]
\end{aligned} \tag{4.107}$$

since $\dot{z}_{2e} - \dot{z}_2 = 0$ and $D_1 = 0$. From equations (4.104) and (4.105) we can compute

$$\ddot{z}_{2e} - (\dot{z}_{2e})' = D_2\left[\lambda(1-G_1) + \tfrac{G_1 F_1}{\phi_1}\right]. \tag{4.108}$$

Substituting into eq. (4.107) above leads to

$$\begin{aligned}
\Delta_3 &= D_3 + D_2\left[\lambda(1-G_1) + \tfrac{G_1 F_1}{\phi_1} + \lambda(1-G_2) + \tfrac{G_2 F_2}{\phi_2}\right] \\
&\quad - \left[\dot{F}_{2e} - \dot{F}_2\right] \text{sat}(\tilde{\mu}_2/\phi_2)
\end{aligned} \tag{4.109}$$

We then must construct an F_3 for use in eq. (4.90) such that $\Delta_3 \leq F_3$. Finally, the designer must choose values for λ, σ, ϕ_1, ϕ_2, and ϕ_3.

For a first try at the design, assume that we have good knowledge of the

plant. Therefore we can let $D_2 = D_3 = 0$. This leads to

$$\begin{aligned} F_1 &= \phi_2 \\ F_2 &= \phi_3 \\ F_3 &= 0. \end{aligned} \tag{4.110}$$

It also means that the $\dot{z}_{3e}$ equation will have no components due to the control, hence $\dot{z}_{3eu} = 0$. Note that this is not totally unrealistic because a larger ϕ_i allows a larger command following error. This could imply a larger uncertainty of the system model. Also, the more acurate the description of the plant, the closer we approach this setup.

For this simulation we chose:

$$\begin{aligned} \lambda &= 10.0 \\ \sigma &= 1.0 \\ \phi_1 &= 0.02 \\ \phi_2 &= 0.1 \\ \phi_3 &= 0.1 \end{aligned} \tag{4.111}$$

Notice from the many equations for $\dot{z}_i$ and Δ_i that we always have internal gains of the form $\frac{F_i}{\phi_i}$. Looking at equations (4.110) and (4.111) we can see that these internal gains are very reasonable with the highest value being $\frac{F_1}{\phi_1} = 5$. Figure 4.7 shows the simulation results for the same pitch up maneuver as before at a speed of Mach 0.9 and an altitude of 30,000 ft. The commanded control rates for the canard and strake were still outside their limits with a maximum rate at 3.9 sec. of 151 deg/sec for the canard and 88 deg/sec for the strake. These are lower than the rates of the (Runga-Kutta (RK)) controller in the last section, however the price paid is a larger error for θ as seen in Figure 4.8. The value of ϕ_1 chosen only ensures that the error approaches a value of $0.02 * \frac{180}{\pi} = 1.146°$.

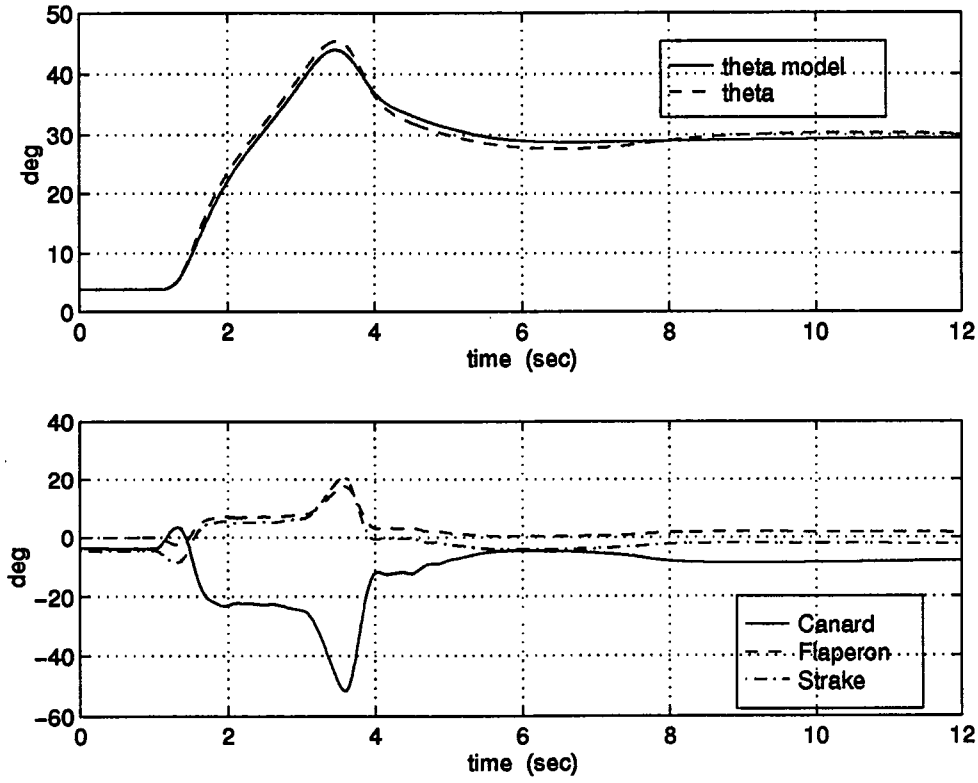


Figure 4.7: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft using the robust controller.

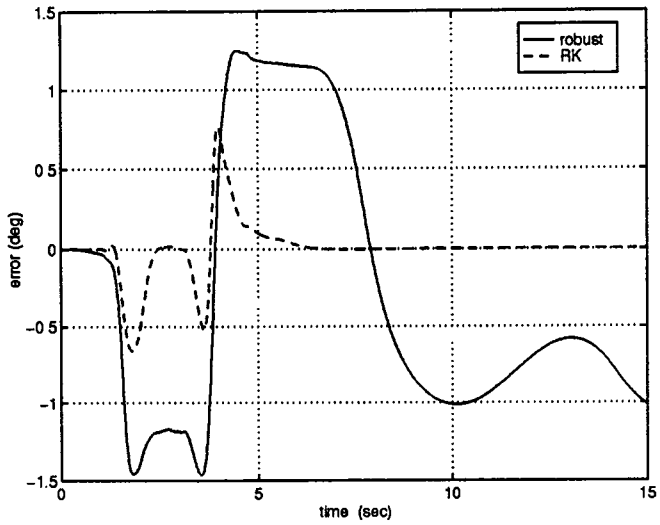


Figure 4.8: Model-following error for a pitch up maneuver at Mach 0.9 and 30,000 ft using the robust controller.

One possible solution to decreasing the error would be to use a lower value for ϕ_1. However, this would lead to higher control rates as we saw before. A better solution would be to add integration to the model-following controller. This is the topic for the next section.

4.3.1 Integral Robust Control

To aid the controller in following a signal, we can redesign the robust controller in the previous section to include integral action. Let us define $\mu_0 = \int \mu_1$ and $z_0 = \int y_d$. We then let

$$s_0 = \tilde{\mu}_0 - \phi_0 \text{sat}(\tilde{\mu}_0/\phi_0) \tag{4.112}$$

where as before $\tilde{\mu}_0 = \mu_0 - z_0 = \int(\mu_1 - y_d)$, and ϕ_0 is a strictly positive quantity. When the $\text{sat}(\tilde{\mu}_0/\phi_0)$ is not in effect ($\tilde{\mu}_0 < \phi_0$) $s_0 = 0$ and hence $\dot{s}_0 = 0$.

Otherwise we get

$$\begin{aligned} \dot{s}_0 &= \dot{\mu}_0 - \dot{z}_0 \\ &= \mu_1 - y_d \\ &= s_1 + \phi_1 \text{sat}(\tilde{\mu}_1/\phi_1) + z_1 - y_d \end{aligned} \tag{4.113}$$

Letting

$$\Delta_0 = \dot{z}_{0e} - \dot{z}_0 + \phi_1 \text{sat}(\tilde{\mu}_1/\phi_1) \tag{4.114}$$

$$= \phi_1 \text{sat}(\tilde{\mu}_1/\phi_1) \tag{4.115}$$

$$\tag{4.116}$$

we get (for $s_0 \neq 0$)

$$\dot{s}_0 = s_1 + z_1 - y_d + \Delta_0 \tag{4.117}$$

If we let $F_0 = \phi_1$ then we are assured that $|\Delta_0| \leq F_0$. We now add another term to the Lyapunov function candidate of equation (4.86) which we will now refer to as V_1 to get

$$V_2 = \frac{1}{2}\sum_{i=0}^{3} \sigma^{3-i} s_i^2 = \sigma^3 s_0^2 + V_1 \tag{4.118}$$

Taking the derivative we then get

$$\begin{aligned} \dot{V}_2 &= \sigma^3 s_0(s_1 + z_1 - \dot{z}_{0e} + \Delta_0) \\ &\quad + \sigma^2 s_1(s_2 + z_2 - \dot{z}_{1e} + \Delta_1) \\ &\quad + \sigma s_2(s_3 + z_3 - \dot{z}_{2e} + \Delta_2) \\ &\quad + s_3(L_f^3 h + L_g L_f^2 h\nu - \dot{z}_{3e} + \Delta_3). \end{aligned}$$

Redefining equations (4.87)- (4.89) and adding z_0 we have

$$z_0 = \int y_d \tag{4.119}$$

$$z_1 = y_d - F_0 \text{sat}(\tilde{\mu}_0/\phi_0) - \lambda_0 s_0 \tag{4.120}$$

$$z_2 = \dot{z}_{1e} - F_1 \text{sat}(\tilde{\mu}_1/\phi_1) - \lambda s_1 - \sigma s_0 \tag{4.121}$$

$$z_3 = \dot{z}_{2e} - F_2 \text{sat}(\tilde{\mu}_2/\phi_2) - \lambda s_2 - \sigma s_1 \tag{4.122}$$

where λ_0 is a strictly positive constant. Choosing the same control as before, i.e. equation (4.90) we still have (in a proof analogous to the previous case)

$$\dot{V}_2 + 2\lambda V_1 + \sigma^3 \lambda_0 s_0^2 \leq 0$$

which shows that the s_i all exponentially converge to zero. We use λ_0 instead of λ so we can adjust the gain on the integrator independently. Using the same logic, we could use a different λ_i for each z_i if we desired.

We now need to check how the new definitions for z_i affect $\dot{z}_i$. With

$$z_1 = y_d - F_0 \text{sat}(\tilde{\mu}_0/\phi_0) - \lambda s_0$$

and $F_0 = \phi_1$, we get

$$\dot{z}_{1e} = \dot{y}_d - [\mu_1 - y_d] \left[\lambda(1 - G_0) + \frac{G_0 F_0}{\phi_0} \right] \tag{4.123}$$

where

$$G_0 = \begin{cases} 1 & |\tilde{\mu}_0| \leq \phi_0 \\ 0 & otherwise. \end{cases} \tag{4.124}$$

This is similar in style to what was previously $\dot{z}_{2e}$. Next we look at

$$z_2 = \dot{z}_{1e} - F_1 \text{sat}(\tilde{\mu}_1/\phi_1) - \lambda s_1 - \sigma s_0.$$

Taking the estimated derivative we get

$$\dot{z}_{2e} = \ddot{z}_{1e} - [\mu_2 - \dot{z}_{1e}] \left[\lambda(1 - G_1) + \frac{G_1 F_1}{\phi_1} \right] - \sigma(\mu_1 - y_d)(1 - G_0) \tag{4.125}$$

since $F_1 = \phi_2$ and hence $\dot{F}_1 = 0$. For $\ddot{z}_{1e}$ we get

$$\ddot{z}_{1e} = \ddot{y}_d - [\mu_2 + D_1 - \dot{y}_d] \left[\lambda(1 - G_0) + \frac{G_0 F_0}{\phi_0} \right]. \tag{4.126}$$

Since $D_1 = 0$ we still have $\ddot{z}_{1e} = (\dot{z}_{1e})'$. Therefore, as in the previous case we have $\dot{z}_{2e} = \dot{z}_2$ so equation (4.99) for Δ_2 is unchanged. Since the equation

for z_3 remains unchanged (equation (4.89) or (4.122)), the only change for $\dot{z}_{3e}$ (equation (4.102)) is the contribution due to $\ddot{z}_{2e}$. Starting from equation (4.125) and taking the estimated derivative we get

$$\ddot{z}_{2e} = z_{1e}^{(3)} - [\mu_3 - \ddot{z}_{1e}]\left[\lambda(1-G_1) + \frac{G_1F_1}{\phi_1}\right] - \sigma(\mu_2 - \dot{y}_d)(1-G_0) \tag{4.127}$$

with

$$z_{1e}^{(3)} = y_d^{(3)} - [\mu_3 - \ddot{y}_d]\left[\lambda_0(1-G_0) + \frac{G_0F_0}{\phi_0}\right]. \tag{4.128}$$

We also need to calculate $\dot{z}_3$ in order to find Δ_3. The same reasoning applies here, so we only need to find the new form for $(\dot{z}_{2e})'$. Starting from equation (4.125) and taking the actual derivative we get

$$(\dot{z}_{2e})' = (\ddot{z}_{1e})' - [\mu_3 + D_2 - (\dot{z}_{1e})']\left[\lambda(1-G_1) + \frac{G_1F_1}{\phi_1}\right] - \sigma(\mu_2 + D_1 - \dot{y}_d)(1-G_0) \tag{4.129}$$

with

$$(\ddot{z}_{1e})' = y_d^{(3)} - [\mu_3 + D_2 - \ddot{y}_d]\left[\lambda_0(1-G_0) + \frac{G_0F_0}{\phi_0}\right]. \tag{4.130}$$

From equation (4.107) we saw that

$$\begin{aligned} \Delta_3 &= D_3 + [\ddot{z}_{2e} - (\dot{z}_{2e})'] - \left[\dot{F}_{2e} - \dot{F}_2\right]\mathrm{sat}(\tilde{\mu}_2/\phi_2) \\ &\quad + D_2\left[\lambda(1-G_2) + \tfrac{G_2F_2}{\phi_2}\right]. \end{aligned} \tag{4.131}$$

Combining equations (4.127) and (4.129) we can then compute

$$\begin{aligned} \ddot{z}_{2e} - (\dot{z}_{2e})' &= (z_{1e}^{(3)} - (\ddot{z}_{1e})') + (\ddot{z}_{1e} - (\dot{z}_{1e})') \\ &\quad + D_2\left[\lambda(1-G_1) + \tfrac{G_1F_1}{\phi_1}\right] + \sigma D_1(1-G_0) \\ &= D_2\left[\lambda_0(1-G_0) + \tfrac{G_0F_0}{\phi_0}\right] + D_2\left[\lambda(1-G_1) + \tfrac{G_1F_1}{\phi_1}\right]. \end{aligned} \tag{4.132}$$

Substituting into equation (4.131) above leads to

$$\begin{aligned} \Delta_3 &= D_3 - \left[\dot{F}_{2e} - \dot{F}_2\right]\mathrm{sat}(\tilde{\mu}_2/\phi_2) \\ &\quad + D_2\left[\lambda_0(1-G_0) + \tfrac{G_0F_0}{\phi_0} + \lambda(1-G_1) + \tfrac{G_1F_1}{\phi_1} + \lambda(1-G_2) + \tfrac{G_2F_2}{\phi_2}\right] \end{aligned} \tag{4.133}$$

With the same assumptions as before, e.g. $D_2 = D_3 = 0$ we use

$$\begin{aligned} F_1 &= \phi_2 \\ F_2 &= \phi_3 \\ F_3 &= 0 \end{aligned} \tag{4.134}$$

along with $F_0 = \phi_0$ (which will always be true). Again, the $\dot{z}_{3e}$ equation will have no components due to the control.

For this simulation we use the same values as before:

$$\begin{aligned} \lambda &= 10.0 \\ \sigma &= 1.0 \\ \phi_1 &= 0.02 \\ \phi_2 &= 0.1 \\ \phi_3 &= 0.1 \end{aligned} \tag{4.135}$$

with the additional gains of

$$\begin{aligned} \lambda_0 &= 1.0 \\ \phi_0 &= 0.01. \end{aligned} \tag{4.136}$$

Figure 4.9 shows simulation results for the identical pitch up maneuver at a speed of Mach 0.9 and an altitude of 30,000 ft. The output follows the command much more closely as seen in Figure 4.10. The error labeled RK is for the control law outlined in the previous section in equations (4.47) and (4.52). A benefit of using integral action to decrease the error is lower control rates. We now have a maximum rate at 3.7 sec. of 128 deg/sec for the canard and 73 deg/sec for the strake. These are still outside their allowable ranges, but closer. The next section outlines a method for dealing with the limits on the actuators.

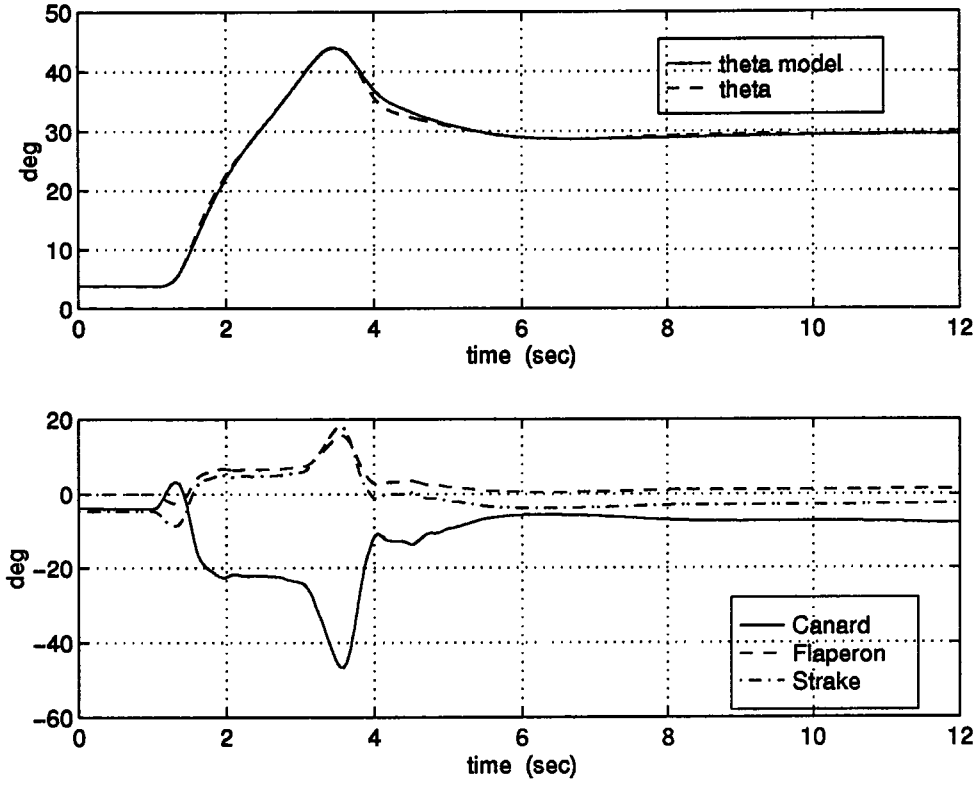


Figure 4.9: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft using the robust controller with integral action.

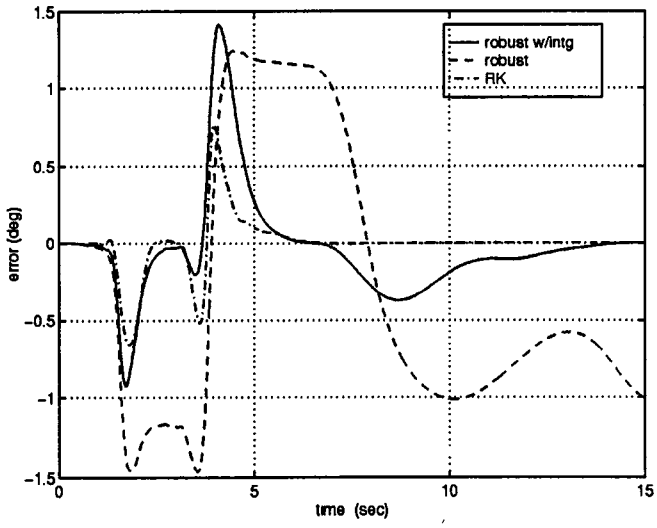


Figure 4.10: Comparison of the model-following error for the robust controller with integral action.

4.3.2 Implementation for limited control effectors

Let us first look at an implementation of the robust control algorithm where we assume that we know the actual system description (at least to within some Δ defined by ϕ_i). However, since we wish to keep the simulation realistic, we will include actuator rate and position limiting in the system description. Since these do not fit into the analytic description of the system, they will be incorporated into the uncertainty information. The difference between the calculated control and that which can actually be used (due to limiting) is added to the D_2 term. This also adds to the D_3 term. The following construction applies to either the plain robust controller or the controller with the integrator added. Only the forms for the derivatives for z_{ie} will change. For simplicity we will show the forms of the derivatives assuming we will be using the controller with the integral added. If not, just use the derivatives shown in section 4.3.

We need to construct the accompanying forms for F_1, F_2, and F_3 in order to

use the control defined by eq. (4.90). From eq. (4.91) we have

$$\begin{aligned} \Delta_1 &= D_1 + \dot{z}_{1c} - \dot{z}_1 + \phi_2 \mathrm{sat}(\tilde{\mu}_2/\phi_2) \\ &= \phi_2 \mathrm{sat}(\tilde{\mu}_2/\phi_2). \end{aligned} \tag{4.137}$$

Therefore we get

$$|\Delta_1| \leq F_1 = \phi_2. \tag{4.138}$$

Since we assume we have a good model of the plant, most of the D_2 term from eq. (4.76) vanishes except the part due to actuator limits. We get

$$D_2 = -\tilde{C}_u(u) \tag{4.139}$$

where $\tilde{C}_u(u)$ is the difference between the control contribution to the $\dot{q}$ equation using the (unlimited) calculated control from eq. (4.90) and the actual (rate and position limited) control contribution and is defined by

$$\tilde{C}_u(u) = \hat{C}_u(u) - C_u(u).$$

Using the reduced set of force and moment equations from eqs. (4.16)-(4.18) we get

$$\hat{C}_u(u) = \frac{1}{I_{YY}} \bar{q} S \bar{c} \left[C_{m_u} u + \frac{\bar{c}}{2V} C_{m_{\dot{\alpha}}} C_{u_{\dot{\alpha}}}(u) \right] \tag{4.140}$$

with

$$C_{u_{\dot{\alpha}}}(u) = \left(1 + \cos\alpha \frac{\bar{q} S \bar{c}}{2mV^2} C_{N_{\dot{\alpha}}} \right)^{-1} \frac{\bar{q} S}{mV} \left[\left(C_{A_\delta} u + C_{A_{\delta^2}} u^2 \right) \sin\alpha - C_{N_\delta} u \cos\alpha \right]$$

and C_{m_u}, C_{A_δ}, $C_{A_{\delta^2}}$, and C_{N_δ} defined by eqs. (4.38)-(4.41), respectively. The u used in the above equations is derived from the control update equation, eq. (4.90), as $u = \int \nu$. We then let the actual control contribution be defined by

$$C_u(u) = \hat{C}_u(u) \mid_{u=u_{lmt}} \tag{4.141}$$

where u_{lmt} is the actual limited value for the control surface commands.

Then as we had before in eq. (4.99) we get

$$\begin{aligned} \Delta_2 &= D_2 + \dot{z}_{2e} - \dot{z}_2 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3) \\ &= D_2 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3). \end{aligned} \tag{4.142}$$

In order to use a less conservative bound for Δ_2 we simply let

$$F_2 \triangleq |D_2| + \phi_3. \tag{4.143}$$

Repeating equation (4.102) we have

$$\begin{aligned} \dot{z}_{3e} &= \ddot{z}_{2e} - \dot{F}_{2e}\text{sat}(\tilde{\mu}_2/\phi_2) - (\mu_3 - \dot{z}_{2e})\left[\lambda(1-G_2) + \tfrac{G_2F_2}{\phi_2}\right] \\ &\quad -\sigma(1-G_1)(\mu_2 - \dot{z}_{1e}) \end{aligned} \tag{4.144}$$

with

$$\ddot{z}_{2e} = z_{1e}^{(3)} - [\mu_3 - \ddot{z}_{1e}]\left[\lambda(1-G_1) + \frac{G_1F_1}{\phi_1}\right] - \sigma(\mu_2 - \dot{y}_d)(1-G_0) \tag{4.145}$$

and

$$z_{1e}^{(3)} = y_d^{(3)} - [\mu_3 - \ddot{y}_d]\left[\lambda_0(1-G_0) + \frac{G_0F_0}{\phi_0}\right]. \tag{4.146}$$

According to the procedure, we need to separate the $\dot{z}_{3e}$ equation into state components and components due to the control. Looking through equations (4.144)–(4.146), the only terms which can possibly have any contributions due to the control are μ_3 and $\dot{F}_{2e}$. However, since our system description is in the extended form this is not possible (by definition of relative degree) for μ_3. Let $u_{lmt} = u - u_\delta$ then we have

$$\begin{aligned} \frac{du_{lmt}}{dt} &= \frac{du}{dt} - \frac{du_\delta}{dt} \\ &= \nu - \dot{u}_\delta. \end{aligned}$$

Looking more closely at $\dot{F}_{2e}$ we get

$$\begin{aligned}
\dot{F}_{2e} &= \operatorname{sgn}(D_2)\tfrac{d}{dt}(D_2) \\
&= \operatorname{sgn}(D_2)\tfrac{d}{dt}\left[\hat{C}_u(u_{lmt}) - \hat{C}_u(u)\right] \\
&= \operatorname{sgn}(D_2)\left[\tfrac{\partial}{\partial u}\hat{C}_u(u_{lmt})(\nu - \dot{u}_\delta) - \tfrac{\partial}{\partial u}\hat{C}_u(u)\nu\right] \\
&= \tfrac{1}{I_{yy}}\operatorname{sgn}(D_2)\left[M_u(u_{lmt})(\nu - \dot{u}_\delta) - M_u(u)\nu\right] \\
&= \tfrac{1}{I_{yy}}\operatorname{sgn}(D_2)\left[\left(M_u(u-u_\delta) - M_u(u)\right)\nu - M_u(u-u_\delta)\dot{u}_\delta)\right]. \\
&\triangleq \dot{F}_{2eu}\nu - \dot{F}_{2ex}
\end{aligned}$$

Substituting in for M_u from equation (4.28) we get

$$\begin{aligned}
\dot{F}_{2eu} &= \tfrac{1}{I_{yy}}\operatorname{sgn}(D_2)\left\{\bar{q}S\bar{c}\tfrac{\bar{c}}{2V}C_{m_{\dot{\alpha}}}\tfrac{\bar{q}S}{mV}(1+\cos\alpha\tfrac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1}\left[2C_{A_{\delta 2}}(-u_\delta)\sin\alpha\right]\right\} \\
&= \operatorname{sgn}(D_2)\left\{\tfrac{S^2\bar{c}^2\rho^2V^2}{8mI_{yy}}C_{m_{\dot{\alpha}}}(1+\cos\alpha\tfrac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1}\left[2C_{A_{\delta 2}}(-u_\delta)\sin\alpha\right]\right\}.
\end{aligned}$$

The constant $\frac{S^2\bar{c}^2\rho^2V^2}{8mI_{yy}} = .0081$ and the term $(1+\cos\alpha\frac{\bar{q}S\bar{c}}{2mV^2}C_{N_{\dot{\alpha}}})^{-1} \approx 1$ throughout the flight envelope. The coefficient $C_{A_{\delta 2}}$ is on the order of 10^{-5} and the remaining terms are on the order of 1. Therefore, $\dot{F}_{2eu}$ is on the order of 10^{-7} throughout the flight envelope and can be neglected.

Turning our attention to Δ_3 we see from equation (4.133) that

$$\begin{aligned}
\Delta_3 &= D_3 - \left[\dot{F}_{2e} - \dot{F}_2\right]\operatorname{sat}(\tilde{\mu}_2/\phi_2) \\
&\quad + D_2\left[\lambda_0(1-G_0) + \tfrac{G_0F_0}{\phi_0} + \lambda(1-G_1) + \tfrac{G_1F_1}{\phi_1} + \lambda(1-G_2) + \tfrac{G_2F_2}{\phi_2}\right] \\
&= D_3 + D_2\left[\lambda_0(1-G_0) + \tfrac{G_0F_0}{\phi_0} + \lambda(1-G_1) + \tfrac{G_1F_1}{\phi_1} + \lambda(1-G_2) + \tfrac{G_2F_2}{\phi_2}\right]
\end{aligned} \tag{4.147}$$

where

$$D_3 = \bar{q}S\bar{c}\frac{1}{I_{YY}}\cdot\frac{\bar{c}}{2V}C_{m_q}D_2. \tag{4.148}$$

Since D_2 is only a function of u, $\dot{F}_{2e} = \dot{F}_2$. For a less conservative bound for Δ_3 we simply define

$$F_3 \triangleq |\Delta_3|. \tag{4.149}$$

The same control law is used, equation 4.90 (repeated here for convenience), only the values for F_3 and $\dot{z}_{3e}$ will change

$$\nu = (L_g L_{\hat{f}}^2 h - \dot{z}_{3eu})^{-1}(-L_{\hat{f}}^3 h + \dot{z}_{3ex} - F_3 \mathrm{sat}(\tilde{\mu}_3/\phi_3) - \lambda s_3 - \sigma s_2).$$

We use the same controller values as before:

$$\begin{aligned} \lambda &= 10.0 \\ \sigma &= 1.0 \\ \phi_1 &= 0.02 \\ \phi_2 &= 0.1 \\ \phi_3 &= 0.1 \\ \lambda_0 &= 1.0 \\ \phi_0 &= 0.01. \end{aligned} \tag{4.150}$$

Figure 4.11 shows the performance of the system with the new controller. As shown the error goes to zero. This time all the control rates are under or on their limits. Both the canard and strake hit their rate limits at 3.9 sec. as we can clearly see in the case of the strake. The flaperons stay below their limits with a maximum of 57 deg/sec. Figure 4.12 shows a comparison of the model-following error for this controller and the one in the last section without rate limits.

Again, verifying that the controller could perform satisfactorily over the entire flight envelope, we used the same control law (with different aerodynamic coefficients for the new operating point) for the same pitch up maneuver at two different speeds, Mach 0.6 and 1.2, both at an altitude of 15,000 ft. Figure 4.13 shows the performance of the X-29 for the pitch up maneuver at Mach 0.6. Both the canard and strake hit their rate limits at 1.5 sec. into the maneuver while the flaperons stay below their limits with a maximum of 59 deg/sec. Figure 4.14

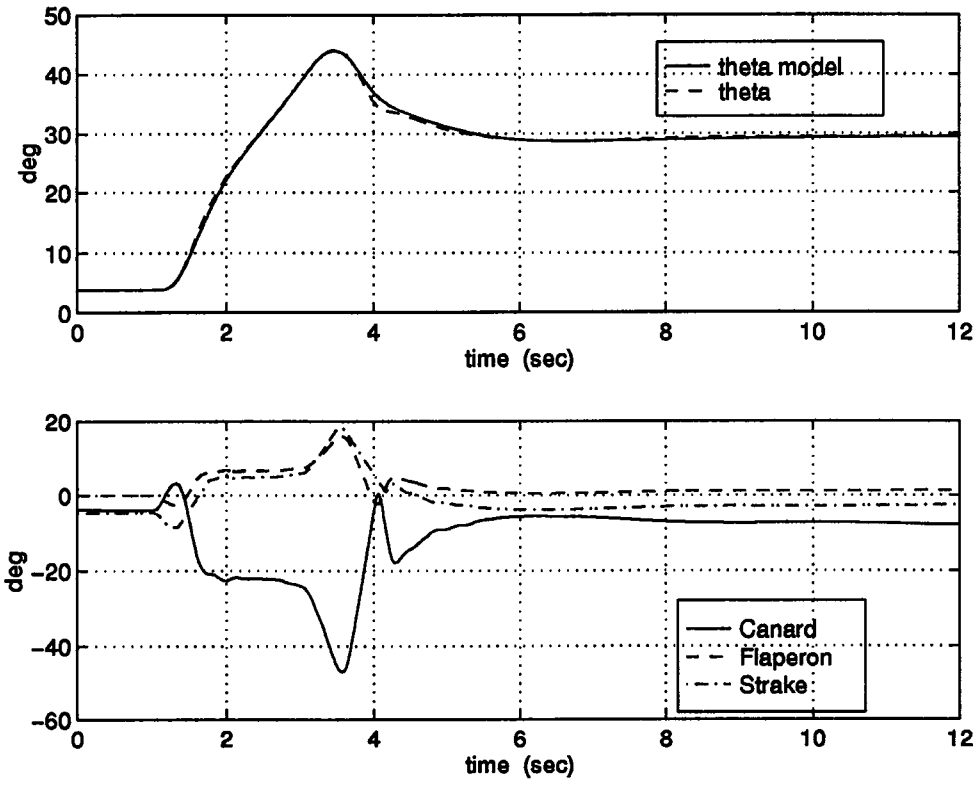


Figure 4.11: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.9 and 30,000 ft using the rate-limited robust controller with integral action.

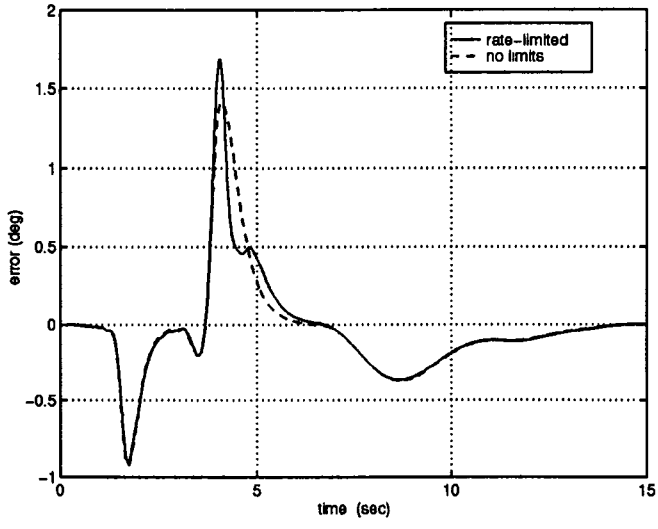


Figure 4.12: Comparison of the model-following error for the robust controller.

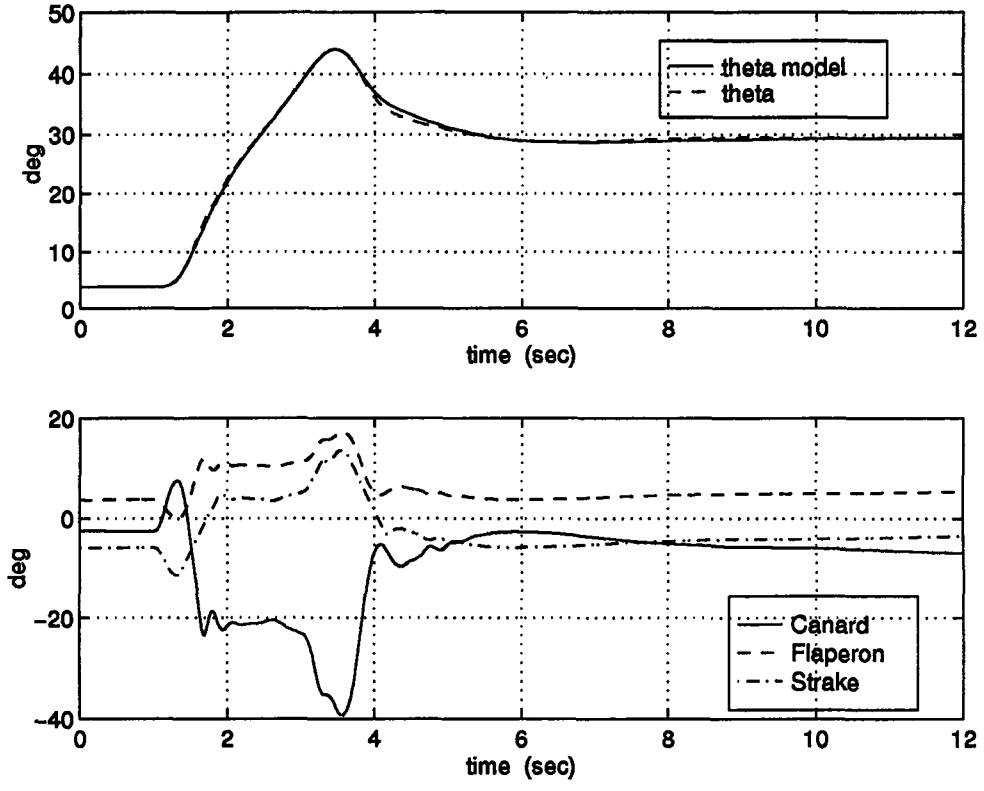


Figure 4.13: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 0.6 and 15,000 ft using the rate-limited robust controller with integral action.

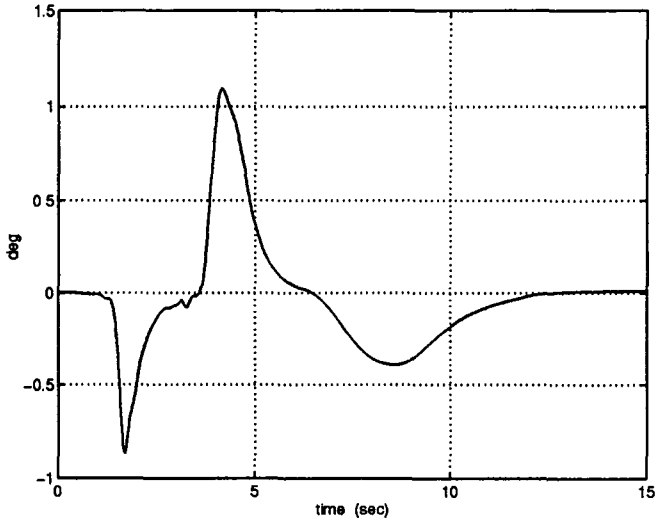


Figure 4.14: Model-following error for the robust controller at Mach 0.6 at 15,000 ft.

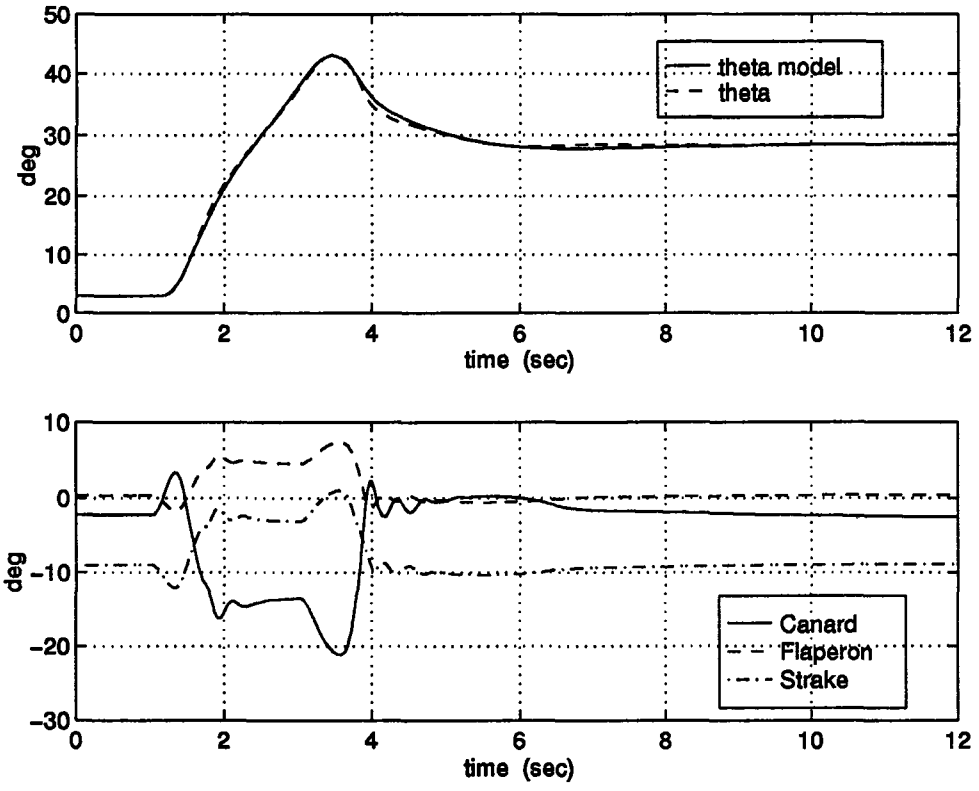


Figure 4.15: Simulation of the X-29 aircraft in a pitch up maneuver at Mach 1.2 and 15,000 ft using the rate-limited robust controller with integral action.

shows the model-following error for this controller. Figure 4.15 shows the performance of the X-29 for the pitch up maneuver at Mach 1.2. Since the X-29 is more maneuverable at the higher speed, and because we are at a higher dynamic pressure, the maneuver is easier to perform. Only the strake hits the rate limits at 3.7 sec. into the maneuver. Figure 4.16 shows the model-following error for this controller. Although the maneuver is easier to perform, the aircraft is close to it's structural limits as seen by the plot of the normal acceleration in Figure 4.17.

As seen in the figures, performance is acceptable. If this were not the case, since we do neglect some of the system dynamics, then we would need to add

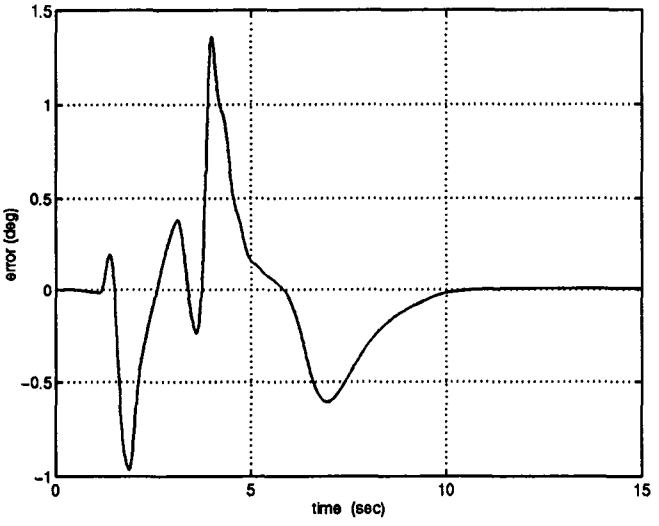


Figure 4.16: Model-following error for the robust controller at Mach 1.2 at 15,000 ft.

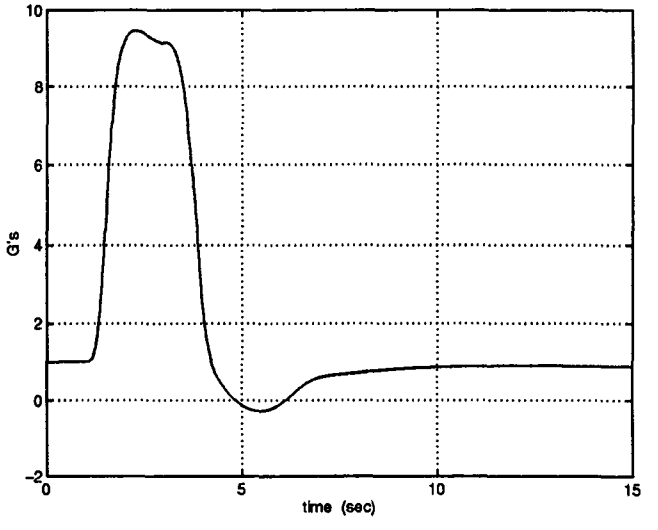


Figure 4.17: Normal acceleration of the X-29 at Mach 1.2 at 15,000 ft.

the additional dynamics to the system description and compute new equations for F_1, F_2, and F_3. This is described in the next section.

4.3.3 Complete implementation

The last section gave a description of the robust controller for the case when we have confidence in the description of our system to within Δ_i. In this section we will outline how to construct the additional terms needed for the controller to account for the neglected dynamics (this is essentially the difference between eqs. (4.1), (4.2), (4.3) and the reduced set of equations (4.16), (4.17), (4.18), respectively) . Ironically, the better we make the description of the neglected dynamics, the more likely it is that they can be used in the system equations for the controller calculations directly. For this reason we will only sketch out the procedure we would use to get a rough estimate of the neglected terms.

By definition of $\tilde{f}$ and y, we still have $D_1 = 0$. Therefore, from eq. (4.92) we have

$$F_1 = \phi_2.$$

From eq. (4.99) we have

$$\Delta_2 = D_2 + \phi_3 \text{sat}(\tilde{\mu}_3/\phi_3)$$

with (eq. (4.76))

$$D_2 = -\tilde{C} - \tilde{C}_\alpha(\alpha) - \tilde{C}_\theta(\theta) - \tilde{C}_q q - \tilde{C}_u(u). \tag{4.151}$$

Looking at the neglected terms from the force and moment equations (4.1), (4.2) and (4.3) we see that equation (4.151) can not be as clearcut as stated. Mach number aside, almost all of the neglected terms have some dependence on α.

Equation (4.151) should actually look more like

$$D_2 = -\tilde{C}_q(\alpha)q - \tilde{C}_u(\alpha, u). \tag{4.152}$$

The first term, $\tilde{C}_q(\alpha)$, is composed entirely from high angle-of-attack terms. The second term, $\tilde{C}_u(\alpha, u)$, can be seperated into a portion due to high angle-of-attack terms and a portion that can be valid at any time. First let us focus on the high angle-of-attack terms. The main contributions to the high angle-of-attack terms comes from the C_m equation:

$$\begin{aligned}
\Delta_{C_mHA} &= HA\Delta C_{m_{\delta_F}}(\alpha, M, \delta_F) \\
&\quad + HA\Delta C_{m_{\delta_S}}(\alpha, \delta_S) + HA^2\Delta C_{m_{\delta_S}}(\alpha, M, \delta_S) \\
&\quad + HA\Delta C_{m_{\delta_C}}(\alpha, \delta_C, \delta_F) + HA^2\Delta C_{m_{\delta_C}}(\alpha, M, \delta_C) \\
&\quad + HA\Delta C_{m_{\delta_{C_{\delta_F}}}}(\alpha, M, \delta_C, \delta_F) + HA\Delta C_{m_{\delta_{C_{\delta_S}}}}(\alpha, M, \delta_C, \delta_S)
\end{aligned}$$

We will also get some contribution from C_A and C_N as

$$\begin{aligned}
\Delta_{C_AHA} &= HA\Delta C_{A_{\delta_F}}(\alpha, M, \delta_F) \\
&\quad + HA\Delta C_{A_{\delta_S}}(\alpha, \delta_S) + HA^2\Delta C_{A_{\delta_S}}(\alpha, M, \delta_S) \\
&\quad + HA\Delta C_{A_{\delta_C}}(\alpha, \delta_C, \delta_F) + HA^2\Delta C_{A_{\delta_C}}(\alpha, M, \delta_C) \\
&\quad + HA\Delta C_{A_{\delta_{C_{\delta_F}}}}(\alpha, M, \delta_C, \delta_F) + HA\Delta C_{A_{\delta_{C_{\delta_S}}}}(\alpha, M, \delta_C, \delta_S)
\end{aligned}$$

and

$$\begin{aligned}
\Delta_{C_NHA} &= HA\Delta C_{N_{\delta_F}}(\alpha, M, \delta_F) \\
&\quad + HA\Delta C_{N_{\delta_S}}(\alpha, \delta_S) + HA^2\Delta C_{N_{\delta_S}}(\alpha, M, \delta_S) \\
&\quad + HA\Delta C_{N_{\delta_C}}(\alpha, \delta_C, \delta_F) + HA^2\Delta C_{N_{\delta_C}}(\alpha, M, \delta_C) \\
&\quad + HA\Delta C_{N_{\delta_{C_{\delta_F}}}}(\alpha, M, \delta_C, \delta_F) + HA\Delta C_{N_{\delta_{C_{\delta_S}}}}(\alpha, M, \delta_C, \delta_S)
\end{aligned}$$

The overall contribution for the high angle-of-attack terms will then be

$$\begin{aligned}\Delta_{HA} &= \bar{q}S\bar{c}\left[\tfrac{1}{I_{YY}}\Delta_{C_mHA} + \tfrac{1}{2V}\tfrac{1}{mV}\cos(\alpha)\Delta_{C_NHA}\right. \\ &\quad \left. -\tfrac{1}{2V}\tfrac{1}{mV}\sin(\alpha)\Delta_{C_AHA}\right].\end{aligned} \tag{4.153}$$

Since Δ_{C_mHA}, Δ_{C_NHA}, and Δ_{C_AHA} are on the same order, and $\frac{1}{I_{YY}} \gg \frac{1}{2mV^2}$ the dominant high angle-of-attack contributions will be

$$\Delta_{HA} = \bar{q}S\bar{c}\left[\tfrac{1}{I_{YY}}\Delta_{C_mHA}\right]. \tag{4.154}$$

Looking at the equation above for Δ_{C_mHA} we see that the Δ_{HA} term would entirely contribute to the control uncertainty term. The remaining control uncertainty would come from the incremental control terms as

$$\begin{aligned}\Delta_u &= \Delta C_{m_{\delta_F}}(\alpha, \delta_F, M) + \Delta C_{m_{\delta_S}}(\alpha, \delta_S, M) \\ &\quad + \Delta C_{m_{\delta_C}}(\alpha, \delta_C, \delta_F, M) \cdot K_{m_{\delta_{C_{\delta_S}}}}(\alpha, \delta_C, \delta_S) \\ &\quad - C_{m_{\delta_C}}(M) \cdot \left[1 - K_{m_{\delta_{C_{\delta_S}}}}(\alpha, \delta_C, \delta_S)\right].\end{aligned} \tag{4.155}$$

We would then get

$$\tilde{C}_u(\alpha, u) = \Delta_{HA} + \bar{q}S\bar{c}\frac{1}{I_{YY}}\Delta_u. \tag{4.156}$$

To find D_3 we would calculate

$$D_3 = \bar{q}S\bar{c}\frac{1}{I_{YY}} \cdot \frac{\bar{c}}{2V}C_{m_q}D_2. \tag{4.157}$$

Figure 4.18 shows a plot of Δ_{HA} for the maneuver flown in the last section at Mach 0.9 at 30,000 ft. Figure 4.19 shows a plot of the remaining neglected control terms, namely, $\bar{q}S\bar{c}\frac{1}{I_{YY}}\Delta_u$ for the same maneuver. The first term in the D_2 equation above would only be comprised of high angle-of-attack terms as

$$\tilde{C}_q(\alpha) = \bar{q}S\bar{c}\frac{1}{I_{YY}}\frac{\bar{c}}{2V}\left[HA\Delta C_{m_q}(\alpha)\right]. \tag{4.158}$$

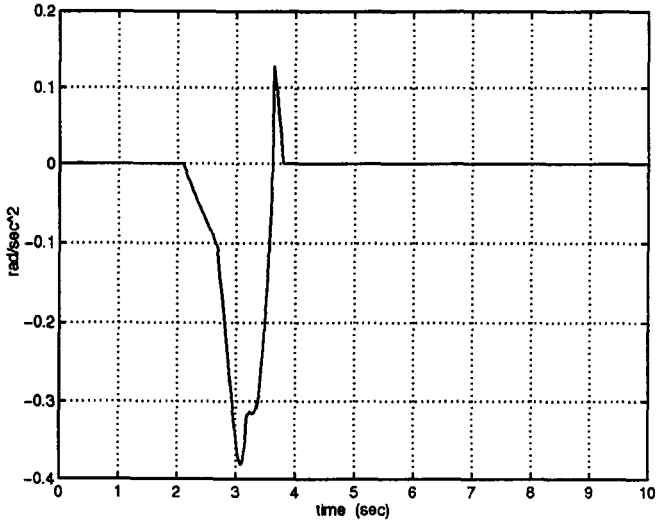


Figure 4.18: Plot of the high angle-of-attack contributions to D_2.

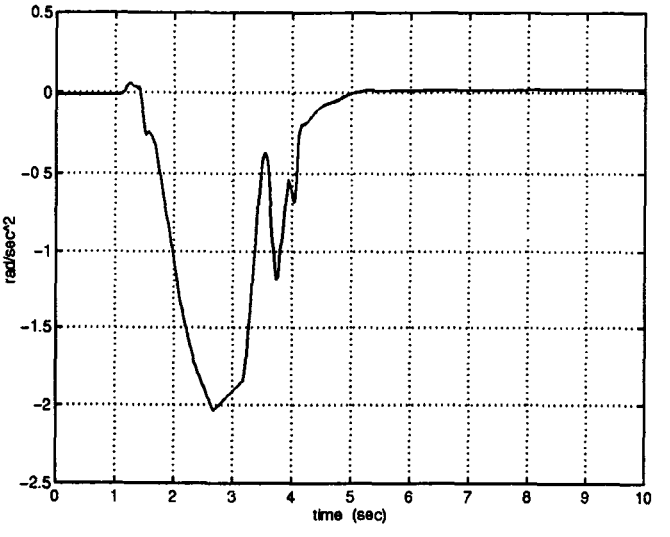


Figure 4.19: Plot of the incremental control contributions to D_2.

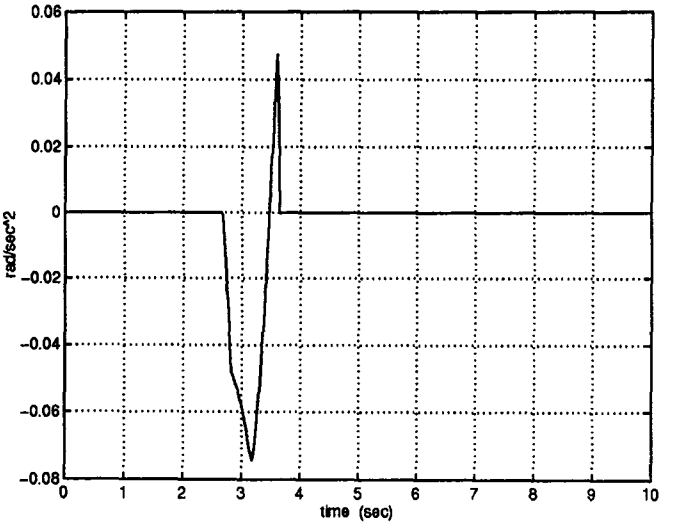


Figure 4.20: Plot of the high angle-of-attack pitch contributions to D_2.

Figure 4.20 shows a plot of the high angle-of-attack terms due to q, namely, $\tilde{C}_q(\alpha) \cdot q$ for the same maneuver.

The figures show the neglected incremental pitching moment terms for a given maneuver. If the performance of the controller in the last section was poor due to neglected terms in the pitching moment equation then we would need to model these contributions. However, as stated earlier, if we can find an analytical form for the neglected terms, we might as well include them directly in the controller calculations and avoid the uncertainty modelling.

Chapter 5

Conclusions and Future Directions

In this dissertation we have outlined the procedure for designing a nonlinear robust controller based on feedback linearization methods. This work was motivated by the need for an algorithmic approach to designing a global controller for nonlinear systems. For application to aerospace systems we needed the ability for the controller to accept desired trajectories as inputs and to be robust to modelling uncertainties. In Chapter 3 we showed preliminary results of using a numerical Newton-Raphson technique for inversion of the X-29 aircraft equations. Since this method cannot guarantee convergence in light of modelling uncertainties, and due to the guessing nature of the algorithm, this method was deemed unsuitable for application to aircaft systems which require a high level of reliability. In Chapter 4 we applied feedback linearization methods, modified by Slotine and Hedrick [28] to include sliding control methods to add robustness, to a high fidelity nonlinear simulation of the X-29. We designed a controller for the

longitudinal axis to follow a set of given trajectories. Since the controller requires state feedback, the actuator models must be left out of the design. Therefore, in addition to possible system model uncertainties, the controller must be able to handle the nonlinearities introduced by the actuators. To complicate the design, the actuators on the X-29 have rate and displacement limits which limit performance and affect stability. By formulating the limits of the actuators into the uncertainty models of the system, we have enabled the controller to account for the nonlinearities added by the actuators. We have shown good performance of our controller as shown by plots of the output of the X-29 for various operating points throughout the flight envelope.

Our contributions to the field of nonlinear control can be summarized as follows:

- An engineering application of a robust nonlinear controller to a high fidelity nonlinear simulation of the X-29 aircraft.

- Formulated a method to describe the nonlinearities introduced by the actuators which encapsulate the limitations of the actual X-29 actuators.

- Modified the method of Slotine and Hedrick [28] to include integral action in the controller to force the position model following errors to zero. This modification is much more desirable for actual implementation than simply increasing the controller gains because higher gains produce higher control rates and decrease stability.

We have designed a controller for the longitudinal axis of the X-29 aircraft. In the actual implementation of this controller, the maneuvers chosen would need to statisfy pilot handling qualities criteria. This can be addressed simply by con-

structing the model generator to output trajectories the pilot would like. We did not address this issue since it would not effect the design of the control law. This work could be continued by also designing a controller for the lateral/directional axis. Because the lateral/directional axis aerodynamic equations were described by a different format, they do not lead to a straightforward application of this control technique. However, the same method would apply to this axis system. The first step in the design would be to fit a more analytic form to the lateral force and moment equations. The fit does not need to be perfect. There could be some remaining Δ differences just as in the longitudinal case. Once this form is acquired, we could perform the same type of controller calculations. In fact, the lateral/directional system would have two outputs, roll and yaw, along with two inputs, aileron and rudder, thus aleviating the redundant control problem. Then with a full 6 degree-of-freedom controller for the X-29, more complex maneuvers could be performed to test the controller.

Bibliography

[1] J. D. Lang and M. S. Francis, "Unsteady aerodynamics and dynamic aircraft maneuverability," *AGARD CP-386*, May 1985.

[2] L. T. Nguyen, R. D. Whipple, and J. M. Brandon, "Recent experiences of unsteady aerodynamic effects on aircraft flight dynamics at high angle of attack," *AGARD-CP-386*, May 1985.

[3] G. Meyer and L. Cicolani, "A formal structure for advanced automatic flight-control systems." NASA TN D-7940, May 1975.

[4] D. McRuer, I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control.* Princeton University Press, 1973.

[5] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation.* John Wiley & Sons, Inc., 1992.

[6] A. Isidori, *Nonlinear Control Systems: An Introduction.* Springer-Verlag, 2nd ed., 1989.

[7] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems.* Springer-Verlag, 1990.

[8] A. van der Schaft, "Linearization and input-output decoupling for general nonlinear systems," *Systems & Control Letters*, vol. 5, pp. 27–33, October 1984.

[9] R. Su, "On the linear equivalents of nonlinear systems," *Systems & Control Letters*, vol. 2, pp. 48–52, July 1982.

[10] W. Wehrend and G. Meyer, "Flight tests of the total automatic flight control system (TAFCOS) concept on a DHC-6 Twin Otter aircraft." NASA TP-1513, Feb. 1980.

[11] G. Smith and G. Meyer, "Application of the concept of dynamic trim control to automatic landing of carrier aircraft." NASA TP-1512, April 1980.

[12] G. A. Smith and G. Meyer, "Aircraft automatic flight control system with model inversion," *J. of Guidance, Control, and Dynamics*, pp. 269–275, May-June 1987.

[13] O. Kato and I. Sugiura, "An interpretation of airplane general motion and control as inverse problem," *J. of Guidance, Control, and Dynamics*, vol. 9, no. 2, pp. 198–204, 1986.

[14] S. H. Lane and R. F. Stengel, "Flight control design using non-linear inverse dynamics," *Automatica*, vol. 24, pp. 471–483, 1988.

[15] S. N. Singh and W. J. Rugh, "Decoupling in a class of nonlinear systems by state variable feedback," *Journal of Dynamic Systems, Measurement, and Control*, pp. 323–329, Dec. 1972.

[16] E. Freund, "Decoupling and pole assignment in nonlinear systems," *Electronics Letters*, vol. 9, pp. 373–374, Aug. 1973.

[17] S. H. Lane and R. F. Stengel, "Nonlinear inverse dynamics control laws - a sampled data approach," *American Control Conf., Minneapolis, MN.*, pp. 1224–1226, 1987.

[18] S. A. Snell, D. F. Enns, and W. Garrard, "Nonlinear inversion flight control for a supermaneuverable aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 15, pp. 976–984, July-August 1992.

[19] S. A. Snell, D. F. Enns, and W. Garrard, "Nonlinear control of a supermaneuverable aircraft," *AIAA Guidance, Navigation, and Control Conference*, pp. 519–530, August 1989.

[20] P. Kokotovic, R. O'Malley, and P. Sannuti, "Singular perturbations and order reduction in control theory - an overview," *Automatica*, vol. 12, pp. 123–132, 1976.

[21] J. Chow and P. Kokotovic, "Two-time-scale feedback design of a class of nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 23, no. 3, pp. 438–443, 1978.

[22] P. K. A. Menon, M. E. Badgett, R. A. Walker, and E. L. Duke, "Nonlinear flight test trajectory controllers for aircraft," *J. of Guidance, Control, and Dynamics*, vol. 10, pp. 67–72, Jan.-Feb. 1987.

[23] D. J. Bugajski, D. F. Enns, and M. Elgersma, "A dyanmic inversion based control law with application to the high angle-of-attack research vehicle," *AIAA Guidance, Navigation, and Control Conference*, Aug. 1990.

[24] D. Enns, "Robustness of dynamic inversion vs. μ-synthesis: Lateral-directional flight control example," *AIAA Guidance, Navigation, and Control Conference*, pp. 210–222, Aug. 1990.

[25] Honeywell Systems and Research Center, "Dynamic inversion based flight control laws for the HARV." Final Report for contract no. ATD-91-HON-7001, March 1992.

[26] Honeywell Systems and Research Center, "Flight control laws for the HARV - Phase II." Final Report for contract no. TSD-92-HON-3801, June 1993.

[27] D. Enns, D. Bugajski, R. Hendrik, and G. Stein, "Dynamic inversion: an evolving methodology for flight control design," *Int. J. Control*, vol. 59, pp. 71–91, Jan. 1994.

[28] J.-J. E. Slotine and J. K. Hedrick, "Robust input-output feedback linearization," *Int. Journal of Control*, vol. 57, no. 5, pp. 1133–1139, 1993.

[29] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, Inc., 1991.

[30] M. Vidyasagar, *Nonlinear Systems Analysis.* Networks Series, Prentice-Hall, 1978.

[31] C. Huang, J. Reilly, G. Knowles, W. Dayawansa, and W. Levine, "Analysis and simulation of a nonlinear control strategy for high angle of attack maneuvers," *AIAA Guidance, Navigation, and Control Conf.*, August 1991.

[32] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control.* Englewood Cliffs, N.J.: Prentice-Hall, 1971.

[33] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis.* New York: Springer Verlag, 1980.

[34] M. Athans, M. L. Dertouzos, R. N. Spann, and S. J. Mason, *Systems, Networks, & Computation: Multivariable Methods.* McGraw-Hill, Inc., 1974.

[35] R. Clarke, J. J. Burken, J. T. Bosworth, and J. E. Bauer, "X-29 flight control system: lessons learned," *Int. J. Control*, vol. 59, pp. 199–219, January 1994.

[36] K. D. Hammett, W. C. Reigelsperger, and S. S. Banda, "High angle of attack short period flight control design with thrust vectoring," *Proc. of American Control Conf.*, pp. 170–174, 1995.

[37] K. D. Hammett, J. M. Berg, C. A. Schwartz, and S. S. Banda, "An analysis of the destabilizing effect of daisy chained rate-limited actuators," *AIAA Guidance, Navigation, and Control Conference*, no. 95-3207, pp. 312–317, 1995.

[38] W. C. Durham and K. A. Bordignon, "Multiple control effector rate limiting," *AIAA Guidance, Navigation, and Control Conference*, no. 95-3208, pp. 318–327, 1995.

[39] G. H. Golub and C. F. Van Loan, *Matrix Computations.* The Johns Hopkins University Press, 1983.