NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL FOR HYPERSONIC VEHICLES

A Dissertation

by

ELIZABETH ROLLINS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	John Valasek
Committee Members,	Helen L. Reed
	Thomas W. Strganac
	Sivakumar Rathinam
Head of Department,	Rodney D. W. Bowersox

December 2013

Major Subject: Aerospace Engineering

ABSTRACT

Because of the widely varying flight conditions in which hypersonic vehicles operate and certain aspects unique to hypersonic flight, the development of control architectures for these vehicles presents a challenge. Previous work on control design for hypersonic vehicles often has involved linearized or simplified nonlinear dynamical models of the aircraft. This dissertation retains the nonlinear dynamics in the design of the controller for a generic hypersonic vehicle model and develops a nonlinear adaptive dynamic inversion control architecture with a control allocation scheme. A robustness analysis is performed on the initial controller design, which shows that the controller is able to handle time delays, perturbations in stability derivatives, and reduced control surface effectiveness while maintaining tracking performance.

One particular safety concern in hypersonic flight is inlet unstarts, which not only produce a significant decrease in the thrust but also can lead to loss of control and possibly the loss of the vehicle. This dissertation focuses on the prevention of inlet unstarts that are triggered by an altered flow that fails to pass through the throat of the engine because the aircraft has exceeded limits on angle-of-attack and sideslip angle. To prevent undesirable inlet unstart events, the nonlinear adaptive dynamic inversion control architecture is given the ability to enforce state constraints. Because several phenomena can cause inlet unstarts, the control architecture also is tested to determine if the controller is able to maintain reference trajectory tracking and to prevent the loss of the vehicle should an inlet unstart occur. Additionally, a fault-tolerant control capability is added to the control architecture so that the vehicle can handle the failure of one or more control surfaces.

The tracking performance of the nonlinear adaptive dynamic inversion control architecture is analyzed for the cases of enforcement of state constraints, control surface failures, and inlet unstarts. In all three cases, the control architecture is able to track reference trajectories with minimal to no degradation in performance. Limitations were discovered in the case of the controller that enforces state constraints in terms of the trajectories that can be tracked when combined with fault-tolerant control. However, the results indicate that the nonlinear adaptive dynamic inversion controller is able to achieve tracking performance in the presence of the uncertainties and inlet unstart conditions studied in this dissertation.

DEDICATION

To Mom, Dad, Katie, and Frances,
who helped me to weather the storm
and who forever will know their elevators from their ailerons.

“Because the road was steep and long,
And through a dark and lonely land,
God set upon my lips a song
And put a lantern in my hand.
- Joyce Kilmer”
- *A Lantern In Her Hand* [1]

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Valasek, for all of his help and guidance throughout my graduate career. Three years ago, after seeing me review an exam with a class for which I was a teaching assistant, he told me that I should consider becoming a teacher. I had thought about teaching before, but it was his words and the opportunity to work with him as his teaching assistant for two years that brought me to the point where I am now. I have gained a solid foundation in the education of undergraduates and in research with Dr. Valasek's help, for which I am grateful.

Second, I would like to thank all of my committee members who worked with my fast-paced schedule during the past year and who have provided insightful input on my dissertation work. Additionally, I would like to thank Dr. Michael Bolender and Dr. Jonathan Muse of the Air Force Research Laboratory at Wright-Patterson Air Force Base. They assisted me during this entire project, from the initial concept to the technical details. I am appreciative of their advice and guidance and believe that I have become a better researcher as a result of their influence. I also would like to thank all of the graduate students who helped me along the way. A special thanks is given to Anshu, who kindly listened to my ideas on state constraints and helped to point me along the right path when I got lost in control theory.

One group whose contribution is not to be overlooked is everyone at Barrington Living History Farm, whom I would like to thank as well. They provided me with a haven at the farm on Saturdays to recharge for the coming week in graduate school. I am thankful to them for all of their encouragement and for the opportunity to live out one of my childhood dreams of being Laura Ingalls Wilder. Many people do not

get to live out their childhood dreams, so I feel especially lucky in that respect.

Finally, I want to thank my family and my friends. Four years ago, I was not sure that I wanted to keep going in graduate school. You all remained with me every step of the way, helping me over the rough spots, encouraging me to keep going, and always believing that I would finish this dissertation. You always knew that I had the strength to finish this degree, and thanks to your belief in me and your prayers, I did. I love you all.

> “ ‘So Potter . . . give us a shout if you need us.’ . . .
>
> Harry nodded. He somehow could not find words to tell them what it meant to him, to see them all ranged there, on his side. Instead he smiled, . . . turned around, and led the way out of the station toward the sunlit street . . . .”
>
> - *Harry Potter and the Order of the Phoenix* [2]

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

On May 1, 2013, the X-51A Waverider, an autonomous hypersonic vehicle demonstrator, reached Mach 5.1 on the fourth and final flight of the latest U. S. Air Force hypersonic vehicle test program [3]. This event is the newest research highlight in a field that has had a long and varied history [4], [5]. In the last decade, a resurgence of interest in the area of hypersonic flight has been seen in the aerospace community, and with this resurgence comes the need for control architectures that are able to meet the unique challenges of hypersonic flight, including the risk of inlet unstarts, which was an issue during the second flight of the X-51A [3].

1.1 Motivation

The design of control architectures for hypersonic vehicles is a current area of research in the field of controls. Flight control of hypersonic vehicles is challenging because of the wide range of operating conditions encountered and certain aspects unique to high speed flight. A particular safety concern in hypersonic flight is the risk of an inlet unstart, which not only produces a significant decrease in thrust but also can lead to loss of the vehicle. There are three main reasons that cause a hypersonic airbreathing engine to unstart: (1) a flow to the inlet that is slower than the required Mach number for the engine to operate, (2) an altered flow that no longer passes through the throat of the engine because of reasons such as exceeding the limits on angle-of-attack (α) and sideslip angle (β), and (3) an increase in the back pressure in the engine that causes the shock wave to move ahead of the throat [6]. This dissertation will focus on the prevention of the second cause and develop a control architecture that will limit angle-of-attack and sideslip angle for hypersonic vehicles to prevent an inlet unstart.

1.2 Literature Search

Many of the previous control designs for hypersonic vehicles have involved the use of linearized models of the aircraft instead of the full nonlinear equations of motion [7], [8], [9], [10]. Annaswamy, et.al. created adaptive controllers for hypersonic vehicles; however, the controllers are designed based on linearized models of the aircraft dynamics and require gain-scheduling for their implementation [7],[8]. Groves, et.al. implemented control designs based on linear models of a hypersonic vehicle for set-point and regulator tracking [9]. Bolender, et.al. designed adaptive control laws for an experimental hypersonic vehicle based on a linearized model of the longitudinal dynamics of the vehicle [10].

In terms of work with nonlinear models for control design in hypersonic vehicles, Johnson, et.al. applied a neural network-based adaptive control architecture to a model of the X-33 vehicle for the generation of ascent and abort trajectories as well as the control of the aircraft [11]. Fiorentini, et.al. [12] and Parker, et.al. [13] both used simplified nonlinear models of a hypersonic vehicle in their control design that exhibited good tracking performance but a slow response. While Parker, et.al. designed an approximate feedback linearization controller, the controller in that paper is not adaptive; however, a case study of their approximate feedback linearization controller showed that the controller was robust to mild plant parameter variations in the moment of inertia I_{yy}, the vehicle length, and the mass of the vehicle [14]. Brocanelli, et.al. created a robust adaptive controller that determined trajectories that would allow the vehicle to remain in hypersonic flight as long as possible following an inlet unstart; however, the nonlinear model used for the hypersonic vehicle only included the longitudinal equations of motion [15].

The need for designing controllers that can enforce state constraints for aircraft has been a recent area of interest in the field of controls as well. Vaddi and Sengupta constructed a model predictive controller for trajectory tracking in a hypersonic vehicle that handled nonlinear state and control constraints [16]. Lavretsky and Gadient examined the addition of state constraints to a dynamic inversion-based control architecture for a second-order system [17], and Gadient, et.al. extended this work to adjust the reference model appropriately when the system neared any state constraints, demonstrating their results in simulation and flight test for an X-48B aircraft [18]. Bürger and Guay designed a switching controller that actively enforces output constraints in one mode for nonlinear systems with parametric uncertainties that are affine in the input [19]. Contrastingly, Muse developed a method for enforcing state constraints in adaptive control laws without switching between modes [20].

1.3 Open Research Issues

- What type of control could be used for systems in which the dynamic equations are nonlinear and in which there is significant parametric uncertainty in the model of the aerodynamics when the system is a hypersonic vehicle?

  Ideally, dynamic inversion would be one choice for a control structure since this method would allow undesired dynamics to be cancelled and replaced with desired dynamics. However, there is significant parametric uncertainty in the model of the aerodynamics for a hypersonic vehicle, which is generated using computational fluid dynamics. Therefore, adaptive dynamic inversion should be used to account for the uncertatinty in the aerodynamics. Also, to retain the most fidelity and accuracy in the inversion, the nonlinear dynamics of a hypersonic vehicle should be used in the design of the control architecture.

- How can an inlet unstart as a result of exceeding limits on states be prevented?

  An inlet unstart can lead to a reduction in thrust and the possible loss of the vehicle. The focus of this dissertation is on the prevention of inlet unstarts that are caused by an alteration of the flow by changes in angle-of-attack and sideslip angle such that the flow no longer reaches the throat of the engine.

- Knowing that it is desired to fly in a manner such as to prevent inlet unstart, how can a vehicle be protected should control surfaces fail?

  In order to protect the vehicle if control surfaces fail, robustness must be added to the control architecture. The addition of robustness should allow the vehicle to continue flying after the loss of control surfaces in such a manner as to avoid an inlet unstart.

- If an inlet unstart should occur, how will the hypersonic vehicle respond given its control architecture?

  For any given flight condition following an inlet unstart, the vehicle must be able to determine the appropriate course of action to ensure its preservation. This dissertation examines the ability of the developed control architecture for a hypersonic vehicle to maintain reasonable tracking performance following an inlet unstart.

1.4 Scope and Contribution

This dissertation presents a design of a nonlinear adaptive dynamic inversion control architecture. A rigid-body model of a generic hypersonic vehicle will be used as the plant for the design and analysis of this controller. Because the dynamic equations for hypersonic vehicles are inherently nonlinear and the aerodynamic and control derivatives for the aircraft have significant uncertainty associated with them, an adaptive dynamic inversion controller is selected as the preferred control architecture. One component of this dissertation will consist of using the complete coupled nonlinear dynamic equations for a hypersonic vehicle in the design of the nonlinear adaptive dynamic inversion control architecture instead of a linearized form of the equations or a subset of the nonlinear dynamics, which is a unique contribution of this research. The second component of this dissertation will involve the addition of state constraints to the control architecture in order to prevent inlet unstarts from occurring. The third component of this dissertation is the inclusion of fault tolerant control to the control architecture. While fault tolerant control is not a novel concept, for the purposes of this dissertation, this component will add a robustness to the control architecture that will be beneficial for the vehicle. The final component of this dissertation will investigate the subsequent performance of the nonlinear adaptive dynamic inversion control architecture after an inlet unstart. The combination of designing an adaptive control architecture that accounts for state constraints, adding fault tolerance into the control architecture, and studying the performance of the control architecture following inlet unstarts constitutes my original contribution to aerospace engineering through this dissertation work.

1.5 Outline of the Dissertation

The following chapters will develop the theory behind the technical approach outlined in the previous section and present simulation results to illustrate the theory. Chapter 2 is the foundational chapter that outlines the nonlinear adaptive dynamic inversion control architecture that is applied throughout this dissertation. The enforcement of state constraints within this control architecture is discussed in Chapter 3. Chapter 4 focuses on the addition of fault-tolerant control to the baseline nonlinear adaptive dynamic inversion control architecture. In Chapter 5, the modeling of an inlet unstart in a simulation and the generation of reference trajectories for flight path angles is presented in conjunction with an examination of the performance of the control architecture after an inlet unstart. Conclusions and recommendations for future research directions are given in Chapter 6.

2. NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL ARCHITECTURE

In order to deal with nonlinear dynamic equations of motion and parametric uncertainties in the aerodynamics model for a hypersonic vehicle, a nonlinear adaptive dynamic inversion control architecture was determined to be a candidate architecture, as stated in Chapter 1. The current chapter describes the derivation of general adaptive control laws and their application using the dynamic equations for a generic hypersonic vehicle, followed by simulation results and a robustness analysis of the control architecture.

2.1 The Generic Hypersonic Vehicle

The Generic Hypersonic Vehicle (GHV) [21], as shown in Figure 2.1, is an academic hypersonic aircraft vehicle model created at the Air Force Research Laboratory as a platform for controls research. The GHV plant simulation is implemented using a Simulink model that contains the nonlinear, 6-DOF equations of motion for an inelastic hypersonic vehicle without rotors. The aerodynamic and thrust forces and moments acting on the vehicle are modeled using look-up tables; the tables for the aerodynamic forces and moments were generated based on computational fluid dynamics data using shock-expansion methods with a viscous correction. Reference [21] contains a detailed description of the equations that are contained in the GHV simulation.

Using two elevons and two ruddervators, it is desired to control angle-of-attack (α), sideslip angle (β), and aerodynamic bank angle (μ). Figure 2.2 shows a diagram of the GHV system with the adaptive dynamic inversion controllers. To simplify the process of designing a nonlinear adaptive dynamic inversion control architecture, it is

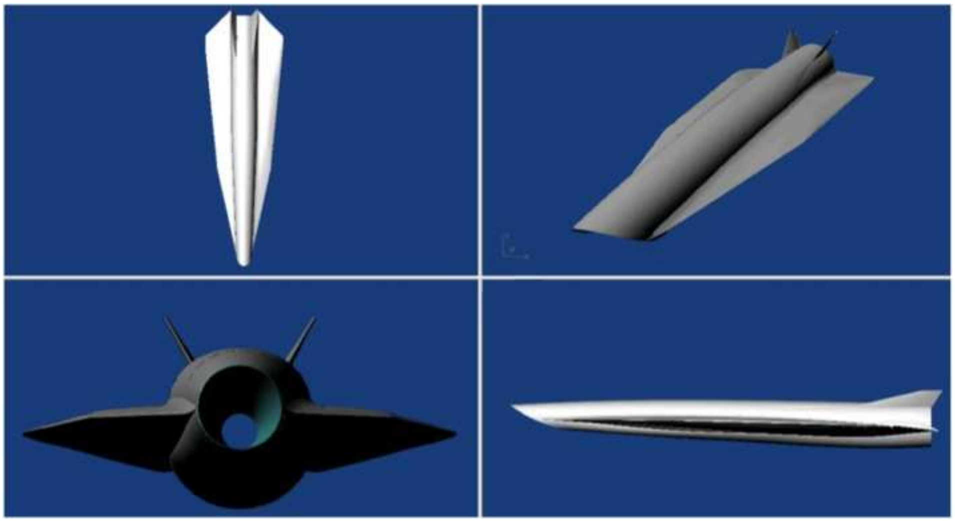


Figure 2.1: The Generic Hypersonic Vehicle (GHV).

assumed that the aircraft states can be divided into two timescale categories, which are the fast states, which consist of the angular rates p, q, and r as noted in [22], and the slow states, which consist of the angles α, β, and μ. An adaptive dynamic inversion controller first must translate α, β, and μ commands into commands for the body axis rates p, q, and r, which then are passed into another adaptive dynamic inversion controller that determines the corresponding control surface deflections to achieve the desired p, q, and r commands.

The following three sections will describe the equations found in the inversion blocks in Figure 2.2. For the equations derived in Sections 2.3 and 2.4, the flat, nonrotating earth assumption [23, p. 43] is made. It is acceptable to make this assumption in this case because while the vehicle is flying fast enough for the round rotating Earth effects to be significant, the time scale of the controlled dynamics are sufficiently fast to neglect them.

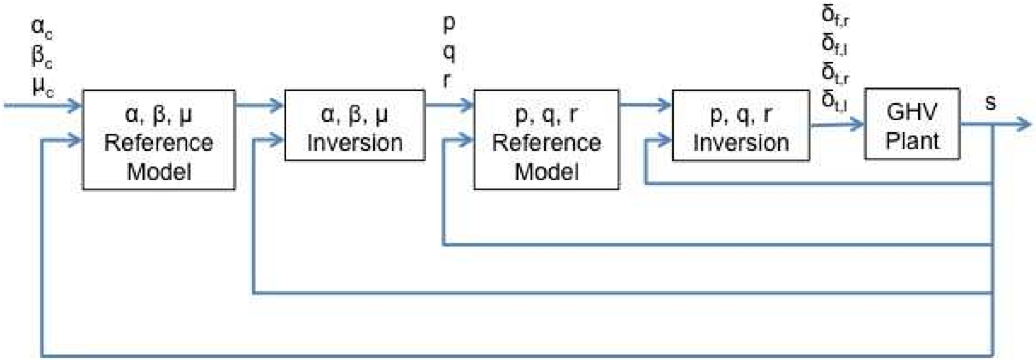


Figure 2.2: Diagram of the adaptive dynamic inversion controllers for the GHV, where s represents all of the states of the system.

2.2 General Adaptive Dynamic Inversion Equations

This section contains the derivation of the control laws for two cases of the adaptive nonlinear dynamic inversion controller. The first case involves dynamic equations containing the same number of controls and controlled variables, and the second case deals with dynamic equations with a greater number of controls than controlled variables. It should be noted in both cases, the general nonlinear equation of the system is assumed to be affine in the control, which is reasonable for small deflection angles.

2.2.1 Case with Equal Number of Controls and Controlled Variables

Consider a general nonlinear equation of a system in the form

$$\dot{x} = f(x) + g(x)u \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^n$ is the control, and $f(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ are locally Lipschitz continuous. It is assumed that $g(x)$ is nonsingular for all $x \in \mathbb{R}^n$. Suppose that the desired reference dynamics for the system are given by

$$\dot{x}_m = Ax_m + Br \tag{2.2}$$

where $x_m \in \mathbb{R}^n$ is the model state, $r \in \mathbb{R}^n$ is a bounded reference signal, $A \in \mathbb{R}^{n \times n}$ is Hurwitz, and $B \in \mathbb{R}^{n \times n}$. The equation for the error between the reference model and the actual system is

$$e = x_m - x. \tag{2.3}$$

Taking the time derivative of equation (2.3) results in

$$\dot{e} = \dot{x}_m - \dot{x} = \dot{x}_m - f(x) - g(x)u. \tag{2.4}$$

If the control u is chosen to be

$$u = [g(x)]^{-1}[\dot{x}_m - \hat{f}(x) + Ke - \nu] \tag{2.5}$$

where $\hat{f}(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a model of the plant dynamics, $K \in \mathbb{R}^{n \times n}$ such that $K = K^T > 0$ are the gains on the tracking errors, and $\nu \in \mathbb{R}^n$ is a pseudo-control signal, then substituting equation (2.5) into equation (2.4) produces the error dynamics

$$\dot{e} = -f(x) + \hat{f}(x) - Ke + \nu. \tag{2.6}$$

Defining the error between the model and the actual system as $\Delta = \hat{f}(x) - f(x)$, equation (2.6) becomes

$$\dot{e} = -Ke + \Delta + \nu. \tag{2.7}$$

In this dissertation, it is assumed that Δ can be represented in the form $\Delta = W^T\beta(x;d)$, where $W \in \mathbb{R}^{p\times n}$ is a set of unknown weights, and $\beta \in \mathbb{R}^{p\times 1}$ is a set of known basis functions composed of the states x and a vector d of bounded continuous exogenous inputs. Using this representation for Δ, ν is chosen to be $\nu = -\widehat{W}^T\beta(x;d)$, where $\widehat{W} \in \mathbb{R}^{p\times n}$ is an estimate of the weights. With these definitions, equation (2.7) can be written as

$$\dot{e} = -Ke - \widetilde{W}^T\beta(x;d) \tag{2.8}$$

where $\widetilde{W} = \widehat{W} - W$ is the weight estimation error.

The stability of the closed loop system under these assumptions can be established using a candidate Lyapunov function of the form

$$V = e^Te + tr(\widetilde{W}^T{\Gamma_W}^{-1}\widetilde{W}) \tag{2.9}$$

where $\Gamma_W \in \mathbb{R}^{p\times p}$ with $\Gamma_W = {\Gamma_W}^T > 0$. In order to determine the adaptation law for the parameters in W and to prove that the error between the states of the actual system and the reference model will converge, first, the derivative of equation (2.9) along the system trajectories is taken, which gives the result

$$\dot{V} = 2e^T\dot{e} + 2tr(\widetilde{W}^T{\Gamma_W}^{-1}\dot{\widehat{W}}^T). \tag{2.10}$$

Substituting equation (2.8) into equation (2.10) produces

$$\dot{V} = -2e^TKe - 2e^T\widetilde{W}^T\beta(x;d) + 2tr(\widetilde{W}^T{\Gamma_W}^{-1}\dot{\widehat{W}}^T). \tag{2.11}$$

Applying the trace identity that $a^T b = tr(ba^T)$, equation (2.11) is determined to be

$$\dot{V} = -2e^T Ke + 2tr(\widetilde{W}^T ({\Gamma_W}^{-1} \dot{\widehat{W}}^T - \beta(x;d)e^T)). \tag{2.12}$$

Then, by choosing $\dot{\widehat{W}}$ as

$$\dot{\widehat{W}} = \Gamma_W \operatorname{Proj}(\widehat{W}, \beta(x;d)e^T) \tag{2.13}$$

where Proj represents the projection operator, which is used to maintain specified bounds on the weights [24], $\dot{V}$ can be upper bounded as

$$\dot{V} \leq -2e^T Ke \leq 0 \tag{2.14}$$

which implies that e is bounded. Because r is bounded by definition above, x_m is bounded. Since e and x_m are bounded, x is bounded. Consequently, $\beta(x;d)$ is bounded as well. In order to use Barbalat's lemma [25] to complete the proof, the second derivative of equation (2.9) along the system trajectories is taken, which gives the result

$$\ddot{V} = -4e^T K\dot{e}. \tag{2.15}$$

Substituting equation (2.8) into equation (2.15) produces

$$\ddot{V} = -4e^T K(-Ke - \widetilde{W}^T \beta(x;d)). \tag{2.16}$$

Because e, $\widetilde{W}$, and $\beta(x;d)$ are bounded as proved above, $\ddot{V}$ is bounded, and therefore $\dot{V}$ is uniformly continuous.

Because V is lower bounded, $\dot{V}$ is negative semi-definite, and $\dot{V}$ is uniformly continuous, by Barbalat's lemma $\dot{V} \to 0$ as $t \to \infty$, and thus $e \to 0$ as $t \to \infty$ as desired.

2.2.2 Case with a Greater Number of Controls than Controlled Variables

Specifically for the GHV, the form of the general adaptive dynamic inversion controller in the previous subsection applies to the α, β, μ inversion component, in which the number of inputs to the system (α, β, μ) is equal to the number of outputs (p, q, r). However, in the p, q, r inversion component, the number of inputs to the system (p, q, r) is not the same as the number of outputs ($\delta_{f,r}$, $\delta_{f,l}$, $\delta_{t,r}$, $\delta_{t,l}$). The fact that the number of outputs is greater than the number of inputs requires a control allocation scheme to be integrated into the inversion control law. To frame the problem in general terms, consider the given nonlinear equation of a system in the form

$$\dot{x} = f(x) + g(x)\Lambda u \tag{2.17}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control, $f(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times m}$ are locally Lipschitz continuous, and $\Lambda \in \mathbb{R}^{m\times m}$ is a constant unknown positive definite matrix. It is assumed that $g(x)$ is full rank for all $x \in \mathbb{R}^n$. Suppose that the desired dynamics of the closed loop system are given by

$$\dot{x}_m = Ax_m + Br \tag{2.18}$$

where $x_m \in \mathbb{R}^n$ is the model state, $r \in \mathbb{R}^m$ is the bounded reference signal, $A \in \mathbb{R}^{n\times n}$ is Hurwitz, and $B \in \mathbb{R}^{n\times m}$.

The derivation of the control law and the adaptive laws, including one for the unknown control effectiveness matrix Λ, proceeds similarly to the derivation in Subsection 2.2.1. The equation for the error between the reference model and the actual system is

$$e = x_m - x. \tag{2.19}$$

Taking the time derivative of equation (2.19) results in

$$\dot{e} = \dot{x}_m - \dot{x} = \dot{x}_m - f(x) - g(x)\Lambda u. \tag{2.20}$$

The desired final form for $\dot{e}$ is

$$\dot{e} = -Ke - \widetilde{W}^T \beta(x; d) + g(x)\widetilde{\Lambda} u \tag{2.21}$$

which is the same as the final form for $\dot{e}$ in Subsection 2.2.1, except for the final term $g(x)\widetilde{\Lambda}u$. With the appropriate choice of adaptive law for $\widehat{\Lambda}$, the choice of the above final form for $\dot{e}$ will allow the stability of the system to be proven. In order to derive this desired form of $\dot{e}$, first the term $g(x)\widehat{\Lambda}u$ is added and subtracted from equation (2.20), where $\widehat{\Lambda} \in \mathbb{R}^{m\times m}$ is an estimate of the control effectiveness matrix, and the error equation becomes

$$\dot{e} = \dot{x}_m - f(x) - g(x)\Lambda u + g(x)\widehat{\Lambda} u - g(x)\widehat{\Lambda} u. \tag{2.22}$$

Let $\widetilde{\Lambda} = \widehat{\Lambda} - \Lambda$, which is the estimation error of the control effectiveness matrix. Then, equation (2.22) simplifies to

$$\dot{e} = \dot{x}_m - f(x) - g(x)\widehat{\Lambda} u + g(x)\widetilde{\Lambda} u. \tag{2.23}$$

Because of the fact that the number of controls is greater than the number of controlled variables in this case, there sometimes are infinite choices for u at any instant in time. In order to determine a specific control law for the system, a constrained optimization problem is solved in which the cost function $J = u^T Q u$, where $Q \in \mathbb{R}^{m \times m}$ with $Q = Q^T > 0$, will be minimized, subject to the constraint $g(x)\widehat{\Lambda}u = \ell$, which must be satisfied at all times. The cost function is chosen to be $J = u^T Q u$ so that the control effort will be minimized, which consequently can be used to reduce the amount of trim drag during flight. it is assumed by this formulation of the problem that the control surfaces do not have position limits, and as a result, sufficient control power will always exist. By choosing the term ℓ in the constraint equation to be

$$\ell = \dot{x}_m - \hat{f}(x) + Ke - \nu \tag{2.24}$$

where $\hat{f}(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is an estimate of the plant dynamics, $K \in \mathbb{R}^{n \times n}$ with $K = K^T > 0$ contains the gains on the errors, and $\nu \in \mathbb{R}^n$ is a pseudo-control signal, the constraint $g(x)\widehat{\Lambda}u = \ell$ will ensure that when the derived control law for this second case is substituted into the equation for $\dot{e}$, and the equation for the error dynamics is simplified, the first two terms of equation (2.21) will appear in the resulting equation for $\dot{e}$ as desired. For simplicity in the control law derivation, equation (2.24) will not be substituted into the constraint equation at the present time.

To derive the control law, first the constraint must be included in the cost function to form the augmented cost function

$$\bar{J} = u^T Q u + \lambda^T (g(x)\widehat{\Lambda}u - \ell) \tag{2.25}$$

where $\lambda \in \mathbb{R}^n$ is a Lagrange multiplier. The necessary conditions for minimizing $\bar{J}$ are given by

$$\frac{\partial \bar{J}}{\partial \lambda} = g(x)\widehat{\Lambda}u - \ell = 0 \tag{2.26}$$

$$\frac{\partial \bar{J}}{\partial u} = 2Qu + \widehat{\Lambda}^T g^T(x)\lambda = 0. \tag{2.27}$$

Rearranging terms in equation (2.27) results in

$$u = -\frac{1}{2}Q^{-1}\widehat{\Lambda}^T g^T(x)\lambda. \tag{2.28}$$

Substituting equation (2.28) into equation (2.26) and solving for λ produces the equation

$$\lambda = -2(g(x)\widehat{\Lambda}Q^{-1}\widehat{\Lambda}^T g^T(x))^{-1}\ell. \tag{2.29}$$

Finally, substituting equation (2.29) back into equation (2.28) results in the control law

$$u = Q^{-1}\widehat{\Lambda}^T g^T(x)(g(x)\widehat{\Lambda}Q^{-1}\widehat{\Lambda}^T g^T(x))^{-1}\ell. \tag{2.30}$$

In order for the control law given in equation (2.30) to be continuous, Q and $g(x)\widehat{\Lambda}Q^{-1}\widehat{\Lambda}^T g^T(x)$ must be invertible. The projection bounds that will be applied in the adaptive law for Λ must ensure that $\widehat{\Lambda}$ remains invertible. It should be noted that for the case where the number of controls equals the number of controlled variables, the control solution is unique, and the control law in equation (2.30) simplifies to

$$u = [g(x)]^{-1}[\dot{x}_m - \hat{f}(x) + Ke - \nu] \tag{2.31}$$

which is the control law that was chosen in Subsection 2.2.1.

Continuing with the derivation of $\dot{e}$, let $\Delta = \hat{f}(x) - f(x)$. Substituting equation (2.30), equation (2.24), and into equation (2.23) produces the equation

$$\dot{e} = -Ke + \Delta + \nu + g(x)\widetilde{\Lambda}u. \tag{2.32}$$

Again, assume that Δ can be represented in the form $\Delta = W^T\beta(x;d)$, where $W \in \mathbb{R}^{p\times n}$ is a set of unknown weights, and $\beta \in \mathbb{R}^{p\times 1}$ is a set of known basis functions composed of the states x and a vector d of bounded continuous exogenous inputs. Also, the representation for ν is chosen to be $\nu = -\widehat{W}^T\beta(x;d)$, where $\widehat{W} \in \mathbb{R}^{p\times n}$ is an estimate of the weights. Then, equation (2.32) can be written as

$$\dot{e} = -Ke - \widetilde{W}^T\beta(x;d) + g(x)\widetilde{\Lambda}u \tag{2.33}$$

where $\widetilde{W} = \widehat{W} - W$, which is the weight estimation error.

As in Subsection 2.2.1, a Lyapunov analysis needs to be performed in order to determine the adaptive laws for $\widehat{\Lambda}$ and $\widehat{W}$ and to prove that the error between the states of the actual system and the reference model will converge. Given the candidate Lyapunov function

$$V = e^Te + tr(\widetilde{W}^T{\Gamma_W}^{-1}\widetilde{W}) + tr(\widetilde{\Lambda}{\Gamma_\Lambda}^{-1}\widetilde{\Lambda}) \tag{2.34}$$

where $\Gamma_W \in \mathbb{R}^{p\times p}$ with $\Gamma_W = {\Gamma_W}^T > 0$, and $\Gamma_\Lambda \in \mathbb{R}^{m\times m}$ with $\Gamma_\Lambda = {\Gamma_\Lambda}^T > 0$, the derivative of equation (2.34) along the system trajectories is taken, which results in the equation

$$\dot{V} = 2e^T\dot{e} + 2tr(\widetilde{W}^T{\Gamma_W}^{-1}\dot{\widehat{W}}^T) + 2tr(\widetilde{\Lambda}{\Gamma_\Lambda}^{-1}\dot{\widehat{\Lambda}}^T). \tag{2.35}$$

Substituting equation (2.33) into equation (2.35) produces

$$\begin{aligned}\dot{V} = -2e^TKe - 2e^T\widetilde{W}^T\beta(x;d) + 2e^Tg(x)\widetilde{\Lambda}\\ + 2tr(\widetilde{W}^T{\Gamma_W}^{-1}\dot{\widehat{W}}^T) + 2tr(\widetilde{\Lambda}{\Gamma_\Lambda}^{-1}\dot{\widehat{\Lambda}}^T)\end{aligned} \tag{2.36}$$

and by applying the trace identity that $a^Tb = tr(ba^T)$ to equation (2.36), the equation for $\dot{V}$ becomes

$$\dot{V} = -2e^T K e + 2tr(\widetilde{W}^T(\Gamma_W{}^{-1}\dot{\widehat{W}}^T - \beta(x;d)e^T)) + 2tr(\widetilde{\Lambda}(\Gamma_\Lambda{}^{-1}\dot{\widehat{\Lambda}}^T + ue^T g(x))). \quad (2.37)$$

Let the equation for $\dot{\widehat{W}}$ in this case be the same as equation (2.13), and let $\dot{\widehat{\Lambda}}$ be

$$\dot{\widehat{\Lambda}} = \Gamma_\Lambda \, \mathrm{Proj}(\widehat{\Lambda}, -ue^T g(x)). \quad (2.38)$$

In this case, the final equation for $\dot{V}$ is is upper bounded by

$$\dot{V} \leq -2e^T K e \leq 0 \quad (2.39)$$

which implies that e is bounded. Since r is bounded by definition above and e is bounded, x_m is bounded, and thus x is bounded. Consequently, $g(x)$ and $\beta(x;d)$ are bounded as well. In order to use Barbalat's lemma [25] to complete the proof, the second derivative of equation (2.34) along the system trajectories is taken, which gives the result

$$\ddot{V} = -4e^T K \dot{e}. \quad (2.40)$$

Substituting equation (2.33) into equation (2.40) produces

$$\ddot{V} = -4e^T K(-Ke - \widetilde{W}^T \beta(x;d) + g(x)\widetilde{\Lambda}u). \quad (2.41)$$

It should be noted that u is bounded because all of the the signals found in u, which is given by equations (2.30) and (2.24) are bounded. Thus, because e, $\widetilde{W}$, $\beta(x;d)$, $g(x)$, $\widetilde{\Lambda}$, and u are bounded as proved above, $\ddot{V}$ is bounded, and therefore $\dot{V}$ is uniformly continuous.

Finally, Barbalat's lemma can be applied. Because V is lower bounded, $\dot{V}$ is negative semi-definite, and $\dot{V}$ is uniformly continuous, by Barbalat's lemma $\dot{V} \to 0$ as $t \to \infty$, and thus $e \to 0$ as $t \to \infty$ as desired.

2.3 P, Q, R Inversion Controller

The first designed controller was the inversion controller for the angular body rates of the GHV since these variables are linked directly to the control surface deflections, which control the vehicle. The reference inputs to the controller are the commanded angular body rates p_c, q_c and r_c, and the output states of the controller are the control surface deflections $\delta_{f,r}$, $\delta_{f,l}$, $\delta_{t,r}$, and $\delta_{t,l}$. Therefore, in this case, Equation (2.17) represents the current system. In order for the adaptive dynamic inversion controller to be designed for the angular body rates, $f(x)$ and $g(x)$ must be determined from the general nonlinear equations for $\dot{p}$, $\dot{q}$, and $\dot{r}$, which in vector-matrix form, are

$$[J]\frac{d\omega_{B,I}}{dt}\bigg|_B + \omega_{B,I} \times J\omega_{B,I} = M_{aero} + M_T \tag{2.42}$$

where

$$[J] = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix} \tag{2.43}$$

and

$$\omega_{B,I} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{2.44}$$

Substituting these equations into Equation (2.42) and simplifying produces the result

$$[J]\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix}+\begin{bmatrix}-J_{xz}pq+(J_z-J_y)qr\\ (J_x-J_z)pr+J_{xz}(p^2-r^2)\\ J_{xz}qr+(J_y-J_x)pq\end{bmatrix}=M_{aero}+M_T. \tag{2.45}$$

Therefore, the nonlinear equations for the angular body accelerations can be written as

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix}=[J]^{-1}\left(-\begin{bmatrix}-J_{xz}pq+(J_z-J_y)qr\\ (J_x-J_z)pr+J_{xz}(p^2-r^2)\\ J_{xz}qr+(J_y-J_x)pq\end{bmatrix}+M_{aero}+M_T\right). \tag{2.46}$$

After having determined the nonlinear equations for the angular body accelerations, the next step is to write those equations in the form of Equation (2.17). In order to accomplish this task, the terms related to the control surfaces, which will form $g(x)$, must be extracted from Equation (2.46). The control surfaces terms are included in the aerodynamic moment terms M_A, which can be written as

$$M_{aero}=\begin{bmatrix}L_A\\ M_A\\ N_A\end{bmatrix}=\begin{bmatrix}\bar{q}SbC_\ell\\ \bar{q}S\bar{c}C_m\\ \bar{q}SbC_n\end{bmatrix} \tag{2.47}$$

where

$$\begin{aligned}C_\ell&=C_{\ell,baseline}+\Delta C_{\ell,surfaces}+\frac{b}{2V_T}\left(C_{\ell_p}p\right)\\ C_m&=C_{m,baseline}+\Delta C_{m,surfaces}+\frac{\bar{c}}{2V_T}\left(C_{m_q}q+C_{m_{\dot{\alpha}}}\dot{\alpha}\right)\\ C_n&=C_{n,baseline}+\Delta C_{n,surfaces}+\frac{b}{2V_T}\left(C_{n_r}r\right)\end{aligned} \tag{2.48}$$

and

$$\begin{aligned}\Delta C_{i,surfaces} &= \Delta C_{i,\delta_{f,r}}(M, \alpha, \beta, \delta_{f,r}) + \Delta C_{i,\delta_{f,l}}(M, \alpha, \beta, \delta_{f,l}) \\ &\quad + \Delta C_{i,\delta_{t,r}}(M, \alpha, \beta, \delta_{t,r}) + \Delta C_{i,\delta_{t,l}}(M, \alpha, \beta, \delta_{t,l})\end{aligned} \tag{2.49}$$

for $i = \ell,\ m,\ n$.

As seen in Equation (2.48), the moment coefficients are comprised of three parts. The baseline term is the moment coefficient for the base airframe, and the second and third terms adjust for the effects on the moment coefficients due to the control surfaces and damping, respectively. In Equation (2.48), the first and third terms do not depend on the control surfaces; therefore, those two terms belong to the $f(x)$ term in Equation (2.17). In order to determine $g(x)$, the second term in each equation in Equation (2.48) must be examined to determine what portion of the term is control-dependent and thus belongs in $g(x)$. For this particular control design for the GHV, it is assumed that a linear approximation with respect to the control surface deflection δ can be made for each of the terms in Equation (2.49). The linear approximation can be expressed as

$$\Delta C_{i,\delta_s}(M, \alpha, \beta, \delta_s) = C_{i,\delta_s}(M, \alpha, \beta, [\delta_s = 0]) + \left.\frac{\partial C_{i,\delta_s}}{\partial \delta_s}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_s \tag{2.50}$$

for $i = N,\ Y,\ A,\ \ell,\ m,\ n$ and $\delta_s = \delta_{f,r},\ \delta_{f,l},\ \delta_{t,r},\ \delta_{t,l}$. In this paper, it is assumed that all interactions between each control surface are negligible, which at high Mach numbers is approximately true. Deflections of the right and left control surfaces will generate summative forces and moments in the XZ-plane of symmetry of the aircraft, whereas in the other planes, the deflections will generate canceling forces and moments. In equation form, for both the flaps and the tail control surfaces, the relationships between right and left elevon deflections are expressed as

$$\begin{aligned} C_{N,\delta_{f,r}} &= C_{N,\delta_{f,l}} & -C_{Y,\delta_{f,r}} &= C_{Y,\delta_{f,l}} \\ C_{A,\delta_{f,r}} &= C_{A,\delta_{f,l}} & -C_{\ell,\delta_{f,r}} &= C_{\ell,\delta_{f,l}} \\ C_{m,\delta_{f,r}} &= C_{m,\delta_{f,l}} & -C_{n,\delta_{f,r}} &= C_{n,\delta_{f,l}} \end{aligned} \tag{2.51}$$

and the relationships between right and left rudder deflections are expressed similarly as

$$\begin{aligned} C_{N,\delta_{t,r}} &= C_{N,\delta_{t,l}} & -C_{Y,\delta_{t,r}} &= C_{Y,\delta_{t,l}} \\ C_{A,\delta_{t,r}} &= C_{A,\delta_{t,l}} & -C_{\ell,\delta_{t,r}} &= C_{\ell,\delta_{t,l}} \\ C_{m,\delta_{t,r}} &= C_{m,\delta_{t,l}} & -C_{n,\delta_{t,r}} &= C_{n,\delta_{t,l}}. \end{aligned} \tag{2.52}$$

Consequently, in Equation (2.50), the term where $\delta_s = 0$ can be written for the combined effect of both the right and left control surfaces collectively as

$$C_{i,\delta_f}(M, \alpha, \beta, [\delta_{f,r}, \delta_{f,l} = 0]) = \begin{cases} 2C_{i,\delta_{f,r}}(M, \alpha, \beta, [\delta_{f,r} = 0]) & \text{for } i = N, A, m \\ 0 & \text{for } i = Y, \ell, n \end{cases} \tag{2.53}$$

$$C_{i,\delta_t}(M, \alpha, \beta, [\delta_{t,r}, \delta_{t,l} = 0]) = \begin{cases} 2C_{i,\delta_{t,r}}(M, \alpha, \beta, [\delta_{t,r} = 0]) & \text{for } i = N, A, m \\ 0 & \text{for } i = Y, \ell, n. \end{cases} \tag{2.54}$$

Given Equations (2.50), (2.53), and (2.54), Equation (2.49) can be rewritten for $i = \ell$, m, n as

$$\begin{aligned} \Delta C_\ell = &\left.\frac{\partial C_{\ell,\delta_{f,r}}}{\partial \delta_{f,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,r} + \left.\frac{\partial C_{\ell,\delta_{f,l}}}{\partial \delta_{f,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,l} \\ &+ \left.\frac{\partial C_{\ell,\delta_{t,r}}}{\partial \delta_{t,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,r} + \left.\frac{\partial C_{\ell,\delta_{t,l}}}{\partial \delta_{t,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,l} \end{aligned} \tag{2.55}$$

$$
\begin{aligned}
\Delta C_m = {} & 2C_{m,\delta_{f,r}}(M, \alpha, \beta, [\delta_{f,r} = 0]) + 2C_{m,\delta_{t,r}}(M, \alpha, \beta, [\delta_{t,r} = 0]) \\
& + \left.\frac{\partial C_{m,\delta_{f,r}}}{\partial \delta_{f,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,r} + \left.\frac{\partial C_{m,\delta_{f,l}}}{\partial \delta_{f,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,l} \\
& + \left.\frac{\partial C_{m,\delta_{t,r}}}{\partial \delta_{t,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,r} + \left.\frac{\partial C_{m,\delta_{t,l}}}{\partial \delta_{t,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,l}
\end{aligned}
\tag{2.56}
$$

$$
\begin{aligned}
\Delta C_n = {} & \left.\frac{\partial C_{n,\delta_{f,r}}}{\partial \delta_{f,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,r} + \left.\frac{\partial C_{n,\delta_{f,l}}}{\partial \delta_{f,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,l} \\
& + \left.\frac{\partial C_{n,\delta_{t,r}}}{\partial \delta_{t,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,r} + \left.\frac{\partial C_{n,\delta_{t,l}}}{\partial \delta_{t,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,l}.
\end{aligned}
\tag{2.57}
$$

Since the first two terms of Equation (2.56) are for fixed values of δ_s, they constitute bias terms and therefore belong in the $f(x)$ portion of Equation (2.17). As a result, only the terms represented by $\left.\frac{\partial C_{i,\delta_s}}{\partial \delta_s}\right|_{M,\alpha,\beta \text{ constant}}$ in Equations (2.55), (2.56), and (2.57) belong in the $g(x)$ term in Equation (2.17).

To complete the analysis of the terms in Equation (2.46), the effect of the center of gravity shift must be accounted for in the nonlinear equations for the angular body accelerations. The shift of a set of moments from a given reference point to the center of gravity is given by the equation

$$
M_{cg} = M_{aero} - r_{cg/aero} \times F_{aero} \tag{2.58}
$$

and in this particular simulation, $r_{cg/aero}$ is defined to be $\begin{bmatrix} x_{cg} & 0 & 0 \end{bmatrix}^T$. In the simulation, F_{aero} is calculated similarly to M_{aero} in Equation (2.47) above, which means that F_{aero} has the form

$$
F_{aero} = \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = \begin{bmatrix} -\bar{q}SC_A \\ \bar{q}SC_Y \\ -\bar{q}SC_N \end{bmatrix}. \tag{2.59}
$$

Therefore, given Equation (2.59) and the definition of $r_{cg/aero}$, Equation (2.58) can be written as

$$M_{cg} = M_{aero} - \begin{bmatrix} x_{cg} \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} -\bar{q}SC_A \\ \bar{q}SC_Y \\ -\bar{q}SC_N \end{bmatrix} \tag{2.60}$$

$$M_{cg} = M_{aero} - \begin{bmatrix} 0 \\ -\bar{q}SC_N x_{cg} \\ -\bar{q}SC_Y x_{cg} \end{bmatrix} \tag{2.61}$$

where M_{aero} is defined in Equation (2.47). It should be noted that the terms C_N and C_Y in Equation (2.61) can be written like the moment coefficients in Equations (2.55), (2.56), and (2.57) as

$$\begin{aligned} \Delta C_N = {} & 2C_{N,\delta_{f,r}}(M, \alpha, \beta, [\delta_{f,r} = 0]) + 2C_{N,\delta_{t,r}}(M, \alpha, \beta, [\delta_{t,r} = 0]) \\ & + \left.\frac{\partial C_{N,\delta_{f,r}}}{\partial \delta_{f,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,r} + \left.\frac{\partial C_{N,\delta_{f,l}}}{\partial \delta_{f,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,l} \\ & + \left.\frac{\partial C_{N,\delta_{t,r}}}{\partial \delta_{t,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,r} + \left.\frac{\partial C_{N,\delta_{t,l}}}{\partial \delta_{t,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,l} \end{aligned} \tag{2.62}$$

$$\begin{aligned} \Delta C_Y = {} & \left.\frac{\partial C_{Y,\delta_{f,r}}}{\partial \delta_{f,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,r} + \left.\frac{\partial C_{Y,\delta_{f,l}}}{\partial \delta_{f,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{f,l} \\ & + \left.\frac{\partial C_{Y,\delta_{t,r}}}{\partial \delta_{t,r}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,r} + \left.\frac{\partial C_{Y,\delta_{t,l}}}{\partial \delta_{t,l}}\right|_{M,\alpha,\beta \text{ constant}} \Delta\delta_{t,l} \end{aligned} \tag{2.63}$$

Similarly to the moment coefficients as shown above, since the first two terms of Equation (2.62) are for fixed values of δ_s, they constitute bias terms and therefore belong in the $f(x)$ portion of Equation (2.17). As a result, only the terms represented by $\left.\frac{\partial C_{i,\delta_s}}{\partial \delta_s}\right|_{M,\alpha,\beta \text{ constant}}$ in Equations (2.62) and (2.63) belong in the $g(x)$ term in Equation (2.17).

Having examined all of the terms in the nonlinear equations for the angular body accelerations, Equation (2.46) can be written in the final form of Equation (2.17) as

$$\begin{aligned}\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = [J]^{-1}\Bigg(-\begin{bmatrix}-J_{xz}pq+(J_z-J_y)qr\\ (J_x-J_z)pr+J_{xz}(p^2-r^2)\\ J_{xz}qr+(J_y-J_x)pq\end{bmatrix} + M_T \\ +\bar{q}SG+\bar{q}SH\begin{bmatrix}\delta_{f,r}\\ \delta_{f,l}\\ \delta_{t,r}\\ \delta_{t,l}\end{bmatrix}\Bigg)\end{aligned} \tag{2.64}$$

where

$$G=\begin{bmatrix} b\Big(C_{\ell,baseline}+\frac{b}{2V_T}(C_{\ell_p}p)\Big)\\ \bar{c}\Big(C_{m,baseline}+\dfrac{\bar{c}}{2V_T}(C_{m_q}q+C_{m_{\dot\alpha}}\dot\alpha)+2C_{m,\delta_{f,r}}(\delta_{f,r}=0)+2C_{m,\delta_{t,r}}(\delta_{t,r}=0)\Big)\\ -2C_{N,\delta_{f,r}}(\delta_{f,r}=0)x_{cg}-2C_{N,\delta_{t,r}}(\delta_{t,r}=0)x_{cg}\\ b\Big(C_{r,baseline}+\frac{b}{2V_T}(C_{n_r}r)\Big)\end{bmatrix} \tag{2.65}$$

and

$$H=\begin{bmatrix} b\frac{\partial C_\ell}{\partial\delta_{f,r}} & b\frac{\partial C_\ell}{\partial\delta_{f,l}} & b\frac{\partial C_\ell}{\partial\delta_{t,r}} & b\frac{\partial C_\ell}{\partial\delta_{t,l}}\\ \Big(\bar{c}\frac{\partial C_m}{\partial\delta_{f,r}}-x_{cg}\frac{\partial C_N}{\partial\delta_{f,r}}\Big) & \Big(\bar{c}\frac{\partial C_m}{\partial\delta_{f,l}}-x_{cg}\frac{\partial C_N}{\partial\delta_{f,l}}\Big) & \Big(\bar{c}\frac{\partial C_m}{\partial\delta_{t,r}}-x_{cg}\frac{\partial C_N}{\partial\delta_{t,r}}\Big) & \Big(\bar{c}\frac{\partial C_m}{\partial\delta_{t,l}}-x_{cg}\frac{\partial C_N}{\partial\delta_{t,l}}\Big)\\ \Big(b\frac{\partial C_n}{\partial\delta_{f,r}}-x_{cg}\frac{\partial C_Y}{\partial\delta_{f,r}}\Big) & \Big(b\frac{\partial C_n}{\partial\delta_{f,l}}-x_{cg}\frac{\partial C_Y}{\partial\delta_{f,l}}\Big) & \Big(b\frac{\partial C_n}{\partial\delta_{t,r}}-x_{cg}\frac{\partial C_Y}{\partial\delta_{t,r}}\Big) & \Big(b\frac{\partial C_n}{\partial\delta_{t,l}}-x_{cg}\frac{\partial C_Y}{\partial\delta_{t,l}}\Big)\end{bmatrix}. \tag{2.66}$$

It should be noted that the partial derivatives in Equation (2.64) are taken with respect to a constant value of M, α, and β from the current flight condition and that the control surface bias terms, where $\delta_s = 0$, are evaluated at a constant value of M, α and β from the current flight condition as well.

Given Equation (2.64), which is now in the form of Equation (2.17), the adaptive dynamic inversion controller can be constructed using Equations (2.13), (2.24), (2.30), and (2.38).

2.4 α, β, μ Inversion Controller

As with the p, q, r inversion controller, equations for $\dot{\alpha}$, $\dot{\beta}$, and $\dot{\mu}$ must be determined in order for the adaptive dynamic inversion controller to be constructed. It should be noted that for this section, S_x will represent $\sin(x)$, C_x will represent $\cos(x)$, and T_x will represent $\tan(x)$, where x is an angle. The derivations for $\dot{\alpha}$ and $\dot{\beta}$ are based on the derivations for those terms on pages 110-112 in Reference [26]. The starting point of the derivations is the basic force equations in the stability axes under the flat Earth assumption, which are

$$^{b}\dot{v}_{rel} = (1/m)F_{A,T} + g - \omega_{b/e} \times v_{rel}. \tag{2.67}$$

Taking the time derivative of the relative velocity in the wind axes instead of in the body axes and converting the right hand side of Equation (2.67) to the wind axes produces the result

$$m\dot{V}_T = F_T C_{\alpha+\alpha_T} C_\beta - D - mgS_\gamma \tag{2.68}$$

$$\begin{aligned} m\dot{\beta}V_T = &- F_T C_{\alpha+\alpha_T} S_\beta - C + mg(C_\alpha S_\beta S_\theta + C_\beta S_\phi C_\theta - S_\alpha S_\beta C_\phi C_\theta) \\ &- mV_T(pS_\alpha - rC_\alpha) \end{aligned} \tag{2.69}$$

$$\begin{aligned} m\dot{\alpha}V_T C_\beta = &- F_T S_{\alpha+\alpha_T} - L + mg(S_\alpha S_\theta + C_\alpha C_\phi C_\theta) \\ &+ mV_T(-pS_\beta C_\alpha + qC_\beta - rS_\beta S_\alpha) \end{aligned} \tag{2.70}$$

where D, L, and C represent drag, lift, and cross-wind force, respectively, in the wind axes.

In order to simplify Equations (2.69) and (2.70) and to express them in terms of μ, which is one of the commanded states, the gravity terms in those equations are transformed using relationships given by the following direction cosine matrices from Chapter 4 of Reference [23] as

$$T_{W,H}(\mu,\gamma,\chi) = T^T_{B,W}(0,-\alpha,\beta)T_{B,H}(\phi,\theta,\psi) \tag{2.71}$$

where W represents the wind axes, B represents the body axes, and H represents the local horizon axes. Each direction cosine matrix has the general form

$$T_{2,1}(\theta_x,\theta_y,\theta_z) = \begin{bmatrix} C_{\theta_y}C_{\theta_z} & C_{\theta_y}S_{\theta_z} & -S_{\theta_y} \\ S_{\theta_x}S_{\theta_y}C_{\theta_z} - C_{\theta_x}S_{\theta_z} & S_{\theta_x}S_{\theta_y}S_{\theta_z} + C_{\theta_x}C_{\theta_z} & S_{\theta_x}C_{\theta_y} \\ C_{\theta_x}S_{\theta_y}C_{\theta_z} + S_{\theta_x}S_{\theta_z} & C_{\theta_x}S_{\theta_y}S_{\theta_z} - S_{\theta_x}C_{\theta_z} & C_{\theta_x}C_{\theta_y} \end{bmatrix}. \tag{2.72}$$

as shown on page 9 of Reference [27]. By examining the elements of the matrices in Equation (2.71), the following relationships involving μ and γ were determined to be

$$T_{W,H}(2,3) = S_\mu C_\gamma = C_\alpha S_\beta S_\theta + C_\beta S_\phi C_\theta - S_\alpha S_\beta C_\phi C_\theta \tag{2.73}$$

$$T_{W,H}(3,3) = C_\mu C_\gamma = S_\alpha S_\theta + C_\alpha C_\phi C_\theta \tag{2.74}$$

which can be substituted into Equations (2.69) and (2.70) in the gravity terms.

Additionally, the thrust force F_T terms are converted into the wind frame and expressed in terms of the vector $\begin{bmatrix} F_{T_x} & F_{T_y} & F_{T_z} \end{bmatrix}^T$, which is given in the body frame.

The transformation of the F_T terms results in

$$\begin{aligned}\begin{bmatrix} F_T C_{\alpha+\alpha_T} C_\beta \\ -F_T C_{\alpha+\alpha_T} S_\beta \\ -F_T S_{\alpha+\alpha_T} \end{bmatrix} &= \begin{bmatrix} C_\alpha C_\beta & S_\beta & S_\alpha C_\beta \\ -C_\alpha S_\beta & C_\beta & -S_\alpha S_\beta \\ -S_\alpha & 0 & C_\alpha \end{bmatrix} \begin{bmatrix} F_{T_x} \\ F_{T_y} \\ F_{T_z} \end{bmatrix} \\ &= \begin{bmatrix} F_{T_x} C_\alpha C_\beta + F_{T_y} S_\beta + F_{T_z} S_\alpha C_\beta \\ -F_{T_x} C_\alpha S_\beta + F_{T_y} C_\beta - F_{T_z} S_\alpha S_\beta \\ -F_{T_x} S_\alpha + F_{T_z} C_\alpha \end{bmatrix}.\end{aligned} \tag{2.75}$$

Finally, the forces D, C, and L must be written in terms of the corresponding forces in the stability axes, which can be calculated directly from information in the model, as

$$\begin{aligned}\begin{bmatrix} D \\ C \\ L \end{bmatrix} &= \begin{bmatrix} C_\beta & S_\beta & 0 \\ -S_\beta & C_\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_s \\ Y_s \\ L_s \end{bmatrix} \\ &= \begin{bmatrix} D_s C_\beta + Y_s S_\beta \\ -D_s S_\beta + Y_s C_\beta \\ L_s \end{bmatrix}.\end{aligned} \tag{2.76}$$

It is assumed that the D_s terms are absorbed into the thrust terms in Equation (2.75).

Substituting Equations (2.73), (2.74), (2.75), and (2.76) into Equations (2.69) and (2.70) gives the final equations for $\dot{\beta}$ and $\dot{\alpha}$ to be

$$\dot{\beta} = \frac{1}{mV_T}\big((Y_s + F_{T_y})C_\beta + mgS_\mu C_\gamma - F_{T_x} C_\alpha S_\beta - F_{T_z} S_\alpha S_\beta\big) + (pS_\alpha - rC_\alpha) \tag{2.77}$$

$$\dot{\alpha} = \frac{1}{mV_T C_\beta}\left(-L_s + mgC_\mu C_\gamma - F_{T_x} S_\alpha + F_{T_z} C_\alpha\right) + \left(-pC_\alpha T_\beta + q - rS_\alpha T_\beta\right) \tag{2.78}$$

which agree with the equations for $\dot{\beta}$ and $\dot{\alpha}$ in Reference [22].

Now, the equation for $\dot{\mu}$ can be derived since the derivation involves the results given in Equations (2.77) and (2.78). Starting from Equation (57) on page 56 of Reference [23], where, for this document $\beta = -\sigma$ in Reference [23], the relationship between the angular body accelerations and the local horizon angular accelerations are expressed as

$$\begin{bmatrix} p - \dot{\beta} S_\alpha \\ q - \dot{\alpha} \\ r + \dot{\beta} C_\alpha \end{bmatrix} = \begin{bmatrix} C_\alpha C_\beta & -C_\alpha S_\beta & -S_\alpha \\ S_\beta & C_\beta & 0 \\ S_\alpha C_\beta & -S_\alpha S_\beta & C_\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & -S_\gamma \\ 0 & C_\mu & S_\mu C_\gamma \\ 0 & -S_\mu & C_\mu C_\gamma \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix} . \qquad (2.79)$$

Taking the inverse of Equation (2.79), the equation for $\dot{\mu}$ is determined to be

$$\begin{aligned} \dot{\mu} = {} & (p - \dot{\beta} S_\alpha)\left(C_\alpha C_\beta - T_\gamma C_\alpha S_\beta S_\mu - T_\gamma S_\alpha C_\mu\right) + (q - \dot{\alpha})\left(S_\beta + T_\gamma C_\beta S_\mu\right) \\ & + (r + \dot{\beta} C_\alpha)\left(S_\alpha C_\beta + T_\gamma C_\alpha C_\mu - T_\gamma S_\alpha S_\beta S_\mu\right) . \end{aligned} \qquad (2.80)$$

Substituting Equations (2.77) and (2.78) into Equation (2.80) and simplifying gives the final equation for $\dot{\mu}$, which is

$$\begin{aligned} \dot{\mu} = \frac{1}{mV_T} \Big(& L_s(T_\beta + T_\gamma S_\mu) + (Y_s + F_{T_y}) T_\gamma C_\mu C_\beta - mg C_\gamma C_\mu T_\beta \\ & + (F_{T_x} S_\alpha - F_{T_z} C_\alpha)(T_\gamma S_\mu + T_\beta) - (F_{T_x} C_\alpha + F_{T_z} S_\alpha) T_\gamma C_\mu S_\beta \Big) \\ & + p C_\alpha \sec(\beta) + r S_\alpha \sec(\beta). \end{aligned} \qquad (2.81)$$

Once the equation for $\dot{\mu}$ has been determined, Equations (2.77), (2.78), and (2.81) are combined together in vector-matrix equation form as

$$\begin{bmatrix}\dot{\beta}\\ \dot{\alpha}\\ \dot{\mu}\end{bmatrix} = \begin{bmatrix} \dfrac{1}{mV_T}\big((Y_s + F_{T_y})C_\beta + mgS_\mu C_\gamma - F_{T_x}C_\alpha S_\beta - F_{T_z}S_\alpha S_\beta\big) \\ \dfrac{1}{mV_T C_\beta}\left(-L_s + mgC_\mu C_\gamma - F_{T_x}S_\alpha + F_{T_z}C_\alpha\right) \\ \dfrac{1}{mV_T}\Big(L_s(T_\beta + T_\gamma S_\mu) + (Y_s + F_{T_y})T_\gamma C_\mu C_\beta - mgC_\gamma C_\mu T_\beta \\ \quad + (F_{T_x}S_\alpha - F_{T_z}C_\alpha)(T_\gamma S_\mu + T_\beta) \\ \quad - (F_{T_x}C_\alpha + F_{T_z}S_\alpha)T_\gamma C_\mu S_\beta\Big) \end{bmatrix} + \begin{bmatrix} S_\alpha & 0 & -C_\alpha \\ -T_\beta C_\alpha & 1 & -T_\beta S_\alpha \\ \sec(\beta)C_\alpha & 0 & \sec(\beta)S_\alpha \end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{2.82}$$

where p, q, and r are the desired angular body rates. It should be noted that it is assumed that the forces due to control surface deflections are negligible, and therefore, the force terms in Equation (2.82) are approximated from look-up tables for the force and moment coefficients at points where the control surface deflections are equal to 0. Also, it is assumed that for the desired angular body rates that the inner loop p, q, r controller is perfect, which means that the desired angular rates equal the commanded angular rates.

Given Equation (2.82), which is now in the form of Equation (2.1), the adaptive dynamic inversion controller can be constructed using Equations (2.5) and (2.13).

2.5 Simulation Results

Based on the control and adaptive laws derived in the previous sections, a simulation of the entire GHV system with the adaptive nonlinear dynamic inversion controllers was created in Simulink. In order to make the simulation more realistic, second-order actuator dynamics with damping ratio $\zeta = 0.7$ and natural frequency

$\omega_n = 25$ Hz are included in the current simulation, and position and rate limits are placed on the control surfaces of 30 deg and 100 deg/s, respectively. Additionally, a time delay of 0.03 s is included in the simulation; however, it should be noted that the simulation can tolerate time delays of up to 0.04 s without the responses becoming significantly oscillatory. Commands to α, β, and μ are given as ramp signals from 0 degrees to a commanded angle in fixed time. For the α, β, μ inversion controller, the basis function $\beta(x; d)$ is chosen to be $\beta(x; d) = \begin{bmatrix} c & \alpha & \beta & \mu & M \end{bmatrix}^T$, where c is a constant bias term. For the p, q, r inversion controller, the basis function $\beta(x; d)$ is chosen to be $\beta(x; d) = \begin{bmatrix} c & p & q & r & \alpha & \beta & M \end{bmatrix}^T$, where c is a constant bias term. For a discussion on the Matlab integration solvers used for the GHV simulation, see Appendix A.

The total velocity of the vehicle is controlled using a PID controller. The input to the controller is the commanded total velocity of the GHV, and the output of the controller is the equivalence ratio. The equivalence ratio is the fifth control, and along with the four control surfaces, completes the the control complement for the vehicle. Additionally, a saturation limiter has been added after the velocity PID controller to constrain the equivalence ratio to be between 0 and 1.

In the derivation of the adaptive dynamic controllers in Section 2.2, a reference model was described. The difference between this reference model and the actual system dynamics constitutes the tracking error of the system. In order to determine the reference states of the system, the reference signal r must be defined. For the α, β, μ inversion controller, the reference signal consists of the commanded values of α, β, and μ. For the p, q, r inversion controller, the reference signal consists of the commanded angular body rates from the α, β, μ inversion controller. Both of the

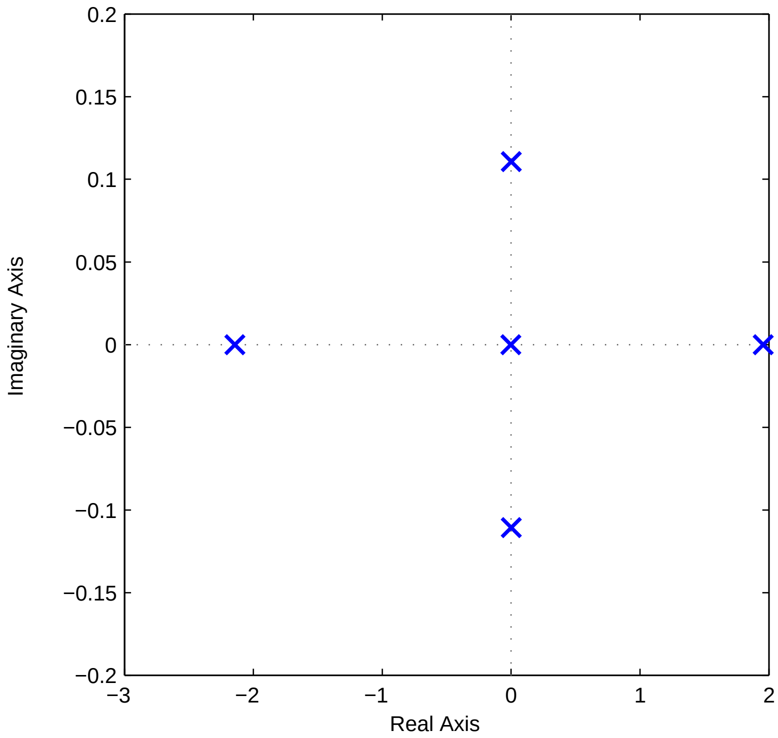


Figure 2.3: Open-loop poles for the linearized longitudinal dynamics.

reference models have the general form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \eta_1 & 0 & 0 \\ 0 & \eta_2 & 0 \\ 0 & 0 & \eta_3 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{2.83}$$

where ξ_1, ξ_2, ξ_3, η_1, η_2, and η_3 are scalars that define the desired time constants of each control channel.

The open-loop poles of the linearized dynamics at the flight condition of Mach 6 at 80,000 ft for both the longitudinal and lateral-directional states are shown in

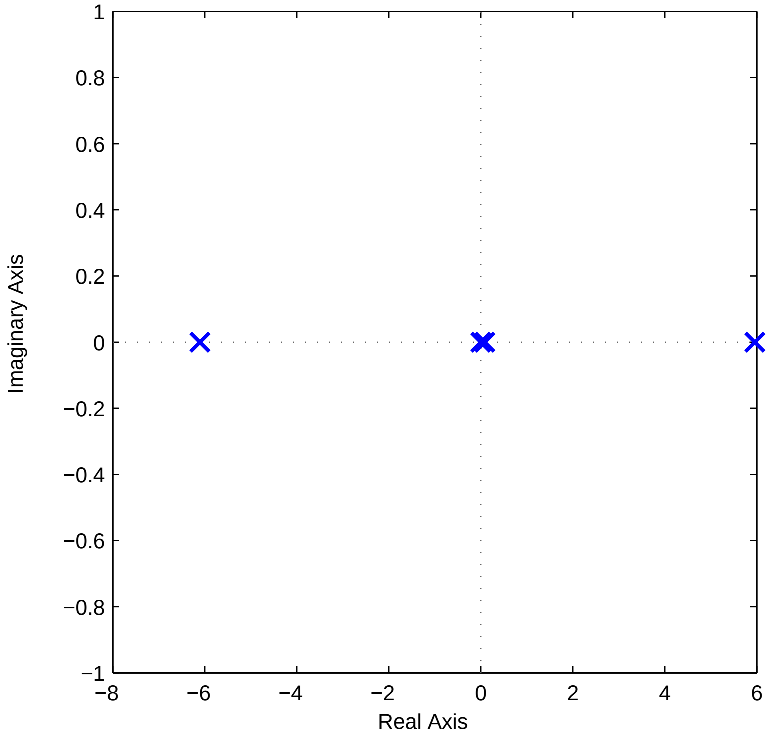


Figure 2.4: Open-loop poles for the linearized lateral/directional dynamics.

Table 2.1: Eigenvalues for the linearized longitudinal dynamics.

Eigenvalue	Damping Ratio	Natural Frequency (rad/s)
-2.14	1.00	2.14
-2.79×10^{-3}	1.00	2.79×10^{-3}
$1.25 \times 10^{-3} \pm 0.111j$	-0.0113	0.111
1.96	-1.00	1.96

Table 2.2: Eigenvalues for the linearized lateral/directional dynamics.

Eigenvalue	Damping Ratio	Natural Frequency (rad/s)
-6.10	1.00	6.10
2.22×10^{-16}	-1.00	2.22×10^{-16}
0.088	-1.00	0.088
5.96	-1.00	5.96

Figures 2.3 and 2.4. It should be noted from the eigenvalues listed in Tables 2.1 and 2.2 that both the longitudinal and lateral-directional states have several eigenvalues in the right-half plane, which indicates that the GHV is an unstable vehicle. A nonlinear adaptive dynamic inversion controller will be able to suppress the unstable dynamics and replace them with desired dynamics for the aircraft.

Figures 2.5, 2.6, 2.7, and 2.8 show representative simulation results with the nonlinear adaptive dynamic inversion controller for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg. The responses are well-behaved, and the controller is able to achieve the desired tracking performance without excessive control effort. It should be noted that following this preliminary analysis of the nonlinear adaptive dynamic inversion controller, pseudo-control hedging ([28],[29]) was added to the simulation in order to protect the nonlinear adaptive dynamic inversion controllers during periods of control surface saturation.

2.5.1 Robustness Analysis

A robustness analysis was performed via simulation on the designed adaptive nonlinear dynamic inversion controller from the previous section. Uncertainties in the plant examined in the analysis include the additive uncertainties ΔC_{m_α}, ΔC_{n_β}, and ΔC_m and multiplicative gains D on the control surface deflections, given in terms of equations as

$$C_m = C_{m_{baseline}} + \Delta C_{m_\alpha} \alpha \tag{2.84}$$

$$C_n = C_{n_{baseline}} + \Delta C_{n_\beta} \beta \tag{2.85}$$

$$C_m = C_{m_{baseline}} + \Delta C_m \tag{2.86}$$

$$C_\delta = D C_{\delta_o}. \tag{2.87}$$

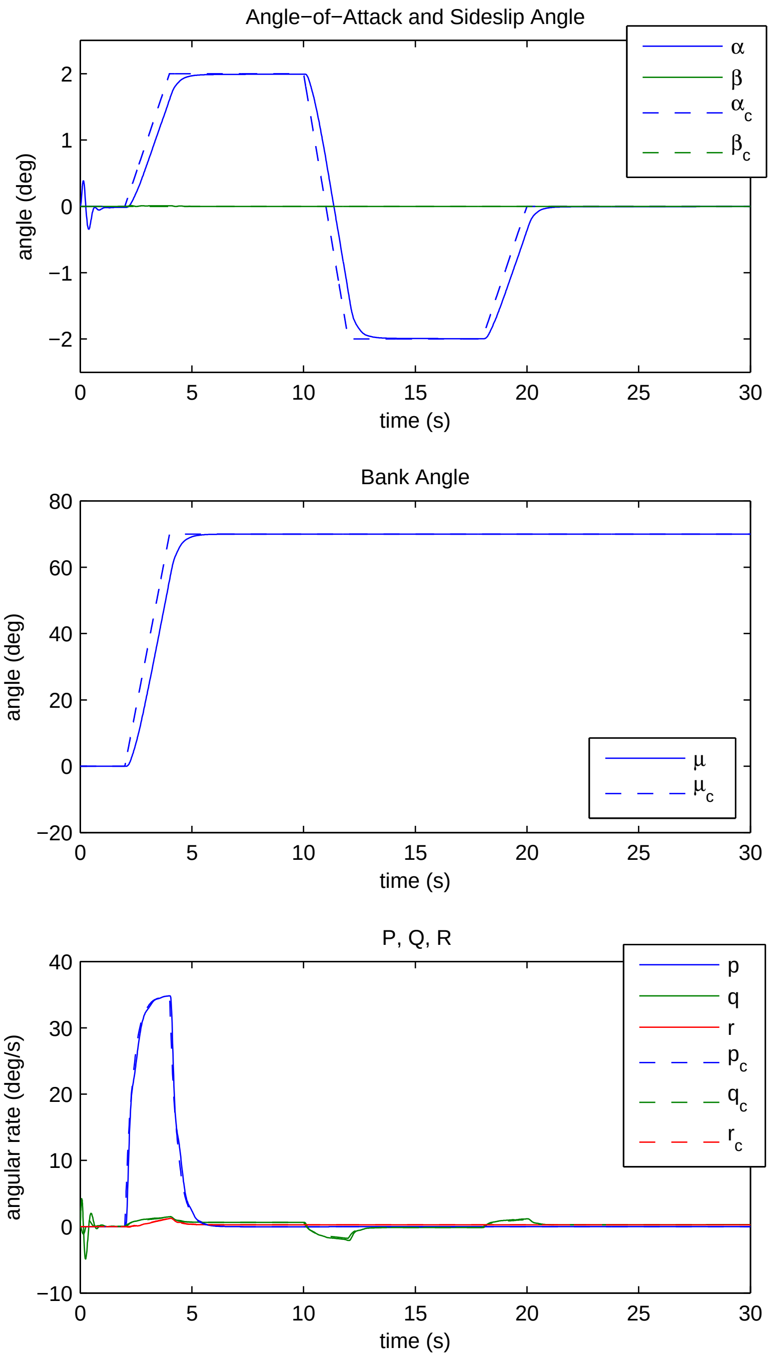


Figure 2.5: State responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg.

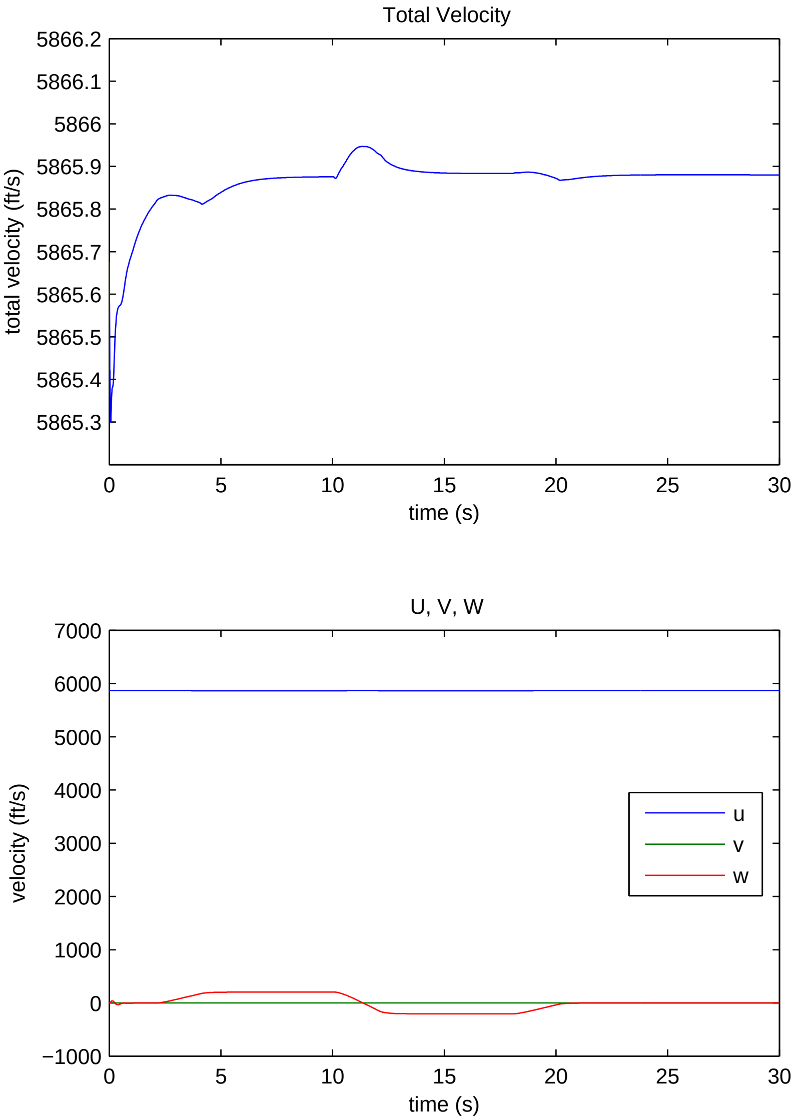


Figure 2.6: Velocity responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg.

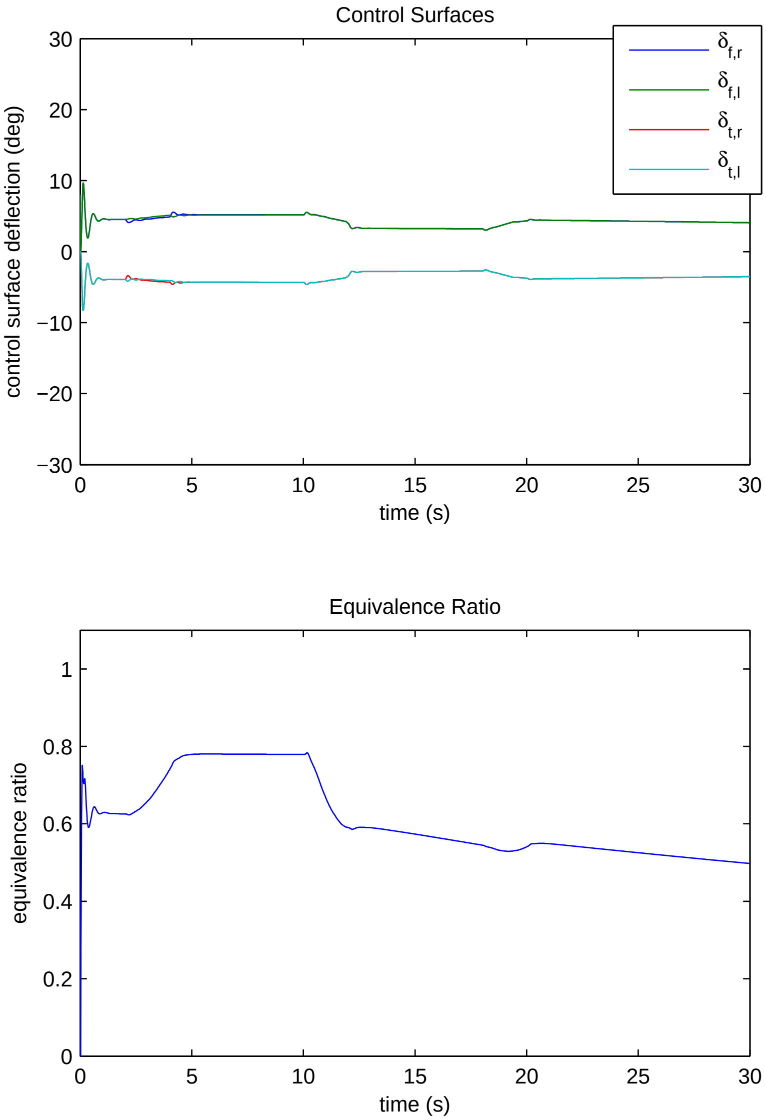


Figure 2.7: Control responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg.

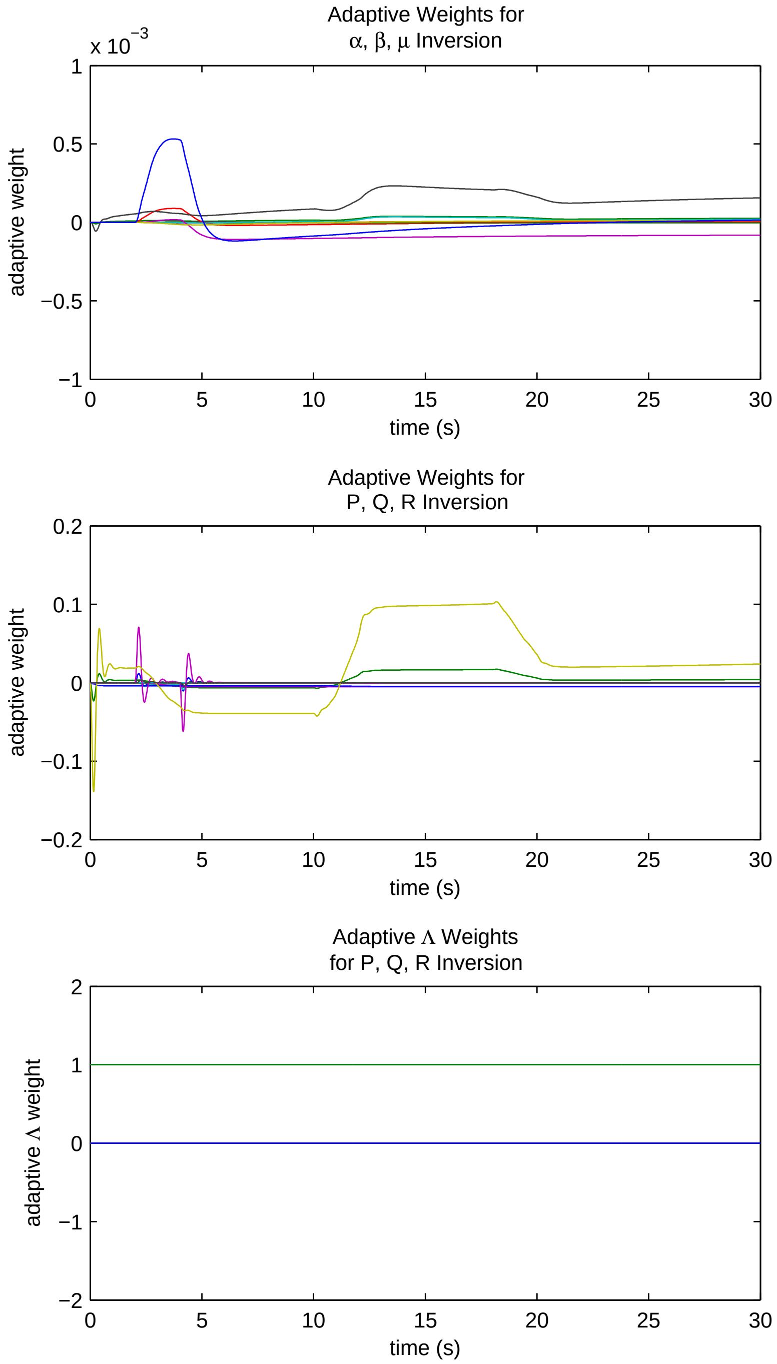


Figure 2.8: Adaptive weight responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg.

Table 2.3: Additive uncertainty ΔC_{m_α} over a 30 s period with 0.03 s time delay.

α (deg)	β (deg)	μ (deg)	max ΔC_{m_α} (deg^{-1})	min ΔC_{m_α} (deg^{-1})
5	0	0	0.0005	-0.0013
5	1	20	0.0003	-0.0011

Table 2.4: Additive uncertainty ΔC_{n_β} over a 30 s period with 0.03 s time delay.

α (deg)	β (deg)	μ (deg)	max ΔC_{n_β} (deg^{-1})	min ΔC_{n_β} (deg^{-1})
0	1	0	0.007	-0.003
5	0	20	0.01	-0.004
5	1	20	0.006	-0.003

The criteria for determining the bounds on the uncertainties is that the states must not demonstrate oscillatory behavior.

Tables 2.3 and 2.4 provide the maximum and minimum values of the additive uncertainties ΔC_{m_α} and ΔC_{n_β} for various α, β, and μ commands. It should be noted that an examination of the maximum and minimum baseline values of C_{m_α} and C_{n_β} show that these values are on the order of 10^{-4}. The maximum and minimum values for ΔC_{m_α} and ΔC_{n_β} in Tables 2.3 and 2.4 are on the order of $10^{-3} - 10^{-4}$, and therefore, the controllers are able to withstand significant uncertainties in C_{m_α} and C_{n_β} and maintain stable tracking flight. Similar results for the additive uncertainty ΔC_m can be found in Table 2.5.

Table 2.6 contains the minimum allowable multiplicative gains D on the control surface deflections for various α, β, and μ commands. These gains represent a loss of control effectiveness for one or more of the control surfaces on the GHV. For all cases, the vehicle is able to tolerate low values of control effectiveness, which shows that the controllers are robust to loss of control effectiveness.

Table 2.5: Additive uncertainty ΔC_m over a 30 s period with 0.03 s time delay.

α (deg)	β (deg)	μ (deg)	max ΔC_m	min ΔC_m
5	0	0	0.0005	-0.003
5	1	20	0.0005	-0.002

Table 2.6: Multiplicative gains D on control surface deflection terms over a 30 s period with 0.03 s time delay.

α (deg)	β (deg)	μ (deg)	$D_{\delta_{f,r}}$	$D_{\delta_{f,l}}$	$D_{\delta_{t,r}}$	$D_{\delta_{t,l}}$
5	0	0	1	0.14	1	1
5	0	0	1	1	1	0.01
5	0	0	0.15	0.15	1	1
5	0	0	1	1	0.15	0.15
5	0	20	1	0.31	1	1
5	0	20	1	1	1	0.01
5	0	20	0.21	0.21	1	1
5	0	20	1	1	0.30	0.30
5	1	20	1	0.42	1	1
5	1	20	1	1	1	0.05
5	1	20	0.38	0.38	1	1
5	1	20	1	1	0.38	0.38

Additionally, the controllers were tested for their ability to withstand a β bias. Because of sensor uncertainty, the true value of β will never be known. Therefore, in the simulation for the commands $\alpha = 5$ deg, $\beta = 1$ deg, and $\mu = 20$ deg, a bias of 1 degree was added to the actual value of β. It was determined that the controllers are able to handle the β bias, allowing the simulation to run to completion and the given commands to be followed.

2.6 Conclusions

Based on the simulation results and the robustness analysis, it can be seen that the objective of designing a control architecture that is robust in order to achieve desired tracking performance was achieved for the GHV. The controllers are robust to decreases in control surface effectiveness, changes in system parameters, and time delays of 0.04 seconds or less. Therefore, it can be concluded that this approach of nonlinear adaptive dynamic inversion control works well as a control architecture for hypersonic vehicles.

3. ENFORCING STATE CONSTRAINTS IN A NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL ARCHITECTURE

The nonlinear adaptive dynamic inversion architecture is able to deal with parametric uncertainty. However, as noted in Chapter 2, there are limits on the amount of variation in parameters such as the static longitudinal stability derivative (C_{m_α}) and the static directional stability derivative (C_{n_β}) beyond which the system response starts to oscillate about the reference trajectory. If the reference trajectory commands angles-of-attack and sideslip angles that are near the limits set on those states to prevent inlet unstarts, oscillations about the reference trajectory can cause the states of the aircraft to exceed their constraints. This chapter presents a method to enforce state constraints within the existing nonlinear adaptive dynamic inversion control architecture. A brief introduction to projection operators is given first, followed by the theory for the inclusion of the state constraints in the control architecture and simulation results.

3.1 Projection Operators

The projection operator for two vectors $\theta, y \in \mathbb{R}^k$ is defined mathematically as

$$\mathrm{Proj}(\theta, y) = \begin{cases} y - \frac{\nabla h(\theta)(\nabla h(\theta))^T}{\|\nabla h(\theta)\|^2} y h(\theta) & \text{if } h(\theta) > 0 \text{ and } y^T \nabla h(\theta) > 0 \\ y & \text{otherwise.} \end{cases} \tag{3.1}$$

where $h(\theta) : \mathbb{R}^k \mapsto \mathbb{R}$ is a convex function, and $\nabla h(\theta) = \left(\frac{\partial h(\theta)}{\partial \theta_1} \dots \frac{\partial h(\theta)}{\partial \theta_k}\right)^T$ [30]. Figure 3.1 depicts the effect of the projection operator in $\mathbb{R}^2$, where $\{\theta | f(\theta) = 0\}$ represents the boundary of the region in which the state constraints are satisfied, $\{\theta | f(\theta) = 1\}$ represents the boundary outside of which the state constraints are violated, and Ω_A

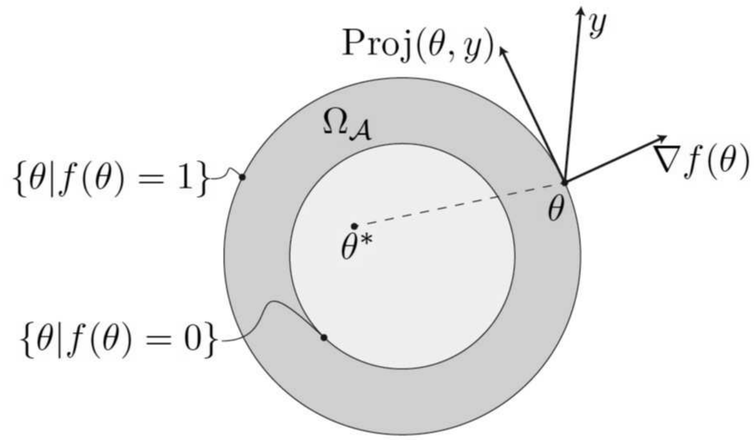

$\mathrm{Proj}(\theta, y)$
y
$\Omega_{\mathcal{A}}$
$\{\theta|f(\theta) = 1\}$
$\nabla f(\theta)$
θ
θ^*
$\{\theta|f(\theta) = 0\}$

Figure 3.1: "Visualization of the Projection Operator in $\mathbb{R}^2$" [30].

represents the region in which the projection operator acts on the variable y. In this figure, $f(\theta)$ is equivalent to $h(\theta)$ in equation (3.1).

In order to use the projection operator given in equation (3.1), a convex function $h(\theta)$ must be defined. One candidate function is

$$h(\theta) = \frac{(\epsilon_\theta + 1)\theta^T\theta - \theta_{max}^2}{\epsilon_\theta \theta_{max}^2} \tag{3.2}$$

where θ_{max} represents the maximum value of the norm of the vector θ, ϵ_θ represents the allowable tolerance of θ beyond θ_{max}, and $0 < \epsilon_\theta \leq 1$ [31].

The following lemma indicates that the projection operator can be used in the enforcement of state constraints.

Lemma 1 (Lavretsky, et.al., 2012) *If an initial value problem, i.e., adaptive control algorithm with adaptive law and initial conditions, is defined by:*

1. $\dot{\theta} = \mathrm{Proj}(\theta, y)$

2. $\theta(t = 0) = \theta_0 \in \Omega_1 = \{\theta \in \mathbb{R}^k | h(\theta) \leq 1\}$

3. $h(\theta) : \mathbb{R}^k \mapsto \mathbb{R}$ is convex

then $\theta(t) \in \Omega_1 \forall t \geq 0$ [30].

The proof for this lemma can be found in [30]. According to this lemma, as long as the value of θ at $t = 0$ is within the region in which $h(\theta) \leq 1$, which means that the state constraints are satisfied initially, then the constraints will be satisfied at all times through the use of the projection operator.

The following lemma will be useful in proving that the closed-loop system involving the control law that enforces state constraints is stable.

Lemma 2 (Lavretsky, et.al., 2012) *One important property of the projection operator follows. Given $\theta^* \in \Omega_0$,*

$$(\theta - \theta^*)^T (\text{Proj}(\theta, y, h) - y) \leq 0 \text{ [30].}$$

The proof for this lemma can be found in [30].

3.2 General Adaptive Control Equations with State Constraints

Consider a general nonlinear equation of a system in the form

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

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^n$ is the control, and $f(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ are locally Lipschitz continuous. It is assumed that $g(x)$ is nonsingular for all $x \in \mathbb{R}^n$. Suppose that the desired reference dynamics for the system are given by

$$\dot{x}_m = Ax_m + Br \tag{3.4}$$

where $x_m \in \mathbb{R}^n$ is the model state, $r \in \mathbb{R}^n$ is a bounded reference signal, $A \in \mathbb{R}^{n\times n}$ is Hurwitz, and $B \in \mathbb{R}^{n\times n}$. The equation for the error between the reference model and the actual system is

$$e = x_m - x. \tag{3.5}$$

Taking the time derivative of equation (3.5) results in

$$\dot{e} = \dot{x}_m - \dot{x} = \dot{x}_m - f(x) - g(x)u. \tag{3.6}$$

In the standard nonlinear adaptive dynamic inversion control architecture, the control u is chosen to be

$$u = [g(x)]^{-1}[\dot{x}_m - \hat{f}(x) + Ke - \nu] \tag{3.7}$$

where $\hat{f}(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a model of the plant dynamics, $K \in \mathbb{R}^{n \times n}$ such that $K = K^T > 0$ are the gains on the tracking errors, and $\nu \in \mathbb{R}^n$ is a pseudo-control signal. Substituting equation (3.7) into equation (3.3) results in the equation

$$\dot{x} = f(x) + \dot{x}_m - \hat{f}(x) + Ke - \nu. \tag{3.8}$$

The nonlinear adaptive dynamic inversion control architecture has been shown to be locally stable in the Lyapunov sense, which indicates that the error eventually will tend to zero. Another part of equation (3.8), $f(x) - \hat{f}(x) - \nu$, also will tend to zero because of the adaptive controller. As a result, in the nonlinear adaptive dynamic inversion control architecture, eventually, $\dot{x} = \dot{x}_m$. Consequently, if the reference model exceeds the state constraints, then the actual system will be commanded to exceed the state constraints as well. Therefore, the control law must be adjusted so that the reference model does not violate the state constraints. This objective is achieved by adding a projection operator to the control law. If the control u now is chosen to be

$$u = [g(x)]^{-1}[\text{Proj}(x, \dot{x}_m) - \hat{f}(x) + Ke - \nu], \tag{3.9}$$

where

$$\text{Proj}(x, \dot{x}_m) = \begin{bmatrix} \text{Proj}(x_1, \dot{x}_{m_1}) \\ \text{Proj}(x_2, \dot{x}_{m_2}) \\ \vdots \\ \text{Proj}(x_n, \dot{x}_{m_n}) \end{bmatrix}, \tag{3.10}$$

then substituting equation (3.9) into equation (2.4) now produces the error dynamics

$$\dot{e} = \dot{x}_m - f(x) - \text{Proj}(x, \dot{x}_m) + \hat{f}(x) - Ke + \nu. \tag{3.11}$$

Defining the error between the model and the actual system as $\Delta = \hat{f}(x) - f(x)$, equation (3.11) becomes

$$\dot{e} = -Ke + \Delta + \nu + \dot{x}_m - \text{Proj}(x, \dot{x}_m). \tag{3.12}$$

Again, in this dissertation, it is assumed that Δ can be represented in the form $\Delta = W^T\beta(x; d)$, where $W \in \mathbb{R}^{p\times n}$ is a set of unknown weights, and $\beta \in \mathbb{R}^{p\times 1}$ is a set of known basis functions composed of the states x and a vector d of bounded continuous exogenous inputs. Using this representation for Δ, ν is chosen to be $\nu = -\widehat{W}^T\beta(x; d)$, where $\widehat{W} \in \mathbb{R}^{p\times n}$. With these definitions, equation (3.12) can be written as

$$\dot{e} = -Ke - \widetilde{W}^T\beta(x; d) - [\text{Proj}(x, \dot{x}_m) - \dot{x}_m] \tag{3.13}$$

where $\widetilde{W} = \widehat{W} - W$ is the weight estimation error. It should be noted that if the state constraints are not within a certain tolerance of being violated, then $\text{Proj}(x, \dot{x}_m) =$

$\dot{x}_m$, and equation (3.13) reverts back to its original form when using the nonlinear adaptive dynamic inversion control architecture without enforcing state constraints.

The stability of the closed loop system under these assumptions can be examined using a candidate Lyapunov function of the form

$$V = e^T e + tr(\widetilde{W}^T {\Gamma_W}^{-1} \widetilde{W}) \tag{3.14}$$

where $\Gamma_W \in \mathbb{R}^{p \times p}$ with $\Gamma_W = {\Gamma_W}^T > 0$. In order to determine the adaptation law for the parameters in W and to determine if the error between the states of the actual system and the reference model will converge, first, the derivative of equation (3.14) along the system trajectories is taken, which gives the result

$$\dot{V} = 2e^T \dot{e} + 2tr(\widetilde{W}^T {\Gamma_W}^{-1} \dot{\widehat{W}}^T). \tag{3.15}$$

Substituting equation (3.13) into equation (3.15) produces

$$\dot{V} = -2e^T K e - 2e^T \widetilde{W}^T \beta(x;d) - 2e^T[\text{Proj}(x, \dot{x}_m) - \dot{x}_m] + 2tr(\widetilde{W}^T {\Gamma_W}^{-1} \dot{\widehat{W}}^T). \tag{3.16}$$

Applying the trace identity that $a^T b = tr(ba^T)$, equation (3.16) is determined to be

$$\dot{V} = -2e^T K e - 2e^T[\text{Proj}(x, \dot{x}_m) - \dot{x}_m] + 2tr(\widetilde{W}^T({\Gamma_W}^{-1} \dot{\widehat{W}}^T - \beta(x;d)e^T)). \tag{3.17}$$

Then, by choosing $\dot{\widehat{W}}$ as

$$\dot{\widehat{W}} = \Gamma_W \, \text{Proj}(\widehat{W}, \beta(x;d)e^T), \tag{3.18}$$

equation (3.17) becomes

$$\dot{V} = -2e^T K e - 2e^T[\text{Proj}(x, \dot{x}_m) - \dot{x}_m]. \tag{3.19}$$

As noted previously, if the state constraints are not within a certain tolerance of being exceeded, then $\text{Proj}(x, \dot{x}_m) = \dot{x}_m$, and as a result, the second term of equation (3.19) is zero, and $\dot{V}$ can be upper bounded as

$$\dot{V} \leq -2e^T K e \leq 0 \tag{3.20}$$

which implies that e is bounded. Because r is bounded by definition above, x_m is bounded. Since e and x_m are bounded, x is bounded. Consequently, $\beta(x; d)$ is bounded as well. In order to use Barbalat's lemma to complete the proof, the second derivative of equation (2.9) along the system trajectories is taken, which gives the result

$$\ddot{V} = -4e^T K \dot{e}. \tag{3.21}$$

Substituting equation (3.13) into equation (3.21) produces

$$\ddot{V} = -4e^T K(-Ke - \widetilde{W}^T \beta(x; d)). \tag{3.22}$$

Because e, $\widetilde{W}$, and $\beta(x; d)$ are bounded as proved above, $\ddot{V}$ is bounded, and therefore $\dot{V}$ is uniformly continuous.

Because V is lower bounded, $\dot{V}$ is negative semi-definite, and $\dot{V}$ is uniformly continuous, by Barbalat's lemma $\dot{V} \to 0$ as $t \to \infty$, and thus $e \to 0$ as $t \to \infty$ as desired.

However, if the state constraints are within a certain tolerance of being violated, the second term of equation (3.19) is not equal to zero, and a separate stability analysis must be done. From Lemma (2), let $\theta = x$, $\theta^* = x_m$, and $y = \dot{x}_m$. Then, Lemma (2) can be applied to equation (3.19) to prove that with the projection operator active, $\dot{V}$ can be upper bounded as

$$\dot{V} \leq -2e^T K e - 2e^T[\text{Proj}(x, \dot{x}_m) - \dot{x}_m] \leq 0. \tag{3.23}$$

which implies that e is bounded. Because r is bounded by definition above, x_m and $\dot{x}_m$ are bounded. Assuming that $\dot{r}$ is bounded as well, then $\ddot{x}_m$ is bounded. Since e and x_m are bounded, x is bounded. Consequently, $\beta(x; d)$ is bounded as well. Also, $\widetilde{W}$ is bounded. In order to use Barbalat's lemma to complete the proof, the second derivative of equation (3.14) along the system trajectories is taken, which gives the result

$$\ddot{V} = -4e^T K\dot{e} - 2(\text{Proj}(x, \dot{x}_m) - \dot{x}_m)^T\dot{e} - 2e^T\frac{d}{dt}(\text{Proj}(x, \dot{x}_m)) - 2e^T\ddot{x}_m. \tag{3.24}$$

Substituting equation (3.13) into equation (3.24) produces

$$\begin{aligned}\ddot{V} = &-4e^T K\left(-Ke - \widetilde{W}^T\beta(x; d) - [\text{Proj}(x, \dot{x}_m) - \dot{x}_m]\right)\\ &- 2(\text{Proj}(x, \dot{x}_m) - \dot{x}_m)^T\left(-Ke - \widetilde{W}^T\beta(x; d) - [\text{Proj}(x, \dot{x}_m) - \dot{x}_m]\right)\\ &- 2e^T\frac{d}{dt}(\text{Proj}(x, \dot{x}_m)) - 2e^T\ddot{x}_m.\end{aligned} \tag{3.25}$$

Consider the first element from the projection operator vector defined in equation (3.10). When the projection operator is active, the equation for the first element can be written as

$$\text{Proj}(x_1, \dot{x}_{m_1}) = \dot{x}_{m_1}(1 - h(x_1)). \tag{3.26}$$

By the definition of the projection operator, $0 < h(x_1) \leq 1$ when the projection operator is active. Therefore, $\text{Proj}(x_1, \dot{x}_{m_1})$ is bounded, and by extension, the vector $\text{Proj}(x, \dot{x}_m)$ given in equation (3.10) is bounded as well.

The time derivative of the projection operator also must be analyzed. Again, considering the first element from the projection operator vector defined in equation (3.10), the time derivative of this element is determined to be

$$\frac{d}{dt}(\text{Proj}(x_1, \dot{x}_{m_1})) = \ddot{x}_{m_1} - \ddot{x}_{m_1} h(x_1) - \dot{x}_{m_1} \dot{h}(x_1). \tag{3.27}$$

$\dot{x}_{m_1}$, $\ddot{x}_{m_1}$, and $h(x_1)$ are bounded as proved above. To prove that $\dot{h}(x_1)$ is bounded, the time derivative of $h(x_1)$, where $h(x_1)$ is defined by equation (3.2), is calculated as

$$\dot{h}(x_1) = \frac{dh}{dx_1} \frac{dx_1}{dt} = \frac{2\left(\epsilon_{x_1} + 1\right) x_1}{\epsilon_{x_1} x_{max_1}^2} \dot{x}_1. \tag{3.28}$$

x_1 is bounded as shown above, and ϵ_{x_1} and $x_{max_1}^2$ are constants. By examining equations (3.3) and (3.9) in light of the analysis above, $\dot{x}_1$ is bounded as well, which means that the term $\frac{d}{dt}(\text{Proj}(x_1, \dot{x}_{m_1}))$ is bounded, and by extension, $\frac{d}{dt}(\text{Proj}(x, \dot{x}_m))$ is bounded as well. Therefore, all of the terms in $\ddot{V}$ in equation (3.25) are bounded, which proves that $\dot{V}$ is uniformly continuous.

Following the analysis of the case where the projection operator was not active, because V is lower bounded, $\dot{V}$ is negative semi-definite, and $\dot{V}$ is uniformly continuous, by Barbalat's lemma $\dot{V} \to 0$ as $t \to \infty$, and thus $e \to 0$ as $t \to \infty$ as desired.

3.3 Application of Adaptive Control with State Constraints to Hypersonic Vehicles

This section defines the alterations to the nonlinear adaptive dynamic inversion control architecture for hypersonic vehicles described in Chapter 2. Because the focus of this dissertation is on preventing inlet unstarts that occur in hypersonic vehicles when the flow to the inlet does not pass through the throat of the engine, state constraints on α and β must be enforced. The control law that is directly affected by the enforcement of state constraints is the law associated with the α, β, μ inversion controller. From equation (3.9), the new control law for α, β, μ inversion that enforces the state constraints on α and β is

$$u = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} S_\alpha & 0 & -C_\alpha \\ -T_\beta C_\alpha & 1 & -T_\beta S_\alpha \\ \sec(\beta) C_\alpha & 0 & \sec(\beta) S_\alpha \end{bmatrix}^{-1} \left[\begin{bmatrix} \mathrm{Proj}(\beta, \dot{\beta}_m) \\ \mathrm{Proj}(\alpha, \dot{\alpha}_m) \\ \dot{\mu}_m \end{bmatrix} - \hat{f}(x) + Ke - \nu \right] \quad (3.29)$$

where $\hat{f}(x)$, K, and ν have the same meanings as they did in Chapter 2. If the state constraints are not exceeded, then the control law u in equation (3.29) will have the same form as in Chapter 2. However, if the state constraints are within a certain tolerance of being violated, indicating that the vehicle is nearing an inlet unstart event, then the control law u provides the p, q, and r values necessary to prevent the state constraints from being exceeded.

3.4 Simulation Results

The control law u given in equation (3.29) was implemented in the GHV Simulink model. For the first simulation, the limits on α and β were specified to be ±8 degrees and ±4 degrees, respectively, at Mach 6. The angle-of-attack of the vehicle was commanded to 7.9 degrees, which is within the state constraints; however, an additive uncertainty of 0.0007 deg^{-1} was appended to C_{m_α}. Figures 3.2, 3.3, 3.4, and 3.5 depict the results using the nonlinear adaptive dynamic inversion control architecture without state constraints. When the system approached the commanded value of α, the response overshot the α value and exceeded the state constraints. Figures 3.6, 3.7, 3.8, and 3.9 show the results using the same control architecture with the addition of state constraints. As seen in Figure 3.6, the value of α approached the maximum state constraint of 8 degrees; however, the revised control law was able to prevent the system from violating the constraints. These figures demonstrate that the control law u given in equation (3.29) is able to prevent the system from exceeding specified state constraints.

For the second simulation, the state constraints on α and β both were specified to be ±4 degrees. The angle-of-attack α was commanded as a doublet with limits at ±3.9 degrees, and the sideslip angle β was commanded to be 3.9 degrees. The additive parametric uncertainties in the system were specified to be $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1}. Figures 3.10, 3.11, 3.12, and 3.13 contain the results from using the nonlinear adaptive dynamic inversion control architecture with state constraints. The figures show that the revised control architecture was able to handle systems in which α and β state constraints were approached simultaneously. The control law also was able to limit the overshoot as a result of parametric uncertainties in the presence of a time delay of 0.02 seconds. However, the system response in the presence of a time delay and state constraints was more oscillatory than the response without a time delay.

It should be noted that the state constraint on α was reduced for the second simulation. While the nonlinear adaptive dynamic inversion architecture with state constraints was able to ensure that the constraint on α was not violated for the orig-

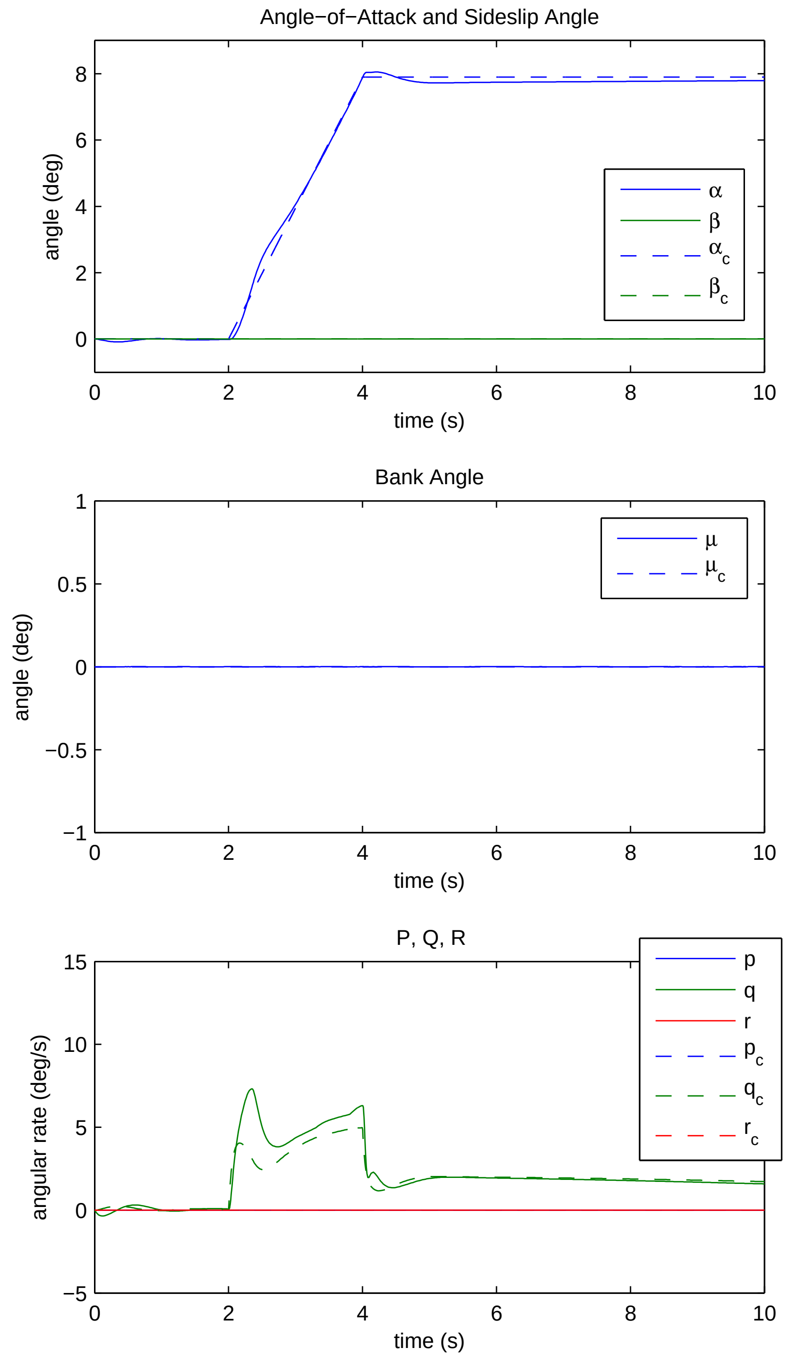


Figure 3.2: State responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} without state constraints.

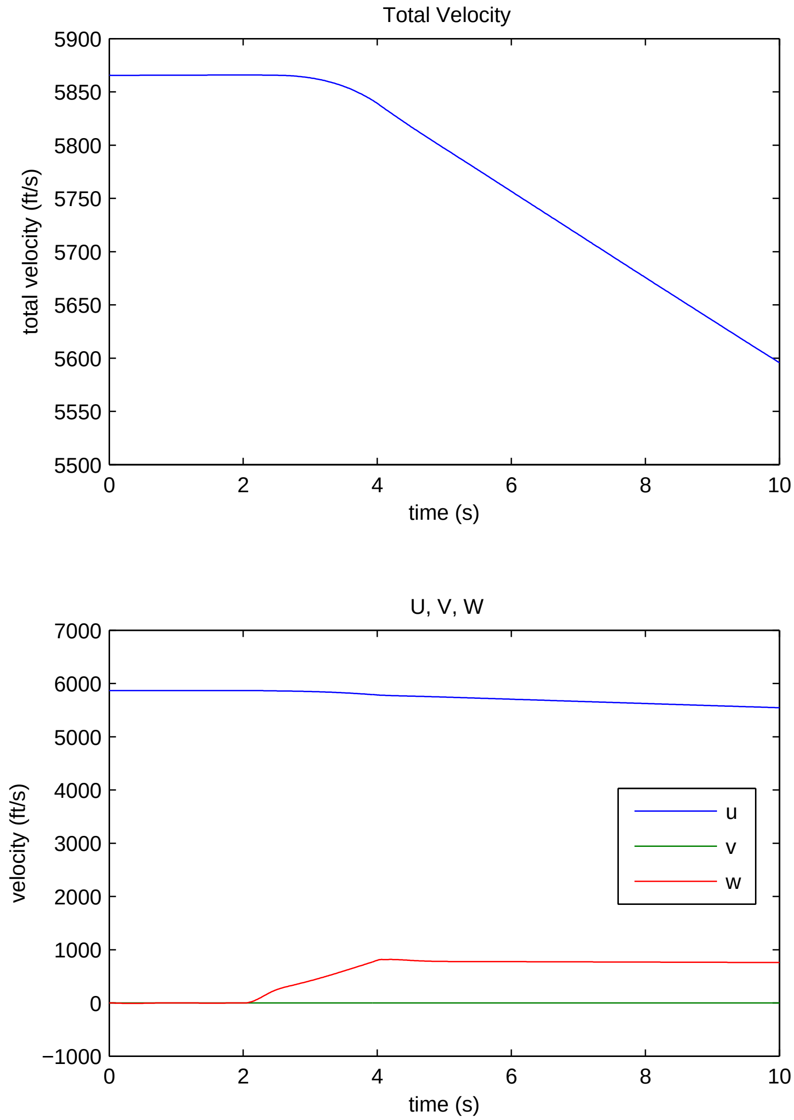


Figure 3.3: Velocity responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} without state constraints.

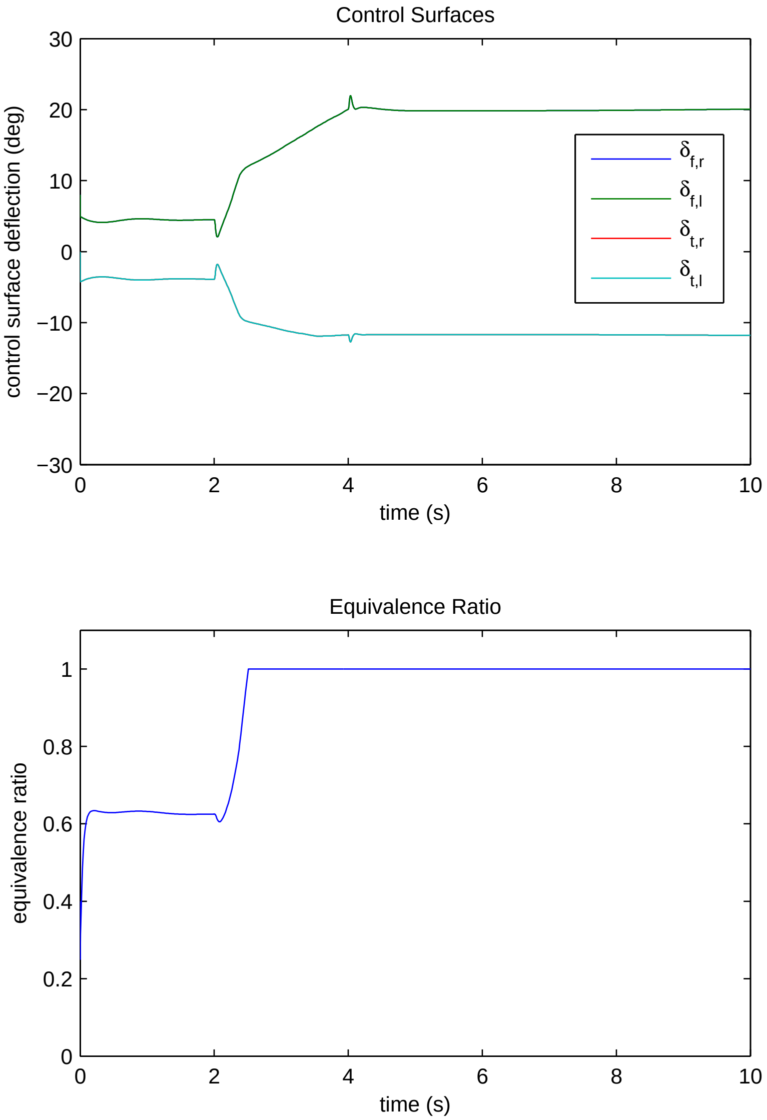


Figure 3.4: Control responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} without state constraints.

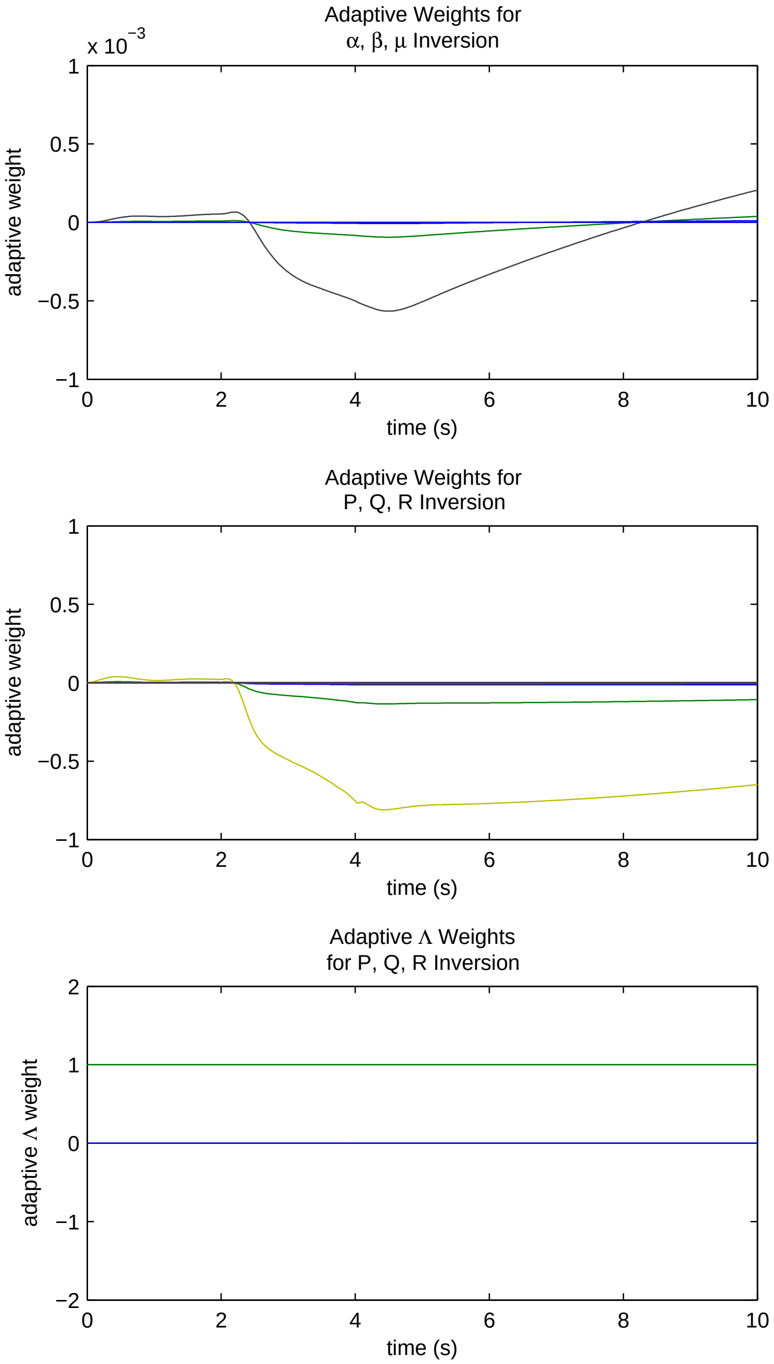


Figure 3.5: Adaptive weight responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} without state constraints.

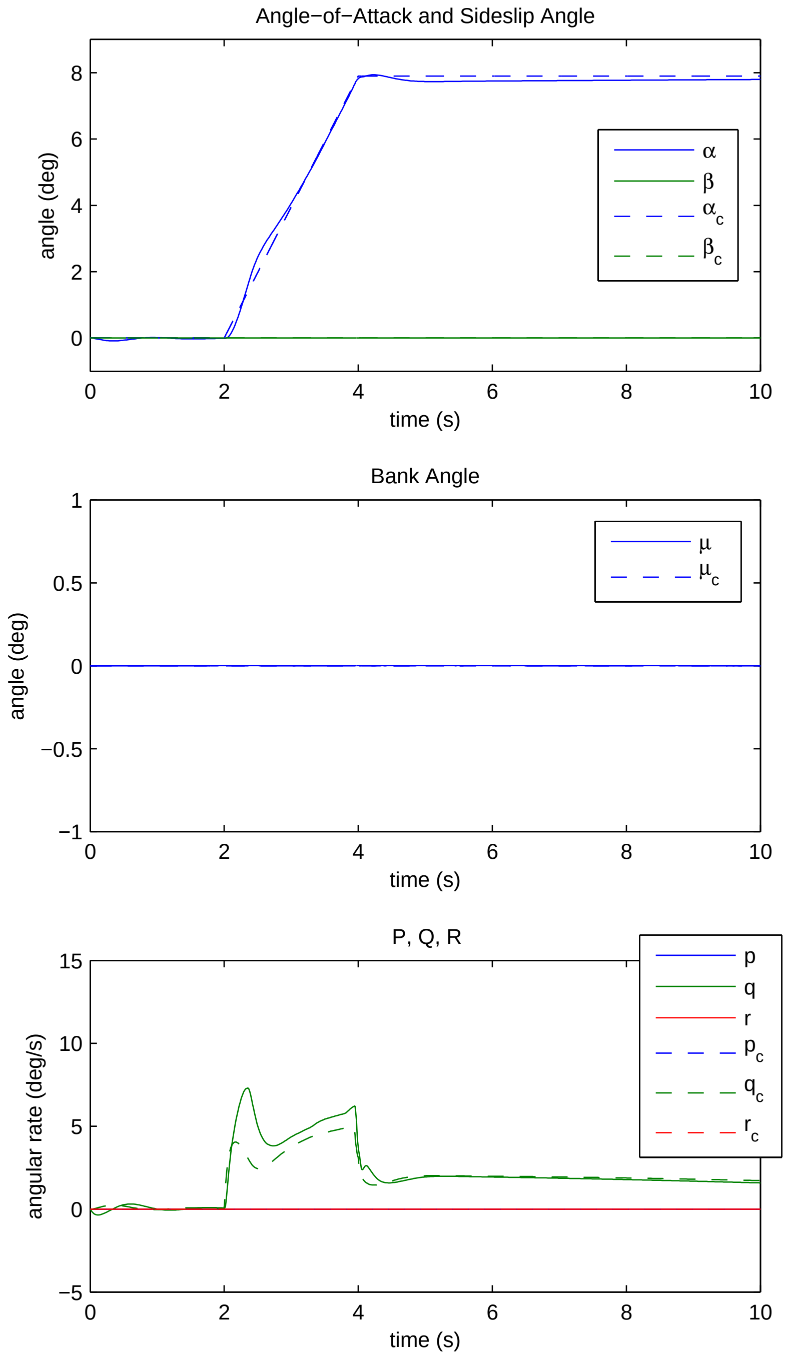


Figure 3.6: State responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} with state constraints.

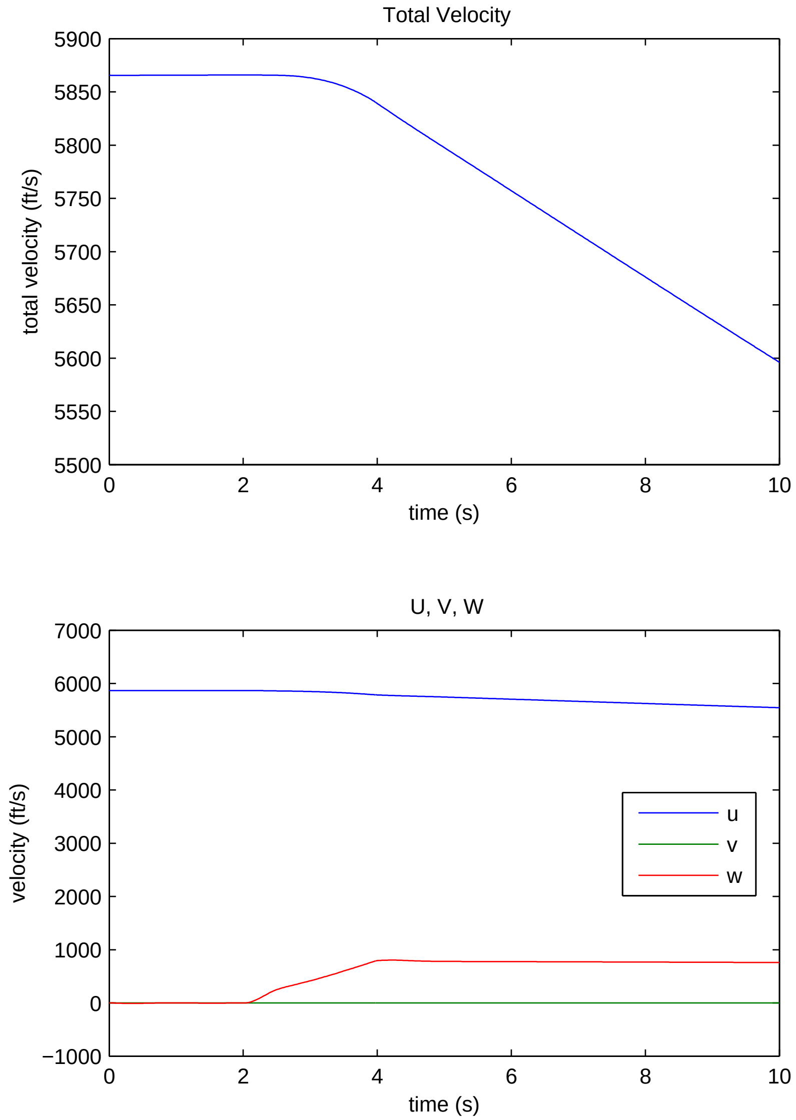


Figure 3.7: Velocity responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} with state constraints.

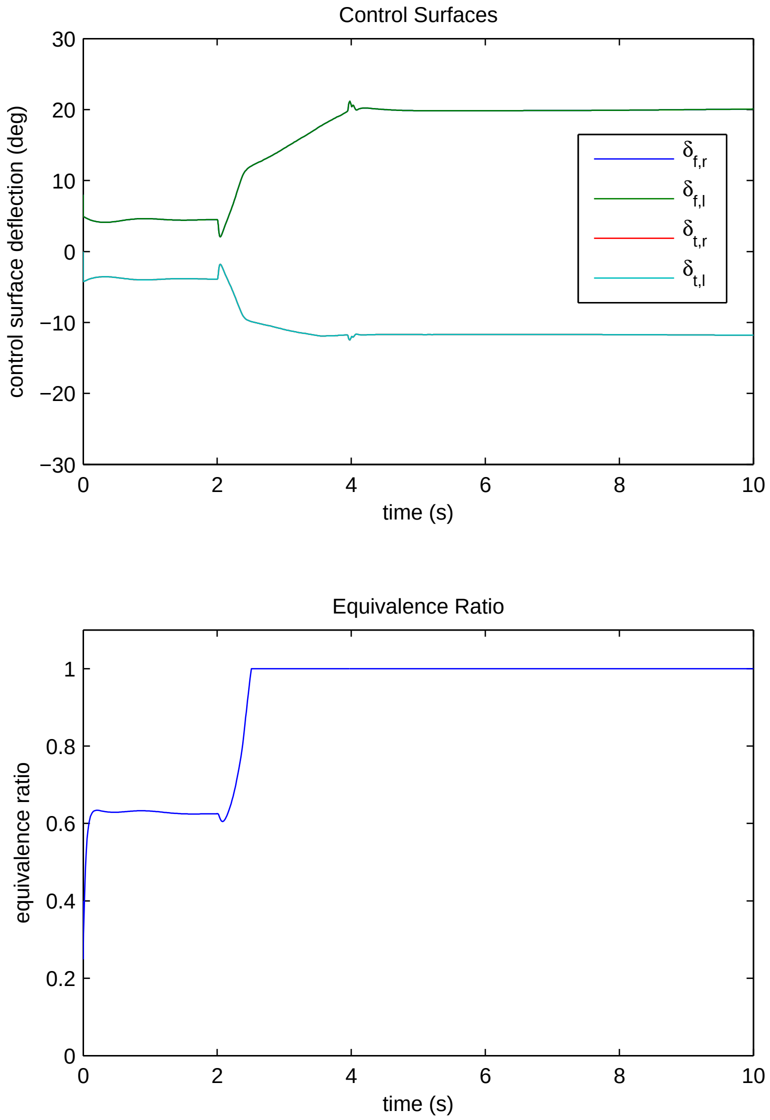


Figure 3.8: Control responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} with state constraints.

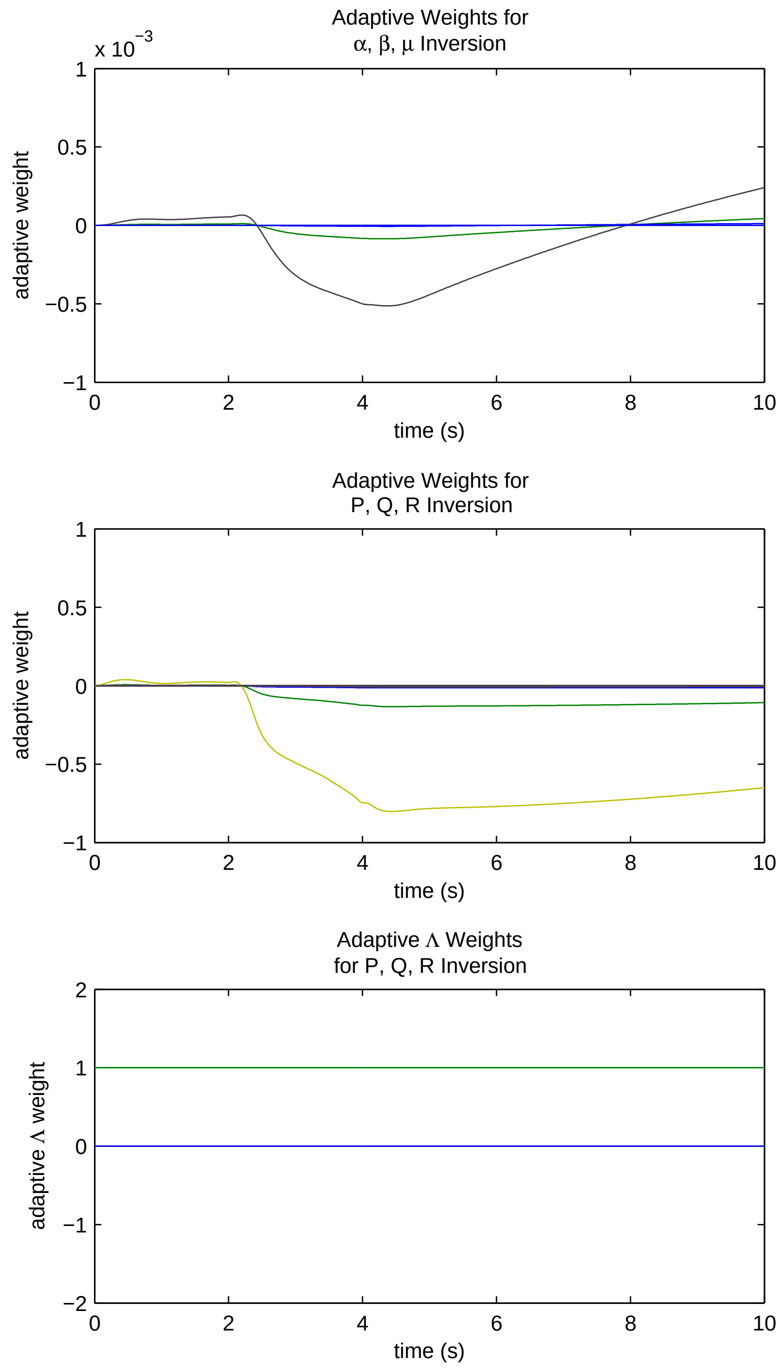


Figure 3.9: Adaptive weight responses for the commands $\alpha = 7.9$ deg, $\beta = 0$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} with state constraints.

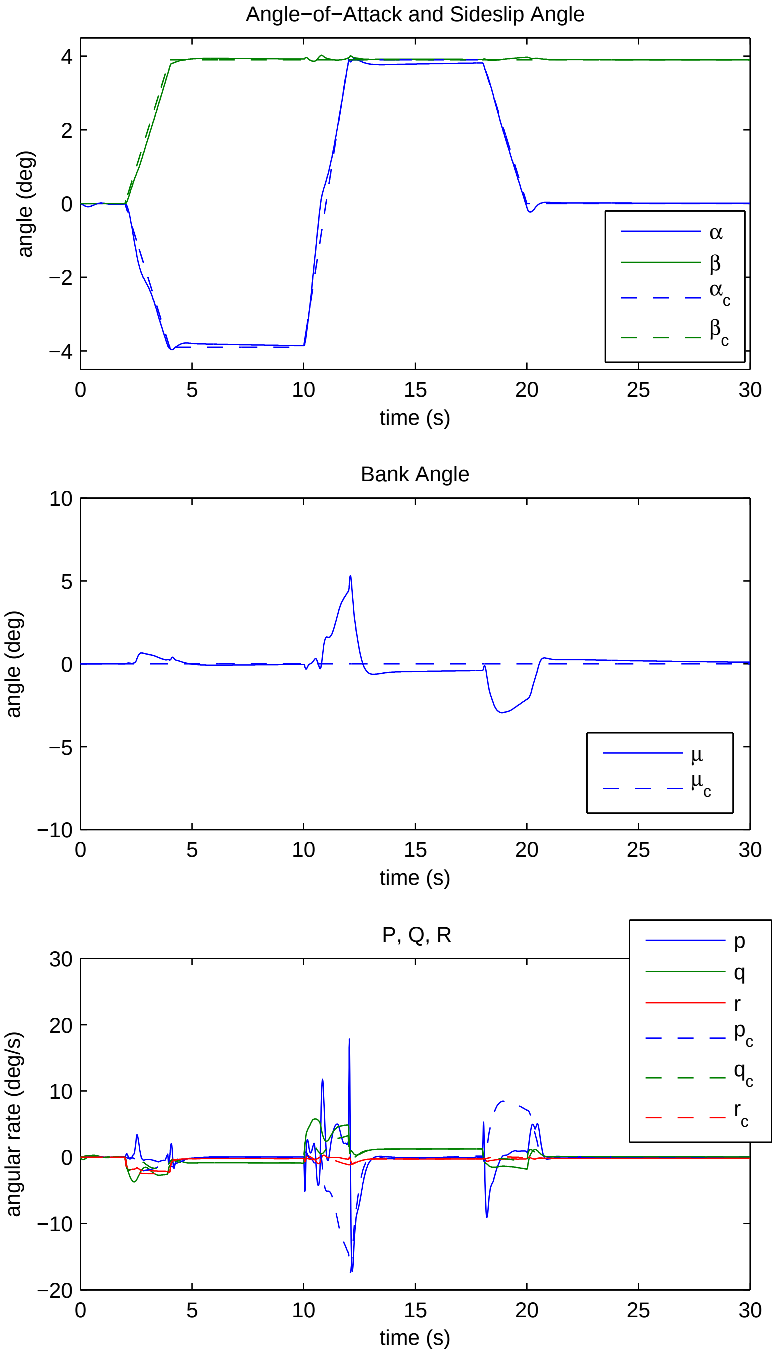


Figure 3.10: State responses for the commands $\alpha = \pm 3.9$ deg, $\beta = 3.9$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

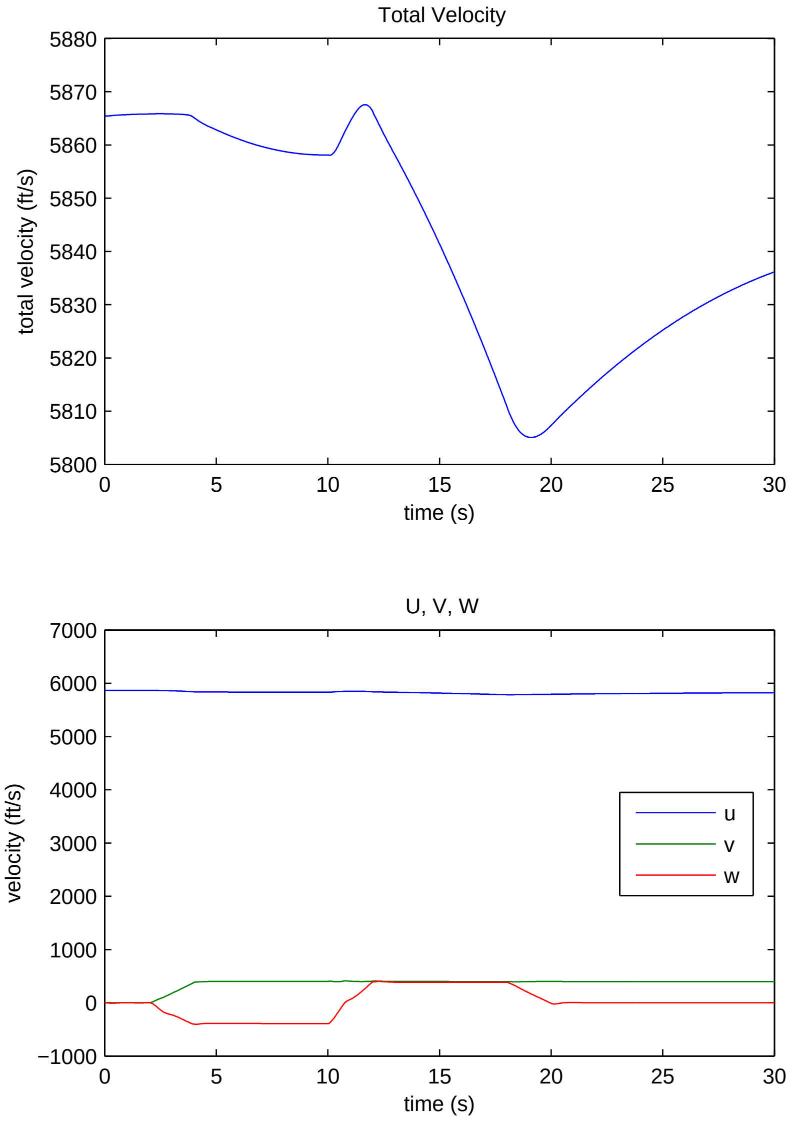


Figure 3.11: Velocity responses for the commands $\alpha = \pm 3.9$ deg, $\beta = 3.9$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

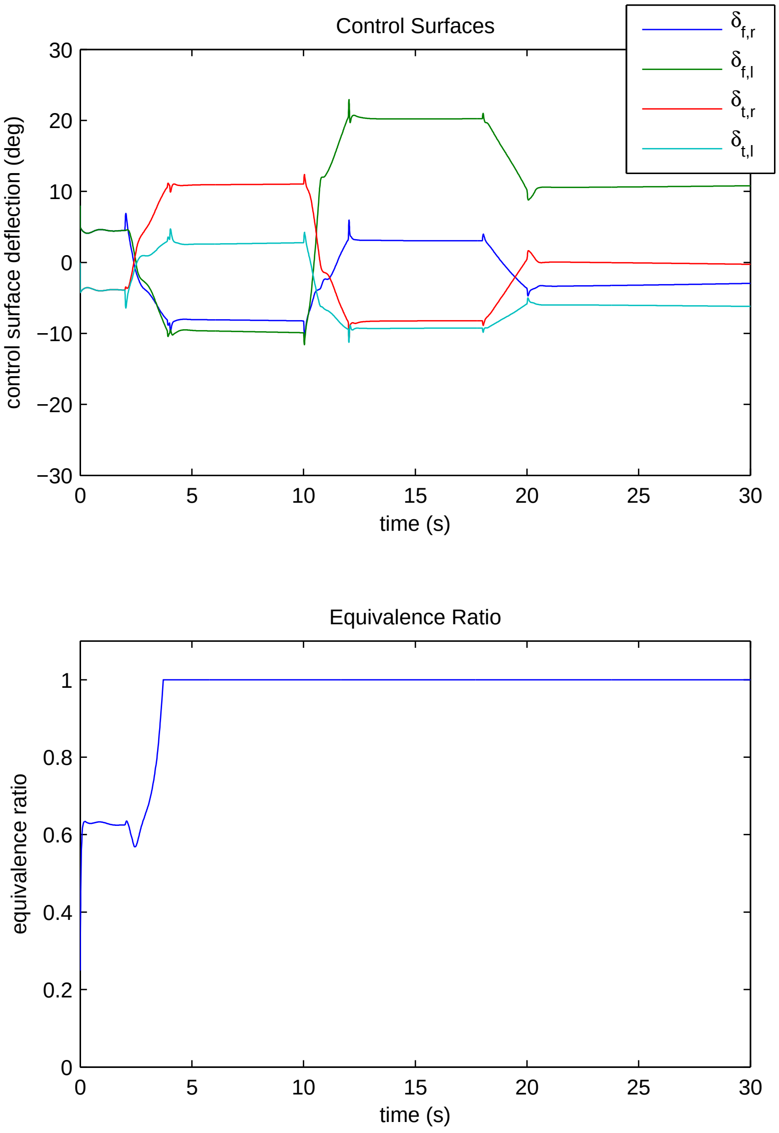


Figure 3.12: Control responses for the commands $\alpha = \pm 3.9$ deg, $\beta = 3.9$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

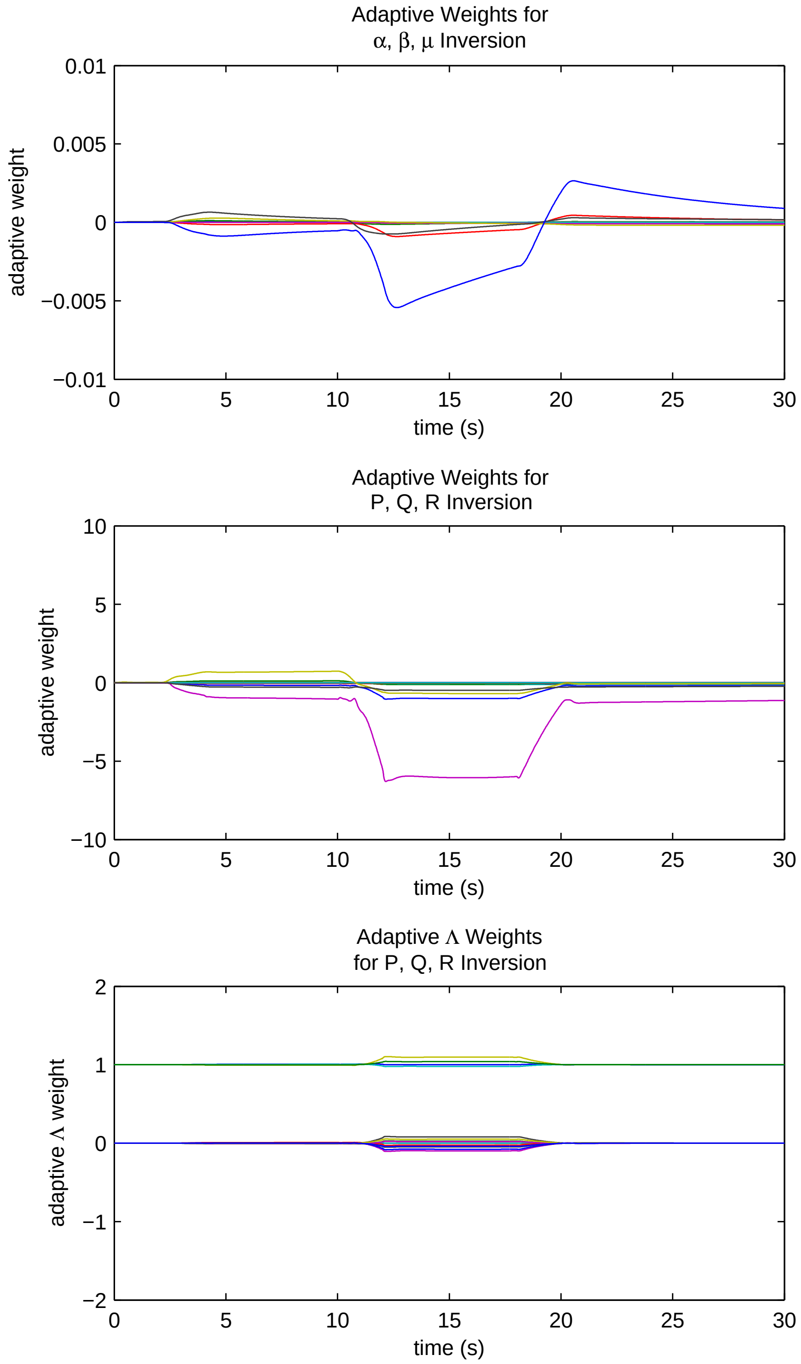


Figure 3.13: Adaptive weight responses for the commands $\alpha = \pm 3.9$ deg, $\beta = 3.9$ deg, and $\mu = 0$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

inal limits on α of ±8 degrees, the state constraint on β of ±4 degrees was exceeded beyond the error tolerance envelope. It is hypothesized that the violation of the β constraint may be linked to the unique control structure of the GHV with its elevons and ruddervators, which leads to coupling in the longitudinal and lateral-directional system response. Figures 3.10 and 3.12 illustrate this coupling, where the commands of the control surfaces to limit α and β induce an aerodynamic bank angle μ, causing μ to deviate from its commanded value of 0 degrees. This coupling may be causing β to exceed its state constraint limits since control surface deflections designed to prevent α from exceeding state constraints also affect the lateral-directional states. Additionally, control surface saturation may be affecting the ability of the control law to prevent state constraints from being violated.

The third simulation highlights one of the limitations of this control architecture for enforcing state constraints. Figures 3.14, 3.15, 3.16, and 3.17 depict the results of a bank angle doublet of ±50 deg with the $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1}. Because the variations in α and β during this maneuver were small, the limits on α and β both were set to ±0.075 deg. In Figure 3.14, α clearly violates the state constraint of 0.075 deg during the doublet maneuver. When α exceeds its limit, the time derivative of the reference trajectory, $\dot{\alpha}_m$ is equal to 0. As a result, the second term of the projection operator, as defined in equation (3.1) becomes 0, and the control law from equation (3.29) does not prevent the states from exceeding their limits.

3.5 Conclusions

The ability to constrain states such as the angle-of-attack and the sideslip angle is critical to preventing inlet unstarts in hypersonic vehicles. In this chapter, it was demonstrated that the control law that was derived in Chapter 2 could be altered with

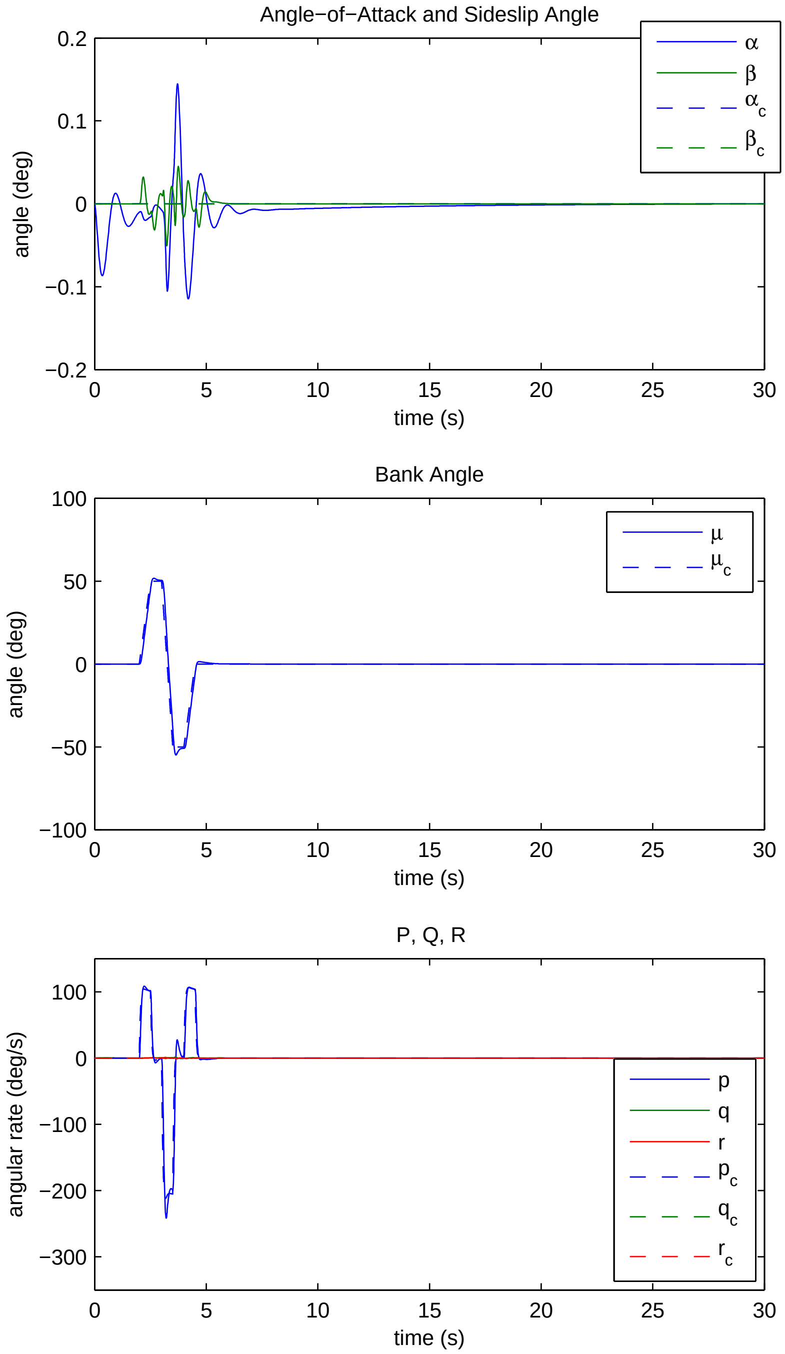


Figure 3.14: State responses for the commands $\alpha = 0$ deg, $\beta = 0$ deg, and $\mu = \pm 50$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

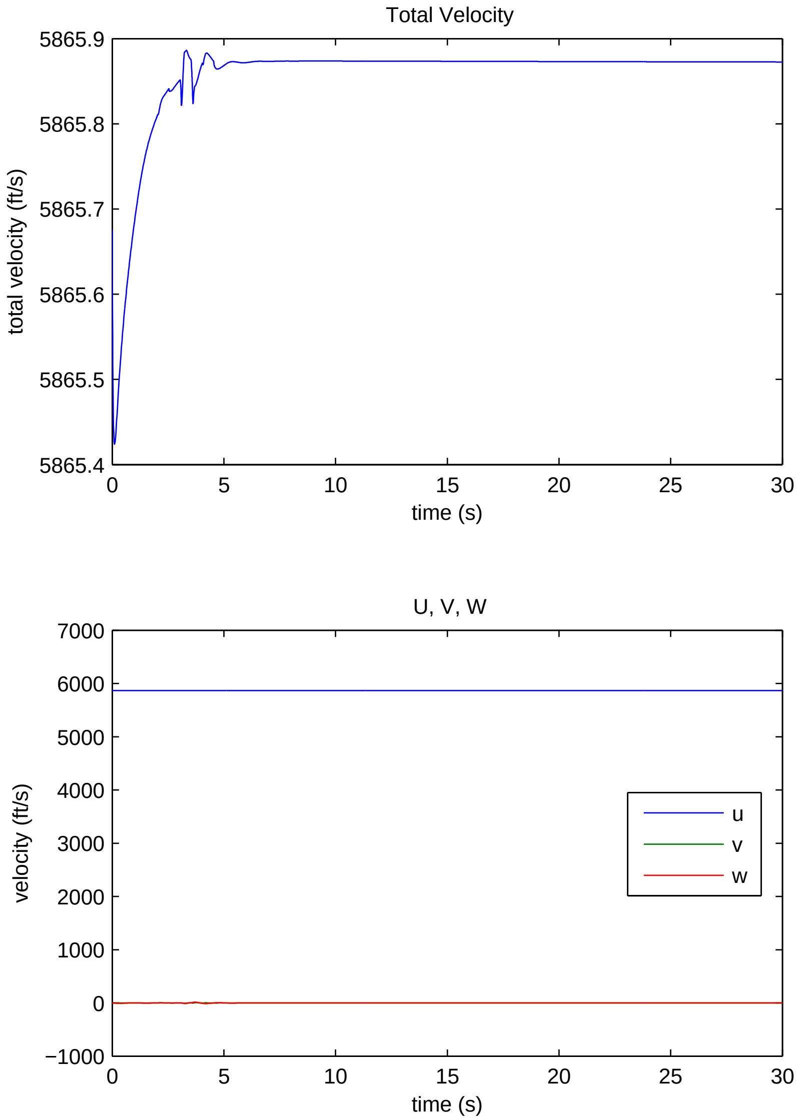


Figure 3.15: Velocity responses for the commands $\alpha = 0$ deg, $\beta = 0$ deg, and $\mu = \pm 50$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

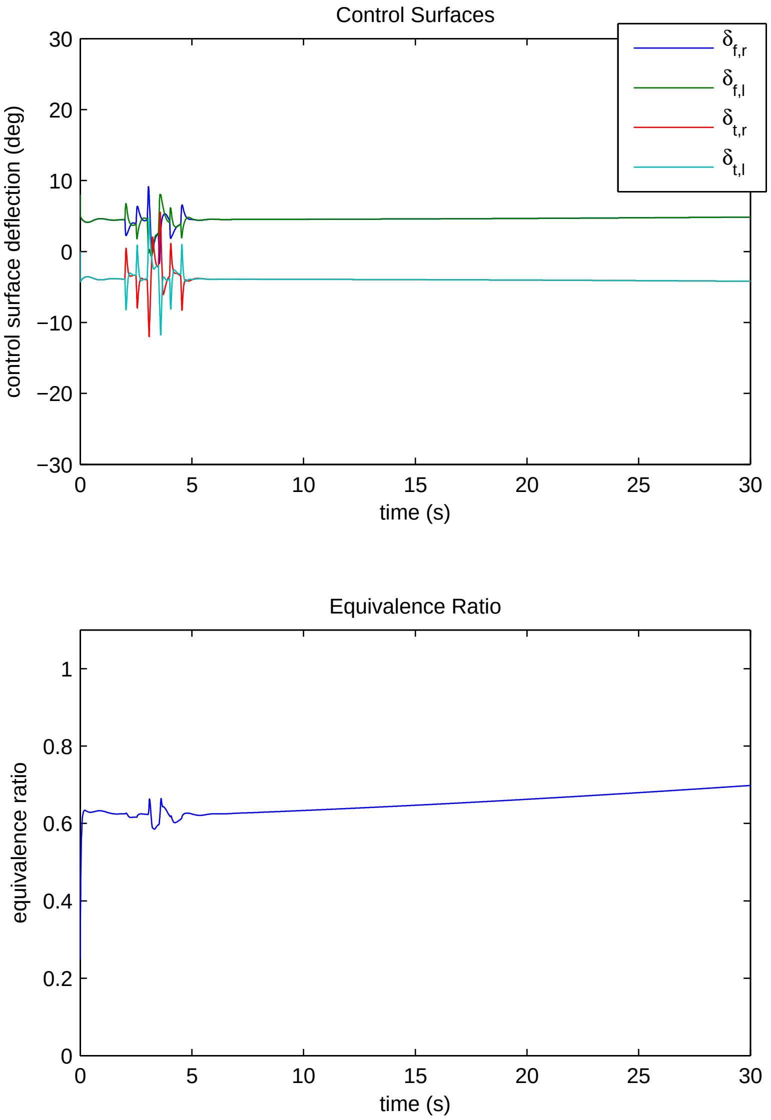


Figure 3.16: Control responses for the commands $\alpha = 0$ deg, $\beta = 0$ deg, and $\mu = \pm 50$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

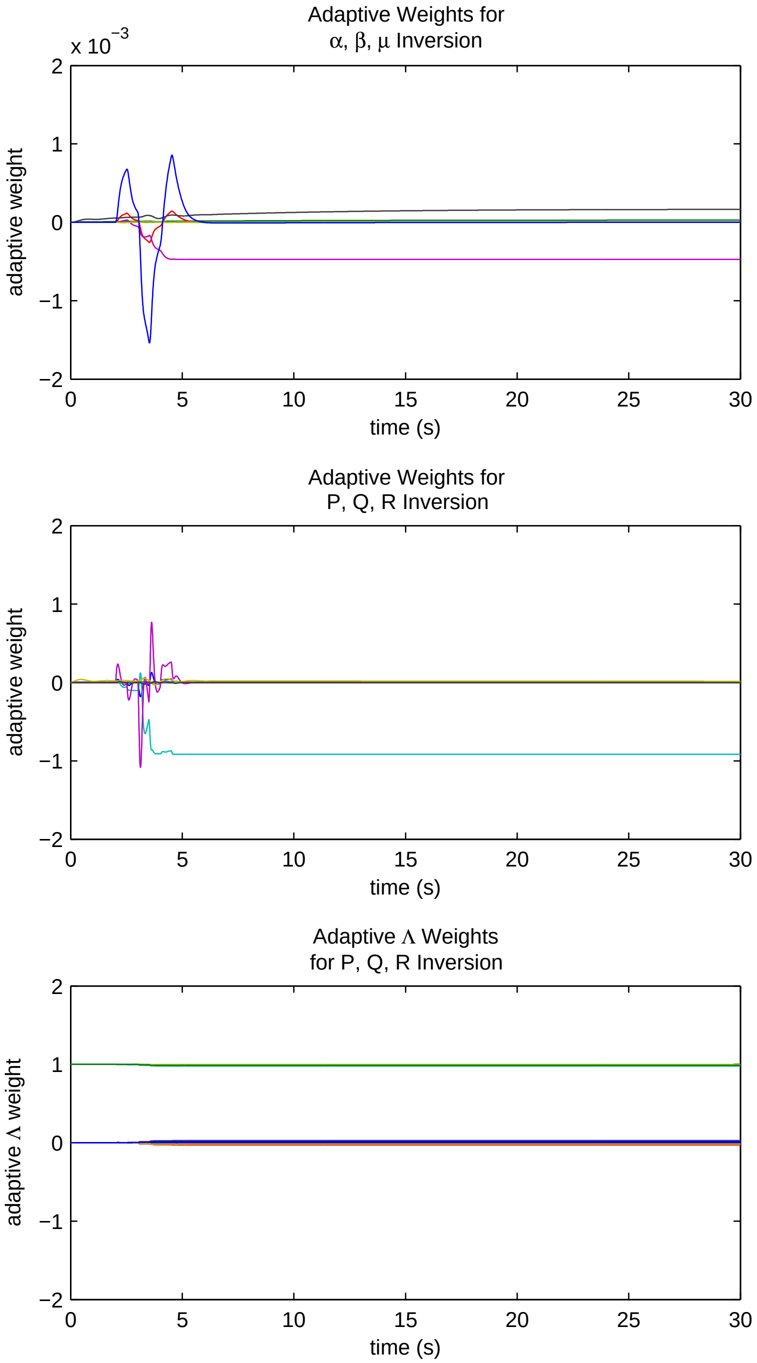


Figure 3.17: Adaptive weight responses for the commands $\alpha = 0$ deg, $\beta = 0$ deg, and $\mu = \pm 50$ deg with $\Delta C_{m_\alpha} = 0.0007$ deg^{-1} and $\Delta C_{n_\beta} = 0.0007$ deg^{-1} with state constraints.

the addition of a projection operator to adjust the time derivative of the reference model in order to successfully constrain certain states to specified limits. The nonlinear adaptive dynamic inversion control architecture that enforces state constraints was able to handle various parametric uncertainties in C_{m_α} and C_{n_β}. However, it should be noted that restrictions exist on the types of reference trajectories on which this control architecture will be able to enforce state constraints.

4. FAULT-TOLERANT NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL

Maintaining control of the aircraft following the failure of a control surface is a significant concern that should be addressed by the control architecture for the vehicle. During the third flight of the X-51A Waverider, a fault in one of the control fins led to the loss of the vehicle [3]. To mitigate the loss of control surfaces on hypersonic vehicles in this disseration, the nonlinear adaptive dynamic inversion control architecture introduced in Chapter 1 is modified to include the fault-tolerant control architecture developed by Tandale and Valasek [32]. This chapter describes the model for the applied control to an aircraft and the modified nonlinear adaptive dynamic inversion control architecture with fault tolerance, followed by simulation results.

4.1 Model for the Applied Control

In order to simulate control faults, a model must be constructed to represent how the calculated control signal is altered by faults to become the actual control signal that is applied to the plant. The model for the control that is applied to a given plant can be expressed as

$$u_{app} = Du_{calc} + E, \tag{4.1}$$

where $u_{app} \in \mathbb{R}^m$ is the applied control, $u_{calc} \in \mathbb{R}^m$ is the calculated control signal, $D \in \mathbb{R}^{m\times m}$ is a diagonal matrix, and $E \in \mathbb{R}^m$. For a system in which all control surfaces are operating normally, the matrix D in equation (4.1) is equal to the identity matrix, and the vector E is a vector of zeros. In order to simulate the complete failure of a control surface, such as a float condition in which the control surface is nonresponsive, the term on the diagonal in the matrix D that corresponds to the failed control surface is set equal to zero, which prevents any of the calculated commands for that control surface from being executed. Further descriptions of simulated control failures using the model in equation (4.1) can be found in [32].

4.2 General Adaptive Control Equations with Fault Tolerance

Consider a general nonlinear equation of a system in the form

$$\dot{x} = f(x) + g(x)\Lambda u \tag{4.2}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control, $f(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times m}$ are locally Lipschitz continuous, and $\Lambda \in \mathbb{R}^{m\times m}$ is a constant unknown positive definite matrix. It is assumed that $m > n$ and that $g(x)$ is full rank for all $x \in \mathbb{R}^n$. Also, it is assumed for this derivation that $\Lambda = I$, the identity matrix, which is a reasonable assumption based on the simulation results in Chapter 2. Suppose that the desired reference dynamics for the system are given by

$$\dot{x}_m = Ax_m + Br \tag{4.3}$$

where $x_m \in \mathbb{R}^n$ is the model state, $r \in \mathbb{R}^n$ is a bounded reference signal, $A \in \mathbb{R}^{n\times n}$ is Hurwitz, and $B \in \mathbb{R}^{n\times n}$. The equation for the error between the reference model and the actual system is

$$e = x_m - x. \tag{4.4}$$

Taking the time derivative of equation (4.4) results in

$$\dot{e} = \dot{x}_m - \dot{x} = \dot{x}_m - f(x) - g(x)u. \tag{4.5}$$

Substituting the model for the applied control given in equation (4.1) into equation (4.5) leads to the equation

$$\dot{e} = \dot{x}_m - f(x) - g(x)Du_{calc} - g(x)E. \tag{4.6}$$

The desired error dynamics have the form

$$\dot{e} = -Ke \tag{4.7}$$

where $K \in \mathbb{R}^{n \times n}$ such that $K = K^T > 0$ are the gains on the tracking errors. The term Ke is added and subtracted from equation (4.6) to produce

$$\dot{e} = \dot{x}_m - f(x) - g(x)Du_{calc} - g(x)E - Ke + Ke. \tag{4.8}$$

The control u_{calc} is chosen to be

$$u = \text{pinv}(g(x)\widehat{D})[\dot{x}_m - \hat{f}(x) - g(x)\widehat{E} + Ke - \nu] \tag{4.9}$$

where

$$\text{pinv}(g(x)\widehat{D}) = (g(x)\widehat{D})^T \left[g(x)\widehat{D}(g(x)\widehat{D})^T\right]^{-1} \tag{4.10}$$

from [32], $\widehat{D}$ is an estimate of the actual D matrix, $\widehat{E}$ is an estimate of the actual E vector, $\hat{f}(x) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a model of the plant dynamics, and $\nu \in \mathbb{R}^n$ is a pseudo-control signal. The terms in equations (4.8) and (4.9) that are definitively known can be represented as

$$\psi = \dot{x}_m + Ke, \tag{4.11}$$

which means that those equations can be rewritten in terms of ψ as

$$\dot{e} = -Ke - f(x) - g(x)Du_{calc} - g(x)E + \psi \tag{4.12}$$

and

$$u = \mathrm{pinv}(g(x)\widehat{D})[\psi - \hat{f}(x) - g(x)\widehat{E} - \nu]. \tag{4.13}$$

Solving equation (4.13) for ψ produces the equation

$$\psi = \hat{f}(x) + g(x)\widehat{E} + \nu + g(x)\widehat{D}u_{calc}. \tag{4.14}$$

Substituting equation (4.14) into equation (4.12) produces the error dynamics

$$\dot{e} = -Ke - f(x) - g(x)Du_{calc} - g(x)E + \hat{f}(x) + g(x)\widehat{E} + \nu + g(x)\widehat{D}u_{calc}. \tag{4.15}$$

Defining the error between the model and the actual system as $\Delta = \hat{f}(x) - f(x)$, and rearranging terms, equation (4.15) becomes

$$\dot{e} = -Ke + \Delta + \nu - g(x)Du_{calc} + g(x)\widehat{D}u_{calc} - g(x)E + g(x)\widehat{E}. \tag{4.16}$$

Again, in this dissertation, it is assumed that Δ can be represented in the form $\Delta = W^T\beta(x;d)$, where $W \in \mathbb{R}^{p\times n}$ is a set of unknown weights, and $\beta \in \mathbb{R}^{p\times 1}$ is a set of known basis functions composed of the states x and a vector d of bounded continuous exogenous inputs. Using this representation for Δ, ν is chosen to be $\nu = -\widehat{W}^T\beta(x;d)$, where $\widehat{W} \in \mathbb{R}^{p\times n}$. With these definitions, equation (4.16) can be written as

$$\dot{e} = -Ke - \widetilde{W}^T\beta(x;d) + g(x)\widetilde{D}u_{calc} + g(x)\widetilde{E} \tag{4.17}$$

where $\widetilde{W} = \widehat{W} - W$ is the weight estimation error, $\widetilde{D} = \widehat{D} - D$ is the estimation error for the matrix D, and $\widetilde{E} = \widehat{E} - E$ is the estimation error for the vector E.

The stability of the closed loop system under these assumptions can be examined using a candidate Lyapunov function of the form

$$V = e^T e + tr(\widetilde{W}^T {\Gamma_W}^{-1} \widetilde{W} + \widetilde{D}^T {\Gamma_D}^{-1} \widetilde{D}) + \widetilde{E}^T {\Gamma_E}^{-1} \widetilde{E} \tag{4.18}$$

where $\Gamma_W \in \mathbb{R}^{p \times p}$ with $\Gamma_W = {\Gamma_W}^T > 0$, $\Gamma_D \in \mathbb{R}^{n \times n}$ with $\Gamma_D = {\Gamma_D}^T > 0$, and $\Gamma_E \in \mathbb{R}^{n \times n}$ with $\Gamma_E = {\Gamma_E}^T > 0$. In order to determine the adaptation laws for the parameters in W, D, and E and to determine if the error between the states of the actual system and the reference model will converge, first, the derivative of equation (4.18) along the system trajectories is taken, which gives the result

$$\dot{V} = 2e^T \dot{e} + 2tr(\widetilde{W}^T {\Gamma_W}^{-1} \dot{\widehat{W}}^T + \widetilde{D}^T {\Gamma_D}^{-1} \dot{\widehat{D}}^T) + 2\widetilde{E}^T {\Gamma_E}^{-1} \dot{\widehat{E}}^T. \tag{4.19}$$

Substituting equation (4.17) into equation (4.19) produces

$$\begin{aligned} \dot{V} = &-2e^T K e - 2e^T \widetilde{W}^T \beta(x; d) + 2e^T g(x) \widetilde{D} u_{calc} + 2e^T g(x) \widetilde{E} \\ &+ 2tr(\widetilde{W}^T {\Gamma_W}^{-1} \dot{\widehat{W}}^T + \widetilde{D}^T {\Gamma_D}^{-1} \dot{\widehat{D}}^T) + 2\widetilde{E}^T {\Gamma_E}^{-1} \dot{\widehat{E}}^T. \end{aligned} \tag{4.20}$$

Applying the trace identity that $a^T b = tr(ba^T)$, equation (4.20) is determined to be

$$\begin{aligned} \dot{V} = &-2e^T K e + 2tr(\widetilde{W}^T ({\Gamma_W}^{-1} \dot{\widehat{W}}^T - \beta(x; d) e^T)) \\ &+ 2tr(\widetilde{D}^T ({\Gamma_D}^{-1} \dot{\widehat{D}}^T + u_{calc} e^T g(x))) + 2(\widetilde{E} {\Gamma_E}^{-1} \dot{\widehat{E}}^T + e^T g(x) \widetilde{E}). \end{aligned} \tag{4.21}$$

Then, by choosing $\dot{\widehat{W}}$ as

$$\dot{\widehat{W}} = \Gamma_W \,\mathrm{Proj}(\widehat{W}, \beta(x;d)e^T), \tag{4.22}$$

$\dot{\widehat{D}}$ as

$$\dot{\widehat{D}} = \Gamma_D \,\mathrm{Proj}(\widehat{D}, -u_{calc}e^T g(x)), \tag{4.23}$$

and $\dot{\widehat{E}}$ as

$$\dot{\widehat{E}} = \Gamma_E \,\mathrm{Proj}(\widehat{E}, -e^T g(x)), \tag{4.24}$$

$\dot{V}$ can be upper bounded as

$$\dot{V} \leq -2e^T K e \leq 0 \tag{4.25}$$

which implies that e is bounded. Because r is bounded by definition above, x_m is bounded. Since e and x_m are bounded, x is bounded. Consequently, $\beta(x;d)$ and $g(x)$ are bounded as well. The term u_{calc} also is bounded since all of the terms in equation (4.9) are bounded. In order to use Barbalat's lemma to complete the proof, the second derivative of equation (4.18) along the system trajectories is taken, which gives the result

$$\ddot{V} = -4e^T K\dot{e}. \tag{4.26}$$

Substituting equation (4.17) into equation (4.26) produces

$$\ddot{V} = -4e^T K(-Ke - \widetilde{W}^T\beta(x;d) + g(x)\widetilde{D}u_{calc} + g(x)\widetilde{E}). \tag{4.27}$$

Because e, $\widetilde{W}$, $\widetilde{D}$, $\widetilde{E}$, $\beta(x;d)$, $g(x)$, and u_{calc} are bounded as proved above, $\ddot{V}$ is bounded, and therefore $\dot{V}$ is uniformly continuous.

Because V is lower bounded, $\dot{V}$ is negative semi-definite, and $\dot{V}$ is uniformly continuous, by Barbalat's lemma $\dot{V} \to 0$ as $t \to \infty$, and thus $e \to 0$ as $t \to \infty$ as desired.

4.3 Simulation Results

In order to implement the fault-tolerant nonlinear adaptive dynamic inversion control architecture in the GHV simulation, the control law for the p, q, r inversion controller defined in Chapter 2 is replaced by the control law given in equation (4.9), and the adaptive laws for the controller from Chapter 2 are replaced by the adaptive laws given in equations (4.22), (4.23), and (4.24). The α, β, μ inversion controller remains unaltered from its form in Chapter 2.

Figures 4.1, 4.2, 4.3, 4.4, and 4.5 depict the simulation results with the fault-tolerant nonlinear adaptive dynamic inversion control architecture for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg with the failure of the left elevon at 7 seconds. While the state responses in Figure 4.1 are slightly oscillatory, the control architecture is able to detect that a fault has occurred at 7 seconds and to compensate accordingly to maintain tracking of the reference trajectories. However, it should be noted that this fault-tolerant control architecture could not tolerate any time delay in the simulation. Also, in a preliminary investigation of combining the fault-tolerant nonlinear adaptive dynamic inversion control architecture of Chapter 4 with the method of enforcing state constraints of Chapter 3, the system appeared to ignore the state constraints, exceeding the set state limits instead of enforcing them. It may not be possible to enforce state constraints with the current control formulation when a control surface initially has failed since the p, q, r inversion controller has to detect the failure before the the controller can begin to compensate for it in the control surface deflection commands.

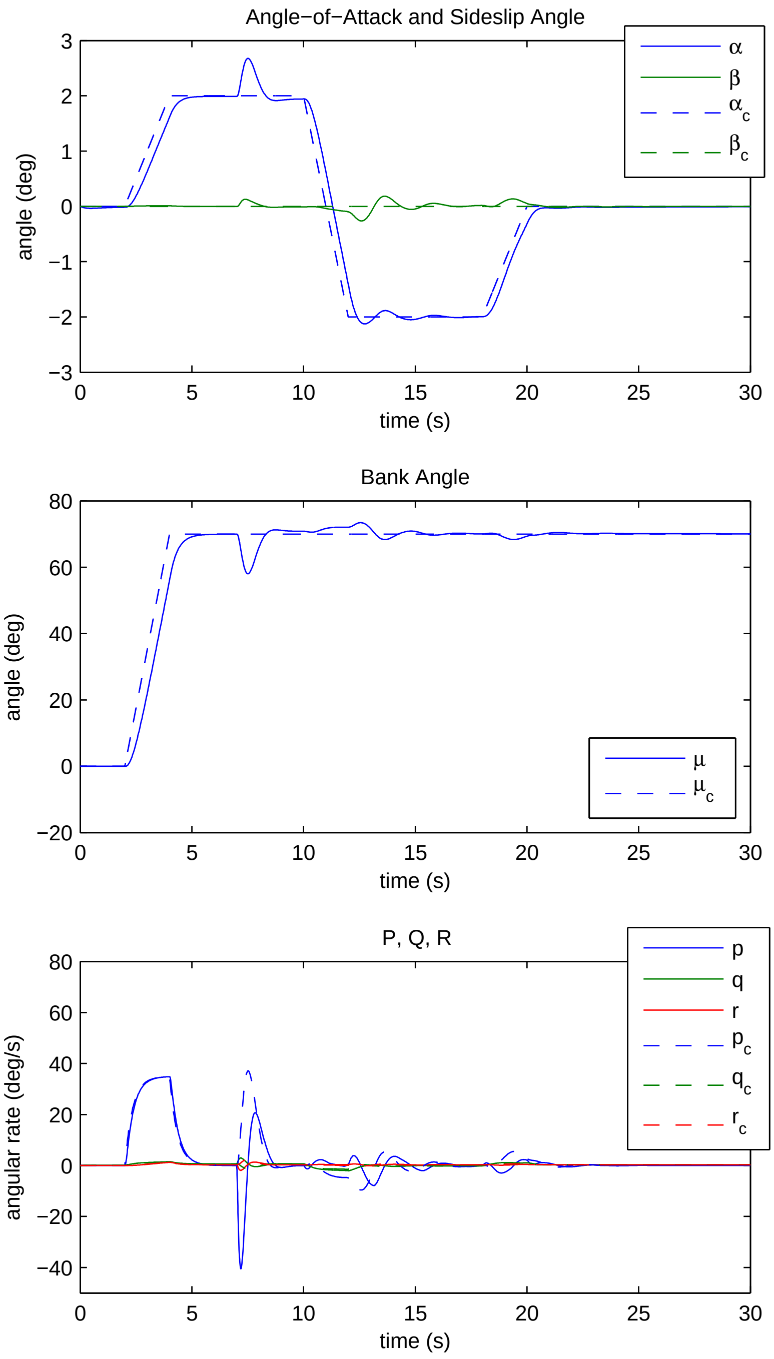


Figure 4.1: State responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg with the failure of the left elevon at 7 sec.

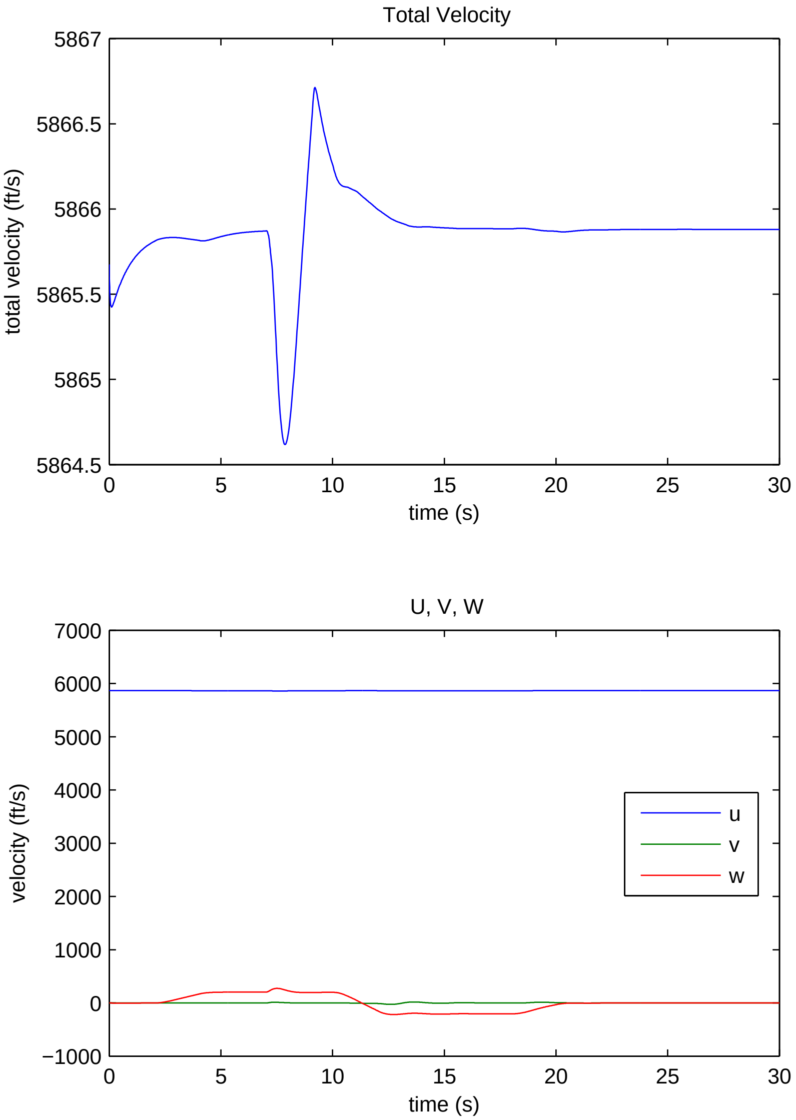


Figure 4.2: Velocity responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg with the failure of the left elevon at 7 sec.

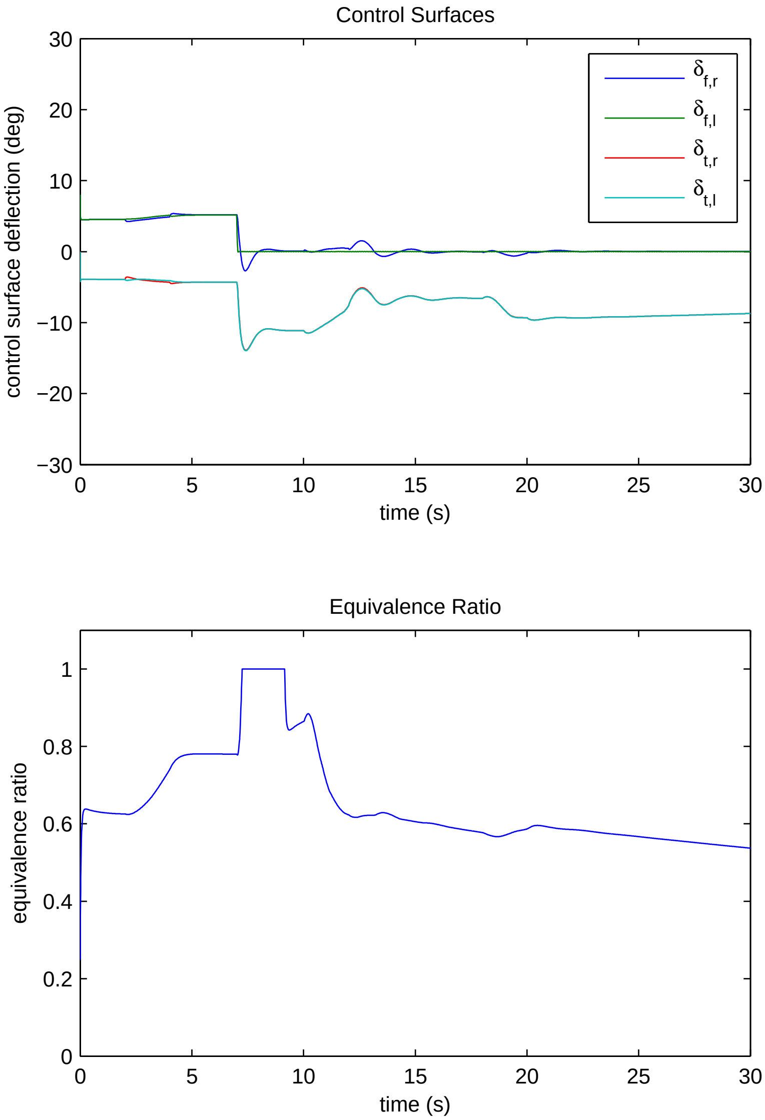


Figure 4.3: Control responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg with the failure of the left elevon at 7 sec.

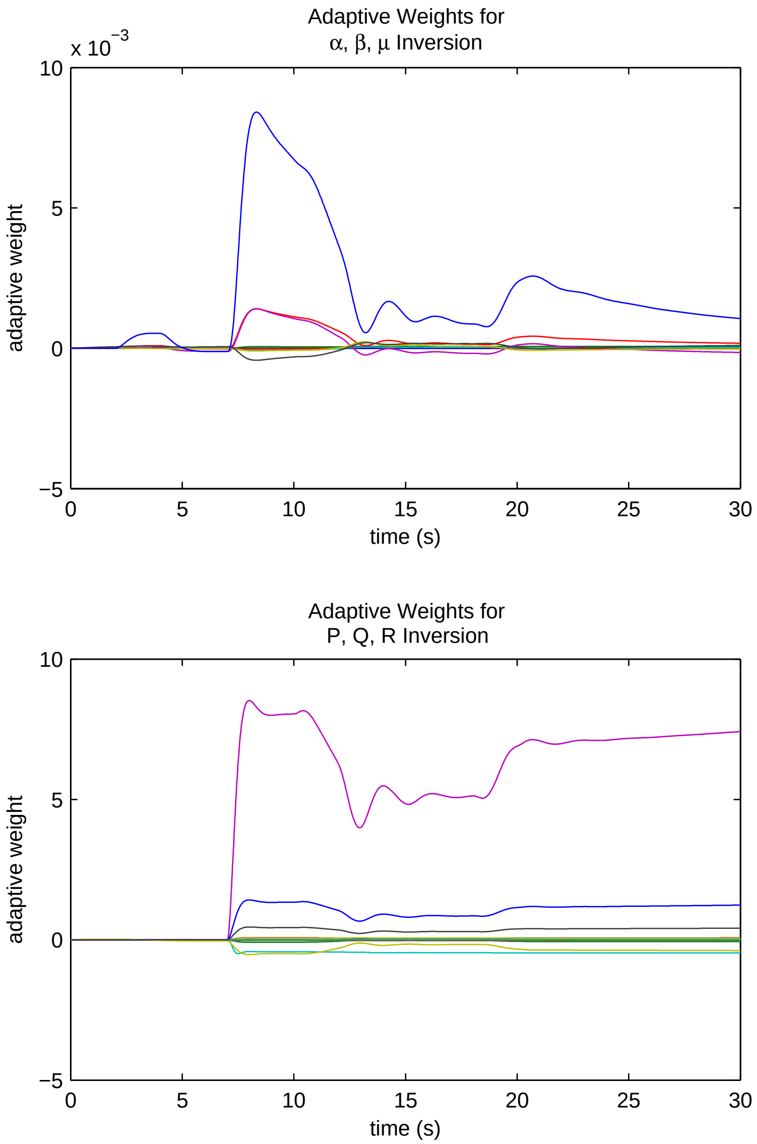


Figure 4.4: Adaptive weight responses for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg with the failure of the left elevon at 7 sec.

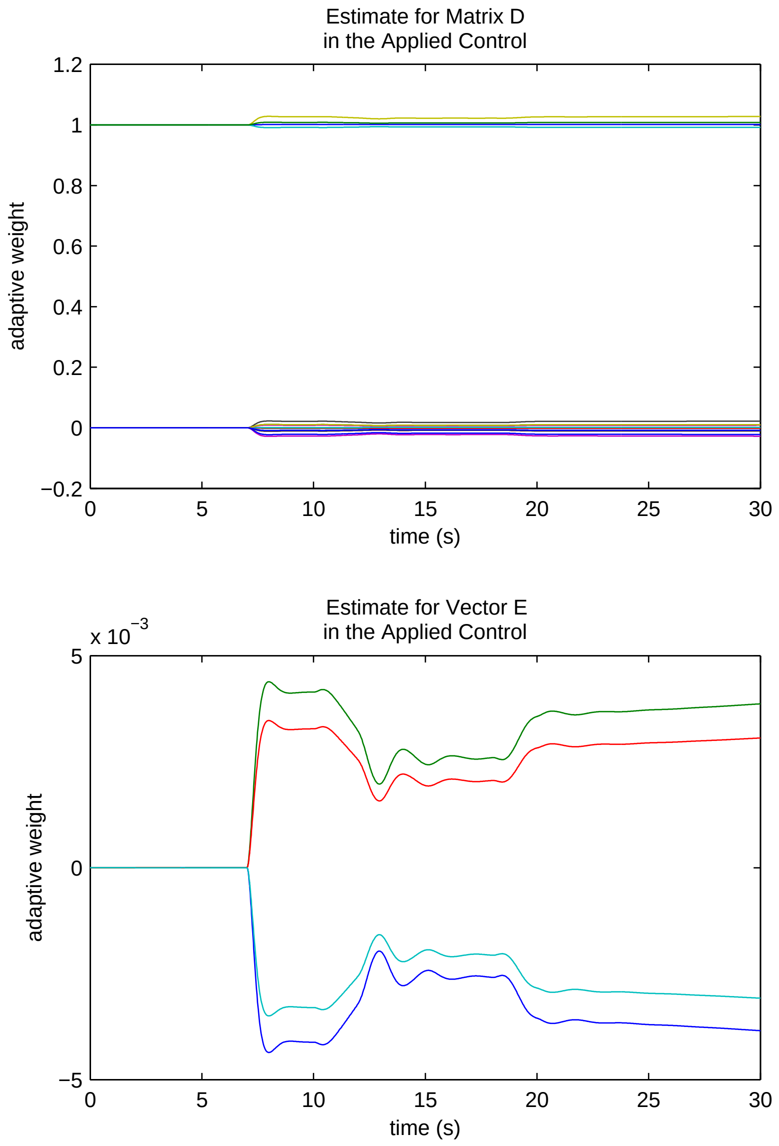


Figure 4.5: Adaptive weight responses for the applied control for the commands $\alpha = \pm 2$ deg, $\beta = 0$ deg, and $\mu = 70$ deg with the failure of the left elevon at 7 sec.

4.4 Conclusions

The combination of fault-tolerant control with the nonlinear adaptive dynamic inversion control architecture of Chapter 2 allowed the GHV to continue tracking a reference trajectory with few oscillations. However, the new fault-tolerant adaptive control architecture cannot tolerate time delay or enforce state constraints at the present. Future work will focus on bringing the ideas of Chapters 3 and 4 of enforcing state constraints and fault-tolerant control together in one controller.

5. ANALYSIS OF THE NONLINEAR ADAPTIVE DYNAMIC INVERSION CONTROL ARCHITECTURE DURING INLET UNSTARTS

Because of the significant decrease in thrust and the potential for the loss of the aircraft, inlet unstarts are a particular concern for hypersonic vehicles. During the second flight of the X-51A Waverider, the vehicle experienced an inlet unstart but was able to continue in controlled flight until the end of the test [3]. The nonlinear adaptive dynamic inversion control architecture that was developed in Chapter 2 also must be capable of maintaining the vehicle on a controlled trajectory in the event of an inlet unstart. This chapter examines the robustness of the control architecture to inlet unstarts along with control surface failures. A description of the modeling of an inlet unstart for the GHV and the theory behind the generation of a flight path angle (γ) reference trajectory are presented first, followed by simulation results.

5.1 Modeling an Inlet Unstart

This dissertation uses a simplified model of an inlet unstart in a hypersonic vehicle described in [15] because of the fact that the propulsive model for the GHV is a low-fidelity model. For this simulation, an inlet unstart is triggered at a specified time, and the loss of thrust and changes to aerodynamic parameters following the unstart are modeled as instantaneous changes. The coefficient of the axial force (C_A) is increased slightly, and the coefficient of the normal force (C_N) is decreased slightly. Additive variations in C_{m_α} and C_{n_β} are included in the plant through the equations

$$C_m = C_{m_{baseline}} + \Delta C_{m_\alpha}\alpha \tag{5.1}$$

$$C_n = C_{n_{baseline}} + \Delta C_{n_\beta}\beta. \tag{5.2}$$

A robustness analysis was performed on the GHV following an inlet unstart to determine the maximum destabilizing variations in C_{m_α} and C_{n_β} that the nonlinear adaptive dynamic inversion control architecture could handle.

5.2 Flight Path Angle Reference Trajectory Generation

While the tracking of α, β, and μ was achieved as demonstrated in Chapter 2, it was desired that the GHV have the ability to track a realistic trajectory instead of selected commands. In order to control flight path angle (γ) as opposed to α, a nonzero setpoint (NZSP) controller ([33], [34]) was designed to generate trajectories for the GHV to follow. The NZSP controller requires a linear model, so the nonlinear GHV plant model was linearized about a flight condition specified by the Mach number and altitude. Assuming that the vehicle remains wings-level during its flight of the trajectory, only the longitudinal dynamics model will be required for the trajectory generation. For the NZSP controller, the longitudinal states are $\begin{bmatrix} u & \theta & q & \alpha \end{bmatrix}^T$, and the controls are $\begin{bmatrix} \delta_T & \delta_e \end{bmatrix}^T$ where δ_T represents the equivalence ratio control, and δ_e represents the elevator control, expressed in terms of the GHV controls as $\delta_e = \left(\delta_{f,r} + \delta_{f,l}\right)/2$. The outputs y_m to be commanded by the NZSP controller are velocity u and flight path angle γ, which can be expressed in matrix-vector form as

$$y_m = \begin{bmatrix} u \\ \gamma \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \theta \\ q \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_e \end{bmatrix}. \tag{5.3}$$

By fitting a polynomial to the trajectory generated for γ, and finding the derivative of that polynomial, the reference model for γ is completely defined for the GHV simulation. In order to implement the γ, β, μ inversion controller, the dynamic equation for $\dot{\gamma}$ must be derived and written in the form of equation (2.1) as

$$\dot{\gamma} = f(s) + g(s) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5.4}$$

where s represents the states in the GHV simulation. The equation for $\dot{\gamma}$ is derived using the same process that was applied to find $\dot{\mu}$ in Section 2.4. Starting from equation (2.79), and taking its inverse, the equation for $\dot{\gamma}$ is determined to be

$$\dot{\gamma} = D(p - \dot{\beta} S_\alpha) + E(q - \dot{\alpha}) + F(r + \dot{\beta} C_\alpha) \tag{5.5}$$

where

$$D = C_\alpha S_\beta C_\mu - S_\alpha S_\mu \tag{5.6}$$

$$E = C_\beta C_\mu \tag{5.7}$$

$$F = \frac{{C_\alpha}^2 S_\beta {C_\mu}^2 - C_\alpha S_\alpha \left({S_\beta}^2 + 1\right) C_\mu S_\mu + \left({S_\alpha}^2 {S_\mu}^2 - 1\right) S_\beta}{C_\alpha S_\beta S_\mu + S_\alpha C_\mu}. \tag{5.8}$$

Consider the equations for $\dot{\beta}$ and $\dot{\alpha}$ in equations (2.77) and (2.78), respectively to have the following form

$$\dot{\beta} = f_\beta + (p S_\alpha - r C_\alpha) \tag{5.9}$$

$$\dot{\alpha} = f_\alpha + (-p C_\alpha T_\beta + q - r S_\alpha T_\beta) \tag{5.10}$$

where f_β and f_α represent the terms in $\dot{\beta}$ and $\dot{\alpha}$, respectively, that do not depend explicitly on the angular rates p, q, and r. Substituting equations (5.9) and (5.10) into equation (5.5) and simplifying the expression gives the resulting equation for $\dot{\gamma}$

that

$$\begin{aligned}\dot{\gamma} = &-Df_{\beta}S_{\alpha} - Ef_{\alpha} + Ff_{\beta}C_{\alpha} \\ &+ p\left(D - DS_{\alpha}{}^{2} + EC_{\alpha}T_{\beta} + FC_{\alpha}S_{\alpha}\right) \\ &+ q\left(0\right) \\ &+ r\left(DC_{\alpha}S_{\alpha} + ES_{\alpha}T_{\beta} + F - FC_{\alpha}{}^{2}\right).\end{aligned} \tag{5.11}$$

Note that there is no dependence on q in the equation for $\dot{\gamma}$. Consequently, when the dynamic equations for β, γ, and μ are expressed in the form

$$\begin{bmatrix}\dot{\beta}\\ \dot{\gamma}\\ \dot{\mu}\end{bmatrix} = f(s) + g(s)\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{5.12}$$

where s represents the states of the GHV, the resulting expression for $g(s)$ is

$$g(s) = \begin{bmatrix} S_{\alpha} & 0 & -C_{\alpha} \\ \left(D - DS_{\alpha}{}^{2} + EC_{\alpha}T_{\beta} + FC_{\alpha}S_{\alpha}\right) & 0 & \left(DC_{\alpha}S_{\alpha} + ES_{\alpha}T_{\beta} + F - FC_{\alpha}{}^{2}\right) \\ \sec(\beta)C_{\alpha} & 0 & \sec(\beta)S_{\alpha}\end{bmatrix}. \tag{5.13}$$

As can be seen in equation (5.13), $g(s)$ is not invertible, which causes a problem with the computation of p, q, and r in the new γ, β, μ inversion block. If $g(s)$ cannot be inverted, then the commands for p, q, and r cannot be determined for the inversion controller. Therefore, substituting the equation for $\dot{\gamma}$ for the equation for $\dot{\alpha}$ in the α, β, μ inversion controller in order to track a trajectory for γ is not possible for the GHV simulation, and another method of including the γ trajectory in the GHV simulation had to be determined.

In order to allow the GHV simulation to track a flight path angle trajectory, a method from Reference [35] was applied in which the equation for $\ddot{h}$, where h

represents the altitude of the aircraft, is written in the form

$$\ddot{h} = f_h(s) + g_h(s)\begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{5.14}$$

Given the equation for $\dot{h}$

$$\dot{h} = V\left(C_\beta C_\alpha S_\theta - S_\beta S_\phi C_\theta - C_\beta S_\alpha C_\phi C_\theta\right), \tag{5.15}$$

where V is the total velocity of the vehicle, the equation for $\ddot{h}$ is determined to be

$$\ddot{h} = \begin{bmatrix} b_0\dot{V} + b_1\dot{\beta} + b_2\dot{\alpha} \end{bmatrix} + \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5.16}$$

where

$$a_0 = b_4$$
$$a_1 = b_3 C_\phi + b_4 S_\phi T_\theta$$
$$a_2 = b_4 C_\phi T_\theta - b_3 S_\phi$$

and

$$b_0 = C_\beta C_\alpha S_\theta - S_\beta S_\phi C_\theta - C_\beta S_\alpha C_\phi C_\theta$$
$$b_1 = V\left(-S_\beta C_\alpha S_\theta - C_\beta S_\phi C_\theta + S_\beta S_\alpha C_\phi C_\theta\right)$$
$$b_2 = V\left(-C_\beta S_\alpha S_\theta - C_\beta C_\alpha C_\phi C_\theta\right)$$
$$b_3 = V\left(C_\beta C_\alpha C_\theta + S_\beta S_\phi S_\theta + C_\beta S_\alpha C_\phi S_\theta\right)$$
$$b_4 = V\left(-S_\beta C_\phi C_\theta + C_\beta S_\alpha S_\phi C_\theta\right).$$

Because $\ddot{h}$ has a nonzero coefficient for q, which means that the term $g(s)$ in equation

(5.12) is invertible, the equation for $\ddot{h}$ can replace the equation for $\dot{\alpha}$ in equation (2.82) for the α, β, μ inversion controller. The original reference trajectory that is generated for γ using the NZSP controller can be converted to $\dot{h}$ using the relation from aircraft kinematics that $\dot{h} = VS_{\gamma}$. Once a polynomial is fitted to the new trajectory for $\dot{h}$, and the derivative of that polynomial is determined, the reference model is defined for $\dot{h}$. The $\dot{h}$, β, μ inversion controller replaces the original α, β, μ inversion controller in the GHV simulation, and now desired trajectories for γ can be tracked.

5.3 Simulation Results

Using a trajectory for γ generated at the flight condition of Mach 6 at 80,000 ft, Figures 5.1, 5.2, 5.3, 5.4, and 5.5 show the results for the GHV simulation during flight path angle tracking with an inlet unstart that occurs at time t = 10 sec and a time delay of 0.03 seconds. Through the robustness analysis, it was determined that the maximum additive variations in C_{m_α} and C_{n_β} that the control architecture could tolerate were $\Delta C_{m_\alpha} = 0.001$ deg^{-1} and $\Delta C_{n_\beta} = -0.001$ deg^{-1}. It should be noted in Figure 5.4 that while the equivalence ratio is commanded to its maximum value following the inlet unstart, thrust is not being generated by the vehicle after time t = 10 sec. While tracking performance is somewhat degraded, the aircraft is still able to nominally track the specified flight path angle trajectory.

In order to investigate the ability of the nonlinear adaptive dynamic inversion control architecture to handle inlet unstarts under various flight conditions, a control failure was introduced into the simulation, and the response of the fault-tolerant adaptive control architecture from Chapter 4 was examined. Figures 5.6, 5.7, 5.8, 5.9, 5.10, and 5.11 depict the results for the GHV simulation during flight path angle tracking with an inlet unstart that occurs at 10 seconds and a failure of the left elevon

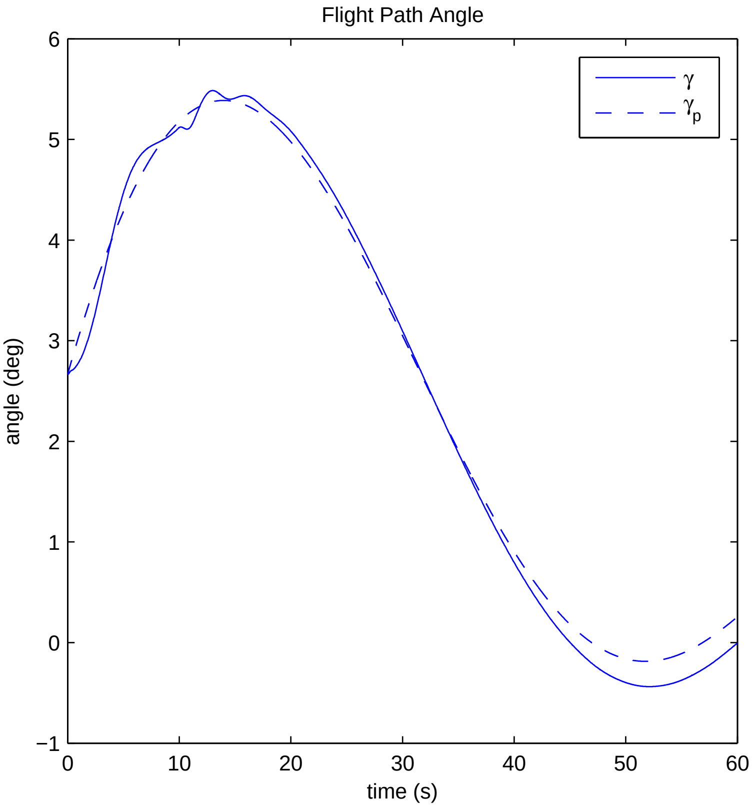


Figure 5.1: Flight path angle response compared with the generated flight path angle trajectory during an inlet unstart at 10 sec. The subscript p represents the flight path angle computed from the polynomial fit of $\dot{h}$.

at 7 seconds. Another robustness analysis of the maximum additive variations in C_{m_α} and C_{n_β} showed that with the addition of a control failure, the control architecture could tolerate only $\Delta C_{m_\alpha} = 0.0001$ deg^{-1} and $\Delta C_{n_\beta} = -0.0001$ deg^{-1}, which are an order of magnitude smaller than in the previous case. Despite a further degradation in tracking performance and the deviation of μ from its commanded angle of 0 deg for a period of time following the control failure, the hypersonic vehicle retains its ability to track the specified flight path angle trajectory.

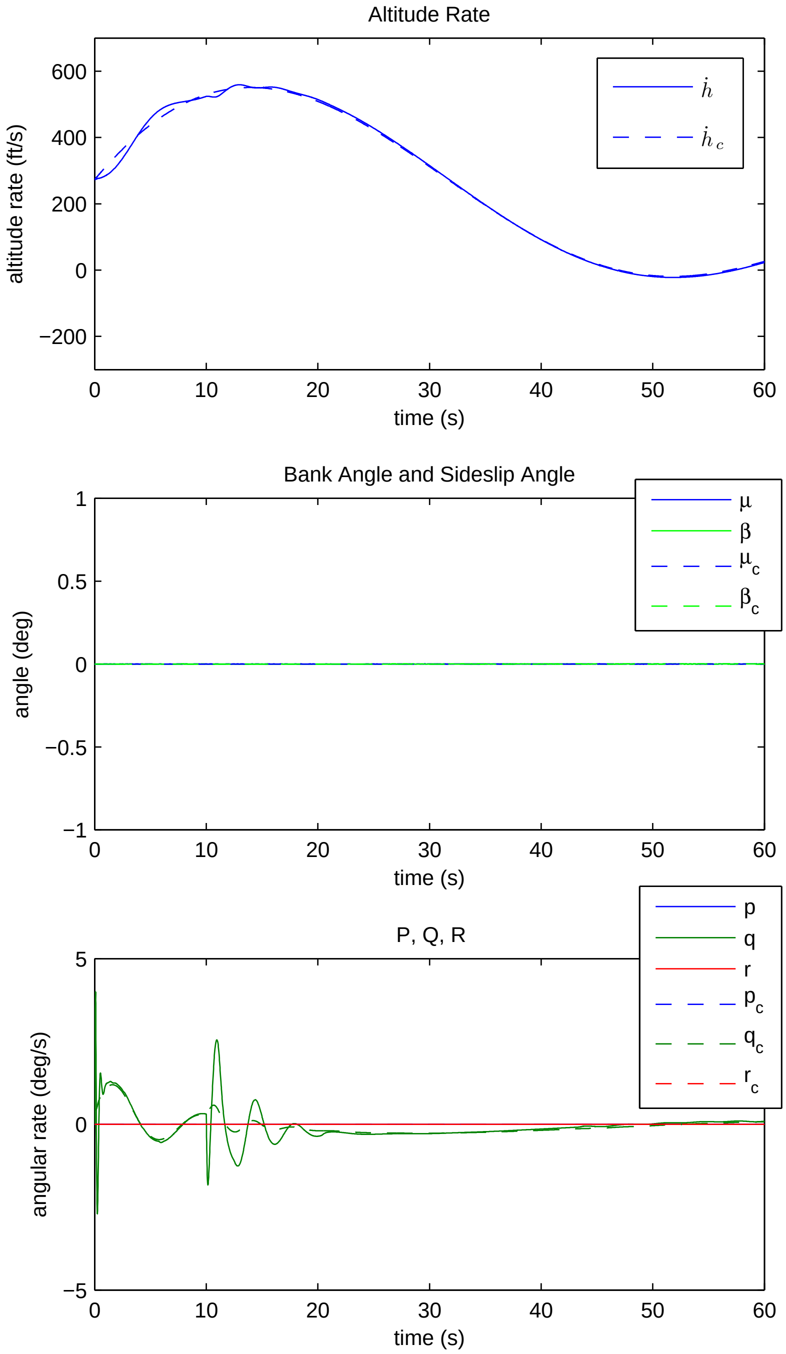


Figure 5.2: State responses for the generated flight path angle trajectory during an inlet unstart at 10 sec.

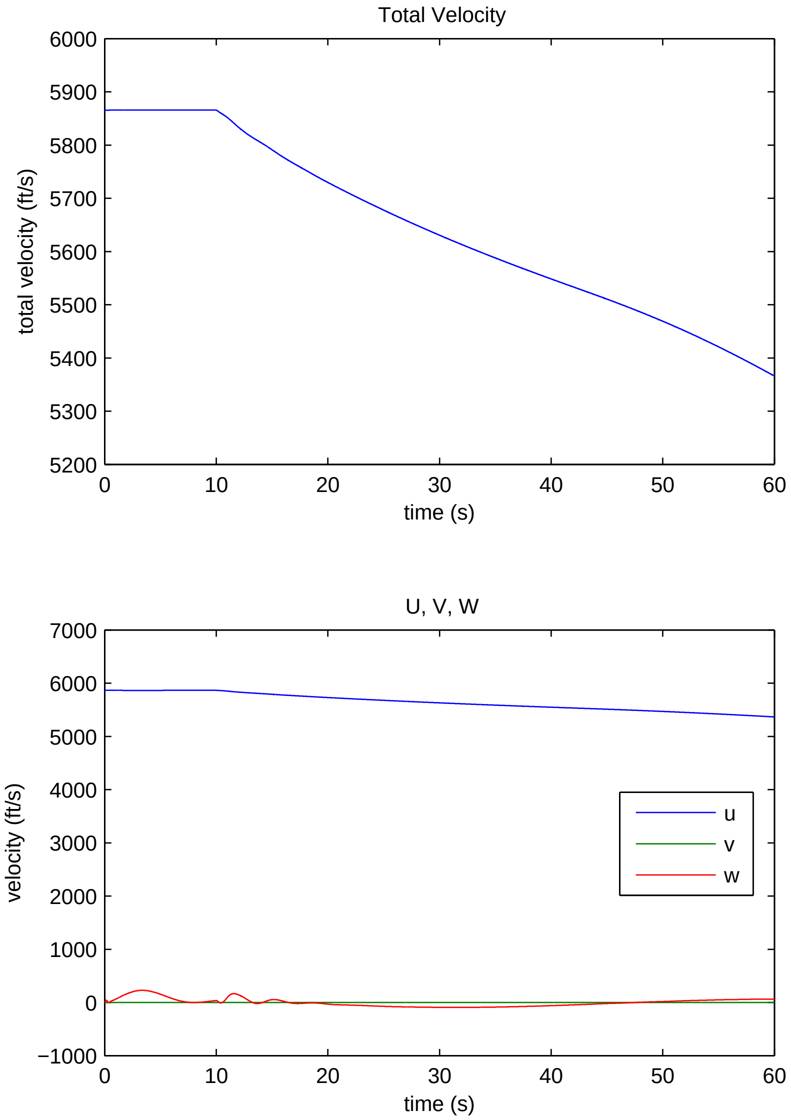


Figure 5.3: Velocity responses for the generated flight path angle trajectory during an inlet unstart at 10 sec.

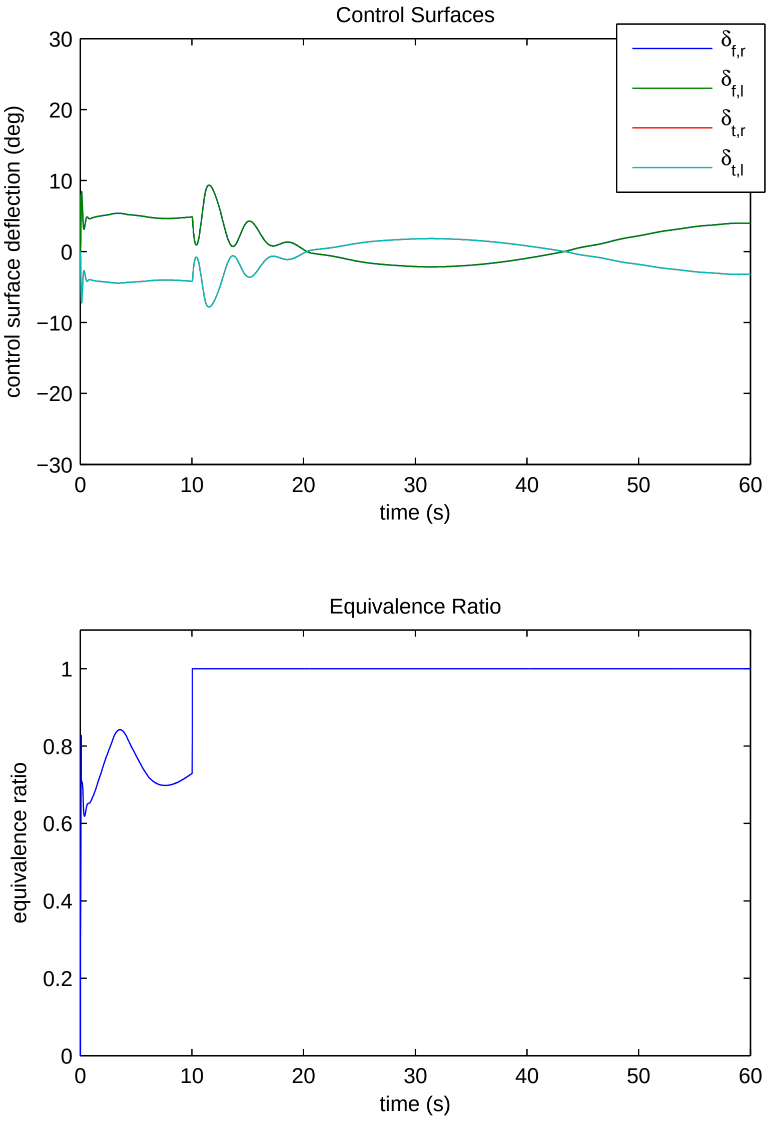
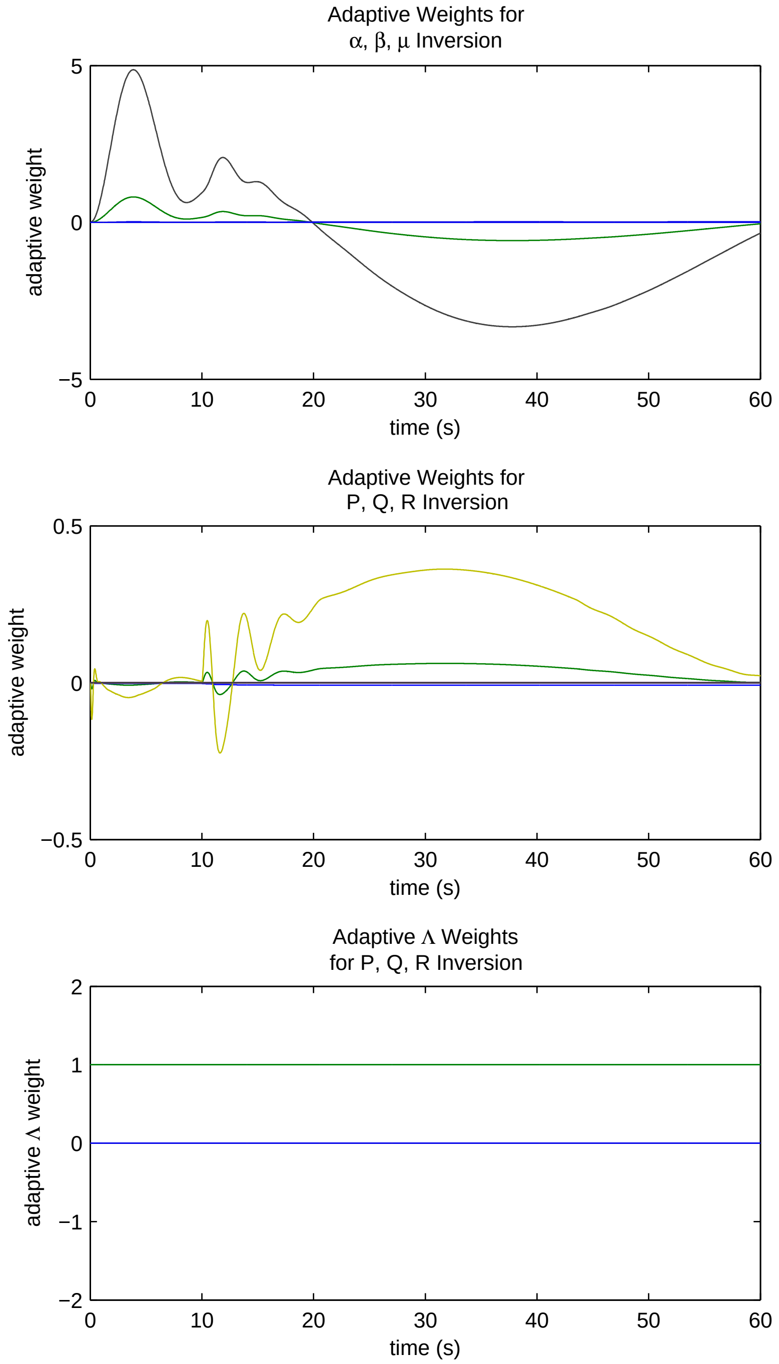


Figure 5.4: Control responses for the generated flight path angle trajectory during an inlet unstart at 10 sec.



Adaptive Weights for
α, β, μ Inversion

Adaptive Weights for
P, Q, R Inversion

Adaptive Λ Weights
for P, Q, R Inversion

Figure 5.5: Adaptive weight responses for the generated flight path angle trajectory during an inlet unstart at 10 sec.

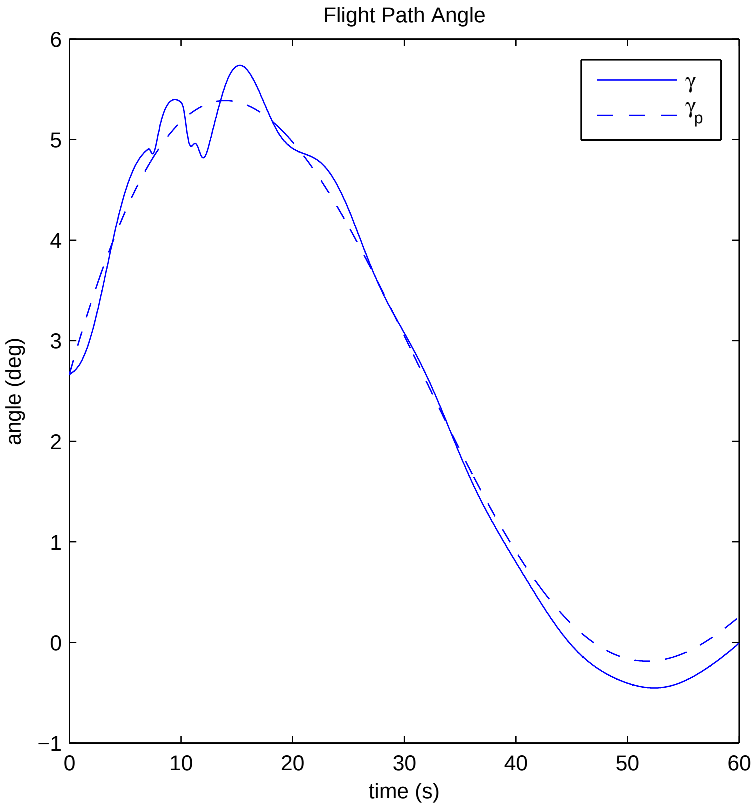


Figure 5.6: Flight path angle response compared with the generated flight path angle trajectory during an inlet unstart at 10 sec with the failure of the left elevon at 7 sec.

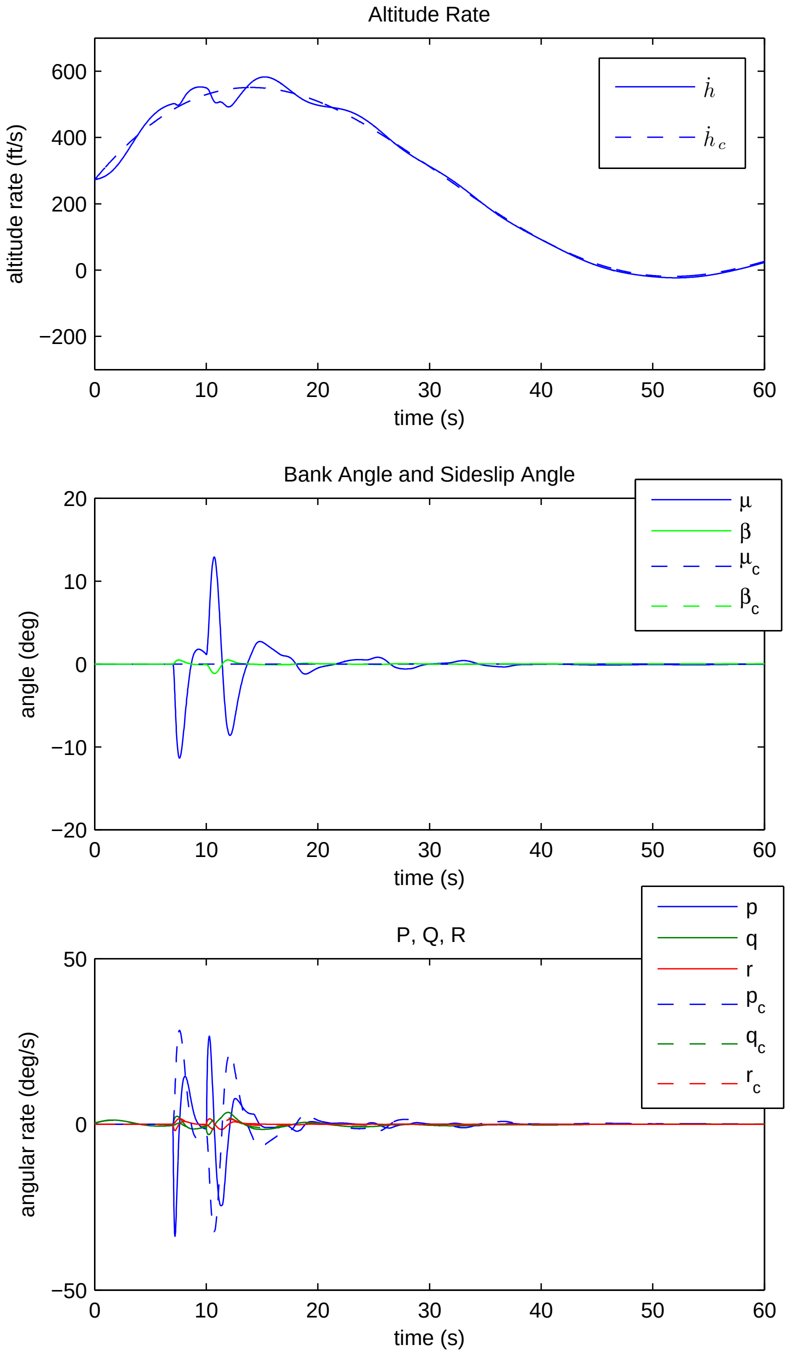


Figure 5.7: State responses for the generated flight path angle trajectory during an inlet unstart at 10 sec with the failure of the left elevon at 7 sec.

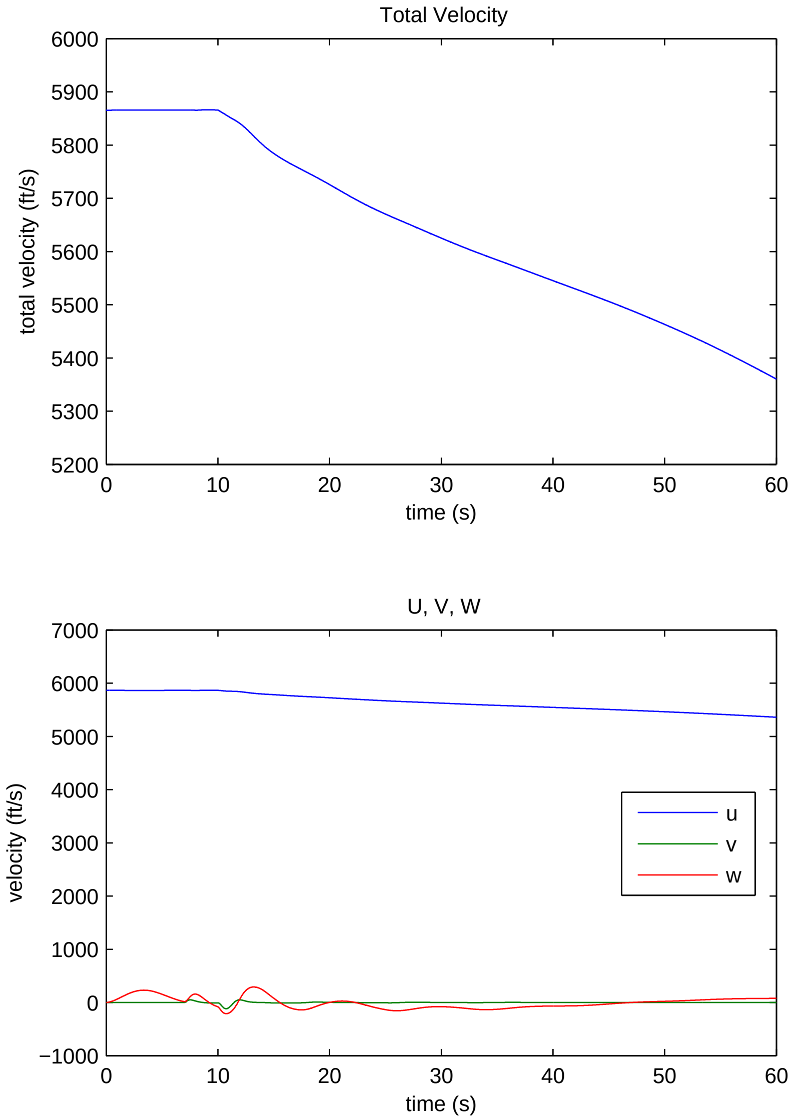


Figure 5.8: Velocity responses for the generated flight path angle trajectory during an inlet unstart at 10 sec with the failure of the left elevon at 7 sec.

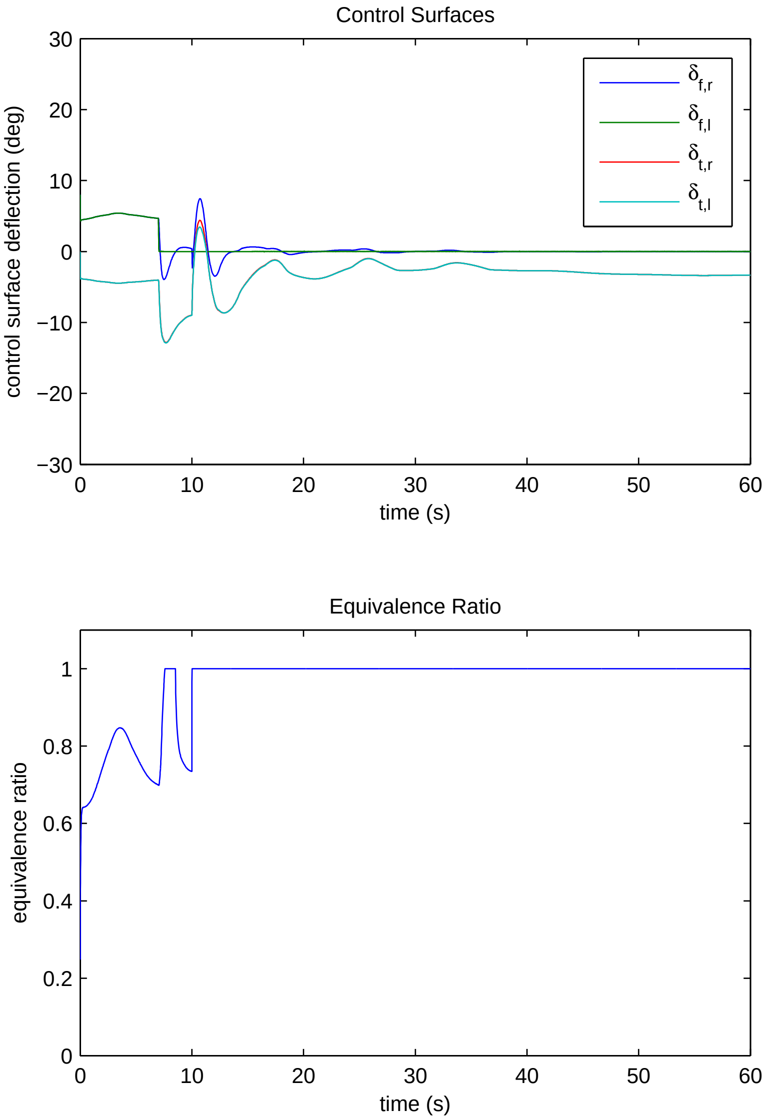


Figure 5.9: Control responses for the generated flight path angle trajectory during an inlet unstart at 10 sec with the failure of the left elevon at 7 sec.

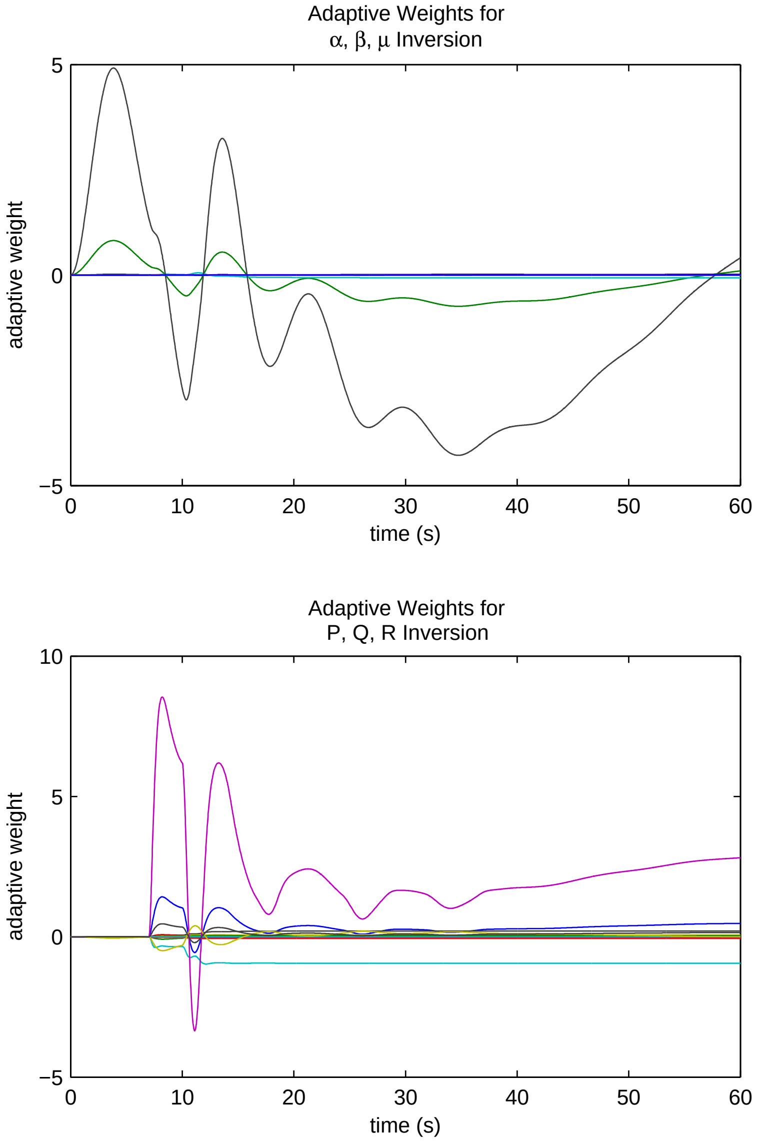


Figure 5.10: Adaptive weight responses for the generated flight path angle trajectory during an inlet unstart at 10 sec with the failure of the left elevon at 7 sec.

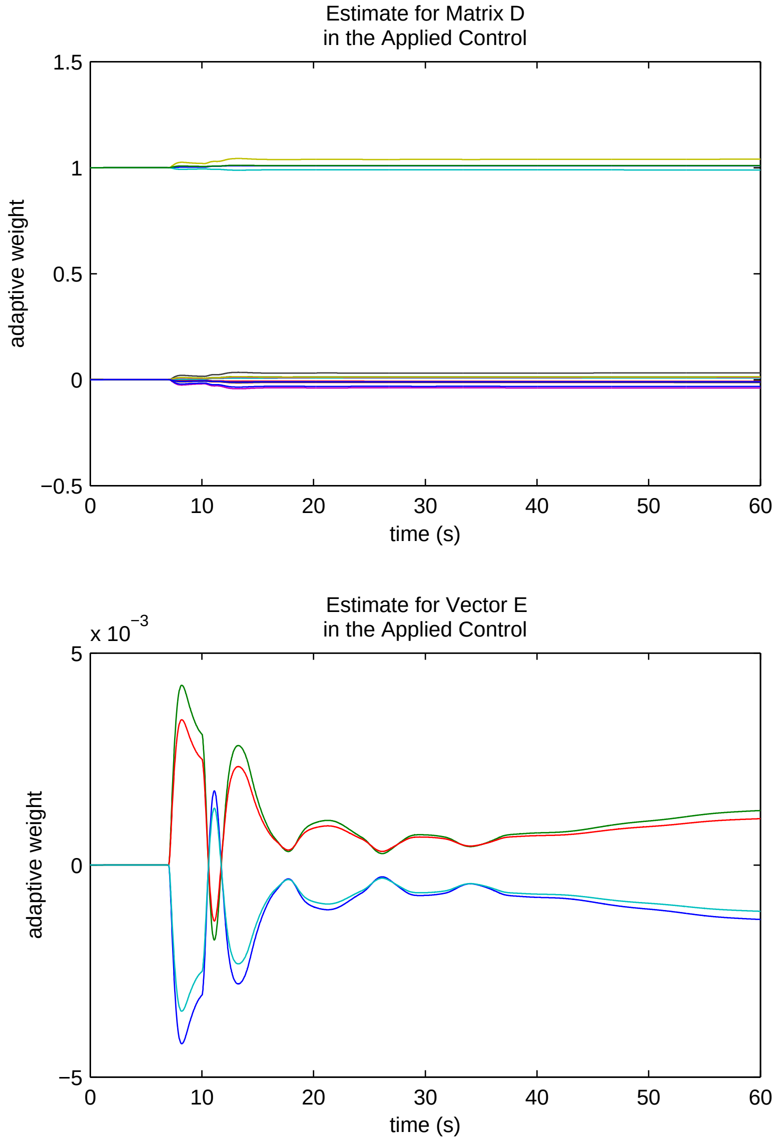


Figure 5.11: Adaptive weight responses for the applied control for the generated flight path angle trajectory during an inlet unstart at 10 sec with the failure of the left elevon at 7 sec.

5.4 Conclusions

Through this analysis of the reaction of the nonlinear adaptive dynamic inversion control architecture to inlet unstarts, it was determined that the controller is able to maintain flight path angle tracking after an unstart has occurred, as well as after the failure of a control surface. A robustness analysis determined that the maximum additive variations in C_{m_α} and C_{n_β} for the case without control failures that the control architecture could tolerate were $\Delta C_{m_\alpha} = 0.001$ deg^{-1} and $\Delta C_{n_\beta} = -0.001$ deg^{-1}. The ability of the nonlinear adaptive control architecture to maintain reference trajectory tracking following an inlet unstart is a valuable characteristic that will be beneficial for the control of hypersonic vehicles.

6. CONCLUSIONS AND RECOMMENDATIONS

The GHV is an unstable vehicle with a maximum unstable longitudinal pole at 1.96 and a maximum unstable lateral/directional pole at 5.96, and consequently, a nonlinear adaptive dynamic inversion control architecture was chosen to stabilize and control the aircraft. Based on the simulation results and the robustness analysis, it can be seen that the objective of designing a control architecture that is robust in order to achieve desired tracking performance was achieved for the GHV. The controllers are robust to decreases in control surface effectiveness, changes in system parameters, and time delays of 0.04 seconds or less. The responses for tracking generated flight path angle trajectories are well behaved, and the necessary control effort for tracking is not excessive. Additionally, the controllers are able to tolerate an inlet unstart and maintain nominal tracking of a specified flight path angle trajectory. Therefore, it can be concluded that this approach of nonlinear adaptive dynamic inversion control works well as a candidate control architecture for hypersonic vehicles.

In order to prevent an inlet unstart that occurs as a result of an altered flow to the engine that does not pass through its throat, the initial nonlinear adaptive dynamic inversion control architecture was modified to include the ability to enforce state constraints on angle-of-attack and sideslip angle. The results of the simulations with the modified control architecture demonstrated that the new control laws were able to constrain the desired states of the GHV in the presence of parametric uncertainties and a time delay of 0.02 seconds or less. However, cases were encountered in which the control architecture was not able to limit all of the states simultaneously. In one simulation, the control architecture was able only to enforce the constraint on angle-of-attack and not the constraint on sideslip angle. It is hypothesized that because of the control surface structure of the GHV with its ruddervators and elevons, which leads to coupling in the longitudinal and lateral-directional system response, control surface deflection commands that are intended to limit one state inadvertently may cause another state to violate its constraints. Also, limitations on the reference trajectories on which the control architecture was able to enforce constraints were discovered. If a constrained state was commanded to 0, a maneuver involving other states could cause the first state to exceed its limits because the time derivative of the reference trajectory for the first state would be 0, and consequently, the projection operator would not be activated in the control architecture in order to constrain the first state.

Fault-tolerant control was added to the nonlinear adaptive dynamic inversion control architecture in order to protect the vehicle in the event of a control failure. The newly revised control architecture again was proven to be stable in the closed loop system. The results of the simulations indicated that the control architecture was able to detect when a control surface had failed and to adjust the controls accordingly so that control of the vehicle and reference tracking could be maintained. However, the fault-tolerant adaptive controller was discovered not to have the ability to withstand time delays or simultaneously enforce state constraints. The fault-tolerant control architecture first must detect the failure and then adjust the control surface deflections in order to compensate for the failure of one of the surfaces. It is hypothesized that state constraints that are encountered while the failure is being detected may be exceeded since the control architecture may not be able to command the appropriate control surface deflections in the presence of a failure to prevent the violation of the state constraints.

The nonlinear adaptive dynamic inversion control architecture was tested to determine if the controller could maintain tracking performance following an inlet unstart. For these simulations, the α, β, μ inversion controller was changed to a $\dot{h}$, β, μ inversion controller so that the control architecture would be able to track more realistic flight path angle trajectories. The simulations demonstrated that while there was a slight degradation in tracking performance, the nonlinear adaptive dynamic inversion control architecture was able to maintain reference trajectory tracking following an inlet unstart, as well as a control surface failure. The loss of tracking performance following an inlet unstart occurred because of the change in aerodynamic parameters, including the coefficient of the axial force, the coefficient of the normal force, the static longitudinal stability derivative, and the static directional stability derivative, which accompanies an inlet unstart. Through a robustness analysis on the GHV, it was determined that the maximum additive variations in the static longitudinal and directional stability derivatives for the case without control failures that the control architecture could tolerate were $\Delta C_{m_\alpha} = 0.001$ deg^{-1} and $\Delta C_{n_\beta} = -0.001$ deg^{-1}.

In summary, this dissertation has shown that the nonlinear adaptive dynamic inversion control architecture is a useful method for the control of hypersonic vehicles. The control architecture can be modified to enforce state constraints and to tolerate control surface failures, and it can maintain tracking performance following an inlet unstart.

6.1 Recommendations

The investigation of the enforcement of state constraints in the nonlinear adaptive dynamic inversion control architecture in Chapter 3 was performed using separate constraints for the angle-of-attack and sideslip angle. However, depending on how the inlet unstart is modeled for a hypersonic vehicle, a combined constraint that involves both angle-of-attack and sideslip angle may be more appropriate to use. Also, in Chapter 3, limitations on the effective use of control laws with projection operators to enforce state constraints for hypersonic vehicles was discussed. One direction for future research in the area of enforcing state constraints in adaptive controllers is the use of the projection operator in the control laws. Further insight into the properties of the projection operator must be gained to understand the complete limitations of the control laws that involve projection operators and before it can be determined if the projection operator can handle a combined constraint on both the angle-of-attack and sideslip angle.

A second direction for future research in the area of enforcing state constraints in adaptive controllers is the merging of controllers that can enforce state constraints with fault-tolerant controllers. The combination of these two types of controllers would be helpful in the prevention of the loss of hypersonic vehicles because of inlet unstarts or the failure of control surfaces.

The effects of an inlet unstart on the tracking performance of the nonlinear adaptive dynamic inversion control architecture were examined in Chapter 5. While the control architecture was determined to be capable of maintaining tracking performance following an inlet unstart, it would be beneficial from an aerodynamic standpoint to be able to recover from an inlet unstart. One direction for future research in the area of inlet unstart and the control of hypersonic vehicles is the development of control logic associated with the inlet unstart envelope for a hypersonic vehicle to determine the appropriate course of action to ensure its preservation. Under certain conditions, such as if the Mach number of the flow into the inlet is too slow, the engine cannot be restarted after an inlet unstart has occurred, and a hypersonic vehicle must fly at sub-hypersonic speeds. Under other conditions, a hypersonic vehicle will be able to perform a maneuver that will restart the engine. The control logic developed through this future research should specify the high level actions to be taken for various Mach numbers, altitudes, and orientations, as well as the manuevers corresponding to these high level actions.

Finally, the simulations in this dissertation were run using a rigid body model of a hypersonic vehicle. One direction for future research in the area of the control of hypersonic vehicles is the development and testing of controllers using an elastic model of a hypersonic vehicle. With a higher-fidelity model that includes aeroelastic effects, situations in which control architectures constructed based on rigid body models might fail can be simulated and analyzed, and controllers that can handle these aeroelastic issues can be developed.

REFERENCES

[1] Aldrich, B. S., *A Lantern In Her Hand*, Puffin Books, New York, New York, 1997.

[2] Rowling, J. K., *Harry Potter and the Order of the Phoenix*, Scholastic Press, New York, New York, 2003.

[3] U.S. Air Force, "Factsheets: X-51A Waverider," `http://www.af.mil/information/factsheets/factsheet.asp?fsID=17986`, May 2013.

[4] Hallion, R. P., "The History of Hypersonics: or, "Back to the Future - Again and Again"," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2005.

[5] Rodriguez, A. A., Dickeson, J. J., Cifdaloz, O., McCullen, R., Benavides, J., Sridharan, S., Kelkar, A., Vogel, J. M., and Soloway, D., "Modeling and Control of Scramjet-Powered Hypersonic Vehicles: Challenges, Trends, & Tradeoffs," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, August 2008.

[6] Heiser, W. H. and Pratt, D. T., *Hypersonic Airbreathing Propulsion*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Washington D. C., 1994.

[7] Annaswamy, A. M., Jang, J., and Lavretsky, E., "Adaptive gain-scheduled controller in the presence of actuator anomalies," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, August 2008.

[8] Gibson, T. E. and Annaswamy, A. M., "Adaptive Control of Hypersonic Vehicles in the Presence of Thrust and Actuator Uncertainties," *AIAA Guidance, Navi-*

gation and Control Conference and Exhibit, Honolulu, Hawaii, August 2008.

[9] Groves, K. P., Sigthorsson, D. O., Serrani, A., Yurkovich, S., Bolender, M. A., and Doman, D. B., "Reference Command Tracking for a Linearized Model of an Air-breathing Hypersonic Vehicle," *AIAA Guidance, Navigation and Control Conference and Exhibit*, San Francisco, California, August 2005.

[10] Bolender, M. A., Staines, J. T., and Dolvin, D. J., "HIFiRE 6: An Adaptive Flight Control Experiment," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Nashville, Tennessee, January 2012.

[11] Johnson, E. N., Calise, A. J., Curry, M. D., Mease, K. D., and Corban, J. E., "Adaptive Guidance and Control for Autonomous Hypersonic Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, May-June 2006, pp. 725–737.

[12] Fiorentini, L., Serrani, A., Bolender, M. A., and Doman, D. B., "Nonlinear Robust Adaptive Control of Flexible Air-breathing Hypersonic Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, Mar.-Apr. 2009, pp. 401–416.

[13] Parker, J. T., Serrani, A., Yurkovich, S., Bolender, M. A., and Doman, D. B., "Approximate Feedback Linearization of an Air-breathing Hypersonic Vehicle," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Keystone, Colorado, August 2006.

[14] Parker, J. T., Serrani, A., Yurkovich, S., Bolender, M. A., and Doman, D. B., "Control-Oriented Modeling of an Air-Breathing Hypersonic Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 3, May-June 2007, pp. 859–869.

[15] Brocanelli, M., Gunbatar, Y., Serrani, A., and Bolender, M. A., “Robust Control for Unstart Recovery in Hypersonic Vehicles,” *AIAA Guidance, Navigation and Control Conference*, Minneapolis, Minnesota, August 2012.

[16] Vaddi, S. S. and Sengupta, P., “Controller Design for Hypersonic Vehicles Accommodating Nonlinear State and Control Constraints,” *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, August 2009.

[17] Lavretsky, E. and Gadient, R., “Robust Adaptive Design for Aerial Vehicles with State-Limiting Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, November-December 2010, pp. 1743–1752.

[18] Gadient, R., Lavretsky, E., and Hyde, D., “State Limiter for Model Following Control Systems,” *AIAA Guidance, Navigation and Control Conference*, Portland, Oregon, August 2011.

[19] Bürger, M. and Guay, M., “Robust Constraint Satisfaction for Continuous-Time Nonlinear Systems in Strict Feedback Form,” *IEEE Transactions on Automatic Control*, Vol. 55, No. 11, November 2010, pp. 2597–2601.

[20] Muse, J. A., “A Method for Enforcing State Constraints in Adaptive Control,” *AIAA Guidance, Navigation and Control Conference*, Portland, Oregon, August 2011.

[21] Vick, T. J., “Documentation for Generic Hypersonic Vehicle Model,” Tech. rep., U.S. Air Force Research Laboratory, Wright-Patterson AFB.

[22] Snell, S. A., Enns, D. F., and Garrard Jr., W. L., “Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft,” *AIAA Guidance, Navigation, and Control Conference Technical Papers*, Vol. Part 1, American Institute of Aeronautics and Astronautics, Washington D. C., August 1990, pp. 808–825.

[23] Miele, A., *Flight Mechanics*, Vol. 1: Theory of Flight Paths, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1962.

[24] Pomet, J.-B. and Praly, L., "Adaptive Nonlinear Regulation: Estimation from the Lyapunov Equation," *IEEE Transactions on Automatic Control*, Vol. 37, No. 6, June 1992, pp. 729–740.

[25] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice Hall, Inc., Upper Saddle River, New Jersey, 1991.

[26] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, Wiley, Hoboken, New Jersey, 2nd ed., 2003.

[27] Schmidt, D. K., *Modern Flight Dynamics*, McGraw-Hill, New York, New York, 2012.

[28] Johnson, E. N., *Limited Authority Adaptive Flight Control*, Ph.D. thesis, Georgia Institute of Technology, November 2000.

[29] Johnson, E. N. and Kannan, S. K., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Monterey, California, August 2002.

[30] Lavretsky, E., Gibson, T. E., and Annaswamy, A. M., "Projection Operator in Adaptive Systems," *arXiv e-Prints*, arXiv:1112.4232v6 [nlin.AO], 2012.

[31] Hovakimyan, N. and Cao, C., $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, Vol. 21 of *Advances in Design and Control*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 2010.

[32] Tandale, M. D. and Valasek, J., "Fault-Tolerant Structured Adaptive Model Inversion Control," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, May-June 2006, pp. 635–642.

[33] Valasek, J., “Optimal Setpoint Controller for a Generic X-29A Aircraft with Forebody Vortex Nozzles,” *Proceedings of the 1996 IEEE International Conference on Control Applications*, Dearborn, Michigan, September 1996.

[34] Stengel, R. F., *Stochastic Optimal Control: Theory and Applications*, Wiley, New York, New York, 1986.

[35] Menon, P., Badgett, M., Walker, R., and Duke, E., “Nonlinear Flight Test Trajectory Controllers for Aircraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 1, Jan.-Feb. 1987, pp. 67–72.

[36] Siddarth, A. and Valasek, J., “Kinetic State Tracking for a Class of Singularly Perturbed Systems,” *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 3, May-June 2011, pp. 734–749.

[37] Shalashilin, V. I. and Kuznetsov, E. B., *Parametric Continuation and Optimal Parametrization in Applied Mathematics and Mechanics*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[38] Shampine, L. F. and Reichelt, M. W., “The MATLAB ODE Suite,” *SIAM Journal on Scientific Computing*, Vol. 18, No. 1, 1997, pp. 1–22.

[39] Moler, C. B., *Numerical Computing with MATLAB*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 2004.

APPENDIX A

TWO TIME-SCALES AND INTEGRATION METHODS

The development of the nonlinear adaptive dynamic inversion control architecture of Chapter 2 is possible because of the fact that the aircraft states can be divided into two timescale categories. Those categories are the fast states, which consist of the angular rates p, q, and r as noted in [22], [35], and [36], and the slow states, which consist of the angles α, β, and μ. In working with systems involving two timescales, the issue of solving stiff differential equations often arises [37]. The presence of stiff equations of motion indicates that the integration method and step size be taken into consideration so that the solution presented is accurate.

Various integration solvers are used by Matlab and Simulink for ordinary differential equations, which are the type of equations in the GHV simulation. A description of these solvers can be found in [38] and [39]. All of the results in this dissertation were determined using the variable step-size ode45 solver in Matlab as the integration method. In order to verify in Chapter 2 that the simulation results computed with the ode45 solver were representative of the actual solution because of the presence of two timescales, the same simulation was run using the ode23s solver, which is designed to solve stiff equations, with a time step between 0.009 and 0.01 sec. At this time step size, the rigid body modes should be captured. The simulation run using the ode23s solver produced results that were identical to the results from the simulation that used the ode45 solver, which indicated that the simulation results in Chapter 2 accurately represent the actual solution of the equations of motion. To verify that the results shown in Figures 3.10, 3.11, 3.12, and 3.13 in Chapter 3 were accurate, the simulation was run using the ode4 solver, which is a fixed time step solver, using time steps of 0.001 and 0.0001 sec. Again, the results from the simulation using the ode4 solver were identical to the results shown in Figures 3.10, 3.11, 3.12, and 3.13. From these two analyses, it therefore can be concluded that the results presented in this dissertation accurately depict the actual solutions to the equations of motion for the GHV.