Gain-Scheduled H-Infinity Control and Analysis of a Nonlinear Generic Hypersonic Vehicle

By

Alec Davis Bowman

Submitted to the graduate degree program in Aerospace Engineering and the
Graduate Faculty of the University of Kansas in partial fulfillment of the
requirements for the degree of Masters of Science.

Dr. Shawn Keshmiri, Chairperson

Dr. Mark Ewing

Dr. Richard Hale

Date Defended: June 9, 2016

The Thesis Committee for Alec Davis Bowman certifies that this is
the approved version of the following thesis:

Gain Scheduled H-Infinity Control and Analysis of a Nonlinear Generic Hypersonic Vehicle

Dr. Shawn Keshmiri, Chairperson

Date Approved: June 14, 2016

Abstract

The concept of hypersonic flight has been around for many years. In recent years, emerging technologies and market forces have renewed latent interest in this challenging field. With many private and government institutions driving new innovations, these concepts are becoming reality. New research is needed to facilitate future innovation and deployment. The complex dynamic behaviors within the hypersonic flight envelope must be studied for designers to either mitigate or compensate for their effects on future vehicles. Control techniques must be adapted to suit the unstable and highly nonlinear dynamics of such systems. This work has two goals: to explore the dynamic characteristics of hypersonic flight and to control such a vehicle in the face of non-linearly changing dynamics. A nonlinear, 6 degree of freedom dynamic model of a Generic Hypersonic Vehicle is developed. The model integrates changing mass, moments of inertia, and center of gravity as a function of fuel burn. A bank of spline interpolation tables generates aerodynamic coefficients dependent on speed, angle of attack, and control surface deflections for the entire flight envelope. The nonlinear model of the full flight envelope is then reduced to a series of linear models to represent the aircraft trimmed under straight and level flight conditions over the range of Mach numbers, Mach 2 to 23. The changing Longitudinal and Lateral dynamics of the linearized system are analyzed as a function of Mach number using standard linear techniques to show the changing vehicle characteristics. A spline-based gain-scheduled, H-infinity controller is also designed for a subset of the linear systems. The controller stabilizes the system between Mach 4.9 and 7.1, with aircraft weight ranging from 160,000 to 230,000 pounds and from 68,000 to 92,000 feet altitude. The controller maintains system stability while commanded to change both Mach number and altitude within the gain-scheduled envelope. Additionally, the controller's performance is assessed in the presence of low frequency disturbances.

Acknowledgments

This work has been many years in the making. I am proud of this final thesis as I feel it successfully conveys the results countless stress filled days and numerous sleepless nights have produced. There are several people that, without your help, this document would have been impossible. **To Dr. Shawn Keshmiri**: Thank you for your combination of patience and relentless drive to get me to this point. Your passion for your work comes through in everything you do and teach—it is inspiring. Under your tutelage, my knowledge of aircraft dynamics has become something innate, well beyond equations in a book. **To Dr. Gonzalo Garcia**: Thank you for our many in depth discussions regarding the ins and outs of control theory. Your knowledge of applied aircraft control theory significantly influenced this work. **To Dr. Keith Santarelli**: Your knowledge and perspective on H-Infinity control changed the way I think about robust control. It transformed my understanding of the equations to something inherently logical and fully understandable, beyond the complex mathematics. **To Dr. Mark Ewing**: Thank you for the showing me the other side of graduate life at KU. I still remember my introduction to KU basketball as your guest to the first game of the season. **To Dr. Richard Hale**: You gave me the opportunity to experience Antarctica. I still have trouble believing I spent two months on the ice forming the most memorable experiences of my life. **To Amy Borton**: I can honestly say you got me through graduate school. Your aid was more than just helping through the administrative processes. Your drive and friendship, getting me through many frustrating times with many stress relieving phone calls and lengthy textual conversations, helped make this work a reality. There is not enough room to express my gratitude. **To those I love**: Thank you for your patience. Thank you for pushing me to keep focused, even when I asked you not to. This work is dedicated to you.

Contents

List of Figures

List of Tables

List of Symbols

A_n	Magnitude of eigenvector contribution due to state n	(-)
b	Wing span	ft
c	Geometric cord	ft
C	Coefficient of (e.g, coefficient of drag due to angle of attack, C_{D_α})	(-)
$d(t)$	Time domain disturbance input	(-)
$D(s)$	Frequency domain disturbance input	(-)
D, Y, and L	Drag, side, and lift force, respectively	lbs
$e(t)$	Time domain error signal	(-)
$E(s)$	Frequency domain error signal	(-)
F	Forces in x, y or z	lbs
F_P	Propulsive force	lbs
g	Gravitational acceleration	ft/s²
$G(s)$	Idealized system model	(-)
h	Elevation/altitude	ft
H	Transformation matrix	(-)
$[H]_A^B$	Transformation matrix from coordinate system A to coordinate system B	(-)
I_{xx}, I_{yy}, and I_{zz}	Moments of inertia	slug-ft²
I_{xy},I_{yx}, I_{xz}, I_{zx}, I_{yz}, and I_{zy}	Products of inertia	slug-ft²
j	Complex number	(-)
$K(s)$	System controller	(-)
K_P, K_I, K_D, and K_N	PID coefficients	(-)
$\boldsymbol{\mathcal{K}}$	Set of all controllers	(-)
$\mathcal{K}$	Controller for a given trim point	(-)

K	Gain-scheduled controller	(-)
K_1, K_2, and K_3	PI and PID controllers	(-)
$\mathcal{L}$	Momentum	slug-ft/s
L, M, and N	Moments about the x, y, and z body axis respectively	ft-lbs
$\mathcal{M}$	Total moments	ft-lbs
M	Mach number	(-)
m	Mass	slugs
$n(t)$	Time domain noise input	(-)
$N(s)$	Frequency domain noise input	(-)
$\mathcal{P}$	Momentum	slug-ft/s
$P(s)$	Augmented plant model	(-)
P, Q, and R	angular rates	deg/s
p, q and r	Perturbed angular rates	deg/s
$\bar{q}$	Dynamic pressure	lbs/ft^2
$r(t)$	Time domain reference input	(-)
$R(s)$	Frequency domain reference input	(-)
S_{ref}	Reference surface area	ft^2
$S(s)$	Sensitivity function	(-)
$T(s)$	Complementary sensitivity function	(-)
t	Time variable	s
T	Temperature	°R
U, V, and W	body velocities	ft/s
u, v and w	Perturbed body velocities	ft/s
$u(t)$	Time domain controller output	(-)
$U(s)$	Frequency domain controller output	(-)

$W_1(s)$, W_2(s), and $W_3(s)$	Weighting transfer functions	(-)
$w(t)$	Combined time domain system input	(-)
$W(s)$	Combined frequency domain system inputs	(-)
$\boldsymbol{X}$	Eigenvector	(-)
X, Y, and Z	Forces along the x, y, and z body axis respectively	lbs
$x(t)$	State variables	(-)
$y(t)$	Time domain P(s) system output	(-)
$Y(s)$	Frequency domain P(s) system output	(-)
$y_G(t)$	Time domain G(s) system output	(-)
$Y_G(s)$	Frequency domain G(s) system output	(-)
$\mathscr{y}(t)$	Time domain real system measurement output	(-)
$\mathcal{Y}(s)$	Frequency domain real system measurement output	(-)
$z(t)$	Time domain weighted system outputs	(-)
$Z(s)$	Frequency domain weighted system outputs	(-)
α	Angle of attack	degs
β	Side slip angle	degs
γ	Flight path angle	degs
δ_{le}	Left elevon deflection	degs
δ_{re}	Right elevon deflection	degs
δ_a	Aileron deflection	degs
δ_e	Elevator deflection	degs
δ_r	Rudder deflection	degs
δ_{ER}	Equivalence ratio	(-)
$\Delta_{x_{cg}}$	Distance between MRC and CG	ft
Δ	Change in	(-)

λ	Eigenvalue	(-)
ρ	atmospheric density	slug/ft^3
σ	Gain-scheduling parameters	(-)
Φ, Θ, and Ψ	Euler angles	degs
ϕ, θ, and ψ	Perturbed Euler angles	degs
$\widetilde{\omega}$	Cross-product-equivalent Matrix	(-)
ω_n	Eigenvector phase shift due to state n	(-)

List of Subscripts

0	Condition at sea level or Nominal Condition
A	Aerodynamic Forces
A+P	Aerodynamic + Propulsive Forces
A+P+G	Aerodynamic + Propulsive + Gravitational Forces
B	Body Coordinate System
cmd	Commanded value
da, de	Total deflection of aileron and elevator
D	Drag
e	Error Signal
E	Inertial or Flat Earth Coordinate System
F	Frobenius Norm
L	Lift
lat	Lateral-Directional
lon	Longitudinal
le, re, dr	Total deflection of left and right elevon and rudder
Mrc	At Moment Reference Center
p	Perturbed state
S	Stability Coordinate System
T	Total state
trm	Trimmed value of state
x	Cross Axis

List of Acronyms

CG	Center of Gravity
DARPA	Defense Advanced Research Projects Agency
ER	Equivalence Ratio
GHV	Generic Hypersonic Vehicle
LTI	Linear Time Invariant
LQG	Linear Quadratic Gaussian
MIMO	Multi-Input Multi-Output
MRC	Moment Reference Center
SISO	Single-Input Single-Output

Chapter 1 Introduction and Technical Background

The concept of the space plane has been around for many years. In the 1980's, The United States, Europe, and Russia invested considerable financial resources to develop their own airplane-like vehicles to reach space. In the late 1980's and into early 2000's there was a renewed push for reusable, single-stage-to-orbit aircraft capable of reaching beyond the earth's atmosphere. Due in part to technical and economic challenges, many of these projects were abandoned, favoring more traditional rocket assisted launches[1]. Recently, however, new technologies and market forces have made the concept of a space plane once again viable. The Defense Advanced Research Projects Agency (DARPA) contracted industry partners to begin developing the next generation of reusable launch vehicles and has recently announced funding for the second round of development during which companies will test their prototype vehicles [2].

Developing a new class of aircraft requires overcoming many new challenges. Of foremost consideration are the complex dynamic behaviors of aircraft flying at high Mach numbers, nonlinear and coupled aerodynamics, high temperatures associated with the shock waves and scramjet engines, and trajectory optimization associated with hypersonic flights. At these speeds (Mach 5 or greater), the dynamics of the system become partially unstable, making the control tasks more challenging with little to no margin for error. For this reason, more advanced, fully automated, and robust controllers become mandatory. This work presents one development path for such a controller.

1.1 Hypersonic Aircraft Flight

Since the beginning of the space race in the 1950's, scientists and engineers have worked to develop effective ways of entering and returning from space. To this day, the only viable means of traveling to and from space is through a modification of the original design of rocket-propelled launch and pod-style reentry vehicles.

In the 1980's Europe, Russia, and the United States explored the idea of a reusable space plane, though only the US's Space Shuttle program became fully operational. Similar to the original launch vehicles, the Space Shuttle used a series of rockets to launch the vehicle into a low earth orbit. To return to earth from its orbit, the Shuttle used reaction-based control until there was sufficient atmosphere to make use of its control surfaces. The vehicle descended at a high angle of attack to dissipate the energy of reentry on its blunt underside, decelerating from a maximum speed of roughly Mach 25 to land like conventional aircraft, making it the first truly hypersonic winged aircraft. By the time the program ended, six shuttles were produced, four of which still exist and are on display throughout the United States (Figure 1) [1], [3].

In the 1980's, NASA began considering a different launch method than the traditional rocket propelled exit vehicle. The conceptual single-stage-to-orbit vehicle would launch horizontally like a conventional aircraft. From there, it would transition out of the atmosphere under its own power using a scramjet engine, after which, it would transition to rocket propulsion as the atmosphere diminished. This solution was all but abandoned due to poor market conditions and lack of mature technology in the 1990's [1].

Figure 1: Space Shuttle Endeavour on Display in Los Angeles, CA. Photograph courtesy of Alec Bowman

Despite the canceled single-stage-to-orbit program, hypersonic aircraft research continued. The X-43A unmanned hypersonic aircraft was a 3.7 meter long, 1300 kg testbed for scramjet propulsion (Figure 2). In 2004, the system was successfully tested, reaching its record speed of 12,144 km/h, or Mach 9.8, during which its scramjet engine operated for 10 seconds [1].

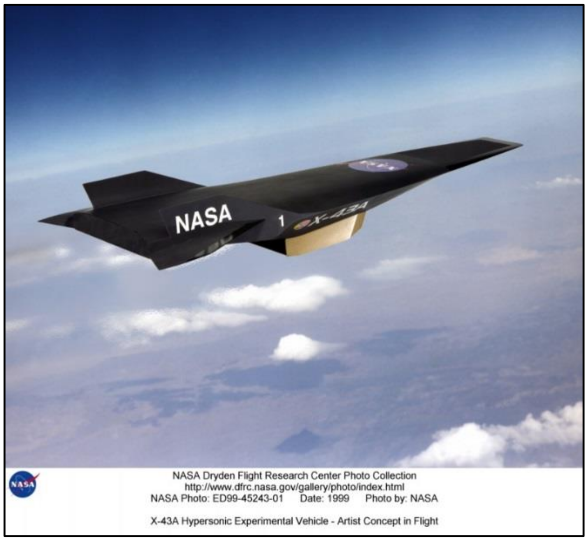

NASA
1
X-43A
NASA Dryden Flight Research Center Photo Collection
http://www.dfrc.nasa.gov/gallery/photo/index.html
NASA Photo: ED99-45243-01   Date: 1999   Photo by: NASA
X-43A Hypersonic Experimental Vehicle - Artist Concept in Flight

Figure 2: X-43A Hypersonic vehicle conceptual model. Illustration from the NASA Dryden Flight Research Center photo collection [4]

While not a direct continuation of the X-43A project, the X-51A provided an additional platform for hypersonic research (Figure 3). Four single use vehicles were developed and tested. The final test took place on May 1, 2013. It reached a top speed of Mach 5.1 in its six minute long successful flight, four minutes of which were under its own scramjet propulsion, at which point, its fuel supply was exhausted [5].

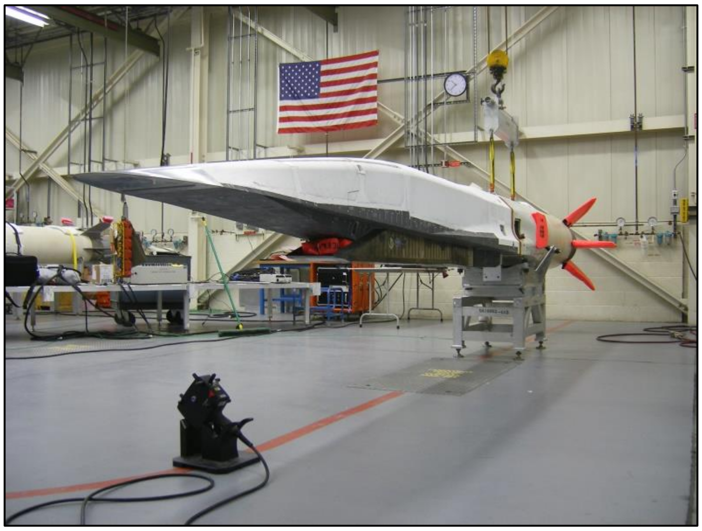


Figure 3: X-51A Hypersonic vehicle being prepared for first flight. Image courtesy of Air Force Flight Test Center Public Affairs[6]

The vast majority of the work done in the development of hypersonic aircraft is not available in open literature and remains proprietary. There are, however, a few models available for review with varying degrees of complexity. One of the first openly available hypersonic aircraft models was developed by Dr. Frank Chavez and Dr. David Schmidt. This system is comprised of a scramjet engine using a one dimensional flow assumption coupled with a two dimensional, Newtonian-based aerodynamic model. A second, nonlinear model for the Longitudinal dynamics was developed by the Air Force Research Laboratory in 2005 [7]–[9]. This three degree of freedom model included propulsion, structural, and aerodynamic coupling.

The model chosen for this work was developed by NASA Langley Research Center using the Aerodynamic Preliminary Analysis System (APAS), a CFD software which produces aerodynamic coefficients from a user defined aircraft model [10],[11]. This model was chosen not for its perfect representation of hypersonic aircraft dynamics, but for its extensive aerodynamic coefficients and its changing mass properties. This model provides a significant amount of aerodynamic detail from which a highly nuanced, nonlinear system may be built. This provides a significant challenge and opportunity to demonstrate advanced control techniques.

1.2 Robust Control

The area of robust control theory is based on the understanding that a control designer never has a perfect mathematical representation of the system to be controlled. It assumes from its outset that there are modeled dynamics within the system that must be tolerated by the controller in order for it to be effective. Additionally, the controller must be able to account for added dynamics due to environmental disturbances or noise present in real world systems. Prior to 1963, designing a controller to tolerate uncertainty and disturbances directly throughout the controller development was not considered. Instead, at the beginning of the design, a designer assumed that the model was “sufficiently accurate” [12].

In 1963, Isaac Horowitz introduced the concept of robust control (though never calling it such) in his book “Synthesis of Feedback Systems”. He presented a method for designing uncertainty-tolerant feedback control for single input, single output systems using the classical root locus approach. Unfortunately, his work went largely disregarded until the mid-70’s [12], [13].

By 1970, significant work in optimal control had been done and was beginning to be applied to complex, real world designs. Linear Quadratic Gaussian (LQG) controllers (a control synthesis approached based on the combination of an optimal Linear Quadratic Regulator and a Kalman Filter) started having applications beyond theory. This was not always successful, however. The lack of robustness caused significant drawbacks. One such application was the controller of a naval submarine. When the submarine was simulated in turbulent seas (an off nominal case for the controller) the system unexpectedly surfaced. A second study done for an F-8C Crusader aircraft had distinctly negative results [12]. It was concluded that there needed to be a "common sense pragmatic (technique) to modify the design based on 'pure' theory" [12].

By 1975, robust control theory was beginning to emerge, including new analysis techniques and design approaches that allowed control designers to tolerate uncertain systems. In 1982, a new approach to the LQG controller was being explored. The classic problem was recast into the frequency domain where the weights used to define the LQG were modeled throughout the system. Instead of applying the typical weighting matrices used in the standard LQG approach, they were replaced with frequency dependent weighting functions, providing resilience against noise in the system. This eventually became known as an H_2 controller [12].

In 1981, another approach was taken to modify the standard LQG framework. Instead of minimizing the 2-norm of the system as is done by the LQG formulation, the ∞-norm (the largest singular value) of the system was minimized. This was done for a single input, single output system. This new formulation was called H-infinity control, hereafter denoted as H_∞. In 1989, Doyle published the H_∞ control problem in a state space format [14]. This extended the control formulation to include Multi-Input Multi-Output (MIMO) systems. With the new formulation, the H_∞ controller surpassed the H_2 controller as the more robust recasting of the LQG controller [12].

1.3 Hypersonic Aircraft Control

Hypersonic flow is a very challenging regime in which to fly an aircraft. Hypersonic aircraft designs typically require an integrated scramjet propulsion system within the airframe. Consequently, the fuselage must create a bow shock wave to generate the pressures needed for the engine to function properly. The resulting pressure distribution causes a nose up tendency. Changing pressure distributions due to the thrust at the aircraft's tail-end add additional dynamics. This has a nonlinear impact on the overall lift, drag, and pitching moments of the final system [9], [10]. Designs having centers of gravity aft of their aerodynamic center adds instabilities into the system. Further complexity is added because hypersonic flight is inherently challenging to model and study as numerous dynamic uncertainties are introduced by the nature of hypersonic flight itself [15]. As a result, hypersonic systems are inherently challenging to control.

In response to the challenge, many control designers have tackled this problem from multiple angles, employing a wide variety of methods. Many variants of H_∞ control have been applied to hypersonic control including pure H_∞, mixed H_∞/H_2, and H_∞ combined with μ-synthesis control techniques.[16]–[19]. In addition, neural network, nonlinear sum-of-squares, sliding mode, and many others control techniques have been successfully employed in simulation [16], [20]–[24].

These models are all generally based on the model presented in Reference [8] or on a model derived therefrom. That specific model only considers Longitudinal aircraft dynamics. In addition, many models only consider a small region within the Longitudinal flight envelope, typically speed around Mach 15 with minimal commanded velocity or altitude changes [19]–[23], [25], [26]. As a result, much of the actual flight envelope remains unexplored.

This presented work aims to expand the studied flight envelope. Both Longitudinal and Lateral-Directional aircraft dynamic are presented and analyzed. The dynamics of the aircraft are studied from Mach 2 to 23, well beyond previously published work. Additionally, a gain-scheduled H_∞ controller is designed to operate on a wide range of Mach numbers and altitudes, beyond what exists today in the open literature.

Chapter 2 Hypersonic Simulation Model

The hypersonic model used is based on the Generic Hypersonic Vehicle (GHV) model developed by NASA Langley Research Center [10]. This model was developed as a method to investigate trajectory optimization, guidance, navigation, stability augmentation, and handling qualities of a single-stage-to-orbit aircraft.

2.1 Vehicle Description

The GHV is a symmetric, winged cone vehicle with a wingspan of 60 feet and length of 200 feet. The wings and vertical tail are in line with the center axis of the aircraft. There are four control inputs to the system: left elevon (δ_{le}), right elevon (δ_{re}), rudder (δ_r), and equivalence ratio (δ_{ER}), or fuel to air ratio. Note that an equivalence ratio of 1 yields the most fuel to air ratio. Values above or below that burn a disproportionate amount of fuel [10].

The elevon deflections are defined in degrees from the hinge line with positive deflection defined as trailing edge up. Note that this is different than the definition defined by Roskam in Reference [27]. The rudder, also measured in degrees from hinge line, is defined as positive deflected to the right. In addition to the four primary control surfaces, there is a canard wing which is deployed at subsonic speeds to assist in Longitudinal stability. Subsonic speeds are not considered in this model so this additional input to the system is not included in the development. A rendering of the vehicle can be seen in Figure 4. The full specifications can be seen in Table 1.

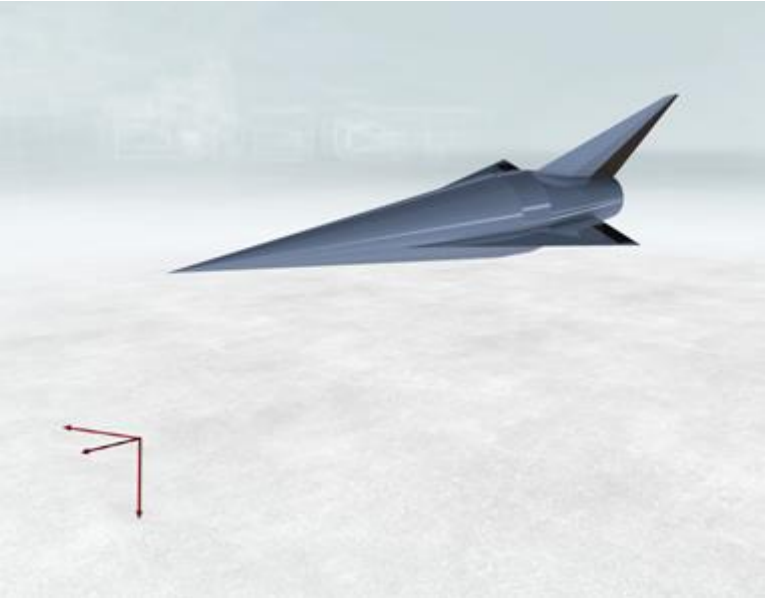


Figure 4: 3D CAD Model of the Generic Hypersonic Vehicle

The center of gravity (CG) of the aircraft is defined with respect to the moment reference center (MRC), the point on the aircraft about which moments are summed. This distance is denoted as $\Delta_{x_{cg}}$. These locations are marked in Figure 5. While the MRC is fixed, the CG changes with fuel burn. Additionally, the Moments of Inertia (I_{xx}, I_{yy}, and I_{zz}) and weight of the aircraft are functions of fuel burn, which, itself, is a function of throttle position (δ_{ER}). The initial gross weight of the aircraft is 300,000 lbs. Note that fuel slosh is not modeled in the simulation. Off axis products of inertia (I_{xy},I_{yx}, I_{xz}, I_{zx}, I_{yz}, and I_{zy}) are assumed negligible [7], [10].

Note that the CG always remains aft of the MRC. This is counterproductive from a stability sense. Whenever positive moments are summed about the MRC, it causes a nose up condition. As the center of gravity is aft of the MRC, it adds additional positive moment. The result is a system that, without any other inputs, will tend to nose up on its own. In order to account for this, additional control forces are needed to keep the system stable.

Table 1: Geometric configuration of the generic hypersonic vehicle

Wings:		
Reference area	3603	ft^2
Aspect ratio	1.00	--
Span	60.0	ft
Leading edge sweep angle	75.97	degs
Trailing edge sweep angle	0.0	degs
Mean aerodynamic chord	80.0	ft
Airfoil section	diamond	--
Airfoil thickness to chord ratio	4.0	%
Incidence angle	0.0	degs
Dihedral angle	0.0	degs
Wing Flaps:		
Area each	92.3	ft^2
Chord	7.22	ft
Inboard section span location	15.0	ft
Outboard section span location	27.28	ft
Vertical Tail:		
Exposed area	645.7	ft^2
Theoretical area	1248.8	ft^2
Span	32.48	ft
Leading edge sweep angle	70.0	degs
Trailing edge sweep angle	38.17	degs
Airfoil section	diamond	--
Airfoil thickness to chord ratio	4.0	%
Rudder:		
Area	161.4	ft^2
Span	22.8	ft
Chord to vertical tail chord ratio	25.0	%
Axisymmetric Fuselage:		
Theoretical length	200.0	ft
Cone half angle	5.0	degs
Cylinder radius (maximum)	12.87	ft
Cylinder length	12.88	ft
Boattail half angle	9.0	degs
Boattail length	40.0	ft
Moment reference center (from tip of cone)	124.01	ft

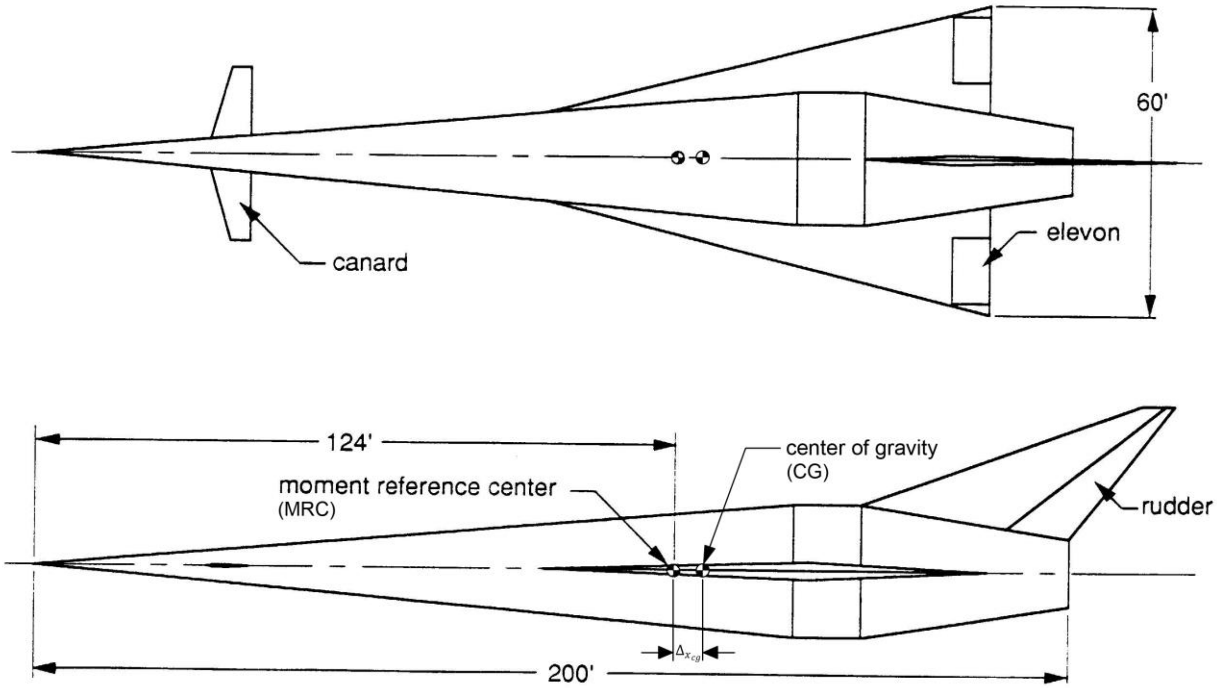


Figure 5: Top and side view of the GHV

2.2 Equations of Motion

This section presents the dynamic relationships used to simulate the GHV. The equations presented herein are derived from basic physical principles in the manner shown in Reference [28].

Coordinate System

Three primary coordinate systems are used in the model. The first is the inertial or flat earth coordinate system denoted with a subscript E. This right hand coordinate system is defined with positive being North, East, and Down as X, Y, and Z respectively. The second coordinate system is the body coordinate system denoted with a subscript B. This coordinate system is affixed to the aircraft with its origin at the center of gravity. X is defined as out the nose of the aircraft parallel to its center line, Y, through the right wing, and Z, down, perpendicular to the centerline (Figure 6). The inertial and body coordinate systems are related by a series of angles Φ, Θ, and Ψ (roll, a rotation about x_I, pitch, rotation about y_I, and yaw, rotation about z_I respectively). Note that all sign conventions are with respect to the right hand rule.

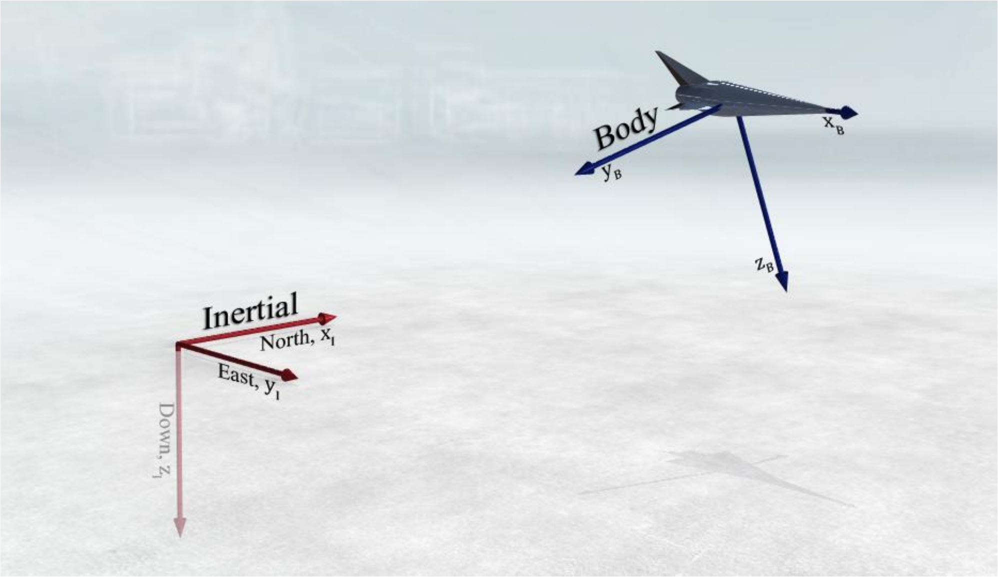


Figure 6: Body and inertial frames

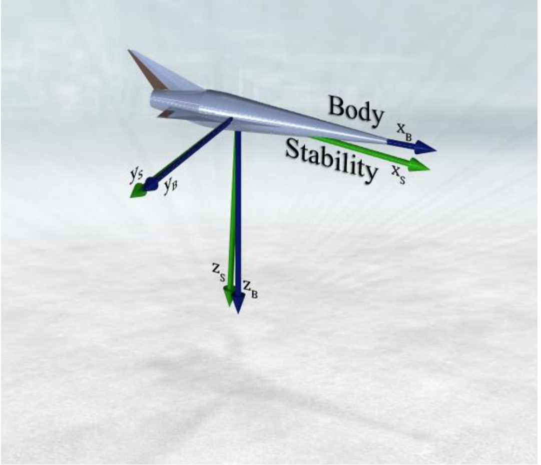


Figure 7: Body and stability frames

The final coordinate reference frame is the stability axis system, denoted with a subscript S. This system is defined as a rotation of the body coordinate system such that the X direction is in line with the velocity vector of the aircraft (Figure 7). These coordinate systems are related by angle of attack, α, and sideslip angle, β. α is defined as a rotation about y_B and β is defined as rotation about z_B. Rotation about the x_B axis is assumed negligible. Once again, positive deflection is with respect to the right hand rule [28].

The principle equations needed to define the equations of motion of the aircraft are defined in the inertial axes, however forces and moments on the aircraft are defined in the body coordinate system. To facilitate this, a transformation matrix is defined to transition between coordinate systems (2-2).

$$[H]_I^B = [H(\phi)]_2^I [H(\theta)]_1^2 [H(\psi)]_I^1 \qquad (2\text{-}1)$$

$$[H]_I^B = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \cos(\theta)\sin(\phi) \\ \cos(\psi)\sin(\theta)\cos(\phi) - \sin(\psi)\sin(\phi) & \cos(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \qquad (2\text{-}2)$$

The forces exerted on the aircraft (lift, drag, and side forces) are represented in the stability axes. A transformation (shown in (2-3)) is used to convert these values into the more convenient body axis system.

$$[H]_B^S = \begin{bmatrix} \cos(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta)\sin(\alpha) & -\sin(\alpha)\sin(\beta) & \cos(\alpha) \end{bmatrix} \qquad (2\text{-}3)$$

Translational Equations

To analyze the motion of the aircraft, first consider Newton’s First Law of Motion which states that the sum of all forces is equal to the change in momentum of a system when examined form an external reference frame:

$$\left[\frac{d\mathcal{P}}{dt}\right]_I = [\sum F]_I \tag{2-4}$$

This can be rewritten with respect to the body reference frame, resulting in Equation (2-5) and the Cross-Product-Equivalent Matrix, Equation (2-6). See [7] for the full derivation.

$$\left[\frac{d\mathcal{P}}{dt}\right]_B = [\sum F]_B - \tilde{\omega}_B \mathcal{P}_B \tag{2-5}$$

$$\tilde{\omega}_B = \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} \tag{2-6}$$

Substituting the definition of momentum, $\mathcal{P} = mV$, into (2-5),

$$\begin{gathered} \left[\frac{d(mV)}{dt}\right]_B = [\sum F]_B - \tilde{\omega}_B m V_B \\ \left[\frac{dm}{dt}\right]_B V + m\left[\frac{dV}{dt}\right]_B = [\sum F]_B - \tilde{\omega}_B m V_B \end{gathered} \tag{2-7}$$

$$\dot{m}_B \begin{bmatrix} U \\ V \\ W \end{bmatrix} + m \begin{bmatrix} \frac{dU}{dt} \\ \frac{dV}{dt} \\ \frac{dW}{dt} \end{bmatrix} = \begin{bmatrix} F_{x_{A+P+G}} \\ F_{y_{A+G}} \\ F_{z_{A+G}} \end{bmatrix}_B - m \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2-8}$$

For many aircraft models, it is safe to assume that the change in mass of the aircraft as it burns fuel is negligible. This is not true in the case of the GHV which burns through a significant portion of its fuel over relatively short flight duration. As a result, the $\dot{m}_B$ term cannot be removed from the equations of motion.

The forces acting on the system can be decomposed into their component parts, which are a summation of the forces acting on the body itself. These forces are aerodynamic (A), propulsive, (P), and inertial gravity (G) forces. Note that this simulation applies propulsive force in only the X-body direction.

Equation (2-8) can be further modified by separating each of the forces acting upon the body:

$$\dot{m}_B \begin{bmatrix} U \\ V \\ W \end{bmatrix} + m \begin{bmatrix} \frac{dU}{dt} \\ \frac{dV}{dt} \\ \frac{dW}{dt} \end{bmatrix} = \begin{bmatrix} F_{x_A} \\ F_{y_A} \\ F_{z_A} \end{bmatrix}_B + \begin{bmatrix} F_{x_P} \\ 0 \\ 0 \end{bmatrix}_B + \begin{bmatrix} F_{x_G} \\ F_{y_G} \\ F_{z_G} \end{bmatrix}_B - m \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2-9}$$

$$\dot{m}_B \begin{bmatrix} U \\ V \\ W \end{bmatrix} + m \begin{bmatrix} \frac{dU}{dt} \\ \frac{dV}{dt} \\ \frac{dW}{dt} \end{bmatrix} = \begin{bmatrix} F_{x_A} \\ F_{y_A} \\ F_{z_A} \end{bmatrix}_B + \begin{bmatrix} F_{x_P} \\ 0 \\ 0 \end{bmatrix}_B + mg \begin{bmatrix} -\sin(\Theta) \\ \cos(\Theta)\sin(\Phi) \\ \cos(\Theta)\cos(\Phi) \end{bmatrix} - m \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2-10}$$

Converting to the standard notation presented in [10], the final equation implemented in the simulation is given in Equation (2-11).

$$\dot{m}_B \begin{bmatrix} U \\ V \\ W \end{bmatrix} + m \begin{bmatrix} \frac{dU}{dt} \\ \frac{dV}{dt} \\ \frac{dW}{dt} \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}_B + \begin{bmatrix} X_P \\ 0 \\ 0 \end{bmatrix}_B + mg \begin{bmatrix} -\sin(\Theta) \\ \cos(\Theta)\sin(\Phi) \\ \cos(\Theta)\cos(\Phi) \end{bmatrix} - m \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2-11}$$

Rotational Equations

Now consider Euler's equation for rigid body rotation as viewed from the Inertial Frame:

$$\left[\frac{d\mathcal{L}}{dt}\right]_I = [\Sigma\mathcal{M}]_I \tag{2-12}$$

This can be rewritten with respect to the body reference frame centered at the CG:

$$\left[\frac{d\mathcal{L}}{dt}\right]_B = [\Sigma\mathcal{M}]_B - \widetilde{\omega}_B \mathcal{L}_B \tag{2-13}$$

Substituting the definition of momentum, $\mathcal{L} = I\omega$, into (2-13),

$$\begin{gathered} \left[\frac{d(I\omega)}{dt}\right]_B = [\Sigma\mathcal{M}]_B - \widetilde{\omega}_B I_B \omega_B \\ \left[\frac{dI}{dt}\right]_B \omega_B + I_B \left[\frac{d\omega}{dt}\right]_B = [\Sigma\mathcal{M}]_B - \widetilde{\omega}_B I_B \omega_B \end{gathered} \tag{2-14}$$

$$\dot{I}_B \begin{bmatrix} P \\ Q \\ R \end{bmatrix} + I_B \begin{bmatrix} \frac{dP}{dt} \\ \frac{dQ}{dt} \\ \frac{dR}{dt} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_{x_A} \\ \mathcal{M}_{y_A} \\ \mathcal{M}_{z_A} \end{bmatrix}_B - \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} I_B \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{2-15}$$

Note again that this simulation assumes the propulsive force is along only the X-axis of the aircraft. Consequently, there is no induced moment due to propulsive force on any axis and all moments are due to aerodynamic effects.

Converting to standard notation of [10], the final equation implemented in the simulation is given in Equation (2-16).

$$\dot{I}_B \begin{bmatrix} P \\ Q \\ R \end{bmatrix} + I_B \begin{bmatrix} \frac{dP}{dt} \\ \frac{dQ}{dt} \\ \frac{dR}{dt} \end{bmatrix} = \begin{bmatrix} L_A \\ M_A \\ N_A \end{bmatrix}_B - \begin{bmatrix} 0 & -R & Q \\ R & 0 & P \\ -Q & P & 0 \end{bmatrix} I_B \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{2-16}$$

One final transformation is needed to complete the rotational equations—specifically, the relationship between the angular rates in the body frame and those on the inertial frame. That relationship is given in Equation (2-17).

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\Theta) \\ 0 & \cos(\Phi) & \sin(\Phi)\cos(\Theta) \\ 0 & -\sin(\Phi) & \cos(\Phi)\cos(\Theta) \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \tag{2-17}$$

2.3 Additional Relationships

In addition to the primary equations of motion of the aircraft, additional relationships are needed to account for atmospheric conditions and a conversion to a standard aircraft model.

Altitude Dependencies

One of the primary outputs of the dynamic model is Mach number. For this calculation, the atmospheric temperature is needed, a parameter dependent on altitude (h). Reference [29] presents the following relationship between temperature and altitude:

$$\begin{gathered} h < 36{,}089\ ft,\ T = T_0(1 - 6.875 \times 10^{-6}\ h) \\ \text{where}\ \ T_0 = 518.7\ °R \\ h \geq 36{,}089\ ft, \qquad T = 389.99\ °R \end{gathered} \tag{2-18}$$

As altitude changes, so too does the air density. Reference [29] gives a relationship between altitude and density:

$$\begin{gathered} h < 36{,}089\ ft,\ \rho = \rho_0(1 - 6.875 \times 10^{-6}\ h)^{4.2561} \\ \text{where}\ \ \rho_0 = 2.377 \times 10^{-3}\ \ slug/ft^3 \\ h \geq 36{,}089\ ft, \qquad \rho = 0.2971\rho_0 e^{-\left(\frac{h-36{,}089}{20{,}806.7}\right)} \\ \text{where}\ \ \rho_0 = 2.377 \times 10^{-3}\ \ slug/ft^3 \end{gathered} \tag{2-19}$$

Aileron/Elevator conversion to Elevon

The standard development for an aircraft controller uses aileron, elevator, rudder, and throttle. The GHV, however, is controlled via left and right elevon, rudder, and equivalence ratio. In order to reconcile these differences and to simplify the controller generation, a conversion must be made. No conversion is needed on either the rudder or the equivalence ratio as the rudder is a standard control surface and the equivalence ratio is a direct analog to throttle. The left and right elevon, however, must be mixed into aileron and elevator. Equation (2-20) shows this procedure.

$$\begin{bmatrix} \delta_{le} \\ \delta_{re} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_e \end{bmatrix} \tag{2-20}$$

Chapter 3 Aerodynamic Coefficients

The dynamic model for the GHV is generated using the Aerodynamic Preliminary Analysis System (APAS) code developed by NASA Langley and Rockwell International Inc. [10]. Force and moment coefficients are estimated based on angle of attack, Mach number, and, when applicable, control deflection. These values provide the substance of the nonlinear model of the GHV. Reference [10] gives all equations and parameters presented in this chapter. Note that the equations shown are taken direction from Reference [10] with only variable names updated for consistency. They represent the hypersonic model as developed within that report. As such, there are some deviations from equations shown in other works such as Reference [27].

3.1 Model Specific Equations

The lift, drag, and side force coefficients are given as follows:

$$\begin{aligned} C_D &= C_{D_\alpha} + C_{D_{\delta_{le}}} + C_{D_{\delta_{re}}} + C_{D_{\delta_r}} \\ C_Y &= C_{Y_\beta}\beta + C_{Y_{\delta_{le}}} + C_{Y_{\delta_{re}}} + C_{Y_{\delta_r}} \\ C_L &= C_{L_\alpha} + C_{L_{\delta_{le}}} + C_{L_{\delta_{re}}} \end{aligned} \tag{3-1}$$

The individual components that make up the lift, drag, and side force are known for any given combination of angle of attack, Mach number, and control surface deflection within the flight envelope. Once these coefficients have been calculated, the total forces on the aircraft as represented in the stability axis system can be calculated:

$$\begin{aligned} D &= \bar{q}S_{ref}C_D \\ Y &= \bar{q}S_{ref}C_Y \\ L &= \bar{q}S_{ref}C_L \end{aligned} \tag{3-2}$$

$\bar{q}$ is the dynamic pressure and S_{ref} is the applicable reference surface area of the aircraft, given in Table 1. To calculate total forces in the body axis, the drag, side force, and lift are transformed to be represented in the body coordinate system:

$$\begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}_B = [H]_B^S \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} = \begin{bmatrix} \cos(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta)\sin(\alpha) & -\sin(\alpha)\sin(\beta) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} \tag{3-3}$$

The total moments on the aircraft due to aerodynamic effects are calculated similarly to the forces above:

$$\begin{gathered} C_l = C_{l_\beta}\beta + C_{l_{\delta_{le}}} + C_{l_{\delta_{re}}} + C_{l_{\delta_r}} + C_{l_P}\left(\frac{P\ b}{2\ V}\right) + C_{l_R}\left(\frac{R\ b}{2\ V}\right) \\ C_m = C_{m_\alpha} + C_{m_{\delta_{le}}} + C_{m_{\delta_{re}}} + C_{m_{\delta_r}} + C_{m_Q}\left(\frac{Q\ c}{2\ V}\right) \\ C_n = C_{n_\beta}\beta + C_{n_{\delta_{le}}} + C_{n_{\delta_{re}}} + C_{n_{\delta_r}} + C_{n_P}\left(\frac{P\ b}{2\ V}\right) + C_{n_R}\left(\frac{R\ b}{2\ V}\right) \end{gathered} \tag{3-4}$$

The terms $\left(\frac{P\ b}{2\ V}\right)$, $\left(\frac{Q\ c}{2\ V}\right)$, and $\left(\frac{R\ b}{2\ V}\right)$ are the computed non-dimensional roll, pitch, and yaw rate, respectively. The total moments about a fixed reference point, the MRC shown in Figure 5, are then calculated:

$$\begin{aligned} L_{mrc} &= \bar{q}bS_{ref}C_l \\ M_{mrc} &= \bar{q}cS_{ref}C_m \\ N_{mrc} &= \bar{q}bS_{ref}C_n \end{aligned} \tag{3-5}$$

c and b are the reference lengths mean geometric chord and wingspan, respectively. Finally, the moments are transferred to the center of gravity:

$$\begin{gathered} L_A = L_{mrc} \\ M_A = M_{mrc} - \Delta_{x_{cg}} Z_A \\ N_A = N_{mrc} + \Delta_{x_{cg}} Y_A \end{gathered} \tag{3-6}$$

3.2 Parameter recovery

The dynamic equations presented in the previous section rely on numerous coefficients which are presented in Reference [10]. However, a digital record of these parameters was not available. All data presented in the report was in the form of plots such as the one in Figure 8. In order for this data to be used to generate the dynamic model of the aircraft, the data had to be extracted from each plot. Recovering these data points by hand would be prone to error as precise scaling is difficult to reproduce on paper. For this model, Siemens NX 7.5 CAD software was used.

A digital version of each plot was imported into the CAD software and a coordinate system was placed on the axis. A scaling factor corresponding to the y-axis of the plot was applied to the coordinate system. This then allowed each individual point to be read directly and copied into a database. This was done for all relevant plots.

To demonstrate the results of this conversion, Figure 9 shows the reproduction of the coefficients shown in Figure 8 using a spline interpolation table. The two data sets match very well. Another option for verifying the conversion process would be to collect the data numerous times using the same technique. Once done, an average value may be taken between all collected data sets and a statistical level of confidence for the conversion may be determined. Potential errors in the conversion may cause slightly different dynamic characteristics. This type of analysis was not performed for this work.

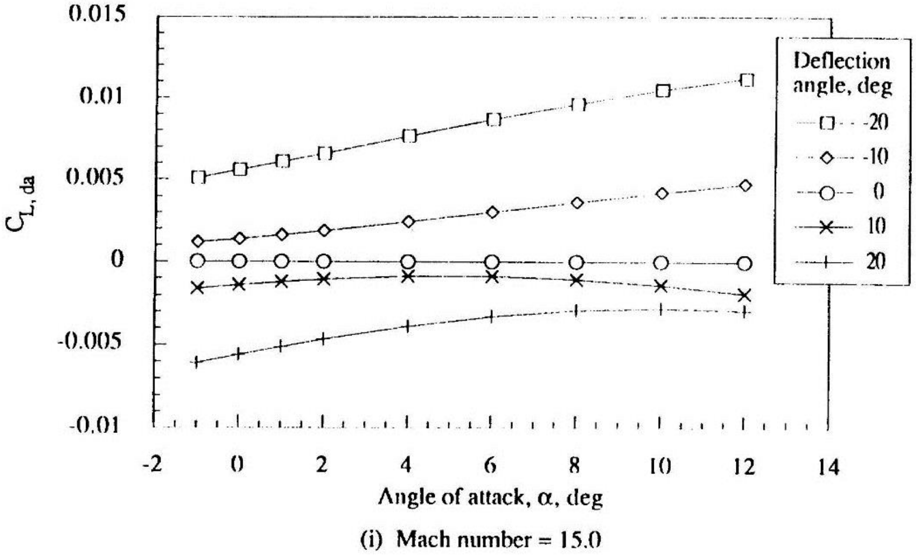


Figure 8: Example original data set, coefficient of lift due to right elevon deflection [10]

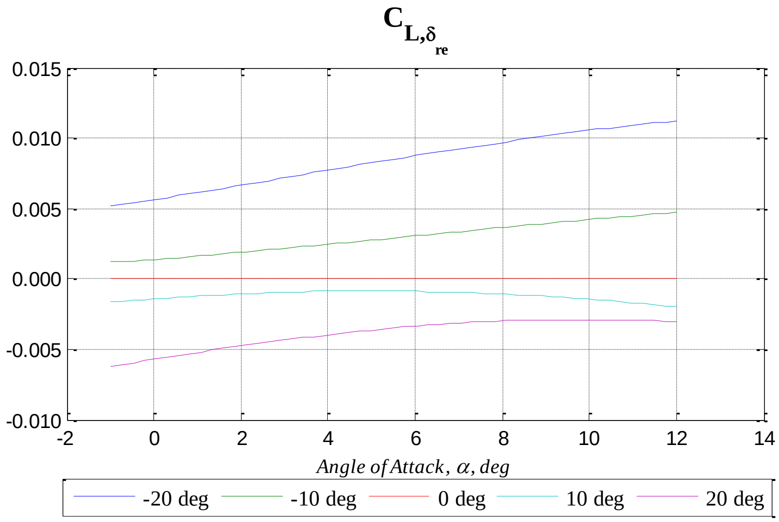


Figure 9: Example spline interpolation, coefficient of lift due to right elevon deflection

3.3 Model Integration

To generate the dynamic model of the aircraft, MATLAB Simulink® R2013a technical computing software was used. As well as being a non-sequential equation solver, the primary benefit of this approach is the ability to use multidimensional lookup tables within the model. Previous implementations of this version of the GHV model have primarily used closed form, nonlinear fit equations to represent the aerodynamic coefficients [7], [25], [26]. This method demands less computational power than the model presented in this work. However, the consequence of using such techniques is that any derived model is only valid for the small region about which the fit equation is applicable. Model fidelity degrades beyond these points. As a result, only a small portion of the flight envelope may be examined.

For this model, all coefficient data was stored in cubic spline lookup tables. Many of these tables are three dimensional, i.e., they depend on three parameters: Mach, angle of attack, and control surface deflection angle. Each of the 34 lookup tables driving the dynamic model use a multidimensional third order interpolation algorithm to interpolate between the various known points collected as described in Section 3.2. More detail on cubic spline interpolation can be seen in Chapter 6.

Each of the equations given in the sections above were converted into a block diagram representation. These blocks were then strung together to form the cohesive nonlinear system model. An example of the converted block diagram can be seen in Figure 10. This is the Simulink® representation of Equations (3-4), (3-5), and (3-6).

Figure 11 illustrates the need for the lookup tables used in this work. The figure shows one of the 34 parameters represented by the spline interpolation. This particular value, C_{m_α}, only varies with Mach number and α. Other parameters, such as $C_{m_{\delta_r}}$ (shown in Figure 10) are even more difficult to represent as they have an additional input variable. It would be challenging, if not impossible, to represent the surface shown with a single, closed form function. The spline interpolation tables, while more computationally expensive, allow for a wider study of the aircraft's performance throughout its entire flight envelope. Additional aerodynamic coefficient plots may be found in Appendix 1.

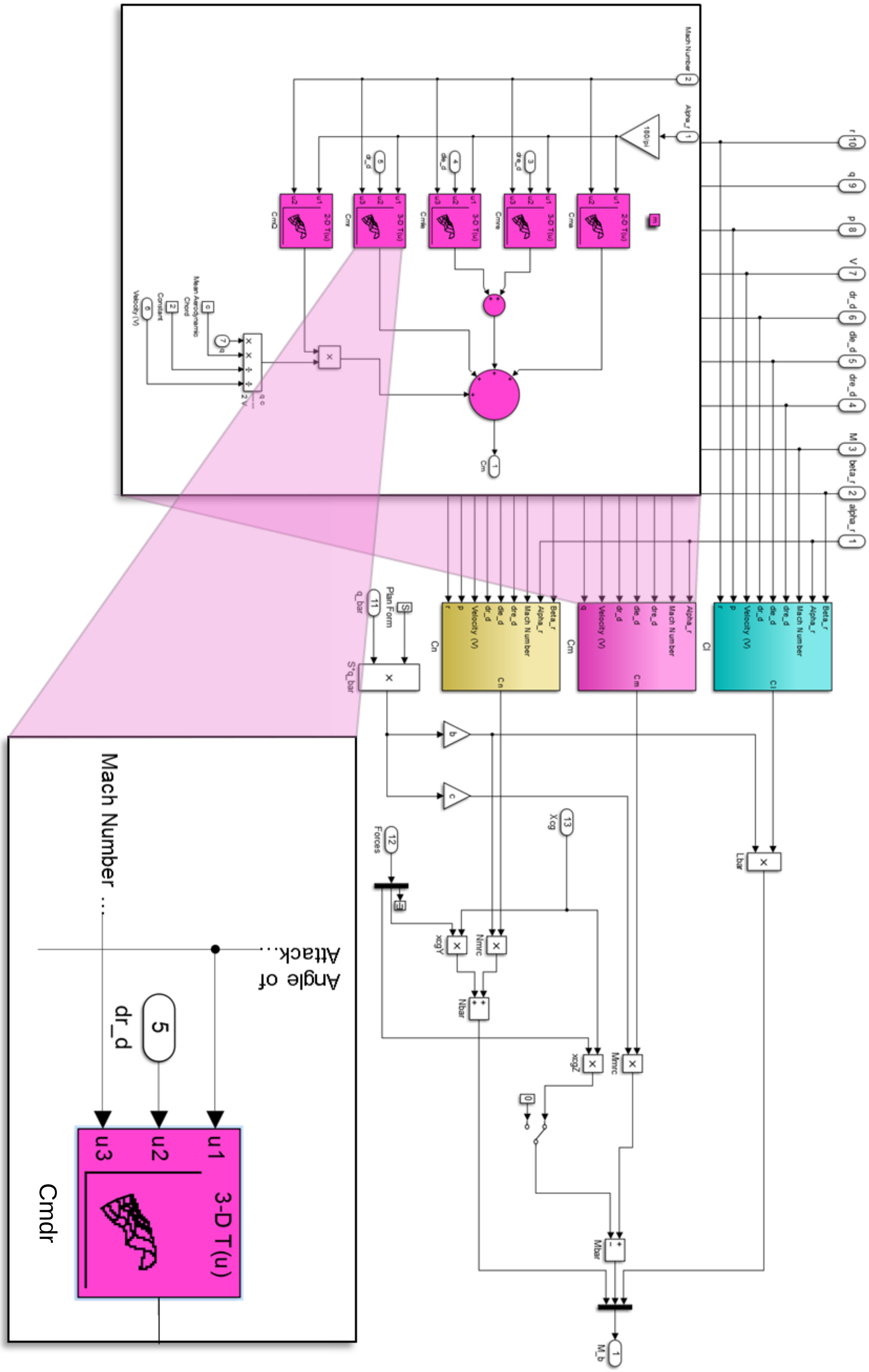


Figure 10: Simulink® block diagram example with cubic spline lookup tables

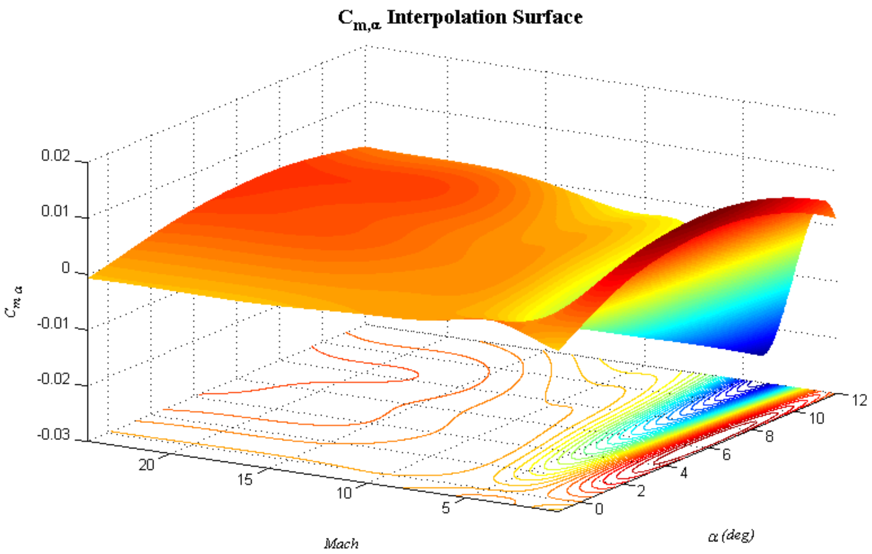


Figure 11: Example aerodynamic coefficient as generated by spline lookup table More coefficients may be found in Appendix 1

Chapter 4 Linear Time Invariant Model of GHV Over a Broad Range of Mach Numbers

The GHV is a highly nonlinear dynamical system. While there are several sine and cosine terms embedded within the dynamic equations, this is not the most significant source of nonlinearity. As can be seen in Figure 11 and Appendix 1, the aerodynamic coefficients vary substantially throughout the possible flight envelope. These variations are made even more significant by the large mass flow rates seen in hypersonic flight.

In order to analyze and control the GHV, a Linear Time Invariant (LTI) model must first be developed for a trim point. Due to the highly nonlinear nature of the GHV, the system must be linearized about multiple trim points in order for the model to be controlled over even a small range within the entire flight envelope.

4.1 State Space Linear Models

While many advances have been made in the development of nonlinear system analysis and control, the majority of techniques available are for linear systems. The GHV, however, has been developed as a nonlinear system. To bridge this divide, the nonlinear system must be placed into the LTI state space formulation given in Equation (4-1). The process for this conversion is given in Section 4.2.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{4-1}$$

The state space formulation is the standard representation for modern control theory. It consists of two parts—the state equation and the output equation. The state equation is a first order differential matrix equation consisting of states in the state vector $x(t)$. In the case of the GHV, there are 13 state variables (Equation (4-2)). The derivatives of the states are equal to a linear combination of the states plus a linear combination of inputs, $u(t)$, to the system, shown in Equation (4-3).

$$x(t) = [\underbrace{North, East, h}_{Position}, \underbrace{u, v, w}_{Velocity}, \underbrace{\phi, \theta, \psi}_{\substack{Rotation\\Angles}}, \underbrace{p, q, r}_{\substack{Angular\\Rates}}, Weight]^T \tag{4-2}$$

$$u(t) = [\delta_a, \delta_e, \delta_r, \delta_{er}]^T \tag{4-3}$$

The output equation, y, is a linear combination of the states plus a linear combination of the system inputs. $y(t)$ typically represents the physically measureable quantities in the system and are, therefore, the only signals available to the control designer when designing a controller.

To analyze the characteristics of any system, the A, B, C, and D matrices are manipulated and examined. These matrices are real value matrices representing the dynamics of the system and the sensors measuring that system.

4.2 Linearization Method

In order to take advantage of many of the tools used to analyze and control a system, a nonlinear system must be reduced to a linear system. Consider the nonlinear system shown in Equation (4-4). This is a generic, nonlinear dynamic system with states vector $x(t)$, system inputs vector $u(t)$, and outputs vector $y(t)$ as well as a time variable t. The functions f and g are nonlinear combinations of the states, inputs, and time variable.

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t), t] \\ y(t) &= g[x(t), u(t), t] \end{aligned} \tag{4-4}$$

Both the states and system inputs may be expressed as the sum of a nominal value and a perturbed value, as shown in (4-5). Substitute these values into (4-4) to get (4-6).

$$
\begin{aligned}
x(t) &= x_0(t) + \Delta x(t) \\
y(t) &= u_0(t) + \Delta u(t) \\
y(t) &= y_0(t) + \Delta y(t)
\end{aligned}
\tag{4-5}
$$

$$
\begin{aligned}
\dot{x}_0(t) + \Delta\dot{x}(t) &= f[x_0(t) + \Delta x(t), u_0(t) + \Delta u(t), t] \\
y_0(t) + \Delta y(t) &= g[x_0(t) + \Delta x(t), u_0(t) + \Delta u(t), t]
\end{aligned}
\tag{4-6}
$$

Equation (4-6) is then linearized by taking a first order Taylor Series expansion of the right hand side of the equation about nominal values as defined in Equation (4-5), the result of which is shown in Equation (4-7) with four partial derivative matrices or Jacobians.

$$
\begin{aligned}
\dot{x}_0(t) + \Delta\dot{x}(t) &\approx f[x_0(t), u_0(t), t] + \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_0(t)\\ u=u_0(t)}} \Delta x(t) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x_0(t)\\ u=u_0(t)}} \Delta u(t) \\
y_0(t) + \Delta y(t) &\approx g[x_0(t), u_0(t), t] + \left.\frac{\partial g}{\partial x}\right|_{\substack{x=x_0(t)\\ u=u_0(t)}} \Delta x(t) + \left.\frac{\partial g}{\partial u}\right|_{\substack{x=x_0(t)\\ u=u_0(t)}} \Delta u(t)
\end{aligned}
\tag{4-7}
$$

In general, the terms $x_0(t)$, $u_0(t)$, and $y_0(t)$ and the time variable t may be nonlinear and time varying. However, when these nominal values are chosen carefully such that they describe a system in steady state, they may be assumed constant (i.e. $x_0(t) = x_0$, $u_0(t) = u_0$, and $y_0(t) = y_0$). Additionally, if all time varying parameters are accounted for by the state variables for a given nominal value, the time term, t vanishes. This results in the Jacobian matrices becoming constant values. For notational simplicity, define each matrix as in Equation (4-8). By equating like terms in Equation (4-7) and making the variable substitutions, Equations (4-9) and (4-10) are formed [28].

$$
\begin{aligned}
\left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_0\\ u=u_0}} \equiv A, &\qquad \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x_0\\ u=u_0}} \equiv B \\
\left.\frac{\partial g}{\partial x}\right|_{\substack{x=x_0\\ u=u_0}} \equiv C, &\qquad \left.\frac{\partial g}{\partial u}\right|_{\substack{x=x_0\\ u=u_0}} \equiv D
\end{aligned}
\tag{4-8}
$$

$$\begin{aligned}\dot{x}_0 &= f(x_0, u_0) \\ y_0 &= g(x_0, u_0)\end{aligned} \tag{4-9}$$

$$\begin{aligned}\Delta\dot{x}(t) &= A\Delta x(t) + B\Delta u(t) \\ \Delta y(t) &= C\Delta x(t) + D\Delta u(t)\end{aligned} \tag{4-10}$$

Equation (4-10) is known as the linear state space model of the system. For notational simplicity, the Δ's are omitted from the equations resulting in the form shown in Equation (4-1). Note that when referring to a specific state variable, a capital letter refers to the total value while a lowercase refers to the perturbed state (for example, $\Phi(t) = \phi_0 + \phi(t)$). In order to recover the total values from the perturbed state, the initial trimmed condition must be added into the states, Equation (4-2), and control deflection, Equation (4-3).

The task of simplifying a system into a linear representation thus consists of two tasks: finding a nominal or trim value about which to linearize the system, and calculate the Jacobians of the system at that trim point (i.e., the $A, B, C,$ and D matrices). There are several methods to accomplish this task. The one chosen for this analysis uses the functionality built into the Simulink® Control Design Toolbox to produce a linearized model of the system.

First, a trim point must be chosen. Aircraft models are typically linearized about an equilibrium point where body accelerations and angular rates are constant. For this analysis, only straight and level flight conditions were considered. Under this assumption, angular rates and body accelerations are held at zero, while angles and velocities are held constant. Note that in order to have straight, coordinated flight, β is held at zero in the absence of external disturbances. Additional trim conditions (such as constant rate turn) may also be considered with this model, however, they are beyond the scope of this work.

In addition to the requirement for straight and level flight, the GHV has three variables which also drive the aircraft's specific trim condition—namely Mach, weight, and altitude. By specifying each of these three parameters, the remaining initial conditions and trimmed inputs to the system can be found numerically.

To determine initial conditions of each state and the value of the trimmed inputs, a numerical optimizer built into Simulink's$^{®}$ linearization tool was used, referred to as the "graddescent_elim" or Gradient Descent with Elimination [30]. The optimizer uses a gradient decent method to find the operating point which yields a steady state condition in the model. To do this, the algorithm sets all the known states, inputs, and outputs (such as the known altitude, weight, and Mach number) as constants then applies the gradient descent technique to find what set of inputs will produce states with derivatives of zeros. It is not desired to have zero derivatives in some states, such as the mass of the aircraft or position, as this would not be representative of reality. In these cases, the quantities are known, but the derivative is allowed to be non-zero [30].

The gradient descent method referenced above consists of numerically calculating the gradient of the system at some initial guess point, taking a step in the direction the steepest negative gradient, then recalculating the gradient at the new point. The algorithm repeats in this manner until a local minimum is found. The details of the specific algorithm employed within the MATLAB tool are well beyond the scope of this work, but may be found in the software documentation, Reference [30], [31].

Once the trim condition has been identified, the second step is to linearize the system. Simulink$^{®}$ uses a block-by-block linearization technique to calculate its linear models. The advantage of this technique is that most of the prebuilt Simulink® blocks have an exact linearized representation already coded. For blocks which do not have a predefined linearization, such as a lookup table, a perturbation linearization technique is used to calculate the Jacobian matrices for the block numerically. Each state and input to the block is perturbed independently by $10^{-5}(1+|\mathcal{X}|)$ where $\mathcal{X}$ is the input or state being perturbed. The result is then recorded as shown in Equation (4-11) where i indicates the state or input being perturbed, x is a system state, u is a control input, y is a system output, and subscript p indicates the perturbed value. Once all blocks have been linearized about the specified trim condition, they are then combined to produce a single linear system to represent the model [32].

$$\begin{aligned} A(:,i) &= \frac{\dot{x}|_{x_{p,i}} - \dot{x}_0}{x_{p,i} - x_0}, \qquad B(:,i) = \frac{\dot{x}|_{u_{p,i}} - \dot{x}_0}{u_{p,i} - u_0} \\ C(:,i) &= \frac{y|_{x_{p,i}} - y_0}{x_{p,i} - x_0}, \qquad D(:,i) = \frac{y|_{u_{p,i}} - y_0}{u_{p,i} - u_0} \end{aligned} \tag{4-11}$$

4.3 Model Verification

Before the dynamic model could be considered complete and accurate, the entire system implementation must first be validated. Reference [25] and [26] present a Longitudinal dynamic model derived from the same data set used to generate this presented model for the hypersonic aircraft. This provides a point against which to verify the developed GHV model.

The model used in [25] and [26] (hereafter referred to as The Stengel Model) are generated using polynomial fit equations for each of the relevant stability and control derivatives. This works well around a specific trim condition. Beyond the trim points, however, the fidelity of the equations degrades. The model is trimmed at $M = 15$, $h = 110{,}000\ ft$, $\alpha = 0.0315\ rad$ (1.80°), $\delta_{ER} = 0.183$, and $\delta_e = -0.0066\ rad$ $(0.378°)$. At this trim state, the model has a lightly damped Phugoid mode of $-0.0001 \pm 0.0263j$, a real Short Period mode pair at -0.8 and 0.687, and a slightly unstable altitude mode at 0.0008. These five modes are detailed in Table 2 and plotted in red in Figure 12. These values will be referred to as the Validation Modes.

To verify the GHV model developed in Simulink®, the fit equations described in the Stengel Model are substituted in place of the lookup table coefficients. If the dynamic models are identical, the Longitudinal modes produced by the Simulink® model with the aerodynamic coefficient equations from the Stengel Model when linearized about the same trim point should be identical to those of the Validation Modes. These modes are shown in green in Figure 12 with additional detail given in Table 3.

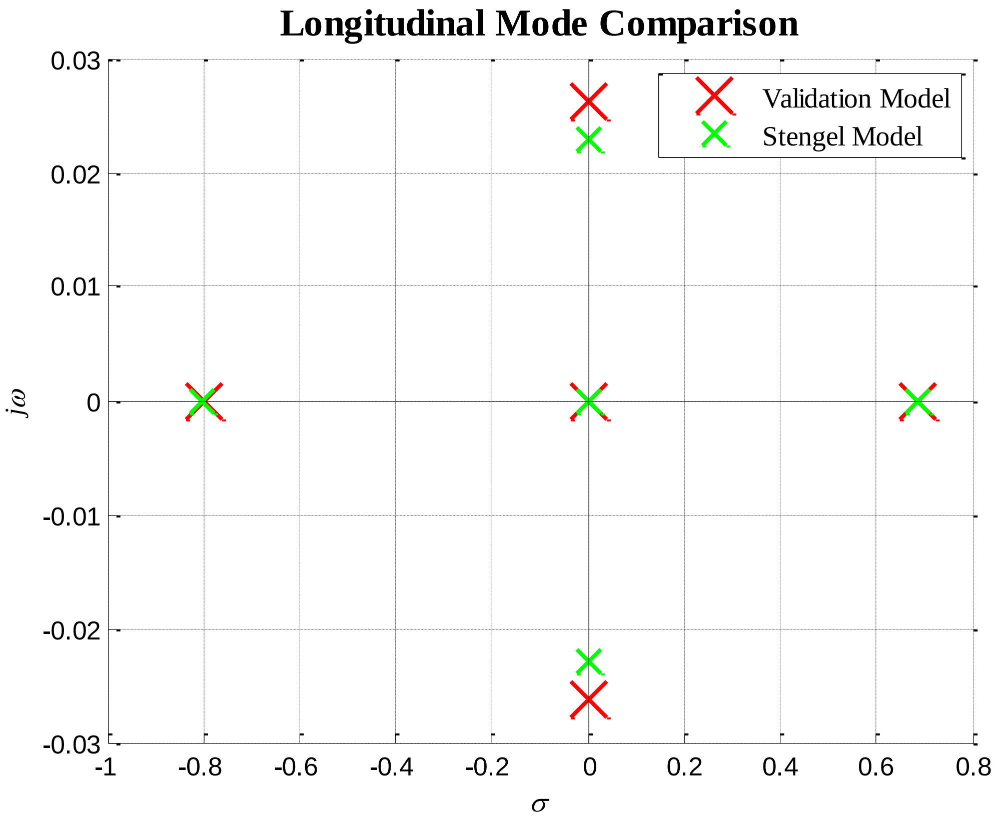


Figure 12: Longitudinal mode comparison

Table 2: Validation mode characteristics

Validation Modes					
Mode	Pole Value	Time Constant	Damping Ratio	Natural Frequency	Damped Natural Frequency
Phugoid	$1\times10^{-4} \pm 0.0263j$	10.00×10^{3}	0.0038	0.0263	0.0263
Real Short Period	-0.8	1.25	1	0.800	0
Real Short Period	0.687	-1.46	-1	0.687	0
Altitude	0.8×10^{-3}	-1250	-1	0.800×10^{-3}	0

Table 3: Stengel Model mode characteristics

Stengel Model Modes					
Mode	Pole Value	Time Constant	Damping Ratio	Natural Frequency	Damped Natural Frequency
Phugoid	$3\times10^{-4} \pm 0.0229j$	3.00×10^{3}	0.0146	0.0229	0.0229
Real Short Period	-0.801	1.25	1	0.801	0
Real Short Period	0.687	-1.46	-1	0.687	0
Altitude	1.60×10^{-3}	-620.	-1	1.60×10^{-3}	0

The Validation Modes and the Stengel Model Modes are very well correlated. The real Short Period modes are identical. The Phugoid mode and the Altitude modes, however, show slight variation. This can be attributed to two subtle variations. First, the Validation model used a non-uniform gravity within its development, while the presented model used a constant gravity assumption. References [33] and [34] show that this variation will affect the location of the Phugoid Mode poles. A second source for the variation is in potential lack of precision in the functions describing the aerodynamic coefficients presented in [26]. The primary difference between the two sets of modes lie in their real component, both of which are very small relative to the other poles. This portion governs the time constant of each mode. Due to their proximity to the origin, very small deviations will appear as large deviations in time constants as the time constant is inversely proportional to the dynamics within the system. As a consequence, very small changes within the system will result in relatively large changes in the final pole location when near the imaginary axis.

From a validation perspective, the near identical linearized models verify that the equations of motions represented in the simulation are equivalent. Note that this verification is only performed on the Longitudinal equations of motion. Unfortunately, no model can be found which reproduces independently the Lateral-Directional behavior of the aircraft, and therefore, there is no reference for comparison. Note, however that the equations of motion implemented in the full simulation are the full development of the dynamic equations, and not just the Longitudinal axis subset. It stands to reason that, if the development of the Longitudinal equations of motions is correct, the Lateral-Directional equations should be similarly accurately reproduced.

An additional simplifying assumption made in the Validation Model was how the aircraft model handles weight. This model uses a constant CG collocated at the MRC of the aircraft. The Simulink® model presented in this document, on the other hand, uses a changing center of gravity with a constantly decreasing weight dependent on the throttle position. In order to compare the Stengel Model and the Validation Model, these added complexities were disabled.

As further comparison, the linear model of the system with all the lookup tables incorporated was calculated. The model was trimmed about the same equilibrium point as the Validation Model. The C.G. = MRC assumption was also retained. The resulting Longitudinal modes were markedly different than The Stengel Model. The results of this linearization can be seen in Table 4.

Table 4: Lookup Table Model mode characteristics

Lookup Table Model Modes					
Mode	Pole Value	Time Constant	Damping Ratio	Natural Frequency	Damped Natural Frequency
Phugoid	$0.0036 \pm 0.0036j$	0.277×10^{3}	0.707	0.0051	0.0036
Real Short Period	1.84	0.544	1	1.84	0
Real Short Period	1.76	-0.568	-1	1.76	0
Altitude	8.40×10^{-3}	-119.	-1	8.40×10^{-3}	0

These variations are understandable, however. It was found that the modes are highly dependent on the specific trimming values within the system. Unlike in the previous comparison where the trim values are identical between the models, this model required a completely different set of trimmed values. The two models have subtle differences which cause a large impact on the system as a whole. The differences arise from how the aerodynamic coefficients have been recovered. The process employed to recover the points used to generate the fit equations in [25] and [26] is not expressly stated, however. As discussed in Section 3.2, this process is prone to errors. As a result, the values used to generate the disparate linear models will be slightly different, resulting in different trimmed state values. This difference is amplified by the implementation equations (shown in Section 3.1). At a trimming Mach number of 15, the mass flow rate over the wings is substantial. As can be seen in Equation (3-4), this amplifies the effects of small variations in aerodynamic coefficients on the overall system model.

Despite these differences, there are some similarities between the models. Both models show a complex Phugoid pair and two real Short Period modes as well as an altitude mode near zero. This shows that, while the magnitude of the responses is different between the models, the dynamic trends are similar.

4.4 Mode Analysis

To analyze the characteristics and stability of an aircraft, it is typical to examine the linearized system's eigenvalues and their associated eigenvectors. The highly nonlinear GHV model was linearized at multiple trim points, resulting in numerous, distinct linear models. As was mentioned in Section 4.2, the trim conditions of the GHV vary independently with Mach, weight, and altitude. After examining the three independent variables, it was found that the greatest deviations in the system are due to changes in Mach number. For this reason, only the linear models relating to changing Mach number are discussed in detail in this document. Altitude and weight are held constant at $70{,}000\ ft$ and $225{,}000\ lbs$ respectively. The effects of the remaining two independent variables are shown in Appendix 2.

To identify the characteristics of each mode of the GHV, the linear models were divided into the Lateral-Directional and Longitudinal components. The modes were then plotted on a single complex plane to see their changing behavior over the range of Mach numbers. The resulting shapes are analogues to the root locus of a feedback system where the point along the root locus is dictated by the Mach number of the point about which the system is linearized.

Unlike many common aircraft, the GHV's modes are not easily categorized into the characteristic aircraft modes such as Phugoid and Short Period for the Longitudinal systems and Roll, Spiral, and Dutch-Roll for Lateral-Directional axes. Instead, a more detailed analysis of each mode's unit eigenvectors is required. Each entry of the complex eigenvectors is represented in its polar form, $A_n e^{j\omega_n}$, where A_n is the relative contribution state n has on the mode and ω_n is the phase shift due to state n. By examining the relative magnitudes of A_n from each state, each mode may be classified. Phase angle does not help to directly classify the mode, but can be used to determine if a state "leads" or "lags" another state for a given mode [28]. As the goal of this section is to classify the modes of the system, only relative eigenvector magnitudes will be studied.

From the eigenvector analysis, each mode of the system is classified. As the linear systems vary as Mach number changes, so too does the magnitude of each eigenvector. The changing values are plotted as a function of Mach number in Figure 14 and Figure 19. It should be noted that the classification applied to a mode at one particular Mach may not be the same as the classification of that same mode at another Mach number. For this reason, only the modes which remain distinct over the entire span of Mach numbers are named in the plot legend.

Also included in the mode analysis is the Altitude mode. This mode is a consequence of including altitude, h, in the states of the system. This mode should be near the origin as it is an integral of the aircraft's vertical velocity in the inertial coordinate system. This mode will not be directly on the origin due to altitude dependent air density, however.

The reader is cautioned that, while this section names each mode present according to the standard aircraft model, this convention may not be wholly applicable to the GHV. Hypersonic flight has significantly different handling qualities than those demonstrated in conventional aircraft. At high speeds, the system has very little resemblance to a typical aircraft. This stretches the applicability of standard naming conventions, particularly in the Lateral-Directional case. For this reason, the author recommends that, in subsequent publications studying hypersonic flight modes, new naming conventions be selected which are more applicable to the hypersonic flight regime.

Lateral-Directional Modes

There have been numerous studies published analyzing the Longitudinal dynamics of hypersonic aircraft. There have been considerably fewer which focus on the Lateral-Directional dynamics of similar systems. What does exist shows that the Lateral-Directional dynamics exhibit some non-standard characteristics. The predominant phenomenon discussed is the presence of a second order Roll/Spiral mode [7], [35], [36]. Note that the dissimilarities between the Lateral-Directional modes of the GHV vs. a more conventional aircraft are not due to the equations of motion, but rather the non-standard aerodynamic coefficients associated with the GHV.

The second order Roll/Spiral mode is referred to as a “Lateral Phugoid” mode. While unrelated to the Longitudinal Phugoid mode, it exhibits the same slow, lightly damped behavior as the Longitudinal analogy. It is typically associated with high angle of attack, low subsonic flight. The result is a coupling between the Roll and Spiral motion [28].

Figure 13 presents the Lateral-Directional modes of the system as they vary with Mach number. At low Mach, all modes of the system are complex. As the Mach number increases, the frequency of each pair increases slightly prior to decreasing to independent real poles. This initial increase and subsequent decrease correlate to a similar rise in the Lateral-Directional aerodynamic coefficients (see Appendix 1). These coefficients each peak at a speed between Mach 2 and 5, as does the frequency of the complex modes, then taper out to more constant values as Mach number increases.

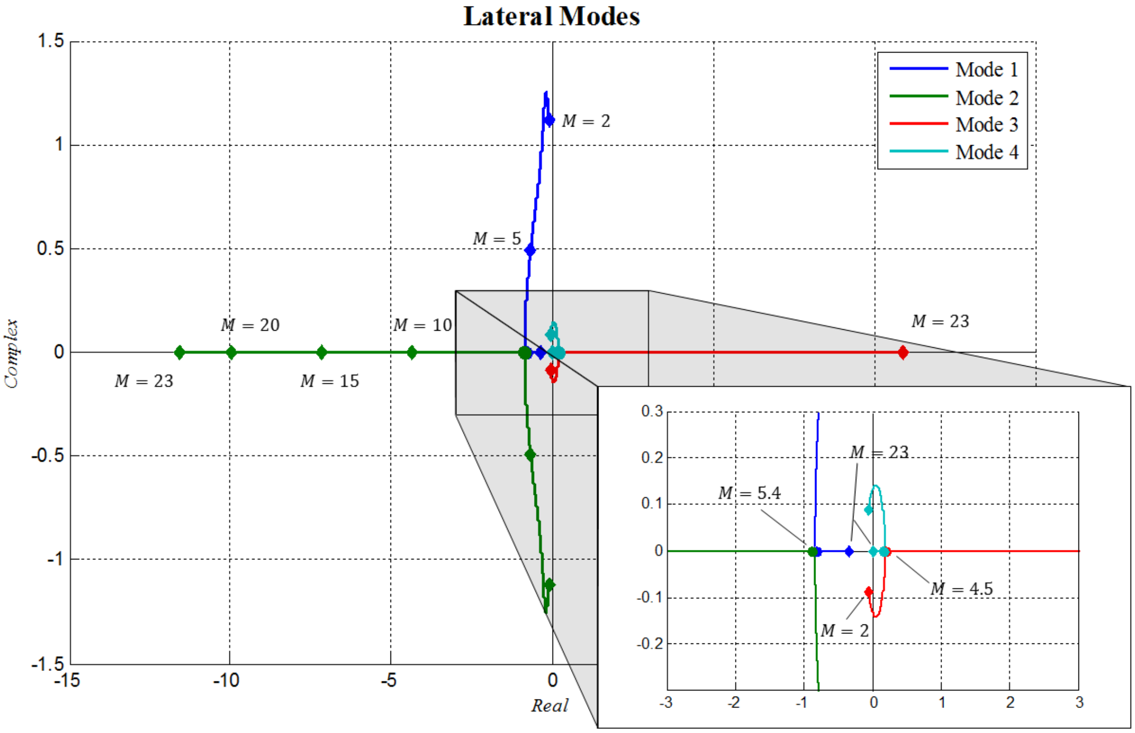


Figure 13: Lateral-Directional modes

The modes shown in Figure 13 can be readily classified below Mach 4.5 due to their relative position in the complex plane. Modes 3 and 4 are recognizable as a Lateral Phugoid mode due to their low damping and frequency. Consequently, modes 1 and 2 are the Dutch-Roll mode. Above Mach 4.5, however, the distinction becomes less apparent. The eigenvectors of the system provide additional insight. Figure 14 gives the normalized, Mach-dependent contribution magnitude from each state to a given mode. The transition point from a complex pair to real mode is also indicated. Each of the relevant Mach ranges are discussed separately.

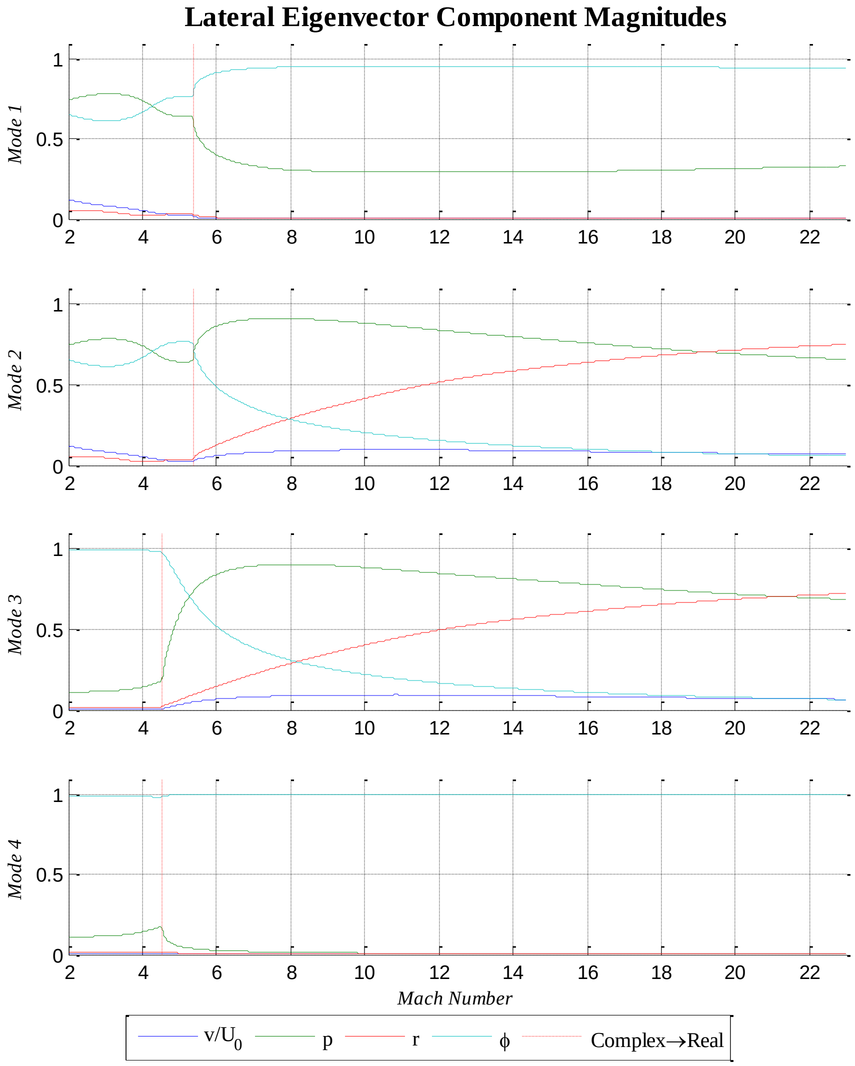


Figure 14: Lateral-Directional mode eigenvectors

First, consider the range of Mach numbers between 2 to 4.5. From examining the location of each mode in Figure 13 above, modes 1 and 2 were identified as Dutch-Roll while 3 and 4 are the Lateral Phugoid. These conclusions are reinforced with the eigenvector plotted in Figure 14. Modes 1 and 2 have a comparatively high contribution from $v/U_0 \approx \beta$, while 3 and 4 are dominated by p and ϕ as is expected from these modes.

As was mentioned earlier, the Lateral Phugoid mode is not typical in aircraft. Reference [7] ties this to a large yawing moment caused by roll rate (C_{n_P}). This is indeed the case with the GHV.

C_{n_P} is large in the region the Lateral Phugoid mode is present (Figure 15). Reference [36] also shows a connection between C_{l_P} and C_{n_R} (Figure 16 and Figure 17). A Lateral Phugoid mode is expected when both of these quantities have large negative magnitudes. Once again, this is the case with the GHV.

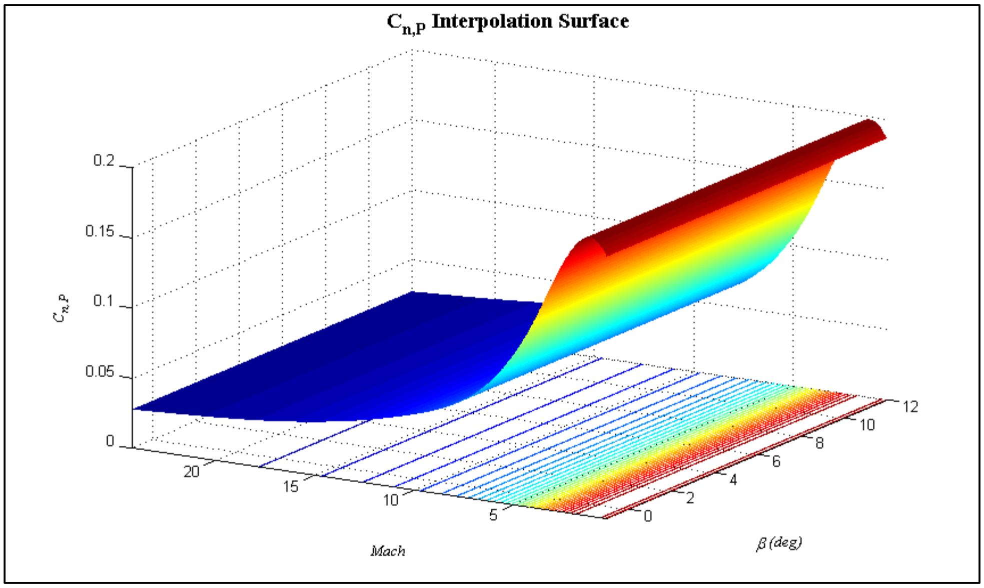


Figure 15: $C_{n,P}$ interpolation surface

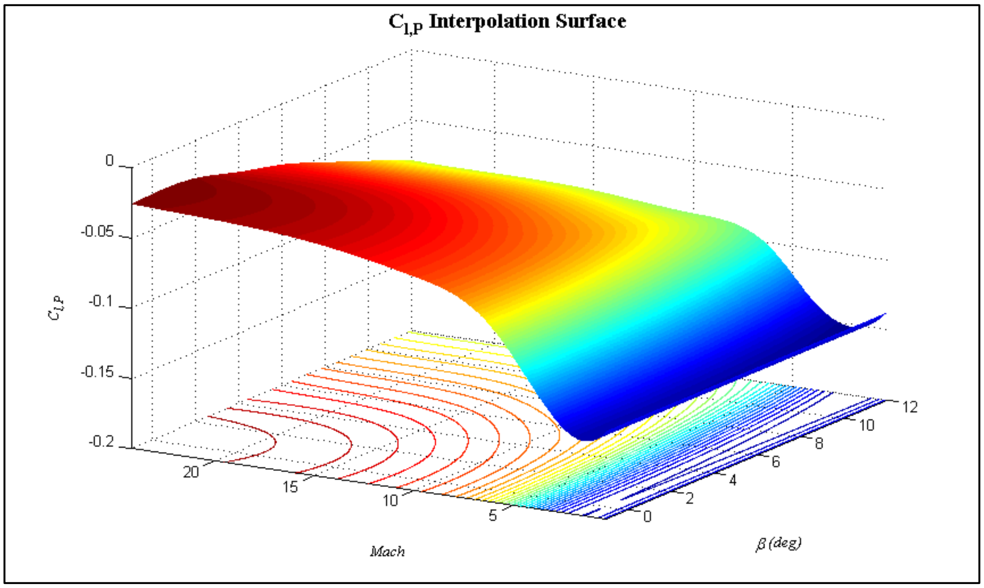


Figure 16: $C_{l,P}$ interpolation surface

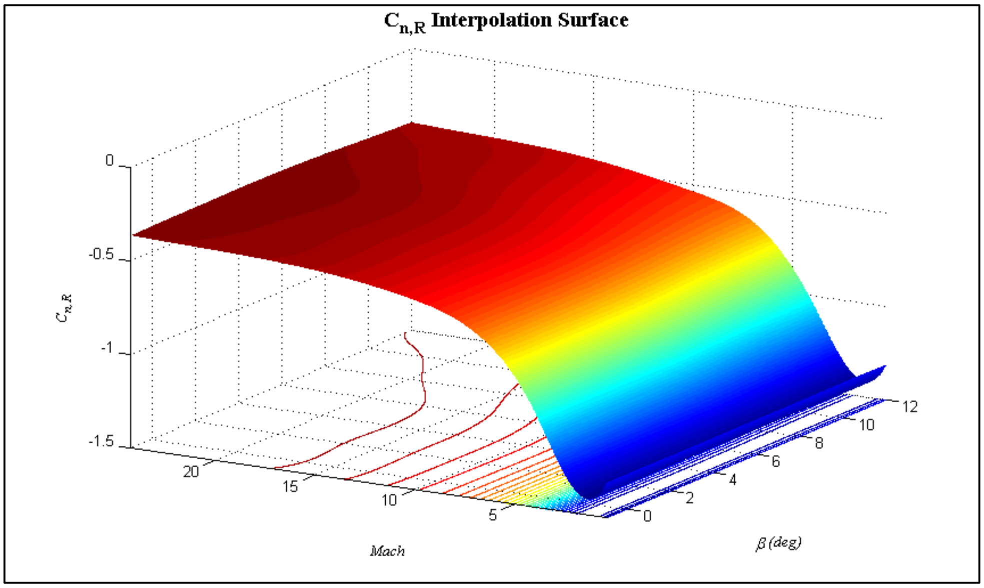


Figure 17: $C_{n,R}$ interpolation surface

Next, consider the Mach numbers ranging from about Mach 6 to 23. From Figure 14, mode 4 is dominated entirely by ϕ. This indicates it clearly as the Spiral mode. Modes 2 and 3 have very similar eigenvectors implying that they are a dynamically coupled mode. (Note that these are not complex conjugates as both mode 2 and 3 are real. Slight variations in the plotted eigenvectors can be observed in these modes.) Compared to the other two modes, mode 2 and 3 have a high contribution from $v/U_0 \approx \beta$. These two factors show that modes 2 and 3 form a real Dutch-Roll mode. Mode 1 is has a comparatively high contribution from p characteristic of the Roll mode.

The region between Mach 4.5 and 6 is harder to classify. While there is the classic complex pair with two real modes of a conventional aircraft between 4.5 and 5.4, their eigenvectors undergo significant changes in this region. In this region, modes 1 and 2 transition from a complex Dutch-Roll mode to two distinctly separate modes (a Roll and a Dutch-Roll mode respectively). Additionally, Modes 3 and 4 transition such that mode 3 joins mode 2 to become a real Dutch-Roll pair, and 4 loses its Roll mode contribution and becomes a pure Spiral mode.

The above conclusions are summarized in Table 5.

Table 5: Lateral-Directional mode classification

<table>
<tr><th>Mach Range</th><th>Mode 1</th><th>Mode 2</th><th>Mode 3</th><th>Mode 4</th></tr>
<tr><td>2-4.5</td><td colspan="2">Dutch-Roll Pair</td><td colspan="2">Lateral Phugoid</td></tr>
<tr><td>4.5-5.4</td><td colspan="2">Dutch-Roll Pair</td><td>Roll to Dutch-Roll</td><td>Spiral</td></tr>
<tr><td>5.4-6</td><td>Dutch-Roll to Roll</td><td colspan="2">Dutch-Roll Pair</td><td>Spiral</td></tr>
<tr><td>6-23</td><td>Roll</td><td colspan="2">Dutch-Roll Pair</td><td>Spiral</td></tr>
</table>

The Lateral-Directional dynamic characteristics can be inferred from these classifications. A real Dutch-Roll pair indicates that, when excited, the aircraft will have a first order unstable response. In other words, the system will not oscillate as it becomes unstable. Rather, it will begin to diverge immediately. The Lateral Phugoid mode will result in an oscillatory rolling effect. The most significant implication of this study, however, is that the region between Mach 4.5 and 6 is the most dynamically nonlinear region of the system. Within this region, the linear models change rapidly and the GHV is, therefore, more challenging to control using linear control techniques.

Due in part to the lack of hypersonic Lateral-Directional studies, previously published models have not identified the transient nature of the poles through a broad spectrum of Mach numbers. It is unique to see modes transition from one well defined state to a different new state such as mode 1 transitioning from Dutch-Roll to Roll and mode 3's transition from Lateral Phugoid to a real Dutch-Roll pair. Additionally, it is unusual to see a real Dutch-Roll characteristic in a system. Both of these phenomenon warrant additional study beyond the scope of this work. This reinforces the idea stated in the opening of this section that standard aircraft mode definitions may not be wholly applicable to aircraft at these speeds.

It should be noted that the Mach ranges presented in this section are only applicable to this particular trim weight and altitude. Reference [36] found the Lateral Phugoid mode at Mach 6.5 for a similar hypersonic aircraft while the presented analysis shows this phenomenon dies out at Mach 4.5. However, this transition Mach number will change as the system is trimmed at different weights and altitudes. Under specific altitude and weight conditions, the GHV has Lateral Phugoid mode at Mach 5 (see Appendix 2). Once again, this shows the GHV's nonlinear response to changing conditions.

Longitudinal Modes

Unlike the Lateral-Directional modes of the system, the Longitudinal modes are much easier to classify using the standard mode definitions. They show predominantly similar characteristics over the entire range of Mach numbers and can, therefore, be classified for the entire range. Figure 18 shows the eigenvalues plotted on the complex plane and Figure 19 gives the magnitude of the unit eigenvectors associated with each mode.

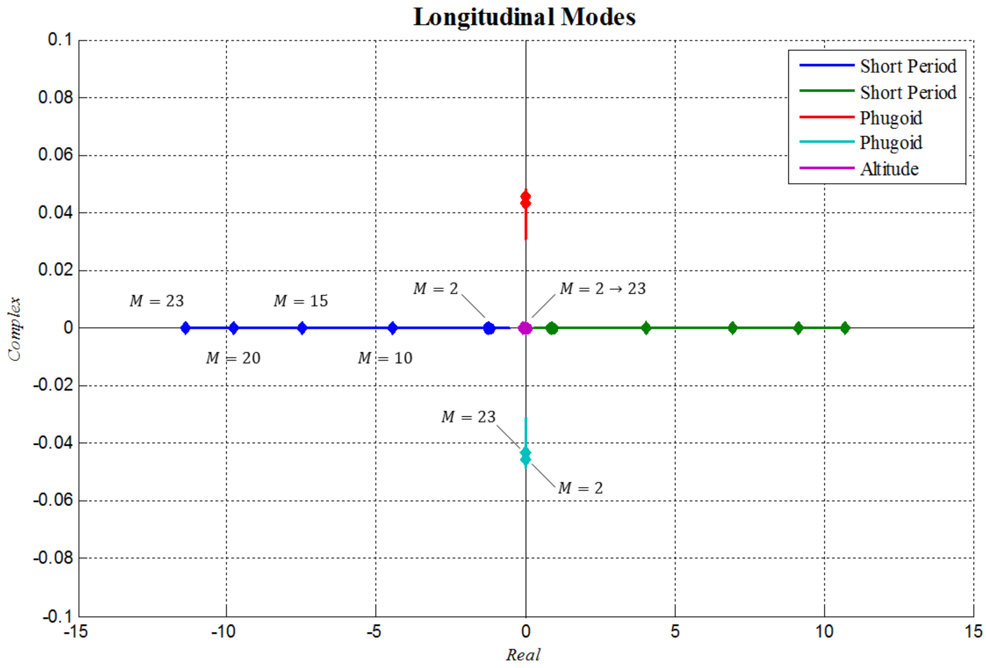


Figure 18: Longitudinal modes

The Short Period mode is classically dominated by q, $w/U_0 \approx \alpha$, and to a lesser extent θ as can be seen in Figure 19. One notable aspect of this mode is that it comprises two real modes. This implies that, when the Short Period mode of the system is excited, it will go unstable in a first order sense and not oscillatory as is typical with a Short Period instability. This same phenomenon was identified in the GHV's dynamics by Stengel as described earlier in this chapter [25], [26]. The primary reason for these two real modes is that the CG is aft of the MRC. In the absence of any other stabilizing characteristics, the aircraft Longitudinal dynamics will not oscillate when becoming unstable. Instead, it will become unstable in a first order manner—in this case as an exponentially increasing pitch up.

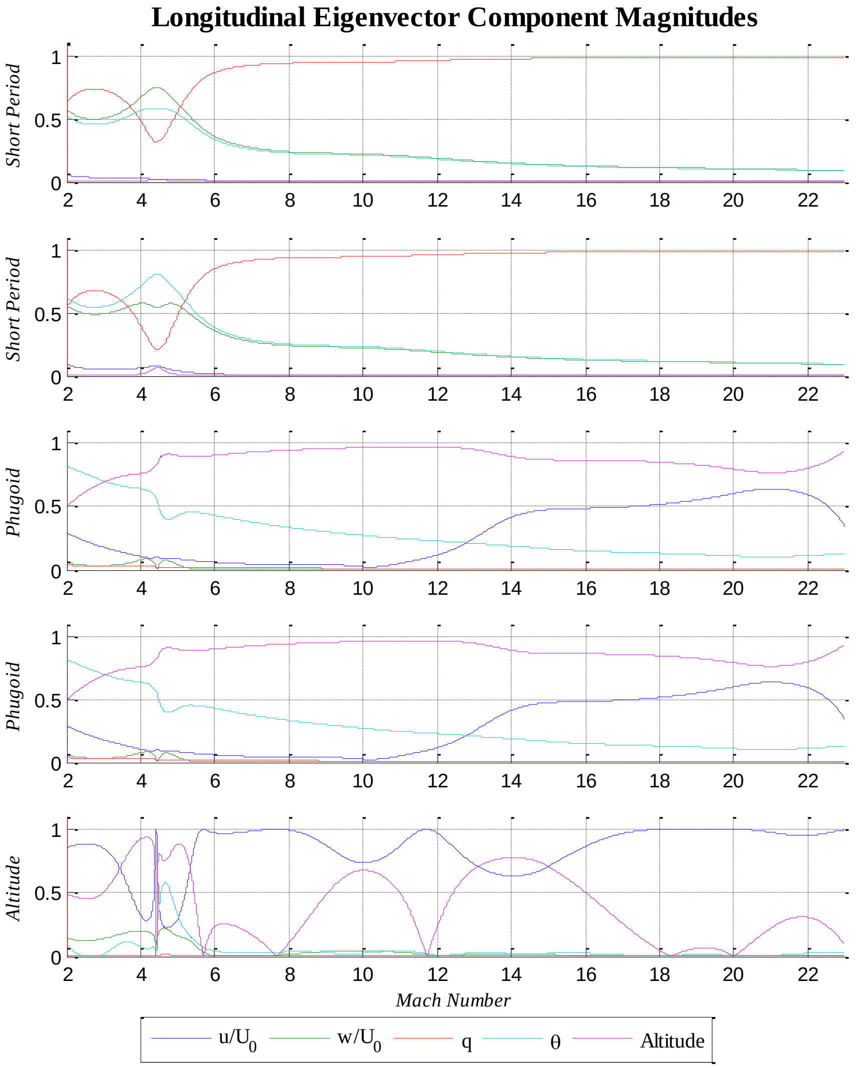


Figure 19: Longitudinal mode eigenvectors

The defining characteristic of a Phugoid mode is that it should be dominated by θ and total velocity, here approximated by u/U_0. These trends are clearly seen in the eigenvectors in Figure 19. Figure 18 shows what would be expected of a Phugoid mode in that it has a low frequency and high time constant. Unlike many of the other modes of the system, the Phugoid mode stays nearly constant throughout the entire speed range. This result is expected. Unlike the Phugoid mode of conventional aircraft which depends on velocity, hypersonic aircrafts' Phugoid mode depends almost entirely on air density gradient through the atmosphere, though even those variations are minimal. Moreover, it is expected that the Phugoid mode has a frequency of about 0.04 and damping ratio near 0 [34]. This is clearly the case with the GHV's Phugoid mode.

This is promising from a linear controls standpoint because there is less need for more complex stabilization to compensate for changing dynamics. This also indicates, however, that the Phugoid mode of this system is more resilient than other modes to changes in the system. It will be challenging to generate a controller which significantly affects this aspect of the system. The final closed loop system is likely to have residual effect from this stationary mode.

Similar to the Lateral-Directional modes, the classification of the Longitudinal modes is distinct above Mach 6, but is less distinct below that speed. Below Mach 6, the eigenvectors change drastically, once again indicating the higher degree of nonlinearity at these speeds. As with the Lateral-Directional case, these variations may be attributed to the large variation in aerodynamic coefficients below Mach 6.

It should be noted that the altitude mode of the system remains near zero for the entire range of speeds. This indicates that this mode is very slow and approximately neutrally stable. This is expected as the altitude mode is inversely proportional to the lift-to-drag ratio. As a result, the mode tends to be small [34].

Another aspect of the altitude mode is its erratic eigenvector magnitude. This behavior is due to the fact that the eigenvalue itself is very small. If we consider the eigenvalue equation, $A\boldsymbol{X} = \lambda\boldsymbol{X}$ where $\boldsymbol{X}$ is an eigenvector of λ and λ is the eigenvalue of A, it can be seen that, if λ is small, any small change in A will result in large changes in $\boldsymbol{X}$.

Cross Axis Effects

In order to separate the system into its Lateral-Directional and Longitudinal components, the cross axis effects of the linear systems are lost. While these effects are not explicitly modeled in the system using cross axis effect coefficients, there is a residual impact due to the nonlinear model implementation. For instance, altitude is considered a Longitudinal parameter. However, as altitude changes, the density of air changes. Air density affects both Longitudinal and Lateral-Directional quantities and therefore has the potential for cross coupling the dynamics. For this reason, a brief examination of the magnitude of the cross coupling effects is performed.

To perform the analysis, The A matrix of the dynamic system was separated into three components—the Lateral-Directional A matrix, the Longitudinal A matrix, and the cross axis A matrix, which is what remains when the other two matrices are removed. The Frobenius Norm is then calculated for each matrix, the equation for which is given in (4-12). The resulting Mach varying norm is given in Figure 20.

$$\left||A|\right|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^2}, \qquad A \in \mathbb{R}^{mxn} \tag{4-12}$$

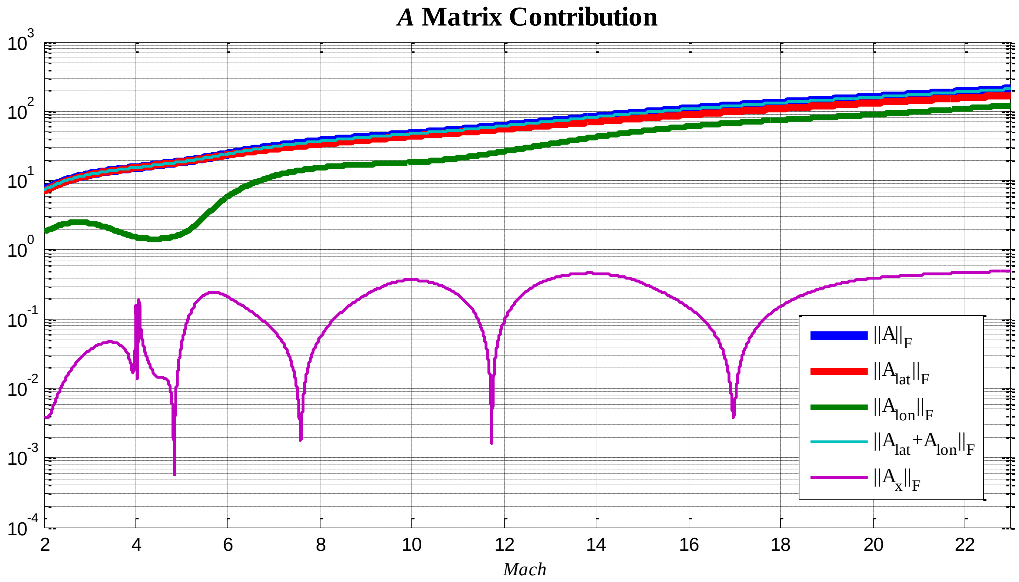


Figure 20: Component matrix contributions

From the above figure, it is evident that the cross axis contributions to the system are minimal. This implies that the approximations made to separate the Lateral-Directional and Longitudinal components are reasonable.

Chapter 5 H_∞ Controller Design

Environmental disturbances, such as wind, vary greatly throughout the entire flight envelope of the GHV. These dynamics are un-modeled within the system, however. Despite this, a system which is to be implemented beyond theory must take these disturbances into account as they have significant impact on the overall performance of the aircraft and controller [37]. Compounding this problem is the GHV's inherent nonlinear behavior. Take, for instance, angle of attack and side slip angle. Both of these quantities are influenced by external aircraft disturbances. These values also drive the aerodynamic coefficients of the system. As a result, external disturbances will cause nonlinear changes to the system.

Logically, the best way to account for these disturbances is to minimize the impact external disturbances have on the system—essentially, decouple external disturbances from the dynamic model. This can be done by including the disturbances into the system model and then designing a controller such that the disturbances are accounted for directly by the controller. In reality, however, this is impractical. Disturbance dynamics are challenging to represent, have limited regions of validity, and can significantly complicate the model [37].

The second alternative to decouple disturbance dynamics is to design a robust controller. The benefit of such an architecture is that it does not require extensive knowledge of the disturbances to be counteracted. Instead, only a limited understanding of their frequency domain spectral content is needed.

To achieve this decoupling in the GHV, an H_∞ control architecture was chosen. The H_∞ controller is a Multi-Input Multi-Output (MIMO), robust controller. The goal of its architecture is to minimize the maximum impact disturbances have on a set of defined outputs. Compared to another robust control architecture based on similar mathematics, the H_2 controller, the H_∞ is more robust. The H_2 controller minimizes the average impact disturbances have on defined outputs. This implies that some disturbances will have above average impact on the outputs. As a result, the H_2 controller is less robust than the H_∞ [38]. The consequence of this added robustness is that performance is degraded. This translates to a less maneuverable, potentially slower responding aircraft.

H_∞ control has been successfully implemented to control the Longitudinal dynamics of hypersonic aircraft in the past. Gao used an H_∞ style observer to produce full state feedback from the system given limited sensor input. From there, a fault tolerant controller was designed to expressly handle system failures [16]. Huang took a mixed H_2/H_∞ approach. This type of controller gives robustness benefits of the H_∞ controller, but only on specific inputs. An H_2 handles more performance based criteria of the controller. The overall combination forms a controller that is less robust overall, but has improved performance characteristics [17], [39]. Gregory a used pure H_∞ controller with the addition of μ-synthesis techniques to directly account for modeling uncertainties [18]. Cifdaloz applies a similar H_∞ approach to that which is presented in this work. It will serve as a reasonable comparison for the final controlled system in Chapter 7 as it shows a related hypersonic aircraft in similar flight conditions [19].

The drawback of all the controllers developed above is that they are only applicable to the Longitudinal dynamics of the system. As a result, the full effects of the hypersonic system cannot be studied. As was explored in Section 4.4, the Lateral-Directional effects are non-standard and by no means trivial. This work aims to resolve this issue.

In order to generate commands for the four independent system inputs (δ_a, δ_e, δ_r, and δ_{ER}), four quantities were chosen as feedback from the system into the controller. ϕ, θ, β, and M were selected as they form a set of preferred parameters for outer loop guidance algorithms [40].

5.1 H_∞ Control Methodology

The H_∞ is an optimal, robust control formulation which can be designed to control a system with both system modeling errors and noise from both external physical disturbances and sensor feedback noise. This work will focus on the disturbance and noise rejecting capability of the H_∞ controller.

First, consider the standard negative feedback loop shown in Figure 21. Included in this representation is a system model, $G(s)$, and a controller, $K(s)$. Additionally, it includes time varying physical disturbances to the system, $d(t)$, sensor noise in the feedback path, $n(t)$, and a commanded reference signal, $r(t)$. The sensible output of the system in the real world is denoted as $\mathscr{y}(t)$.

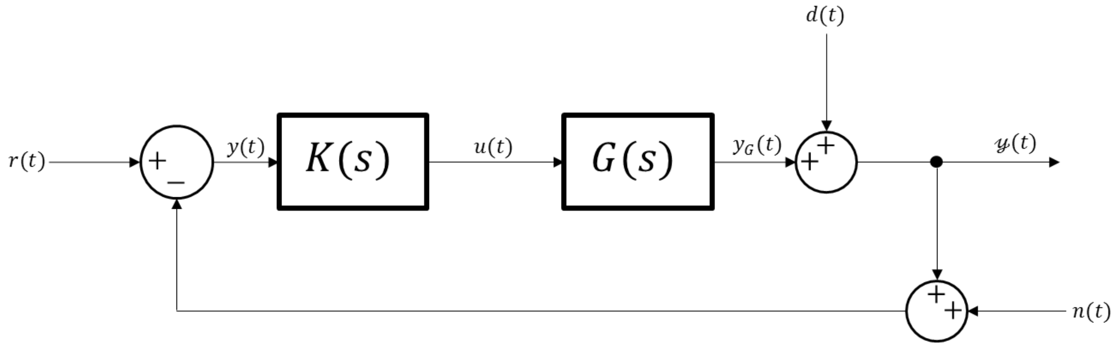


Figure 21: Standard model of a disturbed and noisy feedback system

Figure 21 may be rearranged to the form shown in Figure 22. The combination of the reference, disturbance, and noise have been joined together into a single signal, $w(t)$. For the purposes of this derivation, the signals entering the controller, $y(t)$, the signals being commanded from the controller to the system model, $u(t)$, and the output signals from the system model, $y_G(t)$, have been named. The output to the physical world is omitted from the block diagram for the remainder of this derivation, though it can be represented by $\mathcal{y}(t) = d(t) + y_G(t)$ and will be used in Section 7.3

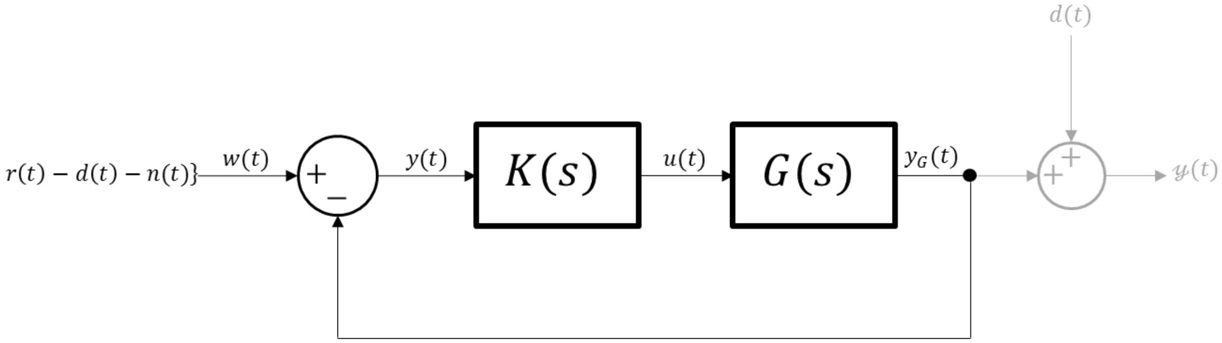


Figure 22: Rearranged disturbed system

Thus far, a standard system model has been manipulated without loss of generality as it relates to any form of control. At this point, the H_∞ structure to the system begins to take shape. In general, the H_∞ problem finds some controller $K(s)$ such that it stabilizes some plant (or system model), $P(s)$, while minimizing the maximum impact external inputs, $w(t)$, have on an arbitrary set of outputs, $z(t)$. This framework is shown in Figure 23. The outputs from the system $z(t)$ are constructed entirely by the designer. Moreover, there is no definitive way in which these outputs must be defined. The designer has free reign over their development. These outputs may even be defined prior to system linearization, enabling the designer to directly manipulate signals within the system [41].

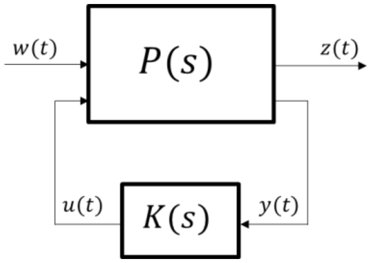


Figure 23: General H_∞ control framework

For this work, the Mixed Sensitivity approach was taken to define signals $z(t)$. This approach generates $z(t)$ such that it weights the Sensitivity and Complimentary Sensitivity (Co-Sensitivity) of the system as opposed to specific signals within the plant itself [42]. The reason behind weighting these specific parameters will be explored in more detail in Section 5.2. The benefit of this approach is that it uses a predefined linear model. With such a formulation, the same weighting matrices may be used for all linear models characterizing the GHV. This lends itself well to the gain-scheduling approach implemented in this work.

Figure 24 shows the augmented system used in the Mixed Sensitivity H_∞ structure. The block diagram has been manipulated from that shown in Figure 22 with the additions shown in red. It has been augmented to produce the new weighted system output $z(t)$. Note that this block diagram now takes the form shown in Figure 23.

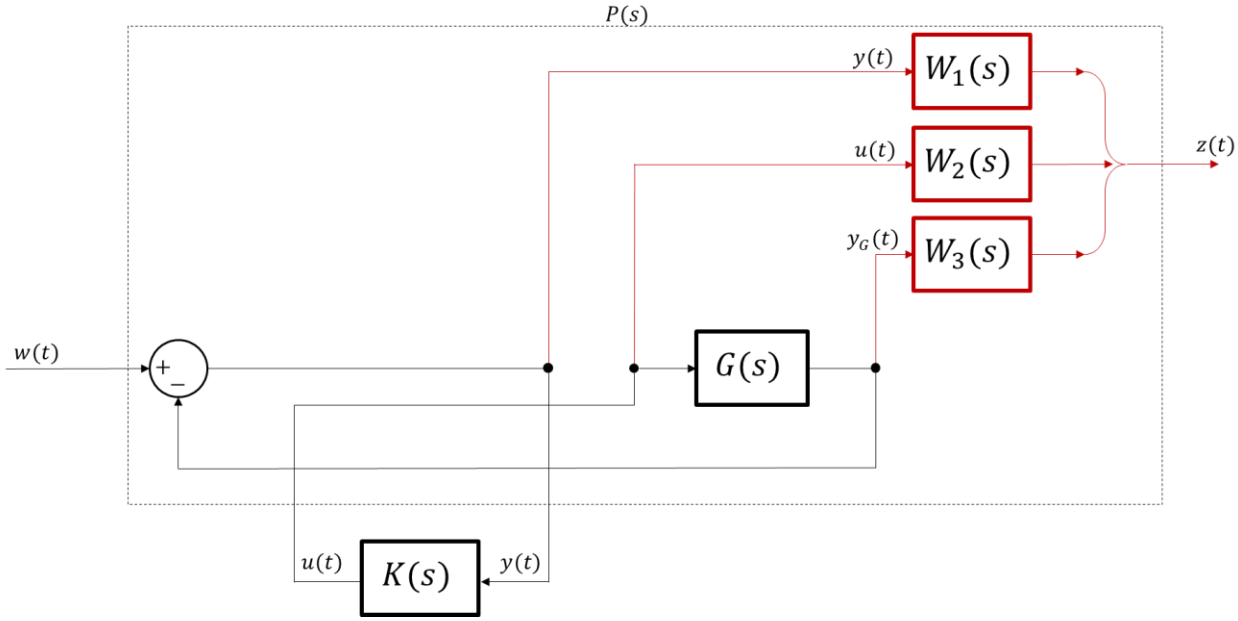


Figure 24: Augmented feedback system with weighting parameters

The new outputs are formed by applying weighting transfer functions which filter various signals to amplify or attenuate specific relevant frequencies of the weighted signals. $W_1(s)$ weights the inputs to the controller, $W_2(s)$ weights the controller command signals, and $W_3(s)$ weights the output of the system being controlled. The combination of the new output signals create the artificial construct $z(t)$ which is only used for the development of the controller and is not used in the implementation of the system in the real world. It should be noted that the three weights on the system are typically frequency dependent transfer functions. The selection of these weighting matrices is examined in more detail in Sections 5.2 and 5.3. For this derivation, it is assumed that W_1 and W_3 are transfer functions while W_2 is a constant.

Now the system $P(s)$ may be formed. As shown in Figure 24, there are two inputs (inputs from the controller, $u(t)$, and external commands, disturbances, and noise, $w(t)$) and two outputs (augmented states, $z(t)$, and outputs to the controller, $y(t)$). Note that the $z(t)$ comprises three signals, $[z_1(t), z_2(t), z_3(t)]^T$. The goal of forming $P(s)$ is to generate a relationship between the two inputs and the two outputs.

To do this, each of the transfer functions $G(s)$, $W_1(s)$, $W_2(s)$, and $W_3(s)$ are converted into their state space representations, given in (5-1) through (5-4).

$$G(s) = \begin{cases} \dot{x}_G(t) = A_G x_G(t) + B_G u(t) \\ y_G(t) = C_G x_G(t) + D_G u(t) \end{cases} \tag{5-1}$$

$$W_1(s) = \begin{cases} \dot{x}_{W_1}(t) = A_{W_1} x_{W_1}(t) + B_{W_1} y(t) \\ z_1(t) = C_{W_1} x_{W_1}(t) + D_{W_1} y(t) \end{cases} \tag{5-2}$$

$$W_2(s) = \{ z_2(t) = D_{W_2} u(t) \tag{5-3}$$

$$W_3(s) = \begin{cases} \dot{x}_{W_3}(t) = A_{W_3} x_{W_3}(t) + B_{W_3} y_G(t) \\ z_3(t) = C_{W_3} x_{W_3}(t) + D_{W_3} y_G(t) \end{cases} \tag{5-4}$$

Define the state vector of $P(t)$ as a combination of the states in Equations (5-1) through (5-4). This gives $x(t) = \left[x_G(t), x_{W_1}(t), x_{W_3}(t)\right]^T$. Note that there is no state added from W_2 as this is a scalar gain and the input is mapped directly to the output.

The final relationship needed to fully define the system $P(s)$ in state space format is to define $y(t) = w(t) - y_G(t)$. By substituting this definition into Equation (5-2), Equations (5-1) through (5-4) can be easily formatted to form $P(s)$ with the states as defined above. This is shown in Equation (5-5). For notational convenience, we write Equation (5-6) by defining the quantities as shown in red in Equation (5-5).

$$P(s) = \begin{cases} \underbrace{\begin{bmatrix} \dot{x}_G(t) \\ \dot{x}_{W_1}(t) \\ \dot{x}_{W_3}(t) \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} A_G & 0 & 0 \\ -B_{W_1}C_G & A_{W_1} & 0 \\ B_{W_3}C_G & 0 & A_{W_3} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_G(t) \\ x_{W_1}(t) \\ x_{W_3}(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \\ B_{W_1} \\ 0 \end{bmatrix}}_{B_1} w(t) + \underbrace{\begin{bmatrix} B_G \\ -B_{W_1}D_G \\ B_{W_3}D_G \end{bmatrix}}_{B_2} u(t) \\ \underbrace{\begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{bmatrix}}_{z(t)} = \underbrace{\begin{bmatrix} -D_{W_1}C_G & C_{W_1} & 0 \\ 0 & 0 & 0 \\ D_{W_3}C_G & 0 & C_{W_3} \end{bmatrix}}_{C_1} \underbrace{\begin{bmatrix} x_G(t) \\ x_{W_1}(t) \\ x_{W_3}(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} D_{W_1} \\ 0 \\ 0 \end{bmatrix}}_{D_{11}} w(t) + \underbrace{\begin{bmatrix} -D_{W_1}D_G \\ D_{W_2} \\ D_{W_3}D_G \end{bmatrix}}_{D_{12}} u(t) \\ y(t) = \underbrace{\begin{bmatrix} -C_G & 0 & 0 \end{bmatrix}}_{D_{21}} \underbrace{\begin{bmatrix} x_G(t) \\ x_{W_1}(t) \\ x_{W_3}(t) \end{bmatrix}}_{x(t)} + \underbrace{[I]}_{D_{21}} w(t) + \underbrace{[-D_G]}_{D_{22}} u(t) \end{cases} \tag{5-5}$$

$$P(s) = \begin{cases} \dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \\ z(t) = C_1x(t) + D_{11}w(t) + D_{12}u(t) \\ y(t) = C_2x(t) + D_{21}w(t) + D_{22}u(t) \end{cases} \tag{5-6}$$

Next, convert the state space representation of P to its transfer function representation using the standard relationship $\frac{Output}{Input} = C(sI - A)^{-1}B + D \equiv H(s)$, the result of which can be seen in Equation (5-7). This is written in more compact notation in Equation (5-8).

$$\begin{aligned} \frac{Z(s)}{W(s)} &= C_1(sI - A)^{-1}B_1 + D_{11} \equiv P_{11}(s) \\ \frac{Z(s)}{U(s)} &= C_1(sI - A)^{-1}B_2 + D_{12} \equiv P_{12}(s) \\ \frac{Y(s)}{W(s)} &= C_2(sI - A)^{-1}B_1 + D_{21} \equiv P_{21}(s) \\ \frac{Y(s)}{U(s)} &= C_2(sI - A)^{-1}B_2 + D_{22} \equiv P_{22}(s) \end{aligned} \tag{5-7}$$

$$\begin{bmatrix} Z(s) \\ Y(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} W(s) \\ U(s) \end{bmatrix} \tag{5-8}$$

Through these transformations, the block diagram shown in Figure 24 becomes Figure 25, the two block representation of the augmented system.

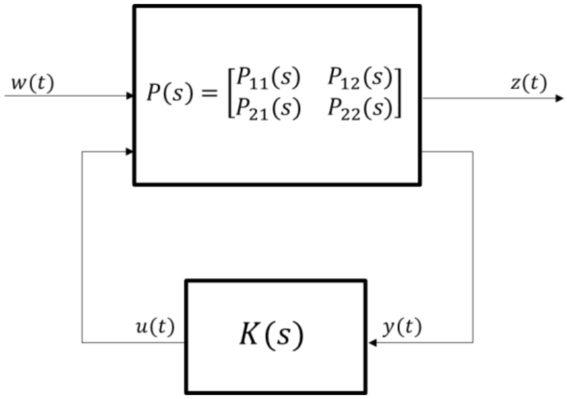

$w(t)$
$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix}$
$z(t)$
$u(t)$
$K(s)$
$y(t)$

Figure 25: Two block representation of the augmented system

Next the controller $K(s)$ must be accounted for in the representation, the goal of which is to have this be a function of only $w(t)$ with output $z(t)$. From Equation (5-8) and Figure 25, we get Equations (5-9) - (5-11).

$$Z(s) = P_{11}(s)W(s) + P_{12}(s)U(s) \tag{5-9}$$

$$Y(s) = P_{21}(s)W(s) + P_{22}(s)U(s) \tag{5-10}$$

$$U(s) = K(s)Y(s) \tag{5-11}$$

By rearranging variables, the transfer function $T_{ZW}(s)$ is built (shown in Equations (5-12)-(5-15)). Note that the frequency variable, s, has been omitted for clarity.

$$Y = P_{21}W + P_{22}KY \Rightarrow Y = (I - P_{22}K)^{-1}P_{21}W \tag{5-12}$$

$$U = K(I - P_{22}K)^{-1}P_{21}W \tag{5-13}$$

$$Z = P_{11}W + P_{12}K(I - P_{22}K)^{-1}P_{21}W \tag{5-14}$$

$$ZW^{-1} = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \equiv T_{ZW} \tag{5-15}$$

The multivariable transfer function, $T_{ZW}(s)$, is the transfer function of $W(s)$ to $Z(s)$ for the closed loop system shown in Figure 25.

One final piece of theory remains to define the H_∞ control problem—that of defining the infinity norm of a system. Consider a generic LTI system, H. The infinity norm of H, denoted $\|H\|_\infty$, is defined as the supremum, or the least upper bound, of the largest singular values of its transfer function for all values along the imaginary axis, $j\omega$. This is written in compact notation in Equation (5-16) [41]. What this means practically is that $\|H\|_\infty$ is a measure of the maximum effect any given input to H has on its output for all input to output combinations over all frequencies.

$$H(s) = \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \Rightarrow H(s) = D + C(sI - A)^{-1}B$$

$$\|H\|_\infty = \sup_{\omega \in \mathbb{R}} \sigma_{\max}[H(j\omega)] \tag{5-16}$$

Finally, the H_∞ control problem can be stated: design a controller $K(s)$ which stabilizes the closed loop system and minimizes the maximum effect the combination of command signals, external disturbances, and feedback noise ($w(t)$) has on a set of defined outputs, $z(t)$. In other words, design a stabilizing controller, $K(s)$, such that $\|T_{ZW}\|_\infty$ is minimized.

Numerically, this is a very challenging problem. In practice, this value is not minimized. Instead, it is customary to modify the problem statement to a sub-optimal result, namely, find $K(s)$ such that $\|T_{ZW}\|_\infty < \gamma$ where $\gamma > 0$. Numerically, this problem is far more conducive to solution by computers [43].

The procedure for numerically calculating $K(s)$ consists of the following: first choose a large value for γ. Next, calculate $K(s)$ for that given value. If a controller is found, choose a new, smaller γ and repeat by calculating a new $K(s)$. This process is repeated until no smaller γ can be found which produces a controller. The details for this procedure can be found in Reference [43].

5.2 Mixed Sensitivity Weighting Matrix Selection Theory

(Note that the derivation presented in this section has been adapted for the H_∞ Mixed Sensitivity problem from a more general discussion of Sensitivity and Co-Sensitivity in Chapter 9.2 of Reference [38])

To design an H_∞ controller using the mixed sensitivity framework, The designer need only select appropriate weighting matrices for $W_1(s)$, $W_2(s)$, and $W_3(s)$ (shown in Figure 24). By doing so, a set of outputs from the system are artificially constructed, denoted as $z(t)$. As the ultimate goal of the H_∞ synthesis process is to minimize the impact $w(t)$ has on $z(t)$, it is worthwhile to consider the makeup of $w(t)$. First, recast each of the signals shown in Figure 21 by taking the Laplace Transform of each (Equation (5-17)). Also, define the error in the system as the difference between the real world output, $\mathscr{y}(t)$ and the reference signal, $r(t)$. This is shown in Equation (5-18). Note that $r(t)$ is not shown in Figure 21 through Figure 23. This is because all the signals shown in these figures are corrupted with noise and disturbance.

$$\begin{gathered}\mathcal{L}\{r(t)\} = R(s), \qquad \mathcal{L}\{d(t)\} = D(s), \qquad \mathcal{L}\{n(t)\} = N(s) \\ \mathcal{L}\{y(t)\} = Y(s), \qquad \mathcal{L}\{y_G(t)\} = Y_G(s), \qquad \mathcal{L}\{\mathscr{y}(t)\} = \mathcal{Y}(s) \\ \mathcal{L}\{w(t)\} = W(s)\end{gathered} \tag{5-17}$$

$$e(t) = \mathscr{y}(t) - r(t) \quad \Rightarrow \quad E(s) = \mathcal{Y}(s) - R(s) \tag{5-18}$$

Next, define the transfer function from the Laplace transform of the combined system inputs, $W(s)$, to the controller input, $Y(s)$, (Equation (5-19)). Similarly define the transfer function from the combined system inputs to the idealized plant model output, $Y_G(s)$, (Equation (5-20)). The two resulting transfer functions, $S(s)$ and $T(s)$, are called the Sensitivity and Complementary Sensitivity (Co-Sensitivity) respectively. Note from Figure 24, $W_1(s)$ weights the Sensitivity function, while $W_3(s)$ weights the Co-Sensitivity of the closed loop system.

$$W(s)^{-1}Y(s) = [I + K(s)G(s)]^{-1} \equiv S(s) \tag{5-19}$$

$$W(s)^{-1}Y_G(s) = K(s)G(s)[I + K(s)G(s)]^{-1} \equiv T(s) \tag{5-20}$$

The expressions for $Y(s)$ and $Y_G(s)$ may be rewritten as in Equation (5-21) and (5-22).

$$Y(s) = W(s)S(s) \tag{5-21}$$

$$Y_G(s) = W(s)T(s) \tag{5-22}$$

Recall that $w(t) = r(t) - d(t) - n(t)$, the Laplace transform of which is $W(s) = R(s) - D(s) - N(s)$. Substitute this expression into Equations (5-22) to get Equations (5-23).

$$Y_G(s) = R(s)T(s) - D(s)T(s) - N(s)T(s) \tag{5-23}$$

From Figure 21, $\mathcal{y}(t) = d(t) + y_G(t)$. Taking the Laplace transform of this, we get Equation (5-24). Substitute Equation (5-23) into (5-24) and group like terms to get Equation (5-25).

$$\mathcal{Y}(s) = Y_G(s) + D(s) \tag{5-24}$$

$$\mathcal{Y}(s) = [R(s) - N(s)]T(s) + D(s)[I - T(s)] \tag{5-25}$$

Next, substitute Equation (5-25) into the Laplace domain representation of system error, Equation (5-18). Rearrange terms to get Equation (5-26).

$$E(s) = N(s)T(s) + D(s)[I - T(s)] - R(s)[I - T(s)] \tag{5-26}$$

Consider the sum of the Sensitivity and the Co-Sensitivity functions. Equation (5-27) shows that they sum to identity.

$$\begin{aligned} S(s)+T(s) &= [I+K(s)G(s)]^{-1}+K(s)G(s)[I+K(s)G(s)]^{-1} \\ S(s)+T(s) &= [I+K(s)G(s)][I+K(s)G(s)]^{-1} \\ S(s)+T(s) &= I \end{aligned} \tag{5-27}$$

Equation (5-27) implies that $I+T(s)=S(s)$. Substituting this relationship into Equations (5-25) and (5-26) to get Equations (5-28) and (5-29) respectively.

$$\mathcal{Y}(s)=[R(s)-N(s)]T(s)+D(s)S(s) \tag{5-28}$$

$$E(s)= N(s)T(s)+[D(s)-R(s)]S(s) \tag{5-29}$$

As the goal of the controller is to have the output equal the reference input, the Sensitivity and Co-Sensitivity should be shaped to attenuate $D(s)$ and $N(s)$ while conducting $R(s)$ through to the system output. Additionally, an ideal controller will have zero error, meaning the Sensitivity and Co-Sensitivity attenuate all signals to the greatest extent possible. However, simultaneously achieving all of these characteristics cannot be achieved with constant magnitude Sensitivity and Co-Sensitivity over all frequencies. Instead, their Bode plots must be shaped.

Consider the desired characteristics of each of the signals in Equation (5-28) and (5-29). For many control problems, the reference signal is constant or very slowly changing, implying that they tend to have large magnitude at low frequencies and low magnitude at high frequencies. A typical Bode plot of $D(s)$ is shown in Figure 26a and one for $N(s)$ in Figure 26b.

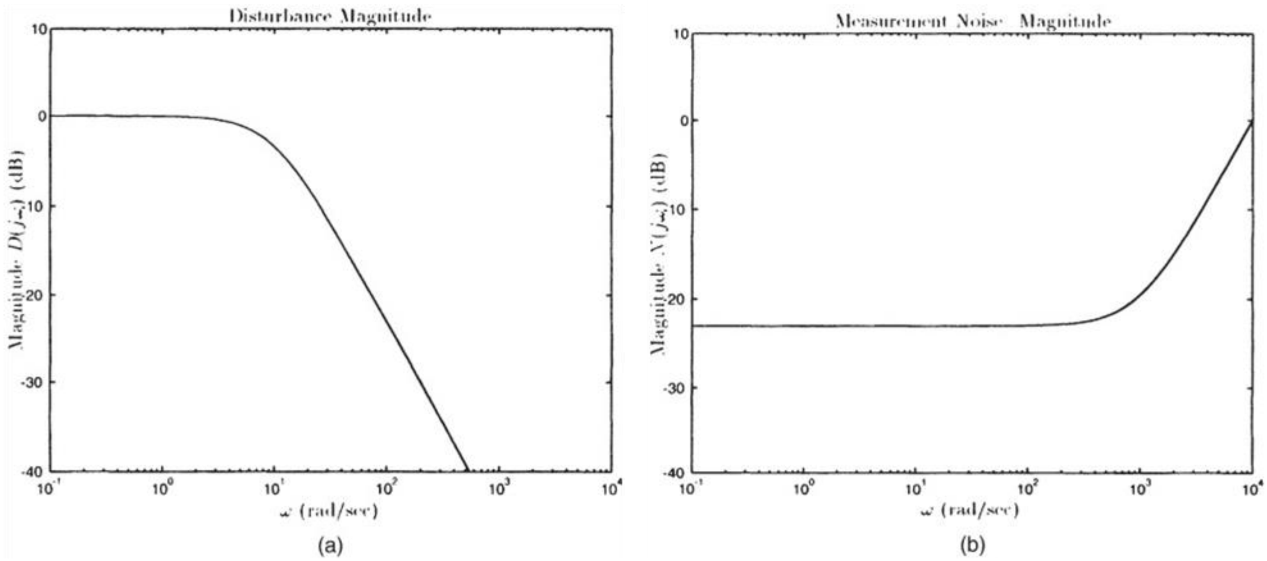


Figure 26: Typical Bode plots of disturbance magnitude (a) and measurement noise (b) [38]

Combining this knowledge with Equation (5-28), for optimal tracking performance, $T(s)$ should have low attenuation at low frequencies to preserve $R(s)$ and high attenuation at high frequencies to degrade $N(s)$. This implies T(s) should take the form of a low pass filter. As a consequence of $S(s) + T(s) = I$, $S(s)$ should take the form of a high pass filter. This is beneficial as, in order to remove the low frequency disturbance from the tracking signal, $S(s)$ should have large attenuation at low frequencies, where $D(s)$ is large.

These properties of $T(s)$ and $S(s)$ are reinforced by the tracking error equation. In order to have low error in the system, all signals in Equation (5-29) should be attenuated. As $S(s)$ filters out low frequency signals (which includes both the disturbance and reference signal) and $T(s)$ filters out the high frequencies (noise) , $E(s)$ should be attenuated over all relevant frequencies.

This lends guidance to how the weighting functions W_1 and W_3 should be shaped. As a consequence of Equation (5-21), W_1 operates directly on $S(s)$. This implies W_1 should have high penalties on low frequency signals, and lower penalties on high frequency signals, forcing $S(s)$ to take the form of a high pass filter. Similarly, because of Equation (5-22), W_3 penalizes

$T(s)$. This means $W_3(s)$ should have a high penalty on high frequencies and low penalty on low frequencies, making $T(s)$ a low pass filter. By shaping the weighting functions, and thus the controller in conjunction with the system in this manner, a system which has good noise and disturbance rejection and good tracking performance can be achieved.

The process of selecting W_2 is more straightforward. It penalizes the relative use of each control actuator. Typically, this is chosen as a diagonal matrix with scalar values on its diagonals. If, however, it is found that the controller drives the actuators at undesired frequencies, a more careful examination of W_2 can be made.

The reader should be cautioned that this is not the only way in which the three Mixed Sensitivity weighting matrices may be defined. The method presented above provides good tracking and disturbance/noise rejection for a broad range of systems. However, if a designer has a known signal which must be attenuated, a more restrictive set of weighting matrices may be designed. An example of such a design may be found in Reference [44].

5.3 Weighing Matrix Selection for the GHV

Designing an H_∞ controller for a system using the Mixed Sensitivity framework is a study in finding the best weighting functions which result in the desired closed loop performance of the system. For many designs, the designer assumes a given structure to the weighting matrices (such as W_1 as a first order low pass filter or W_3 as a high pass filter). The specific values defining the structure are then iterated upon until the desired performance criteria are met [45]. Such design practices are demonstrated in Reference [46]–[49].

Reference [45] presents a method by which the specific shape of each weighting matrix may be selected. However, this method requires that the H_∞ problem may be solved for the resulting weighting matrixes. This was not the case with the GHV.

Owing to the complex nature of the hypersonic dynamic system, the H_∞ synthesis algorithm is highly sensitive to the three sets of frequency weighting functions. Many candidate matrices do not allow the synthesis algorithm to converge. Even if a solution is produced, there is no guarantee that the solution will be stable for any significant input when the closed loop system is formed. To get around this, a Monte Carlo search was conducted to find a reasonable set of weighting parameters which produced the desirable performance. Figure 27 diagrams the procedure taken to determine the controller.

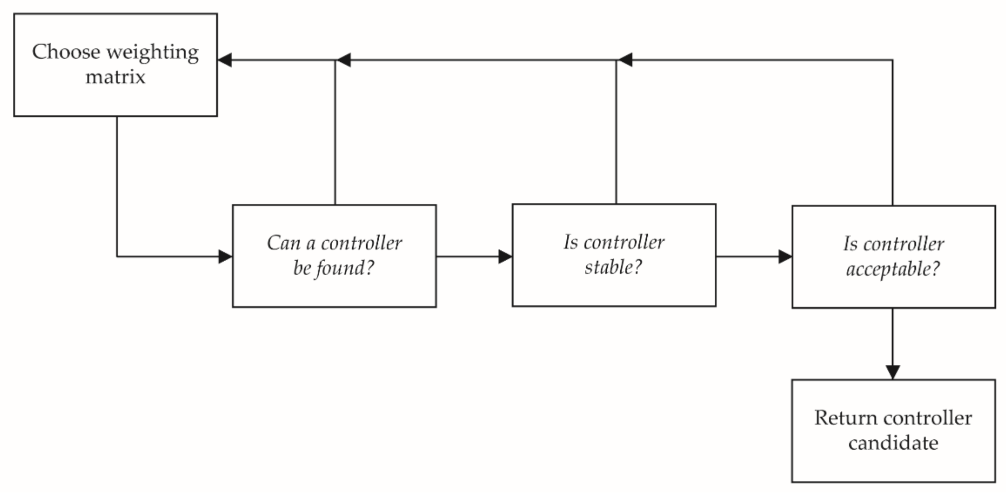

Choose weighting matrix
Can a controller be found?
Is controller stable?
Is controller acceptable?
Return controller candidate

Figure 27: Weighting matrix selection

As the system consisted of four inputs and four outputs, there had to be four frequency dependent entries in each one of the three weighting functions. For simplicity, the weighting matrices associated with the control inputs (W_2) are fixed as constant over all frequencies, resulting in a static gain. Similarly, all entries of W_1 took the form of a low pass filter, while all entries of W_3 were shaped as high pass filters. This reduced the number of random parameters to be selected to seven—pole locations for W_1 and W_3, zero locations for W_1 and W_3, and static gains for W_1, W_2 and W_3. A candidate weighting matrix was composed of four sets of these randomly selected values, one each for every input/output of the system.

To reduce the scope of the possibilities, a search region was defined. Figure 28 shows the selected regions for each weighting matrix and a set of potential candidate weighting matrix entries. All possible weighting functions explored by the Monte Carlo search fall within the shaded regions. The cutoff frequencies shown were chosen such that the system would attenuate noise and disturbances seen by physical systems in the real world (such as those shown in Figure 26).

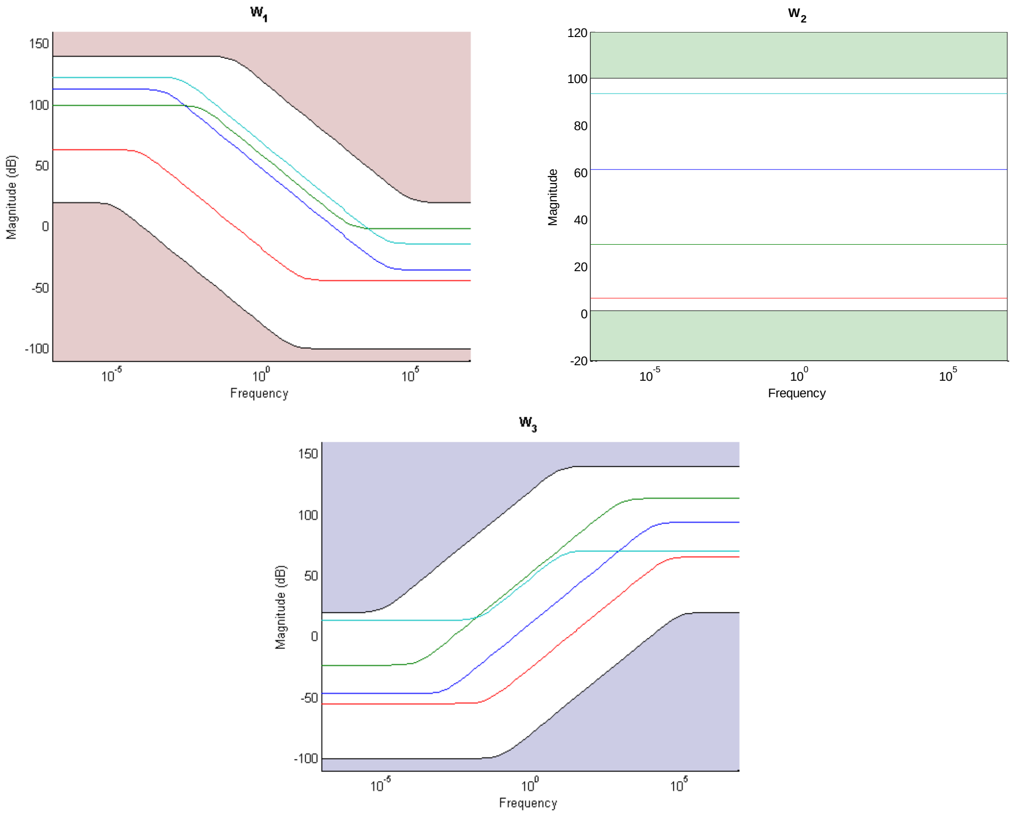


Figure 28: Candidate weighting matrices

With the Monte Carlo search space defined, the H_∞ synthesis algorithm could be run. Only one of the 4,485 linear models of the system could be used to generate the H_∞. In order to have the broadest applicability to all the linear models beyond the one chosen for this exploration, a linear model known to be challenging to control was selected. Each candidate matrix was tested following the procedure given in Figure 27.

First a candidate set of weighting matrices was generated. The H_∞ synthesis algorithm within MATLAB® was executed. If the controller could be found, it was tested by checking the closed loop performance in the presence of a step input. If the step input did not destabilize the controller, its performance was analyzed. If it did not stabilize the system, it was rejected and a new set of weighting functions was generated.

To analyze the successfully stabilizing controller, a cost function, Equation (5-30), was synthesized. The cost function looks at the time history of each output from the system when a step input is injected to each of the controller's inputs separately. Ideally, a controller would completely decouple the dynamics of the system such that a command sent through the controller would only affect that same output in the system. This is not practical in reality, however. As a result, there will always be some amount of off-axis effects. The cost equation takes this into account by comparing the average of the time history for each signal ($\bar{y}_{ij}$) against the ideal, $\bar{y}_{ij}^*$. Note that $\bar{y}_{ij}^* = 1$ for all $i = j$ and $\bar{y}_{ij}^* = 0$ elsewhere. The absolute value of each one of these quantities is weighted by a scalar cost, $c_{y_{ij}}$, and summed.

The time needed to stabilize the system is also weighted by c_t. It weights the magnitude of the difference between the time needed to stabilize the system, $\max(t)$, and the ideal time it should take to stabilize the system, t^*.

$$\mathcal{C} = \left[\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\bar{y}_{ij} - \bar{y}_{ij}^{*}\right|c_{y_{ij}}\right] + |max(t) - t^{*}|c_t \tag{5-30}$$

The cost function parameters were manipulated until the resulting relative magnitude of the costs for each candidate controller matched the objective design parameters.

While it may have been possible to attempt to minimize the cost equation using an algorithm capable of handling discontinuous search spaces, that approach was deemed too onerous. Instead, a threshold was set for an acceptable cost. The Monte Carlo search was terminated after a set number of controllers were found with acceptable costs.

The candidate controllers were then tested on the full range of linear system models. The performance of the controllers capable of stabilizing all the system models were then objectively assessed and the final controller was chosen. As with any control system design, the final closed loop system must be assessed by a human. In this case, the final controller is chosen such that it was the candidate controller with the least overshoot, fastest rise time, least amount of oscillation, and smallest overall stead state error. The final weighting functions used to synthesize the controller can be seen in Figure 29.

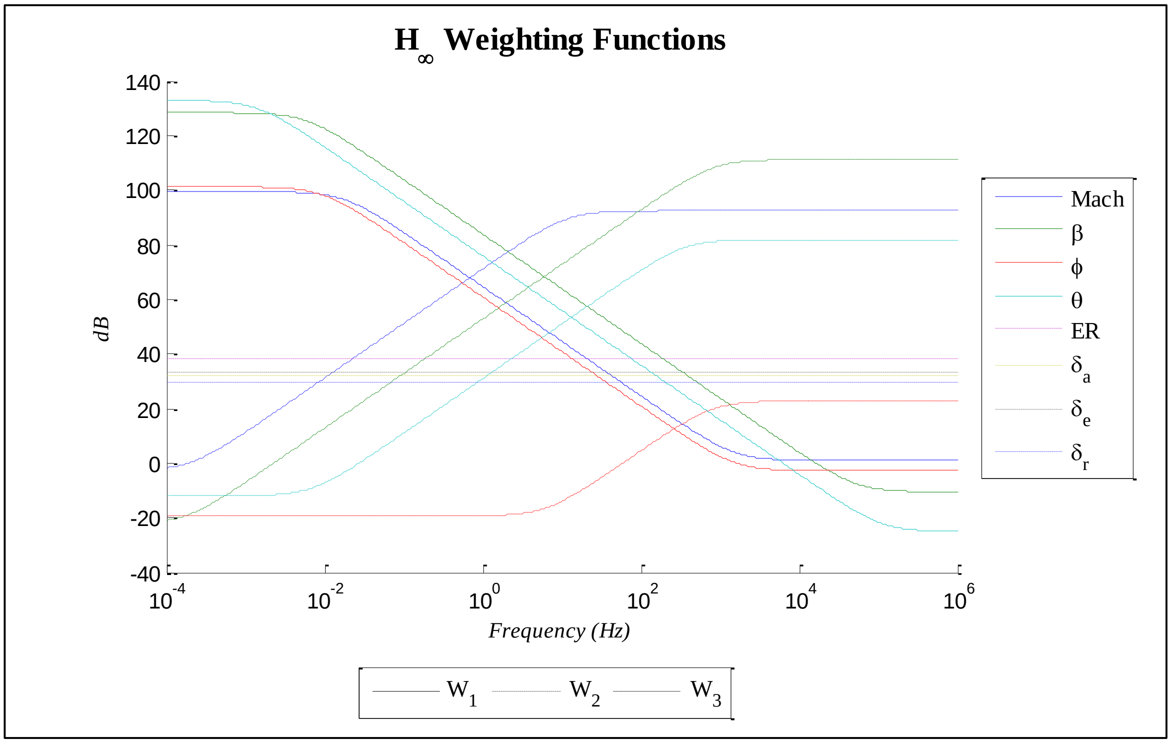


Figure 29: Weighting functions used to develop the H_∞ controller

Chapter 6 Controller Implementation

It is common in aircraft control problems for the designer to opt for a two loop solution to control an autonomous system [50]. Typically, the inner loop controller will handle aircraft stability and robustness compensation, while the outer loop controller adds guidance logic, such as in Reference [51], [52]. The outer loop controller may be as simple or complex as the designer wishes.

For the outer loop controllers to work effectively, the inner loop controller must stabilize the aircraft throughout the entire flight envelope in which the outer loop controller commands. Due to the nonlinearity of the GHV (see Section 4.4) one single inner loop controller will not suffice to control the system over the entire envelope. It is common practice to alleviate this problem by designing multiple inner loop controllers, each designed for a separate LTI model of the system, then link these controllers together using some means of interpolation. This approach is called gain-scheduling [53].

For this work, one gain-scheduled, MIMO, inner loop controller and three outer loop controllers were used. Two of these outer loop controllers are Single-Input Single-Output (SISO), while the third is MIMO. The inner loop controller serves to stabilize the system while the outer loop controllers provide commands to the inner loop control to effect speed and trajectory.

6.1 H_∞ Controller Gain-Scheduling

Gain-Scheduling Architecture

Once the controllers have been designed, the next task is to implement that controller into the system. Note that because the controller was designed using only a LTI model of the aircraft, that controller is only valid while aircraft perturbations are in the proximity of the trim points. In the case of the GHV, these regions are very small, particularly in the lower Mach ranges. This implies many different controllers must be designed, each using a different linear model. To cover a broad range of Mach numbers, the aircraft must be trimmed at several points and a series of controllers must be designed to make the transition smooth. This is the standard gain-scheduling problem. This method transitions between controllers at different trim points based on a gain-scheduling parameter. The simplest application of this method is to schedule based on only one parameter. Equation (6-1) shows such an implementation. Here, $\mathcal{K}_0$ and $\mathcal{K}_1$ are distinct controllers and $f(\sigma)$ is a function on the range $\sigma_0 < \sigma < \sigma_1$. K is the gain-scheduled controller and σ is the gain-scheduling parameter [54].

$$\begin{gathered} K = \mathcal{K}_0 f(\sigma) + \mathcal{K}_1\big(1 - f(\sigma)\big) \\ 0 \leq f(\sigma) \leq 1 \quad \forall \quad \sigma_0 < \sigma < \sigma_1 \end{gathered} \tag{6-1}$$

Note that $\sigma \rightarrow \sigma_0 \Rightarrow K \rightarrow \mathcal{K}_0$. Similarly, $\sigma \rightarrow \sigma_1 \Rightarrow K \rightarrow \mathcal{K}_1$. While the controllers $\mathcal{K}_0$ and $\mathcal{K}_1$ are capable of controlling the aircraft in trim points 0 and 1 respectively, there is no guarantee that gain K can control the aircraft at either point. As σ changes, the intermediate controllers may not produce a controller capable of stabilizing the system. Proper selection of the function, f, and sufficiently closely selected $\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \ldots$ along the scheduling parameter σ will reduce the risk of instability [54].

For application to the GHV, a cubic spline function was chosen as the scheduling function, f. A cubic spline, in this paper referred to simply as a spline, is a piecewise function where each piece is a third order polynomial. For a two dimensional case, let $g(x)$ be a generic function defined over the range $[a, b]$. The cubic spline, $S(x)$ is defined such that:

1. $S(x)$ interpolates $g(x)$ at points $x_0, x_1, x_2, \ldots, x_n$

2. $S(x)$ is continuous on $[a, b]$
3. The first derivative of $S(x)$, $\dot{S}(x)$, is continuous on $[a, b]$
4. The second derivative of $S(x)$, $\ddot{S}(x)$, is continuous on $[a, b]$

All cubic splines in this work have Not-A-Knot end conditions, implying that the third derivative of $S(x)$ is continuous at points $x = x_1$ and $x = x_{n-1}$. For such end conditions, the maximum error of the interpolation is shown in Equation (6-2) [55]–[57].

$$\max_{x \in [x_{i-1}, x_i]} |g(x) - S(x)| \leq \frac{\Delta_\mathrm{x}^4}{4!} \max_{x \in [a,b]} \left| g^{(4)}(x) \right| \qquad (6\text{-}2)$$
$$\Delta_\mathrm{x} = \max_i |x_i - x_{i-1}|$$

Another benefit of using spline interpolation is that it is easily extendable to allow gain-scheduling over multiple scheduling parameters [58]. The GHV must be scheduled along three different scheduling parameters: Mach, weight, and altitude.

Gain-Scheduling Implementation

In the case of the H_∞ controller, each individual $\mathcal{K}_i$ is its own state space model.

$$\mathcal{K}_i = \begin{cases} \dot{x} = \mathcal{A}_i x + \mathcal{B}_i u \\ y = \mathcal{C}_i x + \mathcal{D}_i u \end{cases} = \begin{cases} \dot{x} = \mathcal{A}_i x + \mathcal{B}_i u \\ y = \mathcal{C}_i x \end{cases} \qquad (6\text{-}3)$$

Note that in the case of the H_∞ controller for the GHV, $\mathcal{D}_i = 0$.

The GHV has three primary states that, when changing, cause the system to change nonlinearly—namely speed (Mach number), altitude, and aircraft weight. This means that controllers for each different Mach number, altitude, and weight must be calculated. To facilitate gain-scheduling off of these three parameters, define a set of controllers:

$$\begin{gathered}\boldsymbol{\mathcal{K}} = \{\mathcal{K} | \mathcal{A} \in \boldsymbol{\mathcal{A}}, \mathcal{B} \in \boldsymbol{\mathcal{B}}, \mathcal{C} \in \boldsymbol{\mathcal{C}}\} \\ \boldsymbol{\mathcal{A}} \in \mathbb{R}^{17} \times \mathbb{R}^{17} \times \mathbb{R}^{l} \times \mathbb{R}^{n} \times \mathbb{R}^{m} \\ \boldsymbol{\mathcal{B}} \in \mathbb{R}^{17} \times \mathbb{R}^{4} \times \mathbb{R}^{l} \times \mathbb{R}^{n} \times \mathbb{R}^{m} \\ \boldsymbol{\mathcal{C}} \in \mathbb{R}^{4} \times \mathbb{R}^{17} \times \mathbb{R}^{l} \times \mathbb{R}^{n} \times \mathbb{R}^{m}\end{gathered} \tag{6-4}$$

with $l, m,$ and n being the discrete number of Mach, altitude, and weights under consideration in the flight envelope and $\mathcal{K}$ defined in Equation (6-5).

The individual elements of $\boldsymbol{\mathcal{K}}$ are called the nodes of $\boldsymbol{\mathcal{K}}$, namely $\mathcal{K}[i, j, k]$:

$$\begin{gathered}\mathcal{K}[i, j, k] = \begin{cases} \dot{x} = \mathcal{A}[i, j, k]x + \mathcal{B}[i, j, k]u \\ \qquad y = \mathcal{C}[i, j, k]x \end{cases} \\ \mathcal{A}[i, j, k] \in \mathbb{R}^{17} \times \mathbb{R}^{17} \\ \mathcal{B}[i, j, k] \in \mathbb{R}^{17} \times \mathbb{R}^{4} \\ \mathcal{C}[i, j, k] \in \mathbb{R}^{4} \times \mathbb{R}^{17}\end{gathered} \tag{6-5}$$

where $i, j,$ and k are the index values of Mach, altitude, and weight, respectively. The controller, K, can then be found by interpolating between the nodes of $\boldsymbol{\mathcal{K}}$ using the multidimensional spline interpolation gain-scheduling. This is done element by element where each entry of the matrix $\mathcal{A}, \mathcal{B},$ and $\mathcal{C}$ have a separate interpolation function, resulting in 425 separate gain-scheduling spline functions. The Simulink® implementation of the $\mathcal{A}[i, j, k]$ is shown in Figure 30.

Note that controllers between each element of $\mathcal{K}$ are not necessarily true H_∞ controllers for the system. However, as a consequence of Equation (6-2), the greater the number of controllers represented within $\mathcal{K}$ (i.e., the larger the values of $l, m,$ and n in $\boldsymbol{\mathcal{K}}$) the closer the values of the resulting controller are to a true H_∞ controller.

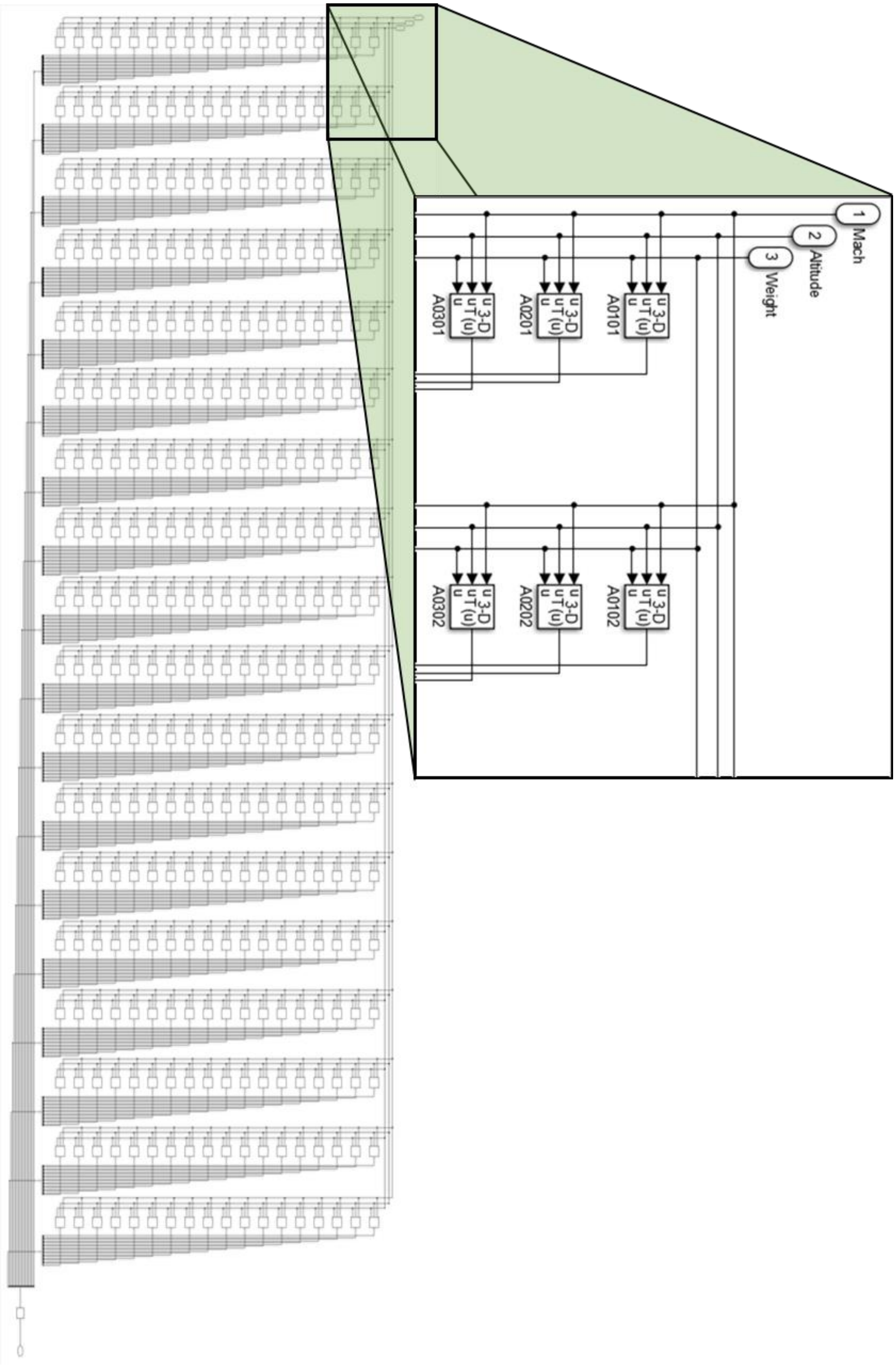


Figure 30: Interpolation table array for $\mathcal{A}[i, j, k]$

Control Region

While the GHV was linearized for speed ranging from Mach 2 to 23, the controller was only developed for a portion of that region. To demonstrate gain-scheduling's resilience to changing dynamics, the controller was developed to stabilize the system between Mach 5 to 7. As can be seen in Section 4.4 and Appendix 1, this region of straight and level flight has dramatically changing dynamics. The control methodology developed for such a region may be readily extended beyond what is presented without difficulty.

23 Mach numbers, 15 weights, and 13 altitudes were selected to develop the nodes of the gain-scheduled controller, resulting in 4,485 nodes within $\boldsymbol{\mathcal{K}}$. The specific spacing is shown in Table 6. These values were chosen as they result in a gain-scheduled controller which demonstrated smooth transitions when simulating the closed loop system.

Table 6: Gain-scheduling node parameters

Scheduling Parameter	Range	Increment
Mach Number	Mach 4.9 to 7.1	Mach 0.1
Weight	160,000 to 230,000 lbs	5000 lbs
Altitude	68,000 to 92,000 ft	2000 ft

6.2 Outer Loop Control Design and Implementation

The purpose of the inner loop controller is to stabilize the unstable aircraft model. This, in turn, allows an outer loop controller to serve as a guidance control to the system. As was mentioned in the opening to this chapter, many different forms of guidance logic may be used, depending on the desired application of the system. The outer loop controller may be a human piloting the system, or a sophisticated collision avoidance system, such as the one shown in Reference [52]. The outer loop controller designed for the GHV serves to command Mach, altitude, sideslip angle, and yaw angle (M, h, β, and ψ) in order to maneuver within the flight envelope. These outer loop controllers also serve to drive zero steady state error on key terms within the system, an attribute not designed into the inner loop controller.

With the inner loop controller fully stabilizing the aircraft, a less advanced, easier to tune controller may be chosen for simple outer loop guidance. A combination of PI (proportion/integrator) and PID (proportional/integrator/derivative) is selected for outer loop control. Note that the outer loop controllers presented in this section are not intended as a final navigation solution for the aircraft. Rather, it is intended to exercise the inner loop controller within simulation. To this end, the automated PID tuning tools within MATLAB® are used to simplify development.

Figure 31 illustrates the arrangements of the controllers incorporated into the system, along with the naming convention of each signal. Note that G is the GHV dynamic model, K_∞ is the H_∞ controller and K_1, K_2, and K_3 are the PI and PID controllers.

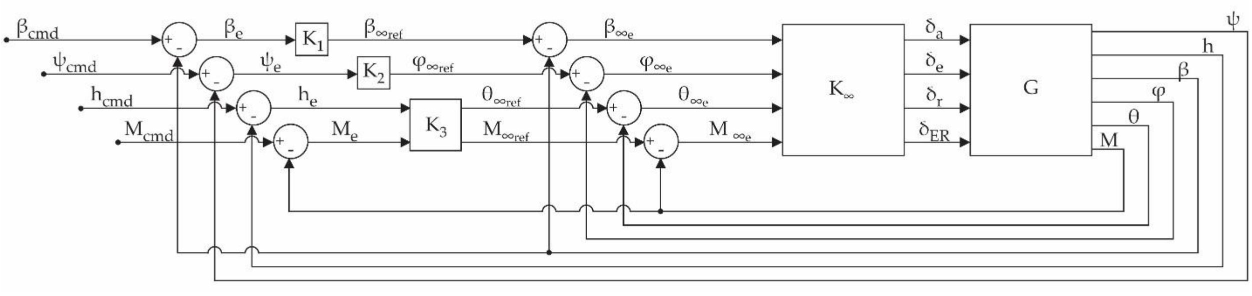


Figure 31: Inner and outer loop control implementation

K_1 *and* K_2

As shown in Figure 31, K_1 generates the inner loop side slip command, $\beta_{\infty_{ref}}$. This is a PID controller which results in zero steady state error, something not guaranteed by the inner loop controller. This loop closure was chosen to exercise the GHV's performance in directionally steady flight, as will be demonstrated in Chapter 7.

K_2 commands roll angle, ϕ, based on a commanded yaw angle, ψ. While a PID controller was tested on this loop closure, a PI controller showed better performance. ψ was selected for command to, once again, enforce directionally steady flight.

The control equation for K_1 and K_2 are given in Equation (6-6). This is the general formulation for a PID controller where K_P, K_I, and K_D are the weighting associated with the proportional, integrator, and derivative, respectively. K_N is the filter coefficient.

$$K_P + K_I \frac{1}{s} + K_D \frac{K_N}{1 + K_N \frac{1}{s}} \tag{6-6}$$

To select the appropriate values for the gain, Simulink's® "PID" block was used. This block incorporates a linearization algorithm similar to that used to generate the linear models for the entire dynamic system (see Section 4.2). The algorithm linearizes about the trip point used to initialize the entire dynamic model, reducing the system down to a single input and single output—namely the input and output to the block. The user then chooses the response characteristic by selecting the response time and the transient characteristics of the newly created linear system's response to a step input. The end results are the four parameters needed to fully define the PID controller. Note that the user never explicitly interacts with the system model, but rather tune based on the response to the internally generated linear system [59].

Once the controller has gone through its initial automated tuning, the entire nonlinear model is run with the resulting characteristics. The individual parameters are then tuned either by another iteration using the automated tuning process described above or manually. This process is repeated until a reasonable controller is found.

K_3

As shown in Figure 31, K_3 is a MIMO controller. This was done because, in order to reach the desired altitude, both Mach number and θ must be commanded simultaneously. To accomplish this, a PI controller is placed on the direct signals (M_e to $M_{\infty_{ref}}$ and h_e to $\theta_{\infty_{ref}}$) and the cross axis signals (M_e to $\theta_{\infty_{ref}}$ and h_e to $M_{\infty_{ref}}$). The implementation equation is shown in Equation (6-7).

$$\begin{bmatrix} M_{\infty_{ref}} \\ \theta_{\infty_{ref}} \end{bmatrix} = \begin{bmatrix} P_{11} + I_{11}\frac{1}{s} & P_{12} + I_{12}\frac{1}{s} \\ P_{21} + I_{21}\frac{1}{s} & P_{22} + I_{22}\frac{1}{s} \end{bmatrix} \begin{bmatrix} M_e \\ h_e \end{bmatrix} \tag{6-7}$$

Unlike K_1 and K_2, automatic tuning was not possible. Instead, this controller was tuned manually by selecting appropriate coefficients using dimensional analysis and iterative modification based on the nonlinear simulation system response. This process does not explicitly require calculating a plant model. As a consequence, however, many controller iterations are needed to arrive at a satisfactory controller. The resulting controller yields a Mach and θ trajectory which the inner controller can follow that will result in the desired commanded Mach and altitude.

Chapter 7 Results

With the fully designed controller implemented in the nonlinear model, the resulting simulation shows a stable system under nominal conditions. For this simulation, the GHV was commanded to transition from Mach 5 to 7 while changing altitude from 70,000 to 90,000 feet beginning at 10 seconds into the simulation. All other inputs of the system are held at zero. This range was chosen to test the controller ability to transition through a region with dramatically changing dynamics.

The controller in use is smoothly transitioning between numerous designed controllers, varying as Mach number, altitude, and weight changes. The weight of the system is constantly changing as a result of changing throttle position. Moments of Inertia and center of gravity are also changing as fuel is burnt. The controller gain-schedules to account for the shifting dynamics.

Note that the system is commanded using a gradual transition as opposed to an abrupt step input. This more realistically represents typical speed increases in the aircraft. This also allows the controller to account for the nonlinearly changing dynamics being driven by the speed increase.

7.1 Controller Responses

Due to the presence of both an inner and outer loop controller, it is reasonable to look at their inputs related to output of the system separately.

Inner Loop Controller

The inner loop controller consists of the outputs from the outer loop controllers as they feed into the H_∞ controller. The commanded signals (denoted with subscript ∞, ref) are generated by the PI and PID controllers of the outer loop controller. As the formulation of the inner loop controller did not have an integrator added to its development, it is expected that there will be a steady state error between the commanded values and the system's outputs. The steady state error can be seen clearly in Figure 32. Note that the commanded roll angle, ϕ_{cmd}, is significantly different than the system's roll angle. This is due to the closed loop system's insensitivity to roll angle commands. The predominant modes which are influenced by ϕ are the Spiral modes (Mode 4 in Figure 13). At this Mach range, this mode is near the origin and is, therefore, harder to influence. The outer loop controller compensates for this by applying greater command values to affect small changes.

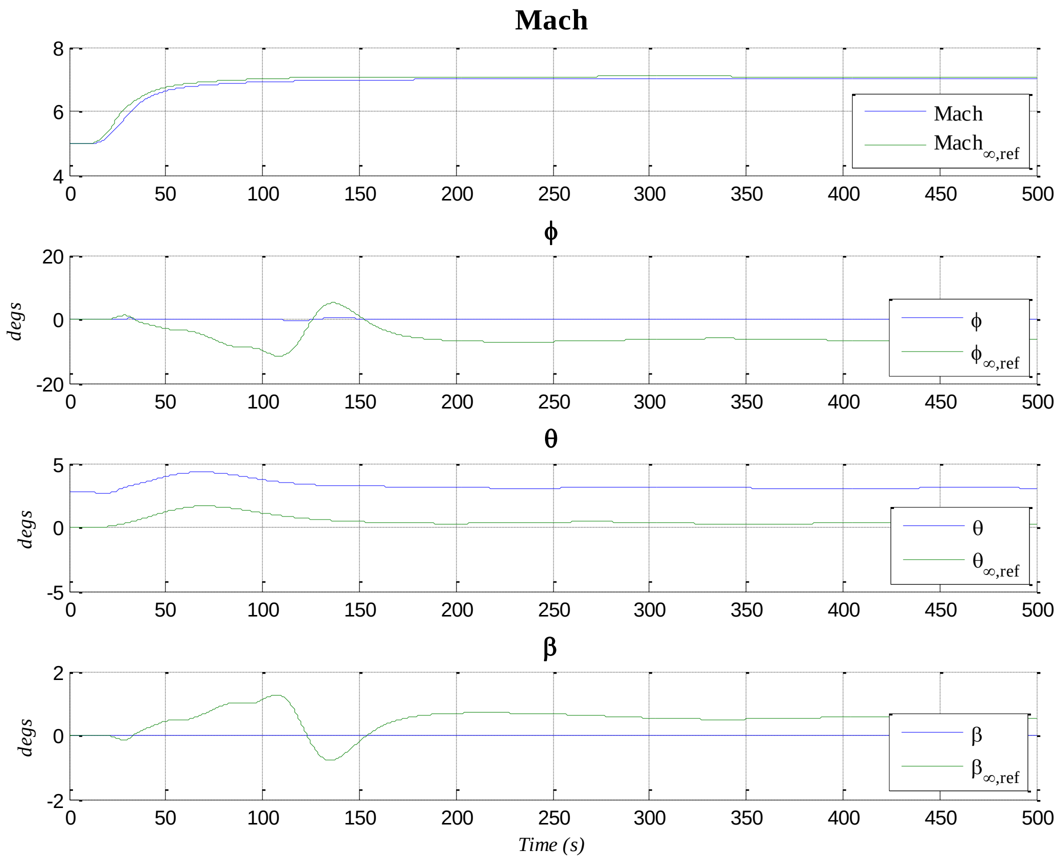


Figure 32: Inner loop controller commands to outputs

Outer Loop Controller

The outer loop feeds inputs into the inner loop H_∞ controller through three different PI or PID controllers. The inputs are Mach number, Ψ, β, and altitude (h). These can be seen in Figure 33. Due to the presence of the integrator on each system, there should be no steady state error from input to output. As the intent of the GHV is as a single-stage-to-orbit vehicle, the commanded real world trajectory would increase speed and altitude simultaneously. For this simulation, Mach is commanded from 5 to 7 while altitude is commanded from 70,000 to 90,000 feet. β and Ψ are held at zero to preserve coordinated flight and trajectory.

Some oscillation is observed in altitude. Figure 34 examines this more closely by extending the simulation to 2500 seconds. Oscillations such as these are typically associated with an excited Phugoid mode. This is not the case here, however. A Phugoid mode oscillation would have a period of about 20 seconds [34]. In this case, the system oscillates with a 150 second period, indicating it has a different source. The oscillations shown are a consequence of the outer loop controller. Recall that the controller governing this state is a MIMO PI controller, and is therefore, more challenging to tune due to the greater number of tunable parameters. Additional tuning may reduce the amount of oscillation, however, it is not necessary. The oscillations seen are only 1.5% of the commanded altitude change while the overshoot is only 1%. Note that a system with optimal damping ($\zeta = 0.707$), has overshoot of 4.33% [60]. While this is only true for SISO systems, the same standard for overshoot tolerance is applicable. The 1% seen here is well within acceptable limits for control systems.

Looking at the control surface deflections, it can be seen that they remain reasonably small. All deflections are within reasonable physical limits and saturation is never seen. It should be noted that the elevator continues to drift as the simulation progresses. This is due to the fact that the aircraft has changing mass. As fuel is burnt, the elevator must continually change in order to keep the aircraft within trim.

One alarming characteristic of the GHV is seen in the elevator deflection (see Figure 35). Even at high speeds, the deflection remains large. This is a consequence of the aft CG position of the GHV. Note again that elevator deflection is defined as positive when trailing edge is up. In order for the aircraft to remain in trim, the system must exert enough control force to counteract the CG position. A negative deflection is required to counteract the pitch up tendency. This same phenomenon was observed in Reference [19]. That model showed deflections of between 9 and 10 degrees at Mach 8 while at 85,000 feet altitude.

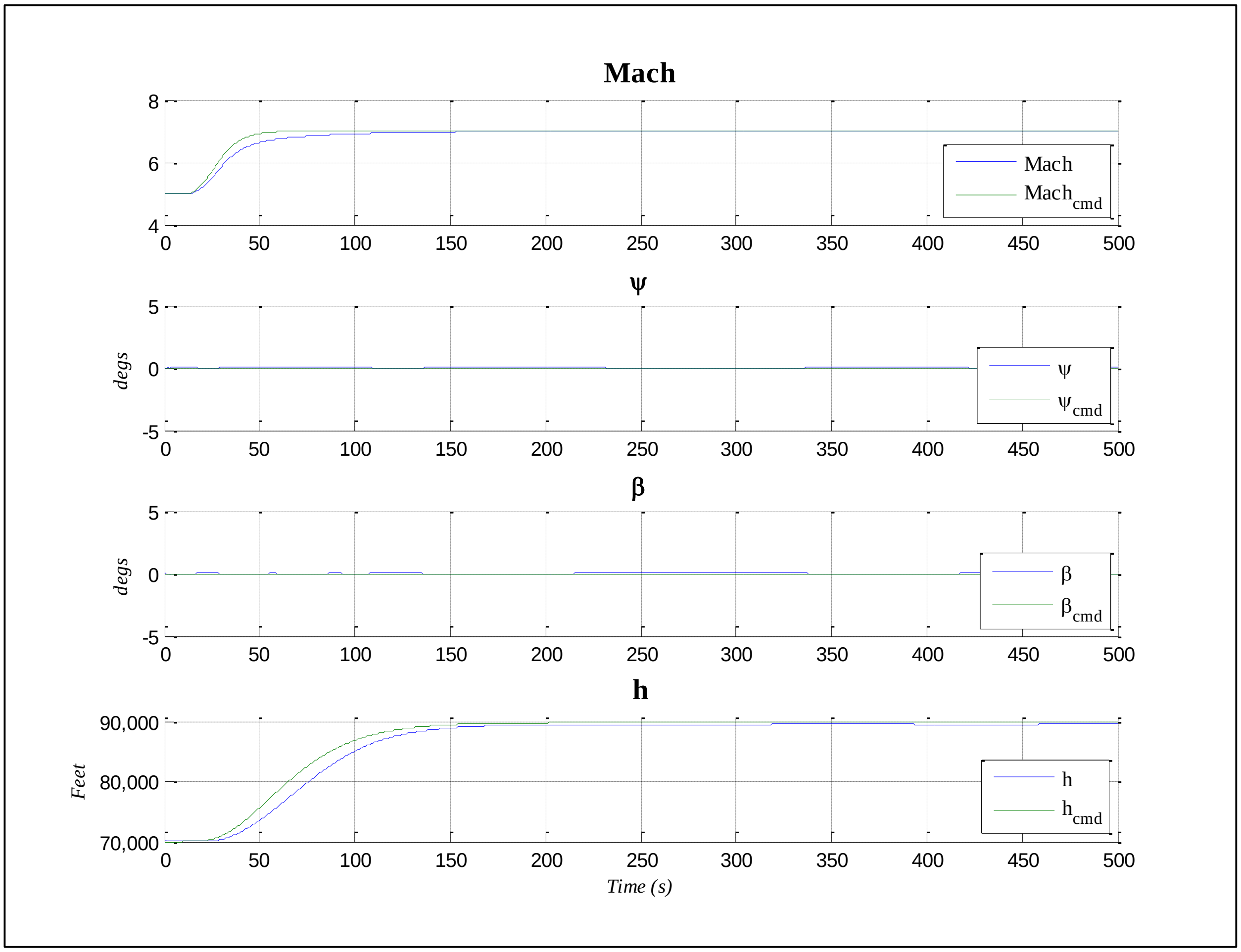


Figure 33: Outer loop controller commands to outputs

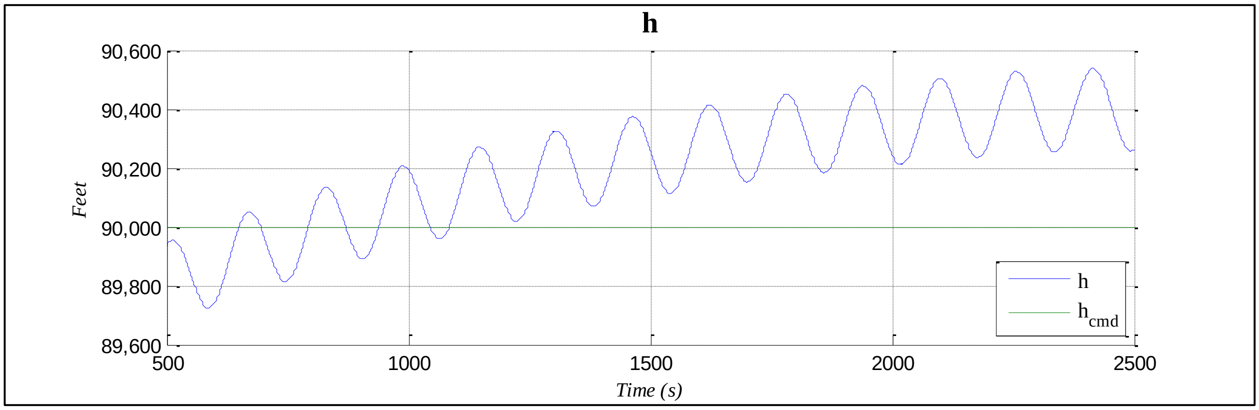


Figure 34: Commanded vs. demonstrated altitude showing marginal stability

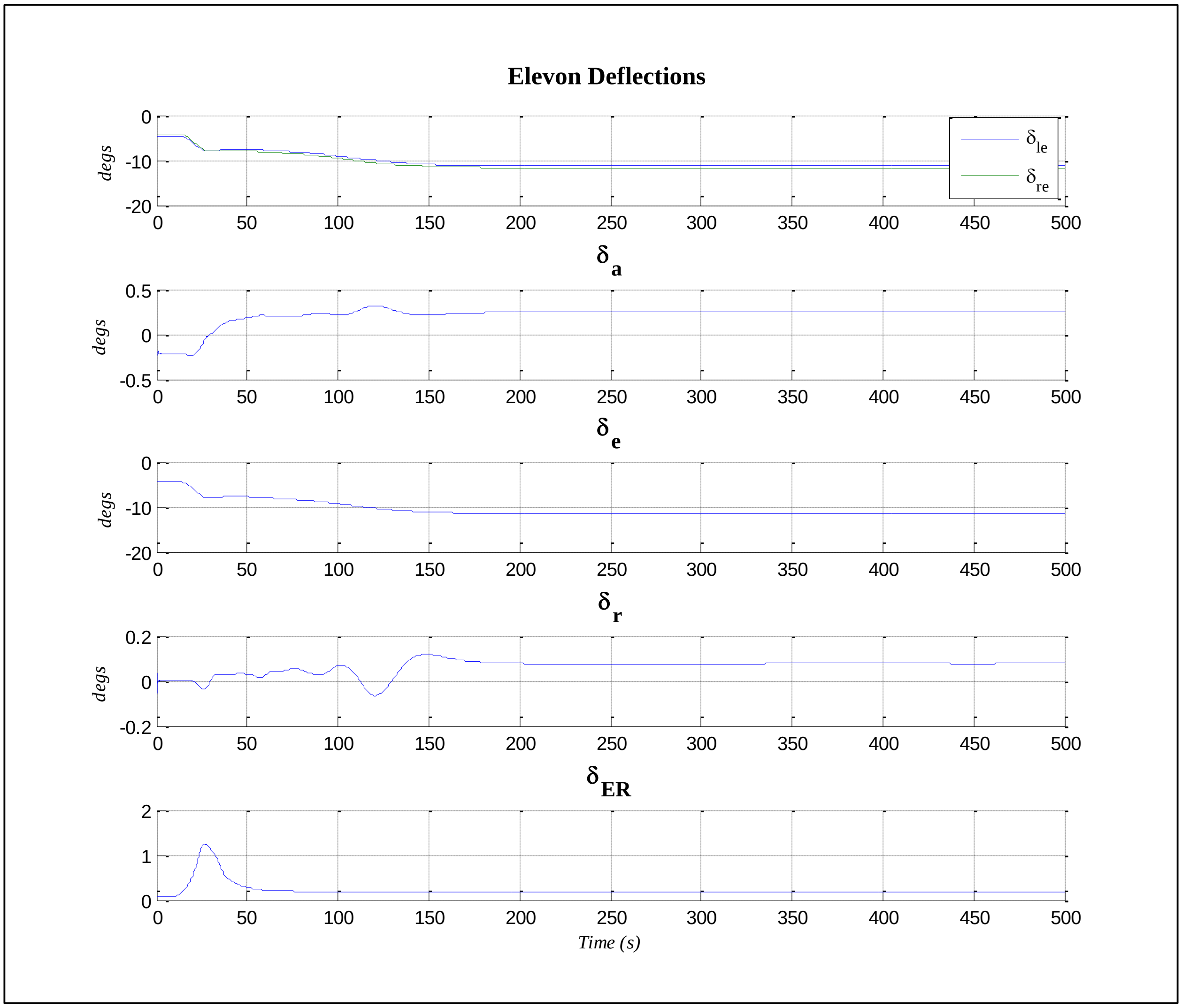


Figure 35: Control surface commanded values

7.2 Overall System Response

From the above figures, it can be seen that the closed loop system tracks inputs well and remains stable throughout. Each control surface responds to the changing commanded states over the time frame in which the state is changing. Beyond this point, the control surfaces remain near constant, only changing to compensate for the changing CG position. Note that the aileron and rudder are not constant throughout the entire simulation. While the magnitude of their deflection remains small, their contribution indicates that the Lateral-Directional dynamics are coupled with the Longitudinal dynamics. This is significant because this effect has been largely disregarded in the open literature.

Below are plotted additional system states. Figure 36 shows the changing body axis angles. Once again, coupled Longitudinal and Lateral-Directional dynamics are seen with small deviations in Φ. For a closer view of this parameter, see Figure 38. Small deviations in Ψ are a consequence of changing Φ. As the magnitude of these changes are very small, the system shows good Ψ tracking. The small oscillation observed in Θ is caused by the MIMO PI outer loop controller as discussed above.

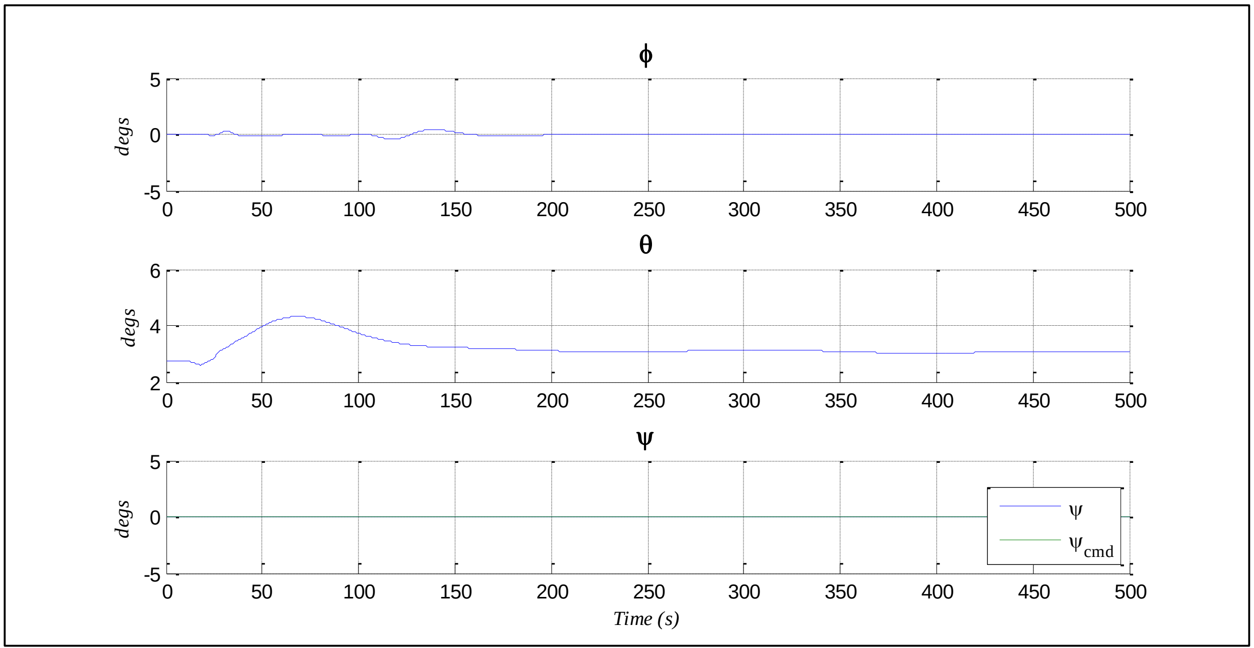


Figure 36: Body axis angles

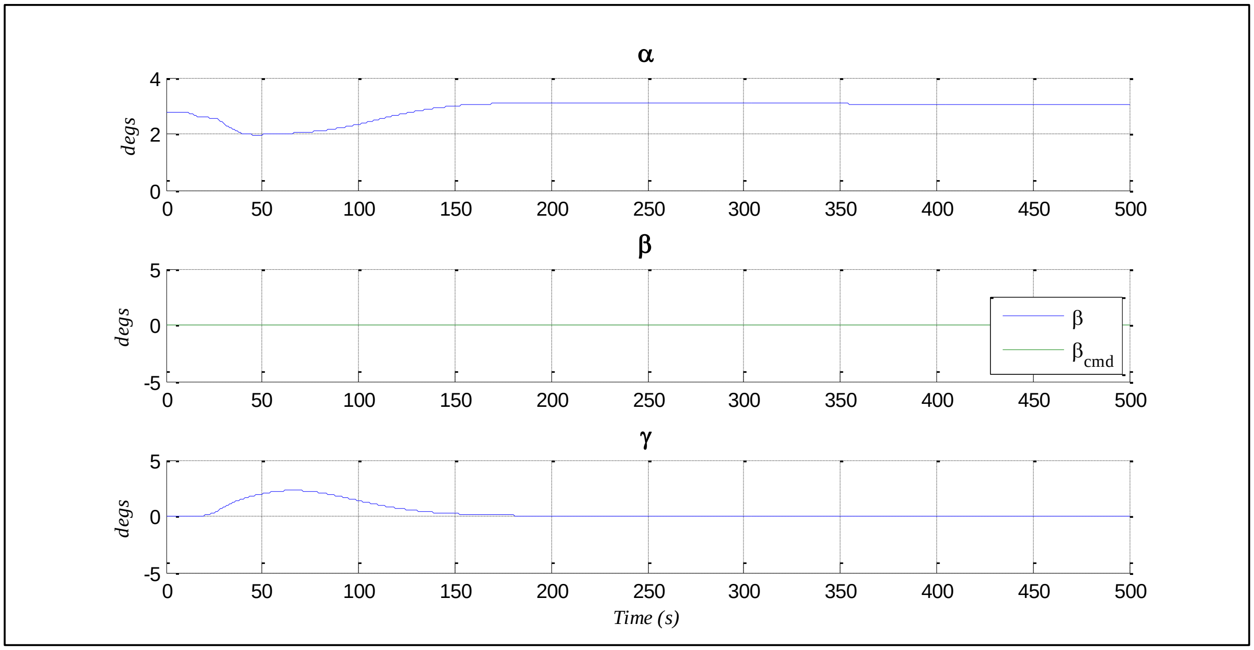


Figure 37: Stability axis angles

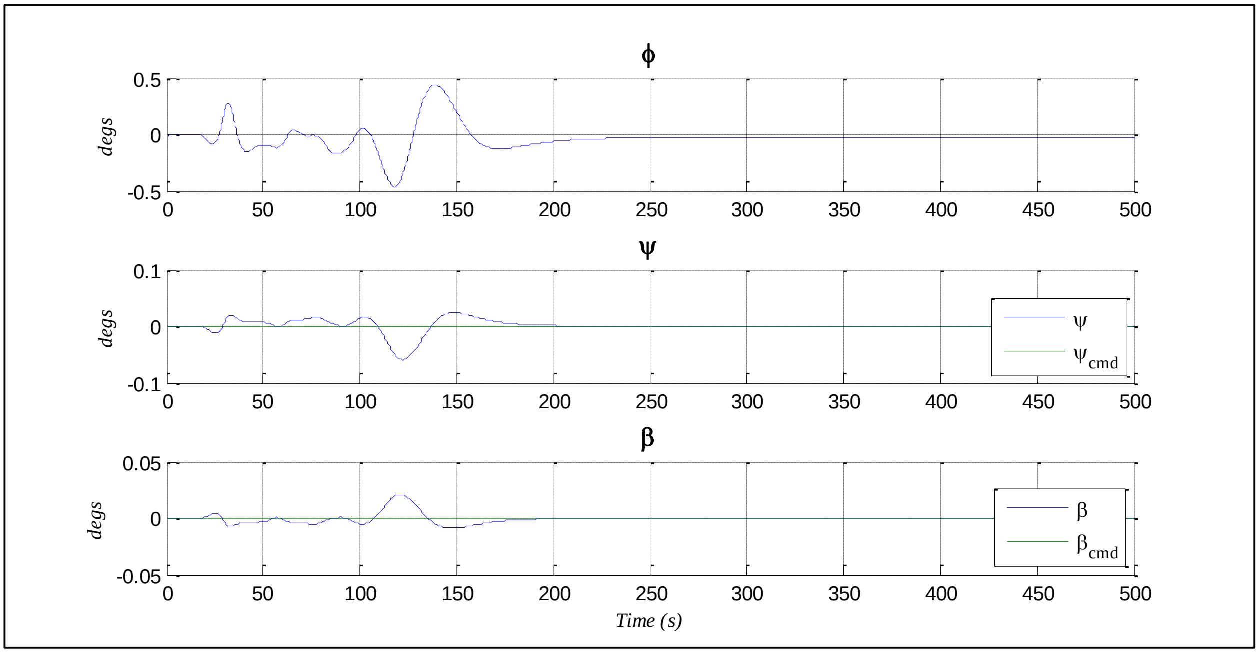


Figure 38: Closer view of angles

The stability axis angles are shown in Figure 37. α initially decreases to compensate for the changing gravity vector resulting from the commanded climb. As the commanded altitude is reached, α stabilizes to a new trim state. The change in velocity is not enough to compensate for the change in air density, resulting in a higher trim magnitude. γ follows a smooth trajectory which is indicative of a smooth transition between altitudes. As altitude reaches its commanded value, γ returns to zero. β, shown more closely in Figure 38, once again demonstrates the coupled dynamics within the system.

The angular rate magnitudes, shown in Figure 39 and Figure 40, are very small, with the maximum change less than 0.1 degrees per second. Additionally, the changes that are evident are slow. For instance, the relatively large oscillation observed in p starting around 25 seconds has a period of roughly 25 seconds. If a human were present on the system, such a small change ($\pm$ 0.08 deg/s) over such a large period would not even be noticed.

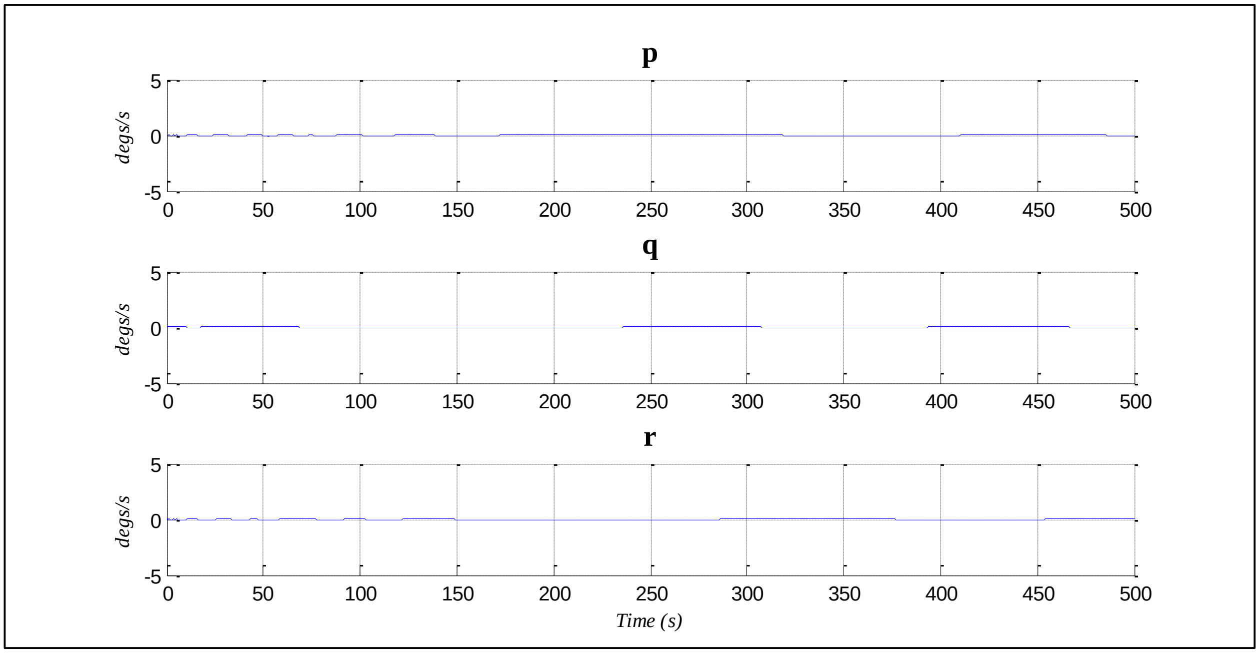


Figure 39: Angular rates

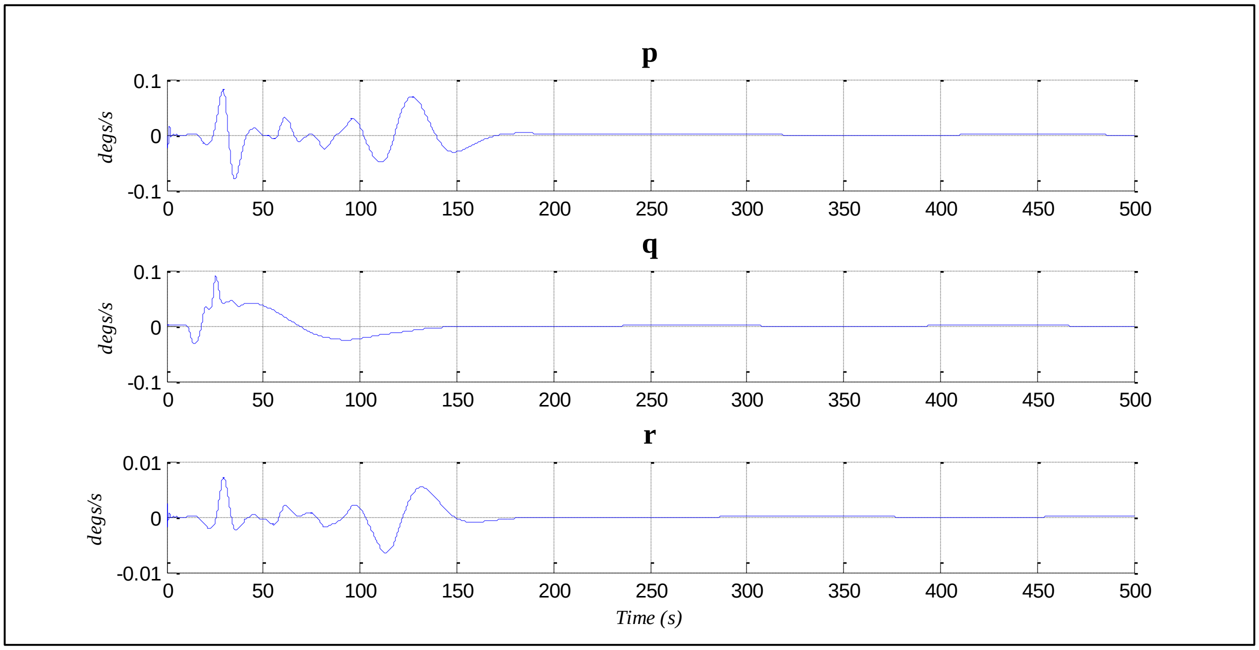


Figure 40: Close view of angular rates

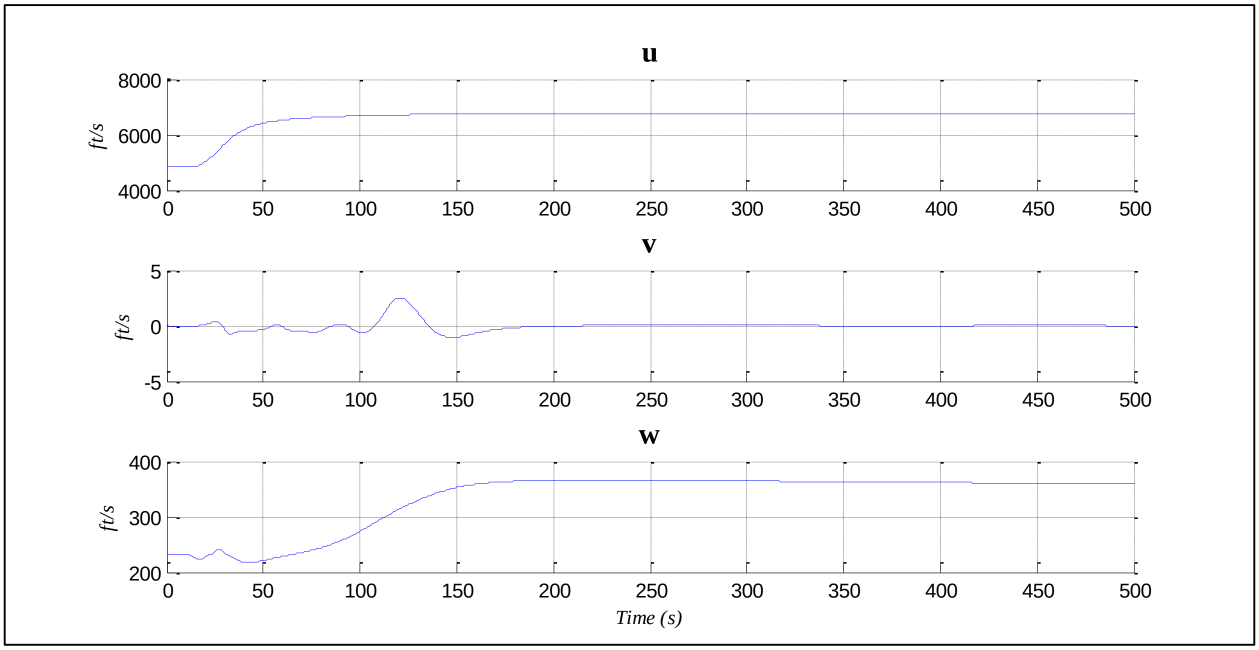


Figure 41: Body velocities

Figure 41 shows the body velocities. Note that the forward velocity, u, transitions smoothly from the lower to higher speed. w has a very slight ripple beginning at approximately 20 seconds. This is the time during which α and γ are balancing to compensate for simultaneously changing altitude and speed. Once this transient is damped out, the change is smooth. w also decreases slightly as time progresses. This is due to the shifting CG position within the aircraft. The side velocity, v, is negligibly small relative to the overall velocity of the GHV.

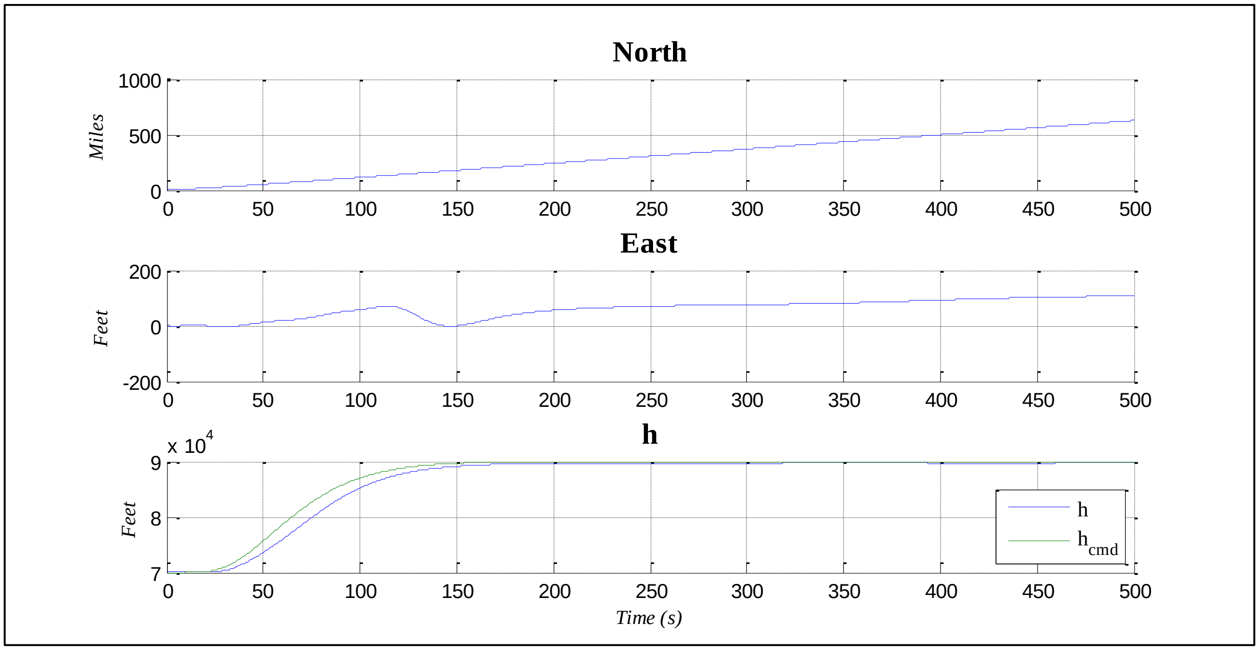


Figure 42: GHV Position within the inertial coordinate system

While trajectory is not explicitly commanded (shown in Figure 42), the aircraft shifts to the east only by about 100 feet over 700 miles (half a vehicle length). This is a consequence of the small deviations in Ψ as the system changes trim conditions. Considering the velocity at which the aircraft is traveling, this change is negligible.

Figure 43 shows the changing mass properties as the weight of the aircraft changes. The propulsive force (F_P) is plotted for reference. When F_P increases, the rate of fuel burn also increases. Consequently, the mass of the aircraft changes more rapidly. This is evident by the large changes seen between 10 and 100 seconds, the same time over which the Mach number is changing. The changes from 100 to 150 seconds are less dramatic, but still evident due the aircraft reaching its final altitude at 150 seconds.

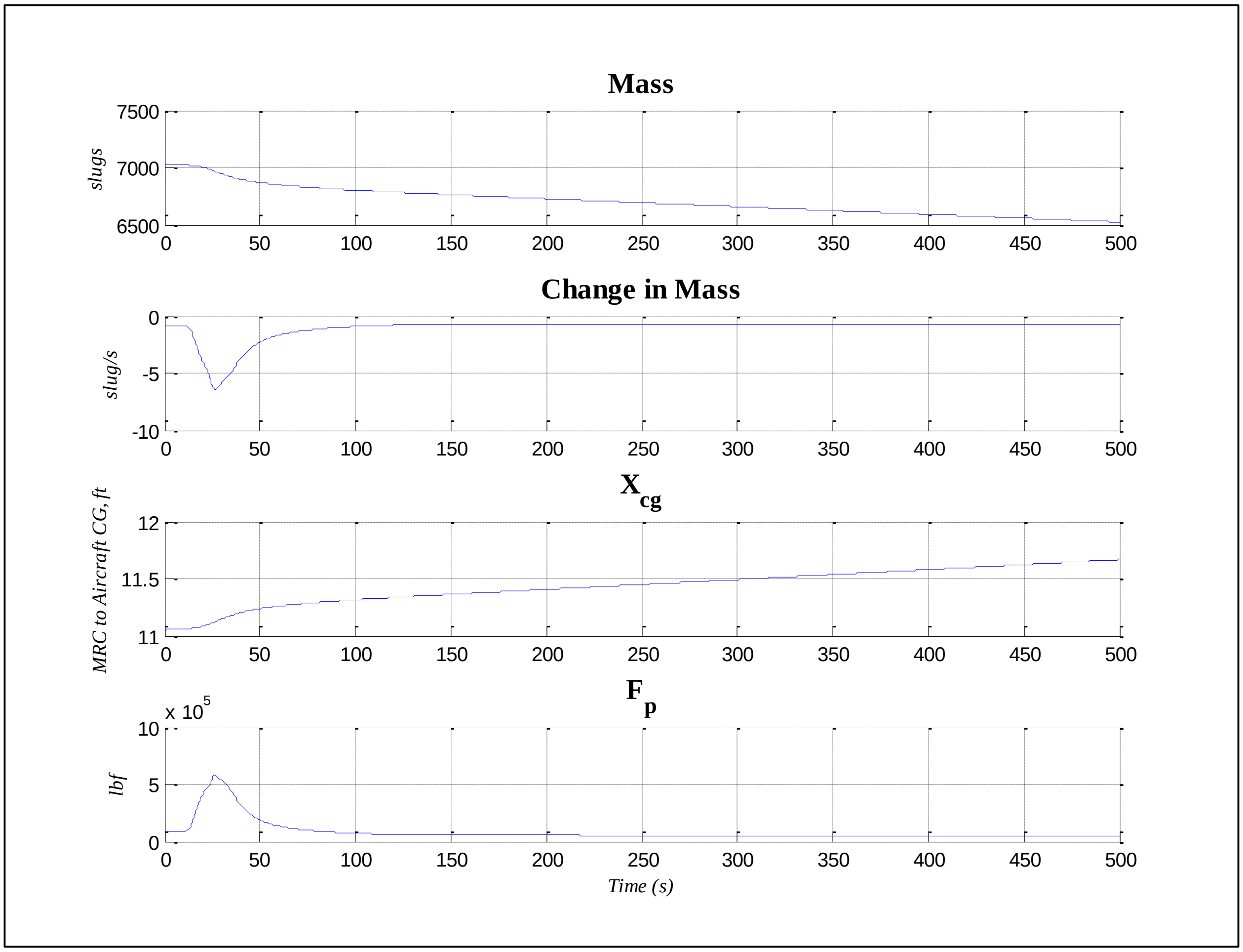


Figure 43: Changing mass parameters

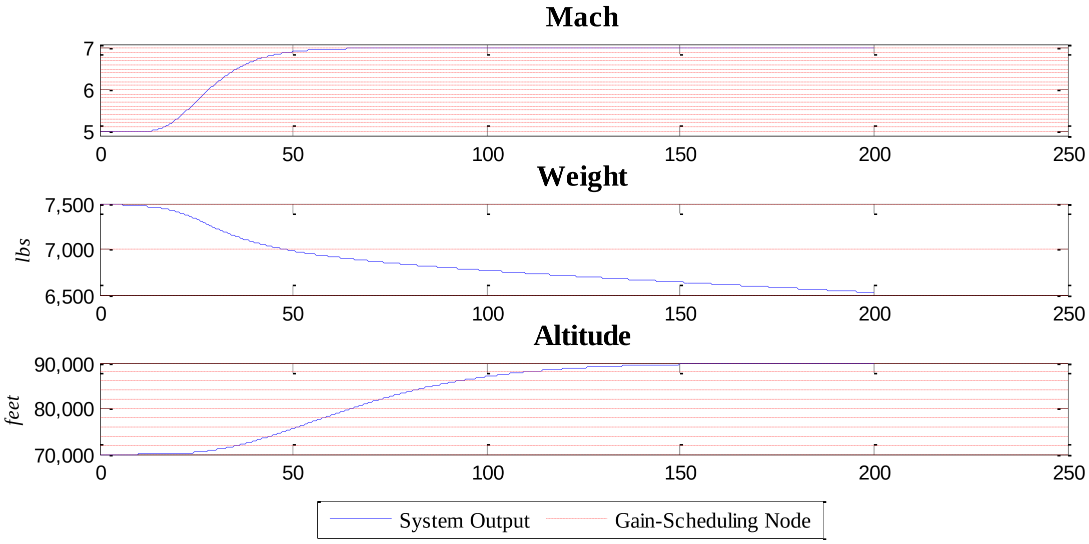


Figure 44: Gain-scheduling throughout the simulation

As shown in Section 6.1, 425 separate parameters of the controller are changing in response to the changing Mach, altitude, and weight. Figure 44 indicates where each of the node controllers, $\mathcal{K}$, fall within the presented simulation. At each time step, the controller is slightly different than the previous time step as it has been interpolated from between each of the nodes shown. Beyond 150 seconds, the controllers do not change as dramatically. However, due to the constantly changing weight, it must still be updated to account for the changing mass, center of gravity, and moments of inertia.

With the gain-scheduling algorithm disabled, changes in any of these three parameters result in the system becoming instantly unstable. The fact that the system remains stable due to commanded changes to Mach and altitude shows that the gain-scheduling is performing as expected. Overall, the system tracks very well in nominal conditions. It is able to handle the transition between multiple linear models through the use of gain-scheduling. The system remains stable despite the instability of the system and the nonlinearities of the model.

Previous works have shown that the H_∞ controller is effective at stabilizing the complex dynamics of hypersonic flight [16]–[19]. As with those works, this system shows that a properly designed H_∞ controller will stabilize the system. This work adds to that result by controlling the system beyond what has already been accomplished—extending the controlled flight envelope to include the cross-coupled Lateral-Directional dynamics as well as increasing the controllable Mach number range.

7.3 Effects of Disturbances on the System

To test the controller's resilience to external disturbances, a low frequency random signal was injected into the closed loop system as a disturbance as shown in Figure 22. Note that only frequency domain representations of the signals are shown in this section. The time domain signals show that the system remains stable, but noise dynamics are not particularly noticeable. As a result, time domain signals are of limited utility for analyzing the system's performance. Consequently, these plots have been relegated to Appendix 3.

The below figures show the frequency domain spectral content of each of the commanded signals within the inner loop controller. Both Mach and θ remain almost completely unaffected by the disturbances. The signals that show the greatest deviation are ϕ and β.

The β and ϕ channels show considerable sensitivity to low frequency disturbances. This can be attributed to the relationship between the weighting functions $W_1$and W_3. As compared to Mach, these two weighting matrices are fairly far apart at lower frequencies. This means that, according to the H_∞ formulation, more low frequencies will be passed through by W_1 without the attenuating effects of W_3 taking effect. As higher frequencies are considered, the W_3 attenuates to a greater level and the system output of the noisy system begin to resemble the nominal system again.

It should be noted that the weighting functions associated with θ are also considerably different at the shown frequencies, though the noise effects on the system are less dramatic than those of β or ϕ. This can be attributed to the natural filtering effects of the GHV dynamics. This shows the system is less susceptible to pitching disturbances.

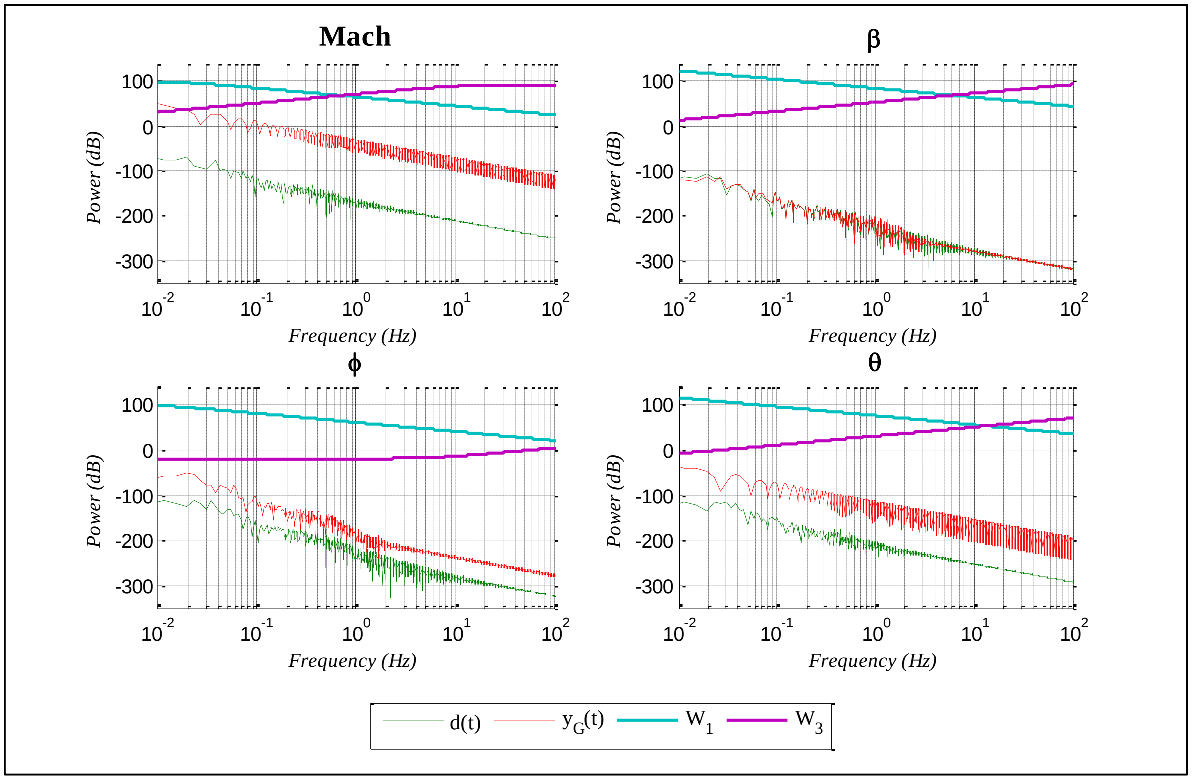


Figure 45: Spectral density of system with disturbances

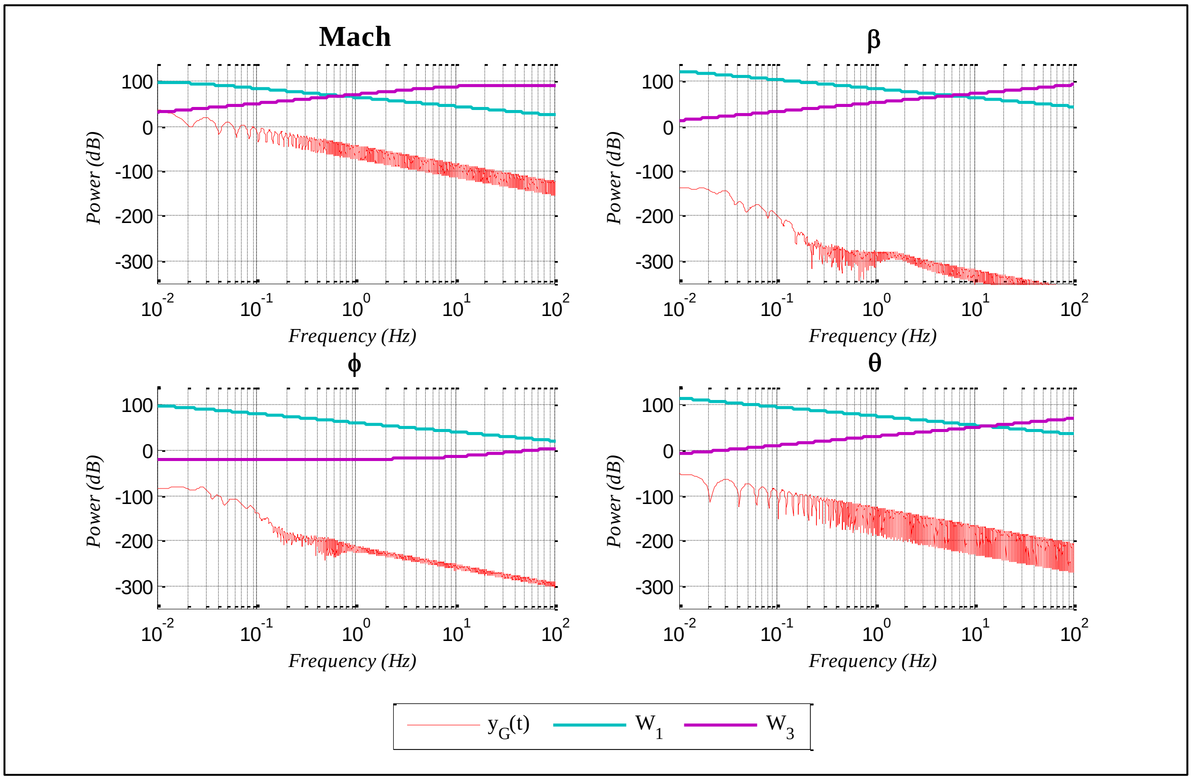


Figure 46: Spectral density of the nominal system

Another important aspect of the noisy system to consider is the control surface spectral content. Note that aileron and equivalence ratio show a greater susceptibility to noise than do the rudder and elevator. This is due to several factors.

The equivalence ratio is penalized to the highest extent among the system inputs by W_2. The result of this penalty is seen in Figure 47 as a lower magnitude signal. This makes changes in the magnitude more noticeable. The frequency band where the greatest increase occurs is between 0.2 and 4 Hz (Figure 48). This band corresponds to a similar, albeit less pronounced rise in Figure 47. This region is highlighted in both figures. The equivalence ratio has a relatively large increase in this region. This indicates that the throttle control in the system is susceptible to these frequencies. The consequence of this is that external noise will be conducted into the engine. As this is a disturbance range typically seen in aircraft system, this trait is undesirable overall.

The rise in aileron magnitude can be attributed to the increase in ϕ magnitude seen in Figure 45. The frequencies below 1 Hz are the most significantly affected in ϕ, corresponding to a similar rise in δ_a in Figure 48. As aileron is the principal control surface affecting ϕ, it is expected that more aileron will be used to counteract it.

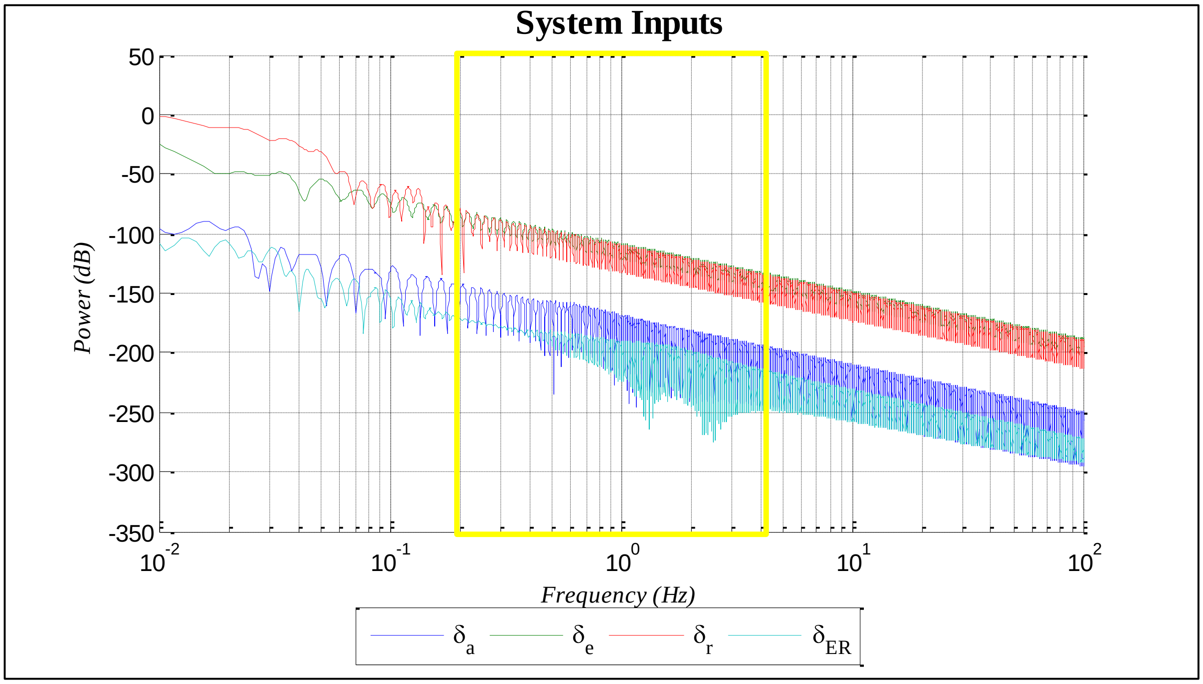


Figure 47: Spectral density of nominal system's control surfaces

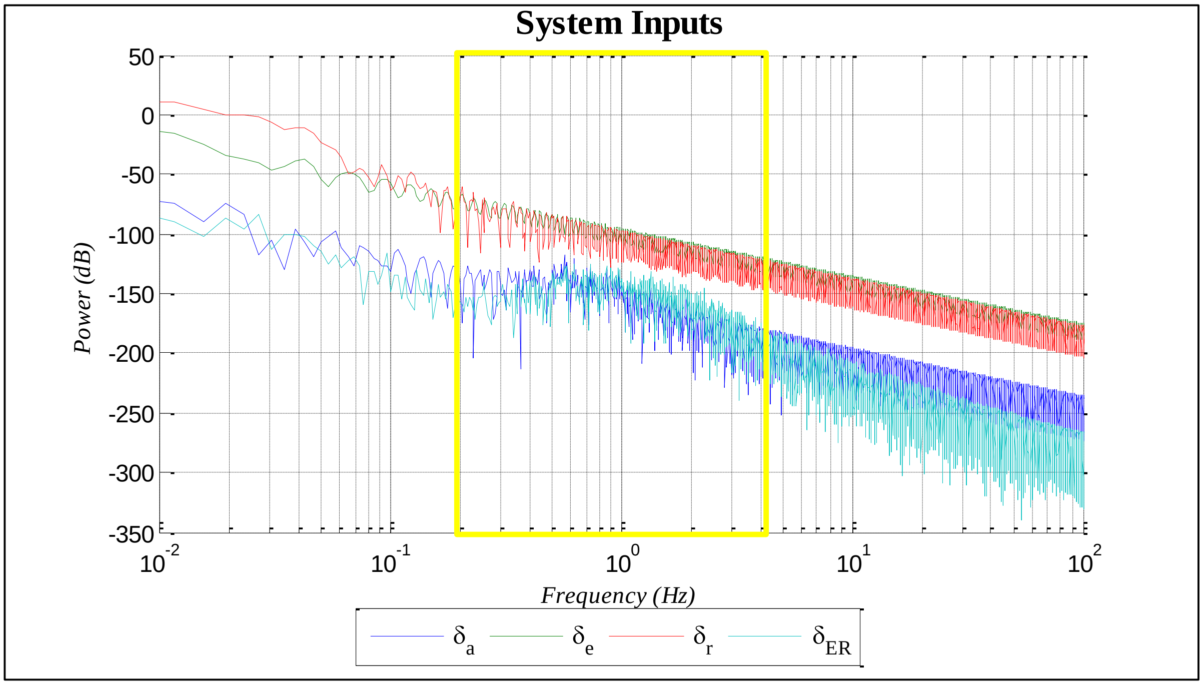


Figure 48: Spectral density of disturbed system's control surfaces

Another important trait to point out is that all signals in Figure 48 are slightly greater than those in Figure 47. This is a consequence of the weighting function W_2. This function weights all controls equally over all frequencies. Thus, in the controller synthesis, no filtering effect was imparted into the input signal dynamics.

Note that the model development does not include actuator dynamics. If present, they would serve to filter out higher frequency content in the control surfaces. In a real system, these dynamics should be measured and incorporated in the model. If not accounted for, these un-modeled dynamics may destabilize the closed loop system.

In general, the controller performs as expected as it maintains system stability in the presence of external disturbances. Some noise is conducted into each signal, though it is primarily seen in the control surfaces. This may be mitigated to a certain extent in a real system by actuator dynamics, however additional steps should be taken to mitigate noisy commanded values. As the weighting matrices associated with system inputs (W_2) do not penalize these signals based on frequency, noisy commanded signals are to be expected. Some amount of noise (particularly at low frequencies) in the system is acceptable, however. Without it, external disturbances would destabilize the system. It is recommended that future iterations of the controller be constructed to penalize control surfaces at high frequencies.

This section shows a preliminary study of the effects of disturbances in the system. It is by no means exhaustive. The noise injected into the system may not truly characterize what would be seen by physical hypersonic aircraft. Additionally, there is no objective noise threshold which the system must remain within. Further study into hypersonic flight is needed to determine what disturbances act on the system and to what level they may be tolerated.

Chapter 8 Summary and Conclusion

This work presents a model of the Generic Hypersonic Vehicle, taking advantage of modern software for advanced implementation and simulation. It incorporates multiple cubic spline lookup tables to create a nonlinear dynamic model across a broad range of Mach numbers. The presented model includes both the Lateral-Directional and Longitudinal dynamics in order to fully simulate the system as well as provide a means of generating one unified, stabilizing controller as opposed to two axis-separated controllers. In addition, the effects of changing altitude and fuel burn are added into the model.

The nonlinear model is also linearized about straight and level flight over the broad range of Mach numbers. An analysis of the changing linearized dynamic characteristics of the system as a function of Mach number is presented along with a comparison to previously implemented version of the same system using alternative techniques.

This work also shows a design approach for developing a gain-scheduled H_∞ controller. The synthesis takes advantage of available computer processing power to perform a 28-degree of freedom Monte Carlo search to find an stabilizing H_∞ synthesis using a defined cost function. The control synthesis is expanded beyond the single linear system designs previously seen in other works, to a broader range of linear models within a portion of the linearized flight envelope. The result is a 3 dimensional, spline interpolated, gain-scheduled controller, dependent on Mach, altitude, and weight. The resulting system is further augmented with the use of PI and PID controllers for managing trajectory. The final closed loop system shows complete stability when commanded throughout the flight envelope. The system performance partially degrades when disturbances are introduced, though it still maintains stability despite the added dynamics.

The model has been generated in such a way that it can be used by others with the appropriate files to develop alternative design solutions to the chosen H_∞ controller. This makes it possible for the GHV to be implemented as a teaching tool to give students an opportunity to test different controllers on a system known to be nonlinear and highly challenging to control.

The principle impacts or this work are as follows:

1. Previous works have focused primarily on studying the Longitudinal dynamics of hypersonic flight. This work reinforces their findings while extending the analysis to a six degree of freedom model valid over a broad range of Mach numbers. This expands the studied dynamics of the system, revealing non-standard Lateral dynamic characteristics previously not reported in the open literature. In addition, the model demonstrates that the Lateral and Longitudinal dynamics are coupled. This implies that future controllers developed for hypersonic flight need to be developed from more than a planar representation of the system.
2. Other controllers developed for hypersonic flight in the open literature primarily focus on controlling the three degree of freedom Longitudinal dynamics of the aircraft around a single trimmed Mach number with constant mass. This work expands the area under control considerably. Here, the controller is developed for a six degree of freedom model with axis cross-coupling. This work also expands considerably the region for which the system may be commanded. No previously published work commands a change to either Mach or altitude to the magnitude shown here. Additionally, the complexity of the parent nonlinear model allows for the study and design of controllers capable of handling changing mass properties including, center of gravity, mass, and moment of inertia.

3. The high fidelity of the model over a broad range of Mach numbers allows for a controller to be designed for the most dynamically changing region. By selecting the most difficult region to control, the spline gain-scheduling method demonstrates its ability to stabilized dramatically changing dynamics. Subsequent extensions of the envelope will prove less challenging.

Chapter 9 Future work

While the gain-scheduled H_∞ control approach successfully stabilizes the GHV and, when coupled with external controllers, can command trajectory, many improvements could be made. Four additional areas have been identified which could expand upon the work undertaken in this document. The four options for additional study range from the most easily implemented, through to the most technically challenging.

9.1 Weighting Matrix Tuning

The art of designing an H_∞ controller is all about designing the weighting matrices. Ideally, the designer should have complete knowledge of how the system is affected by the real world in order to design the controller to withstand those environments. With this knowledge, the designer may adjust the weighting matrices to coax the desired performance from the system.

The first improvement that can be made to the presented controller is to continue to fine tune the weighting matrices. While the noise present in the system is tolerable, there is still room for improvement. Before any changes are made, however, a few design criteria must be considered. First, what noise and disturbance is likely to be seen by the system and how strong is that noise? What is meant by a disturbance? To what level are wind gusts going to affect the system? How powerful is the sensor feedback noise? What is a reasonable amount of allowable noise in the system? Is it enough that the system maintains trajectory control and stability, or should noise and disturbances be attenuated to the greatest extent possible? These and many other questions must be identified and addressed to make any modification to the system.

Once these criteria have been identified, the weighting functions may be shaped. These weights should leverage answers to the above questions. For instance, if the sensor feedback noise profile is known for a given type of sensor, it should be incorporated into the weighting function. The standard shapes of the W_1 and W_3 matrices are only guidelines. There may be a different, higher order transfer function that better represents the disturbances or noise in the system. As long as an understanding of how these weighting matrices relate to the Sensitivity and Co-Sensitivity Functions, and in turn, the noise, disturbances, and reference signals are maintained, these weights can be manipulated.

Additionally, it may even be necessary to modify the shape of the W_2 weighting function to something more than just a constant value. For instance, in the current version of the controller when noise is injected into the system, a 0.2-4Hz signal is propagated to the control surfaces. While it does stabilize the system, this may cause other problems to the aircraft structure. This problem may have been mitigated if W_2 penalized higher frequency signals more heavily than it does lower frequencies.

For a complicated system such as the GHV, it may be necessary to break away from the mixed sensitivity formation of the H_∞ controller altogether. A more complicated, but potentially more beneficial, version may be possible which would weight pertinent signals directly within the nonlinear model. This may improve overall performance and robustness.

9.2 Eliminate Outer Loop Controllers

While the H_∞ controller fully stabilizes the system, it does not fully control all the desired states. Additional outer loop controllers had to be added to augment the inner loop to achieve practical performance. While functional, this strategy is not ideal. The presence of the additional controller, in essence, adds additional dynamics to the system which were not present when the linear models for the GHV were derived. This means that the inner loop controller was not developed for the system as it is implemented, but, instead for a reduced form of the system. Using this two loop technique, there is no guarantee that the resulting system is stable for all given inputs. Though the possibility is remote, under a certain set of inputs, the two sets of controllers could act against each other and cause instability.

To get around this unintended dynamic coupling of multiple controllers, a new, single controller may be developed which would include the effects of both sets of controllers of the presented design. This is achieved with two primary modifications to the system.

One of the reasons for the outer loop controller was to drive the system to zero steady state error by including an integrator in the feedback path. It is common for this task to be handled within the inner loop controller. There is no reason that the H_∞ controller could not have this built in inherently. There are two ways of accomplishing this: feed back and command rates or modify the weighting matrices. Feeding back and commanding the rate value of a signal adds an integrator into the feedback loop and thus causes zero steady state error in the signal. To get zero steady state error by modifying the weighting functions, W_1 should be shaped such that it has a pole at the origin for all signals for which zero steady state error is desired. As frequency of the signals $y(t)$ approaches DC, the penalty from W_1 becomes infinite. The only way the control synthesis algorithm has to compensate for this infinite penalty is for the DC offset to always be zero. This translates to an inherent integrator in the controller [61].

The second change that should be made to eliminate the outer loop controllers would be to add or change which quantities are fed into the control system. For instance, instead of feeding back θ which can then be used to command altitude, feedback altitude itself. Similarly feeding back ψ instead of ϕ would result in a more easily commanded trajectory. This creates new challenges, however. It is conceptually easier to develop a controller when the feed back signals correspond (without cross coupling) directly to the command inputs to the system. Feeding back altitude would, as a result, require commanding both elevator and throttle. While this was achieved by the final implementation of the presented control solution, a combined approach would be more difficult to troubleshoot. It may also be found that replacing some feedback paths causes less ideal performance elsewhere. There are also difficulties inherent with adding new feedback paths without removing others. As more channels are added in the feedback loop, it becomes more challenging to find the appropriate weighting matrices and, inevitably, more challenging to develop a single controller. Additionally, the number of values being commanded is limited to the number of actuators within the system.

A balance should be reached between number of states commanded or regulated and what is needed to attain satisfactory performance. Multiple feedback paths should be explored to determine the optimal set of feedback states.

9.3 Modeling Error and Parameter Uncertainty

The H_∞ controller has built into its architecture the ability of withstanding many types of disturbances. This work has looked at two types of disturbances in detail already: noise and low frequency physical disturbances. However, there is another type which has not been fully explored—system modeling error. The H_∞ architecture handles these uncertainties as unknown state variables in the system and can be designed around.

There are two types of modeling errors to be considered—additive and multiplicative, represented in Figure 49 and Figure 50 respectively. The Δ blocks in the system represent the amount of deviation seen on select variables as defined by the system $G(s)$.

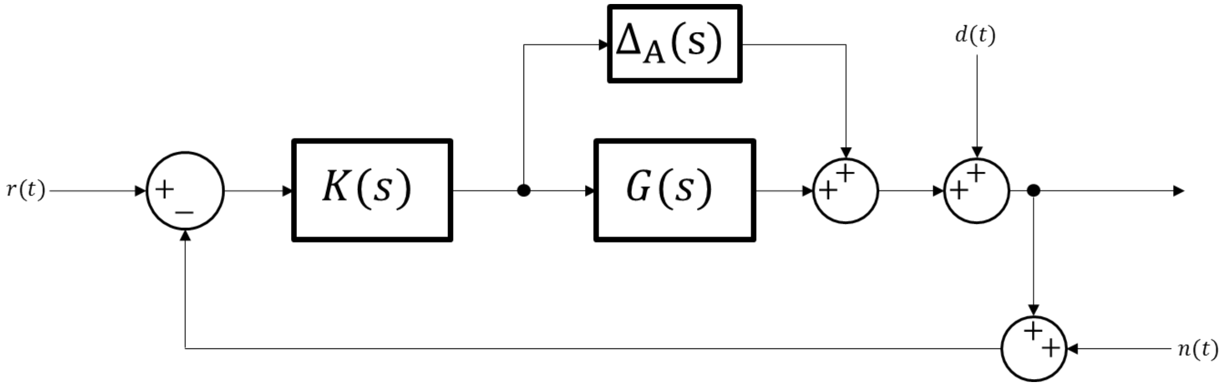


Figure 49: Disturbed and noisy feedback system with additive model uncertainty

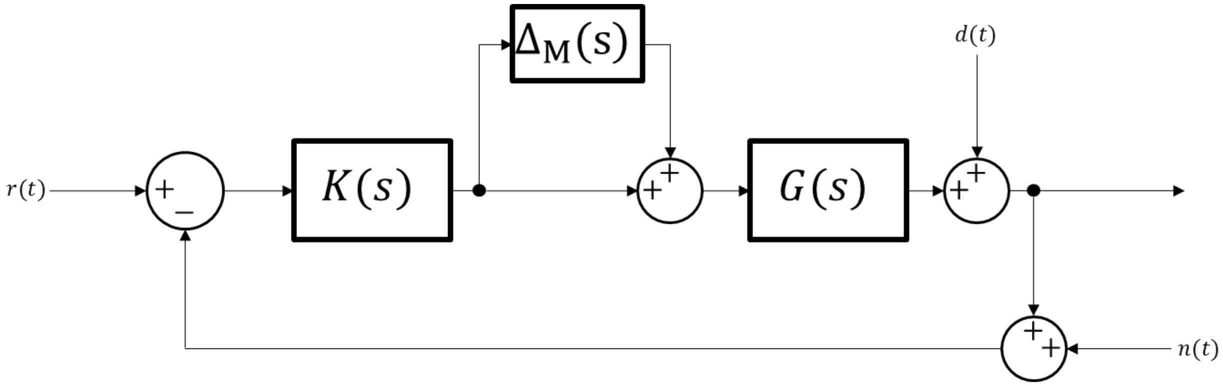


Figure 50: Disturbed and noisy feedback system with multiplicative model uncertainty

These structures can be reshaped into the three block representation of the system shown in Figure 51 similar to that described in Section 5.1. The top two blocks can be combined so that the system takes the shape shown in Figure 25. The uncertain parameters can then be treated as new states of the system which the controller must tolerate [62].

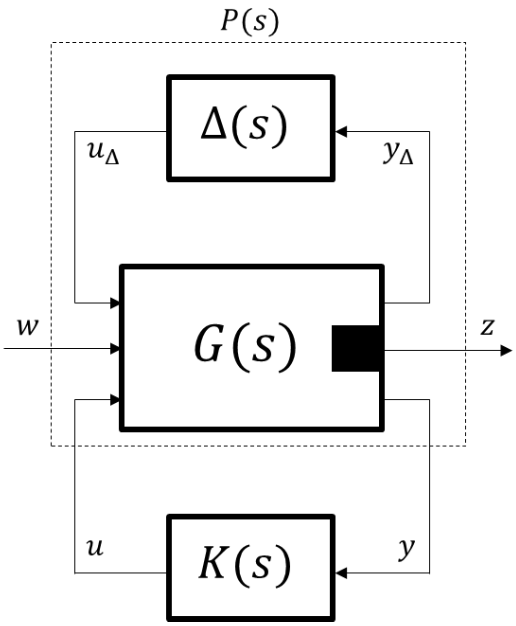


Figure 51: Three block representation

The benefit of this approach is that the controller can now control a system with unknown parameters. This could be leveraged in the GHV's H_∞ controller design. Uncertainty may be applied to many of the parameters that vary with Mach, weight, or altitude. By designing for these uncertainties, there would be less need for gain-scheduling as the controller would inherently be able to handle the changing system.

There are tradeoffs to this approach, however. By increasing the robustness of the controller, system performance is lost. If too many uncertainties are designed into the system, there may not even exist a controller capable of stabilizing and controlling the augmented system. The designer needs to take these tradeoffs into account as the controller architecture moves toward greater robustness.

9.4 Nonlinear Control

The entire focus of this thesis has been on analyzing and controlling the nonlinear system through linear techniques. This is not the only approach, however. While the available nonlinear options for analysis and control are not as numerous as that of linear theory, purely nonlinear strategies exist. If the possibilities enumerated above do not satisfy the ultimate performance, robustness, and implementability required for any final system, nonlinear options may be useful.

There are several advantages of analyzing and controlling a nonlinear system through nonlinear means. To analyze the stability of a nonlinear system in a linear framework, the system must first be linearized. This gives only a snapshot at a given time. While accurate for that instantaneous moment, the linear analysis will not tell the full story over the entire range of the system. Nonlinear techniques such as Lyapunov stability can be applied to a nonlinear system to prove ultimate stability.

Nonlinear controllers can also be designed to take full advantage of the nonlinear dynamics available in the model. Numerical techniques for generating predictive controllers based on a model in real time can have significant performance improvements over static controllers.

The drawbacks to these techniques may be substantial however. The Lyapunov stability criteria requires finding a specific Lyapunov function specific to the system in order to prove stability. These tend to be challenging to find at the best of times. The nonlinear model predictive controller requires significant computing power to calculate the controller in real time.

Several options exist for improving the control of the GHV. This document presents a piece of this analysis and gives guidance on further measures that can be made to improve the system in the future. As with any design, there are many tradeoffs that must be considered. Each of the presented options could solve known problems while simultaneously introducing new ones. Ultimately, the best design is the one that can successfully be implemented and satisfy the design criteria.

References

[1] A. G. Panaras, *Aerodynamic principles of flight vehicles*. Reston, VA: American Institute of Aeronautics and Astronautics, 2012.

[2] Z. Rosenberg, “DARPA’s Spaceplane of the Future,” *Air & Space Magazine*, 28-Apr-2016.

[3] Center for History and New Media, “Zotero Quick Start Guide.” [Online]. Available: http://zotero.org/support/quick_start_guide.

[4] NASA Dryden Flight Research Center Photo Collection, *X-43A Hypersonic Experimental Vehicle - Artist Concept in Flight*. 1999.

[5] M. Wall, “Air Force’s X-51A Hypersonic Scramjet Makes Record-Breaking Final Flight,” *Space.com*, 03-May-2013.

[6] Edwards Air Foirce Base, Air Force Flight Test Center Public Affairs, *X-51 WaveRider*. 2010.

[7] S. Keshmiri, “Modeling and Simulation of a Generic Hypersonic Vehicle,” University of Kansas, 2007.

[8] M. Bolender and D. Doman, “A Non-Linear Model for the Longitudinal Dynamics of a Hypersonic Air-breathing Vehicle,” 2005.

[9] M. A. Bolender and D. B. Doman, “Nonlinear Longitudinal Dynamical Model of an Air-Breathing Hypersonic Vehicle,” *J. Spacecr. Rockets*, vol. 44, no. 2, pp. 374–387, Mar. 2007.

[10] J. D. Shaughnessy, S. Z. Pinckney, J. D. McMinn, C. I. Cruz, and M. Kelley, “Hypersonic vehicle simulation model winged-cone configuration,” National Aeronautics and Space Administration, Langley Research Center ; National Technical Information Service, distributor, Hampton, Va.; Springfield, Va., 1990.

[11] G. Sova, P. Divan, and L. Spacht, “Aerodynamic preliminary analysis system 2. Part 2: User’s manual,” NASA, Rockwell International Corp., NASA-CR-182077, Apr. 1991.

[12] M. G. Safonov, “Origins of robust control: Early history and future speculations,” *Annu. Rev. Control*, vol. 36, no. 2, pp. 173–181, Dec. 2012.

[13] IsaaC.M. Horowitz, *Synthesis of Feedback Systems*. New York, New York: Academic Press Inc, 1963.

[14] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, “State-space solutions to standard H2 and H ∞ control problems,” *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 831–847, Aug. 1989.

[15] B. R. Cobleigh, *Development of the X-33 Aerodynamic Uncertainty Model*. 1998.

[16] Z. Gao, B. Jiang, P. Shi, J. Liu, and Y. Xu, “Passive Fault-Tolerant Control Design for Near-Space Hypersonic Vehicle Dynamical System,” *Circuits Syst. Signal Process.*, vol. 31, no. 2, pp. 565–581, Apr. 2012.

[17] Huang Huang and Yong Wang, “Mixed H2/H∞ robust adaptive control of hypersonic vehicles based on the characteristic model,” presented at the Control Conference (CCC), 2012 31st Chinese, Hefei, China, 2012, pp. 2883–2888.

[18] Irene M. Gregory, Rajiv S. Chowdhry, John D. McMinn, and John D. Shaughnessy, “Hypersonic vehicle model and control law development using H-infinity and mu-synthesis,” NASA Langley Research Center, NASA-TM-4562, Oct. 1994.

[19] O. Cifdaloz, A. A. Rodriguez, and J. M. Anderies, “Control of distributed parameter systems subject to convex constraints: Applications to irrigation systems and Hypersonic Vehicles,” 2008, pp. 865–870.

[20] B. Xu, D. Gao, and S. Wang, “Adaptive neural control based on HGO for hypersonic flight vehicles,” *Sci. China Inf. Sci.*, vol. 54, no. 3, pp. 511–520, Mar. 2011.

[21] X. Li, B. Xian, C. Diao, Y. Yu, K. Yang, and Y. Zhang, “Output feedback control of hypersonic vehicles based on neural network and high gain observer,” *Sci. China Inf. Sci.*, vol. 54, no. 3, pp. 429–447, Mar. 2011.

[22] A. Ataei and Q. Wang, “Non-linear control of an uncertain hypersonic aircraft model using robust sum-of-squares method,” *IET Control Theory Appl.*, vol. 6, no. 2, pp. 203–215, 2012.

[23] H. Liu, Q. Zong, B. Tian, and J. Wang, “Hypersonic Vehicle control based on integral sliding mode method,” 2012, pp. 1762–1766.

[24] H. Duan and P. Li, “Progress in control approaches for hypersonic vehicle,” *Sci. China Technol. Sci.*, vol. 55, no. 10, pp. 2965–2970, Oct. 2012.

[25] C. I. Marrison and R. F. Stengel, “Design of Robust Control Systems for a Hypersonic Aircraft,” *J. Guid. Control Dyn.*, vol. 21, no. 1, pp. 58–63, Jan. 1998.

[26] Q. Wang and R. F. Stengel, “Robust Nonlinear Control of a Hypersonic Aircraft,” *J. Guid. Control Dyn.*, vol. 23, no. 4, pp. 577–585, Jul. 2000.

[27] J. Roskam, *Airplane flight dynamics and automatic flight controls*. Lawrence, Kan: DARcorporation, 2011.

[28] R. F. Stengel, *Flight dynamics*. Princeton, NJ: Princeton University Press, 2004.

[29] J. Roskam and C. T. Lan, *Airplane Aerodynamics and Performance,* Revised edition. Lawrence, Kan.: Darcorporation, 2000.

[30] *findopOptions*. MathWorks, Simulink R2013a Software Documentation.

[31] *fmincon*. MathWorks, R2016a Software Documentation, 2016.

[32] *Exact Linearization Algorithm*. MathWorks, Simulink R2013a Software Documentation.

[33] N. Markopoulos, Mease, Kenneth D., and Vinh, Nguyen X., “Thrust law effects on the long-period modes of aerospace craft,” presented at the AIAA Atmospheric Flight Mechanics Conference, Boston, MA, 1989.

[34] G. Sachs, “Longitudinal Long-Term Modes in Super- and Hypersonic Flight,” *J. Guid. Control Dyn.*, vol. 28, no. 3, pp. 539–541, 2005.

[35] R. Colgren, S. Keshmiri, and M. Mirmirani, “Nonlinear Ten-Degree-of-Freedom Dynamics Model of a Generic Hypersonic Vehicle,” *J. Aircr.*, vol. 46, no. 3, pp. 800–813, May 2009.

[36] Christian Breitsamter, Tomislav Cvrlje, Boris Laschka, Matthias Heller, and Gottfried Sachs, “Lateral-Directional Coupling and Unsteady Aerodynamic Effects of Hypersonic Vehicles,” *J. Spacecr. Rockets*, vol. 38, no. 2, pp. 159–167, 2001.

[37] William Vaughan, Dale Johnson, and L. Ehernberger, “An overview of the atomospheric modeling for aeronautical and aerospace vehicles simulation applications,” 1999.

[38] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*, 3rd Edition. US: John Wiley & Sons, 2012.

[39] Carsten Scherer, “Mixed H2/H∞ Control,” in *Trends in Control,* Springer London, 1995, pp. 173–216.

[40] Gonzalo A. Garcia-Garreton, “Decentralized Robust Nonlinear Model Predictive Controller for Unmanned Aerial Systems,” Ph.D. Dissertation, University of Kansas, Lawrence, Kan, 2013.

[41] Alexandre Megretski, “6.245 Multivariable Control Systems,” Massachusetts Institute of Technology, MIT OpenCourseWare, Spring 2004.

[42] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, 2nd ed. Hoboken, NJ: John Wiley, 2005.

[43] G. Ray, “Optimal Control,” presented at the National Programme on Technology Enhanced Learning, 2012.

[44] A. D. de Corcuera, A. Pujana-Arrese, J. M. Ezquerra, E. Segurola, and J. Landaluze, “H∞ Based Control for Load Mitigation in Wind Turbines,” *Energies*, vol. 5, no. 12, pp. 938–967, Apr. 2012.

[45] M. G. Ortega and F. R. Rubio, “Systematic design of weighting matrices for the H∞ mixed sensitivity problem,” *J. Process Control*, vol. 14, no. 1, pp. 89–98, Feb. 2004.

[46] V. V. Athani and S. Agarwal, “Design of a robust controller for a supersonic aircraft using H∞ approach,” *Control Eng. Pract.*, vol. 2, no. 6, pp. 1051–1061, Dec. 1994.

[47] J.-S. Yee, J. L. Wang, and N. Sundararajan, “Robust sampled-data H∞-flight-controller design for high α stability-axis roll maneuver,” *Control Eng. Pract.*, vol. 8, no. 7, pp. 735–747, Jul. 2000.

[48] P. V. N. Gade and D. J. Inman, “Two-Dimensional Active Wing/Store Flutter Suppression Using H-Infinity Theory,” *J. Guid. Control Dyn.*, vol. 20, no. 5, pp. 949–955, Sep. 1997.

[49] G. Garcia, S. Keshmiri, and R. Colgren, “Advanced H-Infinity Trainer Autopilot,” 2010.

[50] C. Tournes and C. D. Johnson, “Application of linear subspace stabilization and linear adaptive techniques to aircraft flight control problems. II. The outer loop,” 1998, pp. 151–155.

[51] Gonzalo A. Garcia and Shahriar Keshmiri, “Nonlinear Model Predictive Controller for Navigation, Guidance and Control of a Fixed-Wing UAV,” presented at the AIAA Guidance, Navigation, and Control Conference, Portland, Oregon, 2011.

[52] T. J. Stastny, G. Garcia, and S. Keshmiri, “Robust Three-Dimensional Collision Avoidance for Fixed-Wing Unmanned Aerial Systems,” 2015.

[53] A. Marcos, “A gain scheduled H-infinity controller for a re-entry benchmark,” 2010.

[54] G. A. Garcia, S. Keshmiri, and R. D. Colgren, “H-Infinity gain scheduling design for the meridian UAS for a broader range of operation and for fault tolerant applications,” 2011, pp. 1174–1180.

[55] Brian Bradie, *A Friendly Introduction to Numerical Analysis*. Upper Sandy River, New Jersey: Pearson Prentice Hall, 2006.

[56] *n-D Lookup Table*. MathWorks, Simulink R2013a Software Documentation, 2016.

[57] *interpn*. MathWorks, Simulink R2013a Software Documentation, 2016.

[58] C. Habermann and F. Kindermann, “Multidimensional Spline Interpolation: Theory and Applications,” *Comput. Econ.*, vol. 30, no. 2, pp. 153–169, Aug. 2007.

[59] *PID Controller, Discrete PID Controller*. MathWorks, Simulink R2013a Software Documentation, 2016.

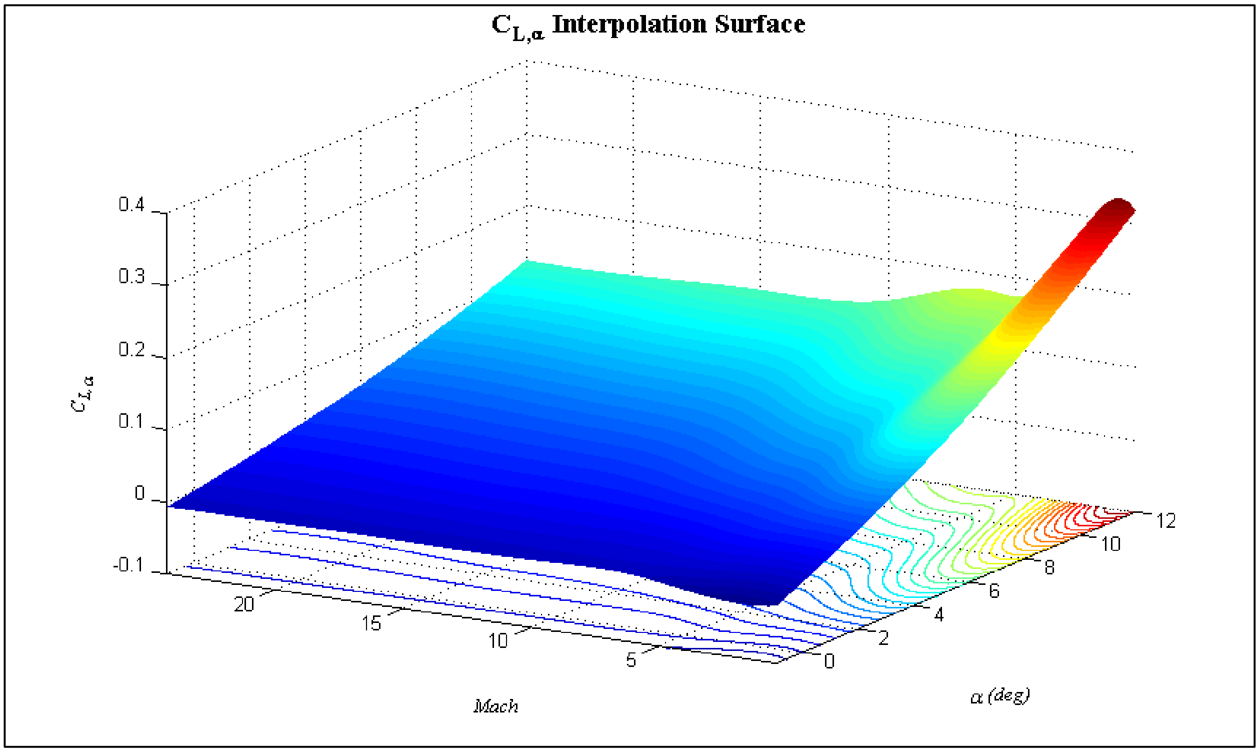
[60] Katsuhiko Ogata, *Modern Control Engineering*, 3rd ed. Upper Sandy River, New Jersey: Prentice-Hall, Inc.

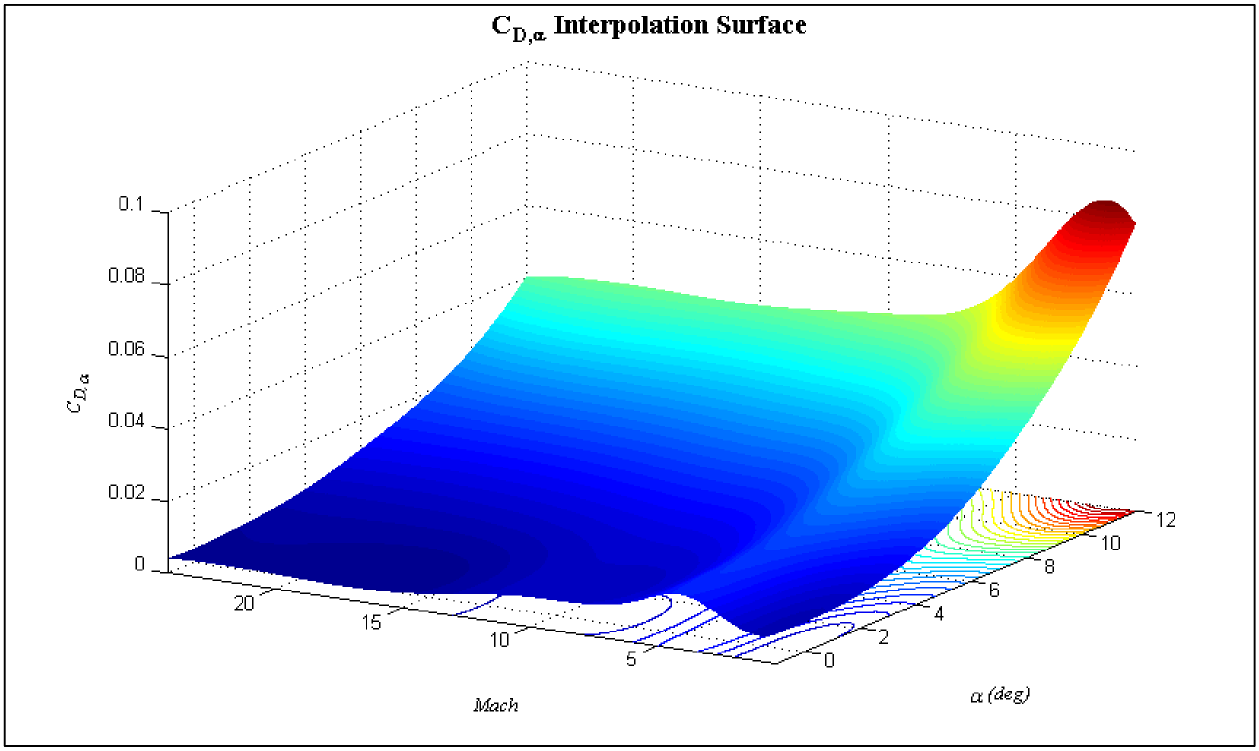
[61] M. F. Miranda, R. H. C. Takahashi, and F. G. Jota, “On H∞ Controllers with integral action: An experimental evaluation,” *Proc. Inst. Mech. Eng. Part J. Syst. Control Eng.*, vol. 224, no. 1, pp. 21–29, Feb. 2010.

[62] L. Huo, G. Song, H. Li, and K. Grigoriadis, “H∞ robust control design of active structural vibration suppression using an active mass damper,” *Smart Mater. Struct.*, vol. 17, no. 1, p. 15021, Feb. 2008.

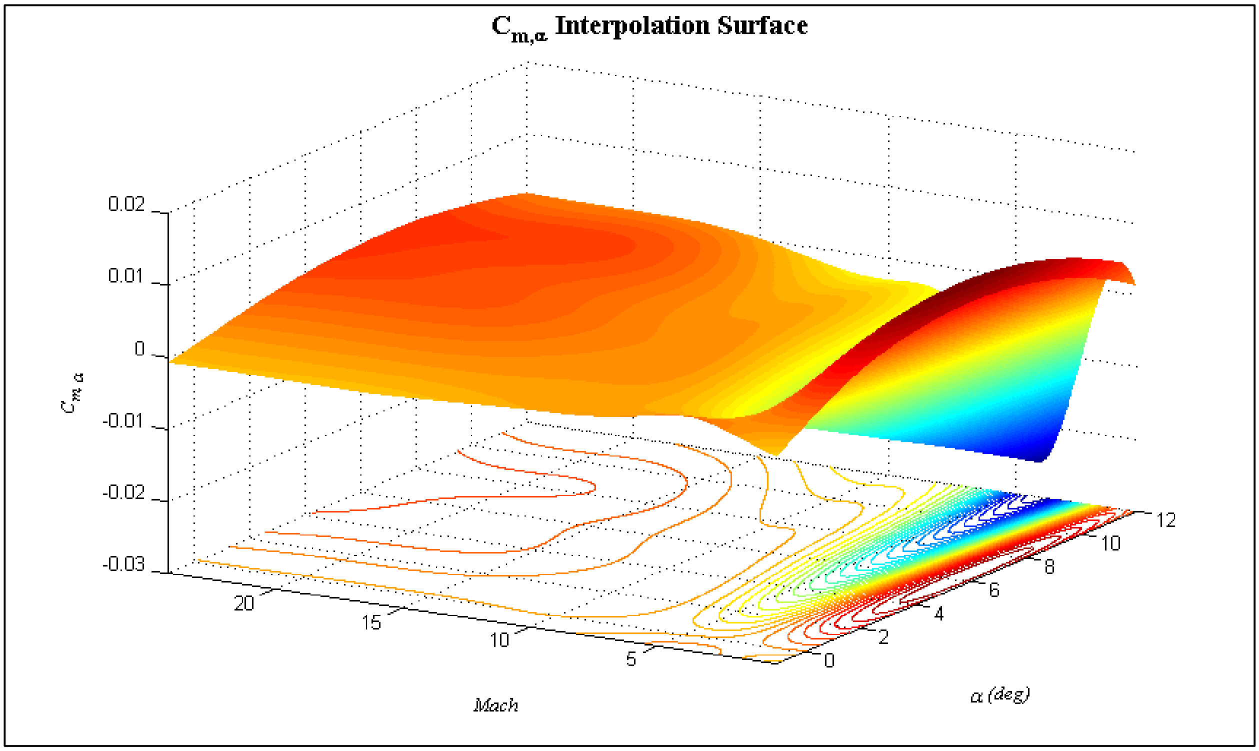
Appendix 1 Selected Aerodynamic Coefficients

Longitudinal Coefficients

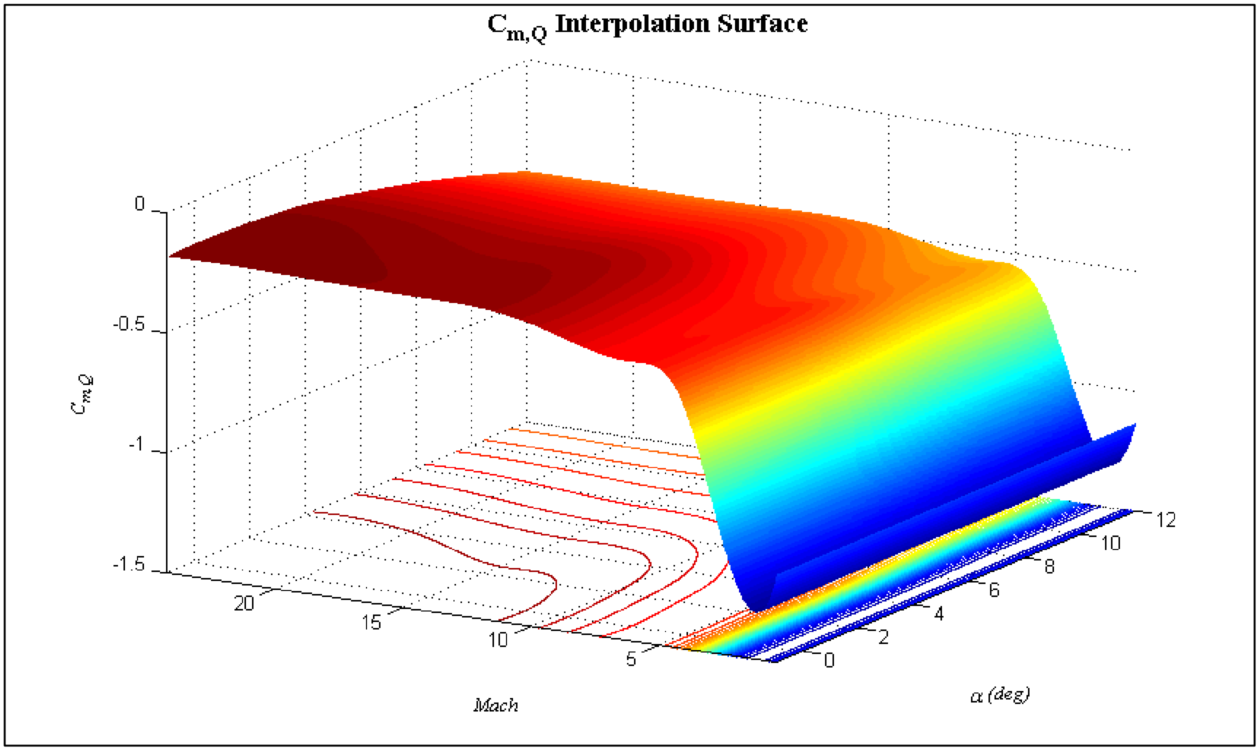




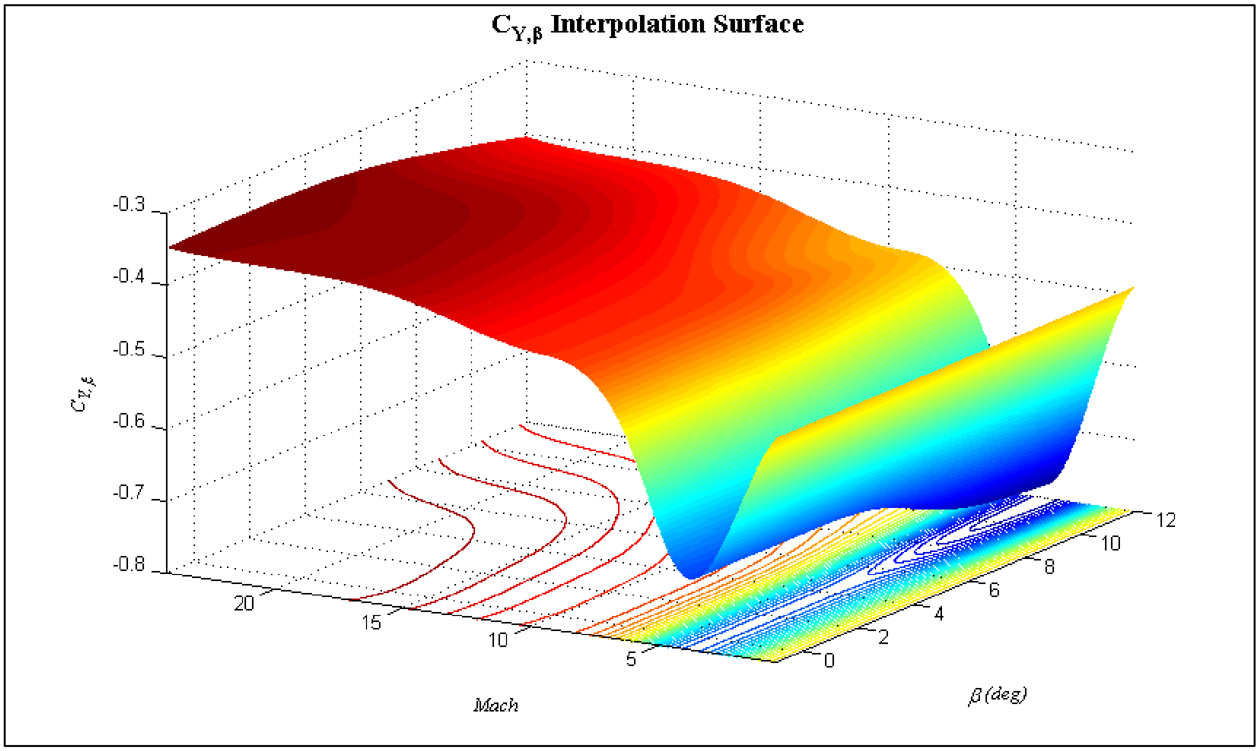
$C_{m,\alpha}$ **Interpolation Surface**

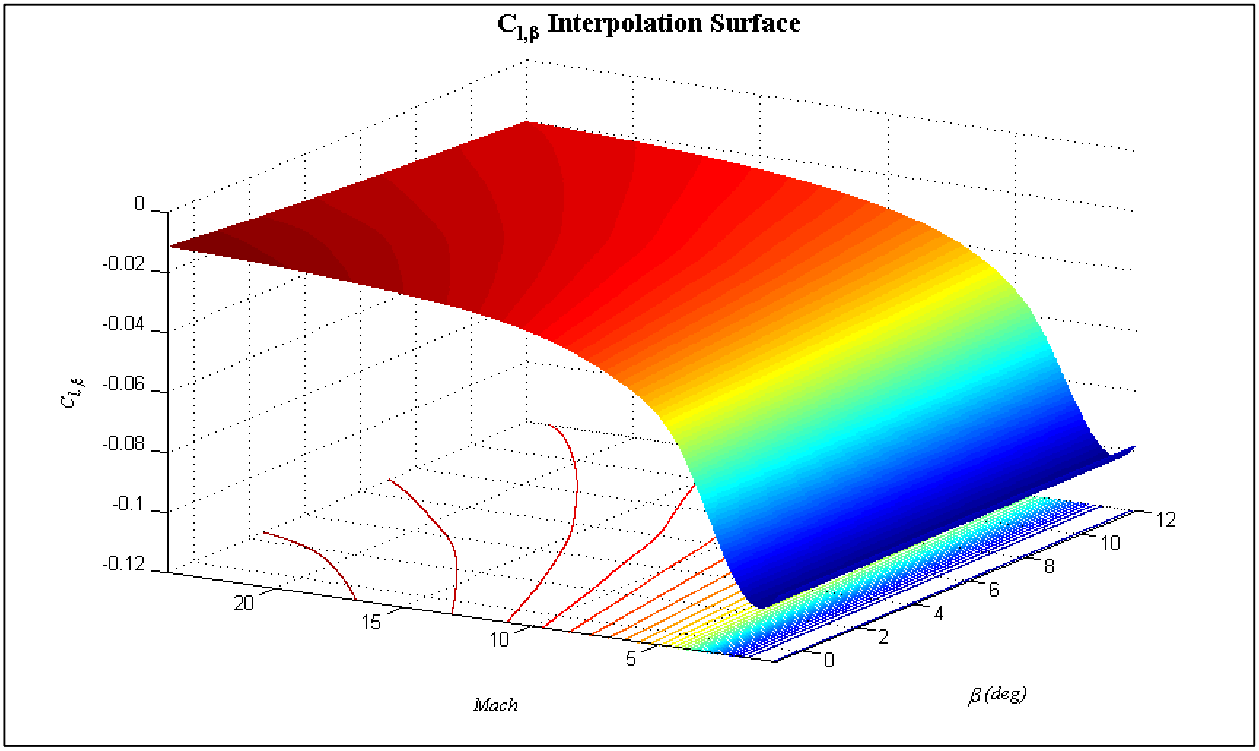


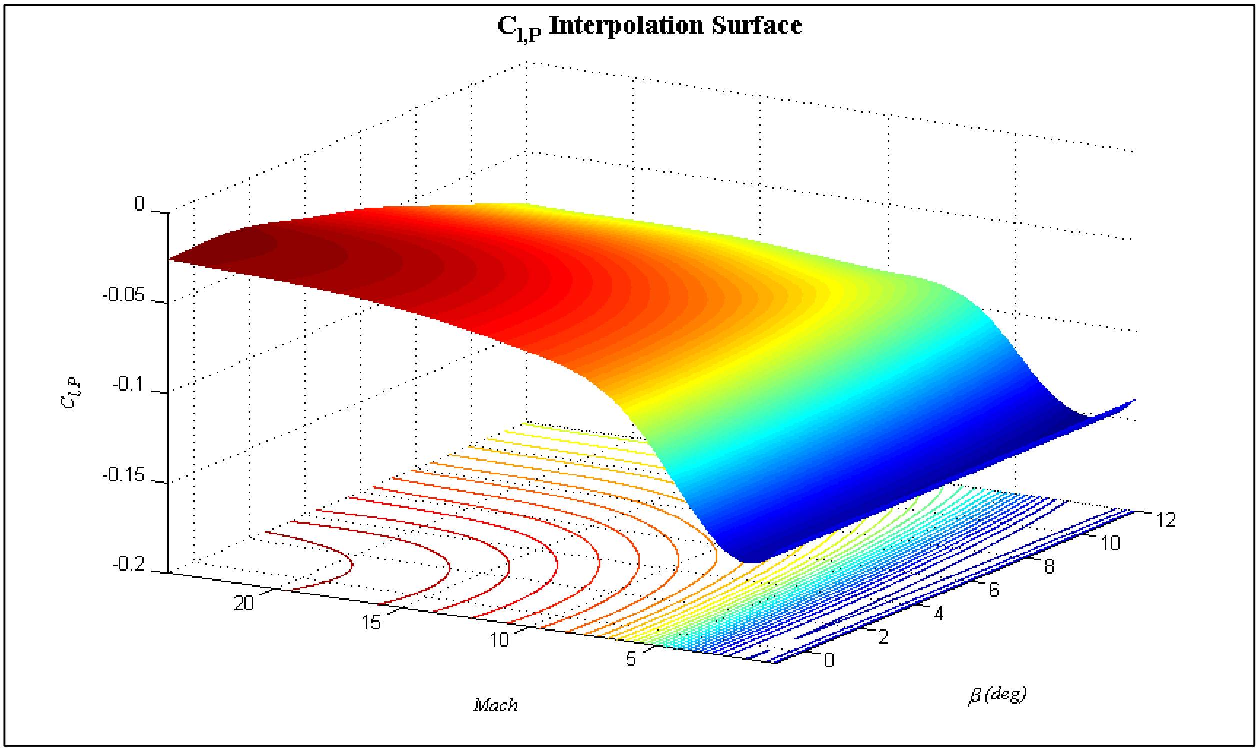
$C_{m,Q}$ **Interpolation Surface**



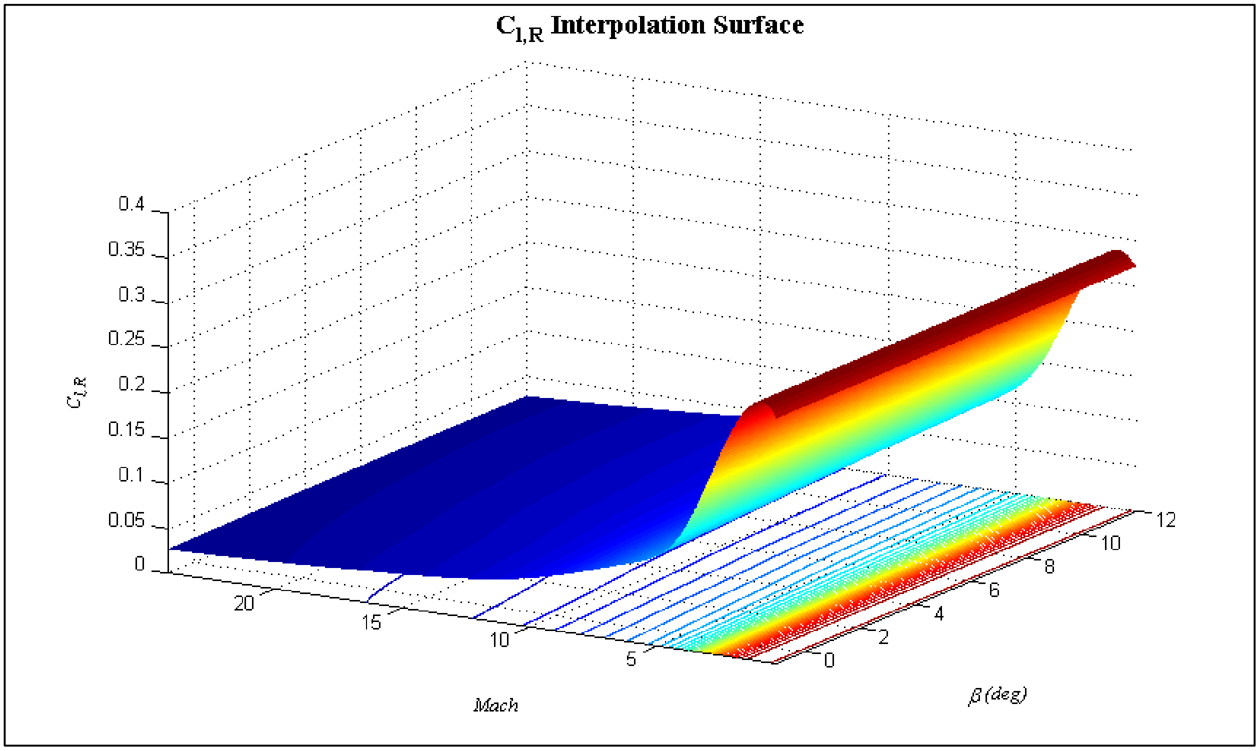
Lateral-Directional Coefficients



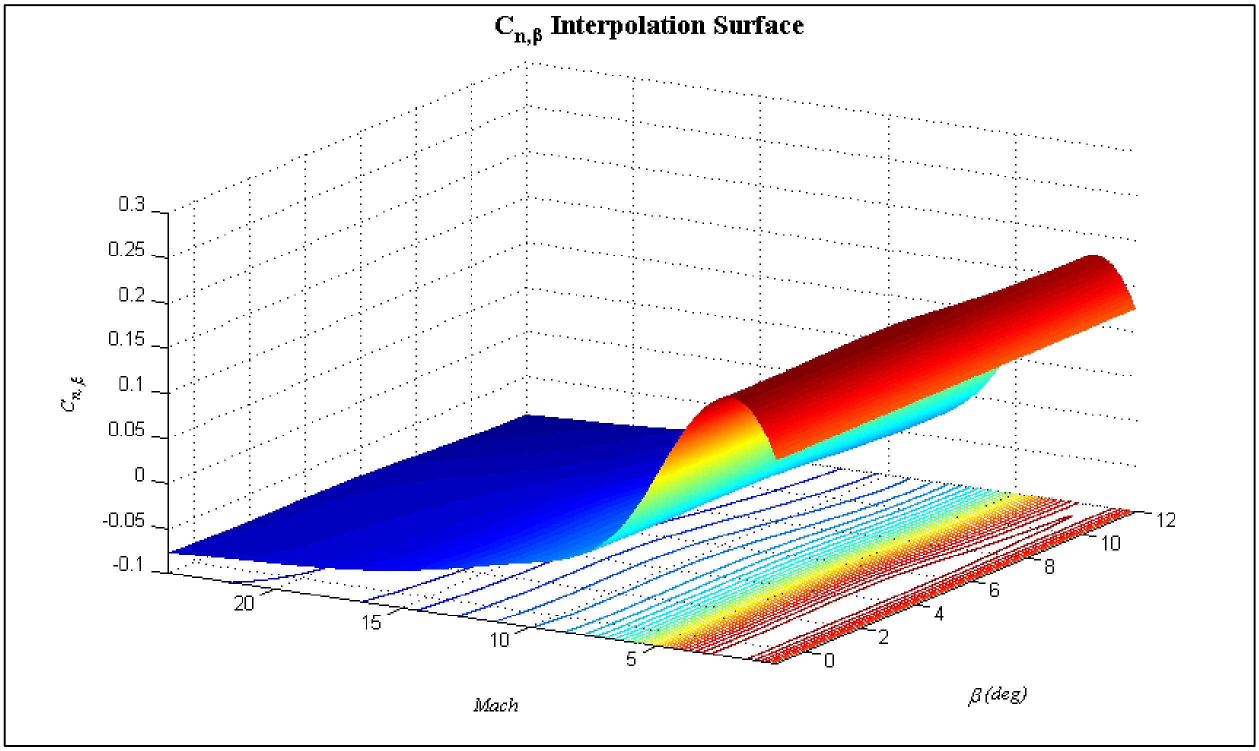


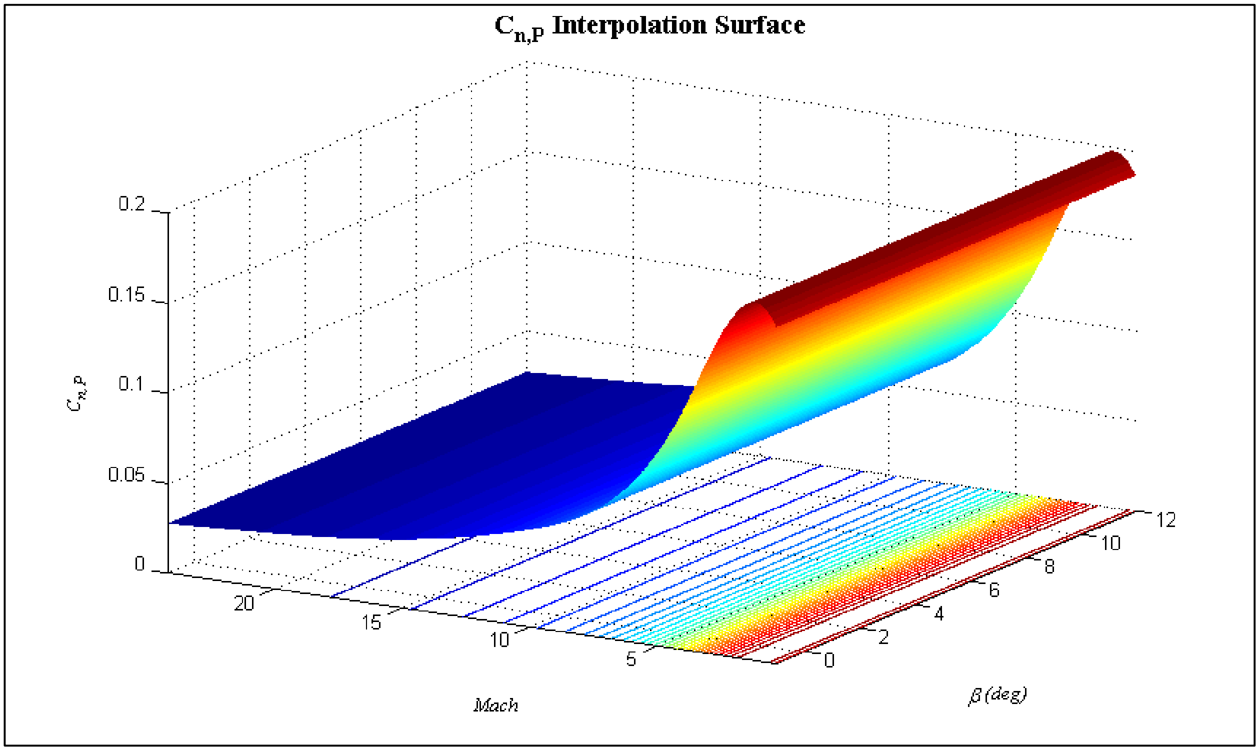


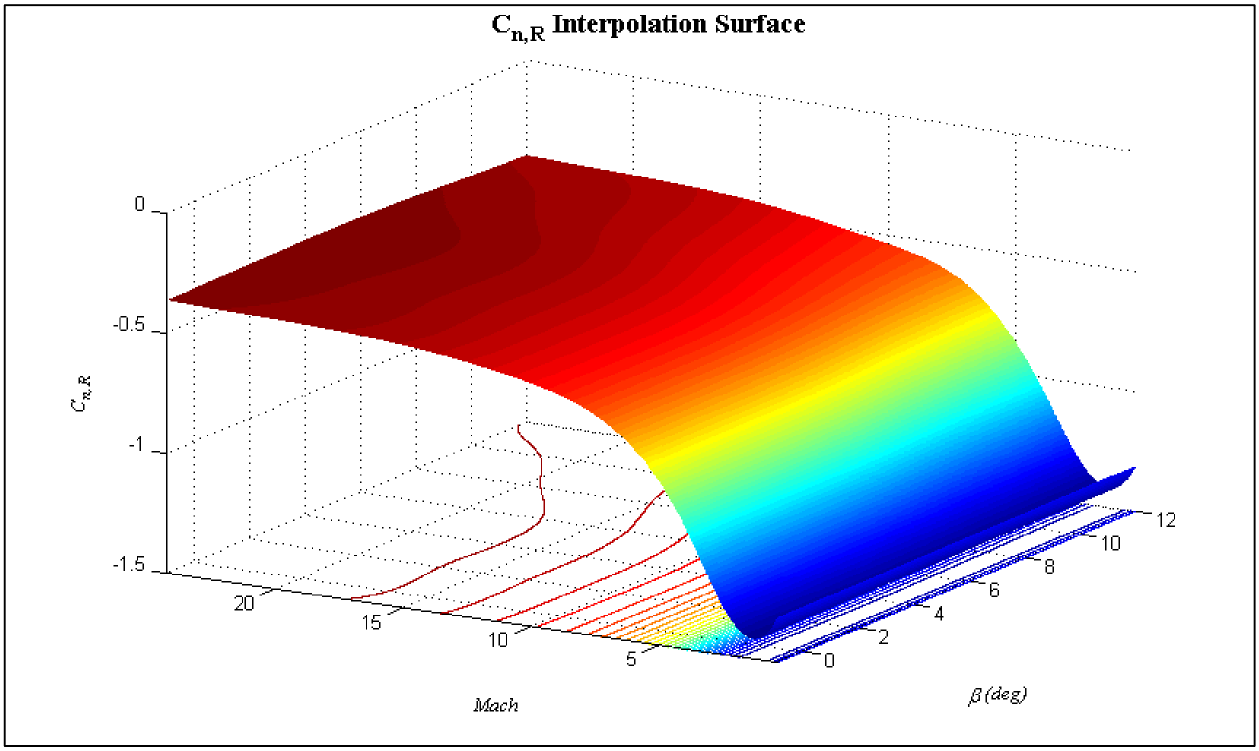
$C_{l,P}$ Interpolation Surface



$C_{l,R}$ Interpolation Surface

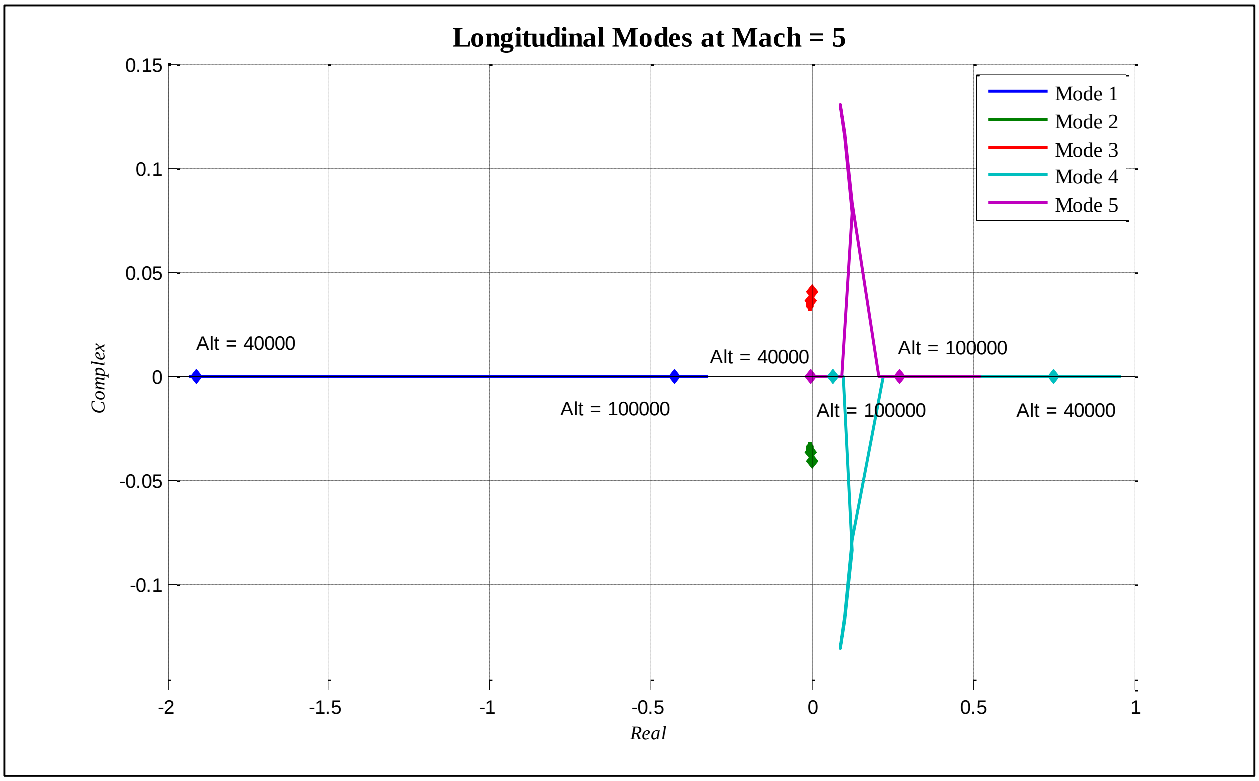

$C_{n,\beta}$ Interpolation Surface
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
$C_{n,\beta}$
20
15
10
5
Mach
0
2
4
6
8
10
12
β (deg)

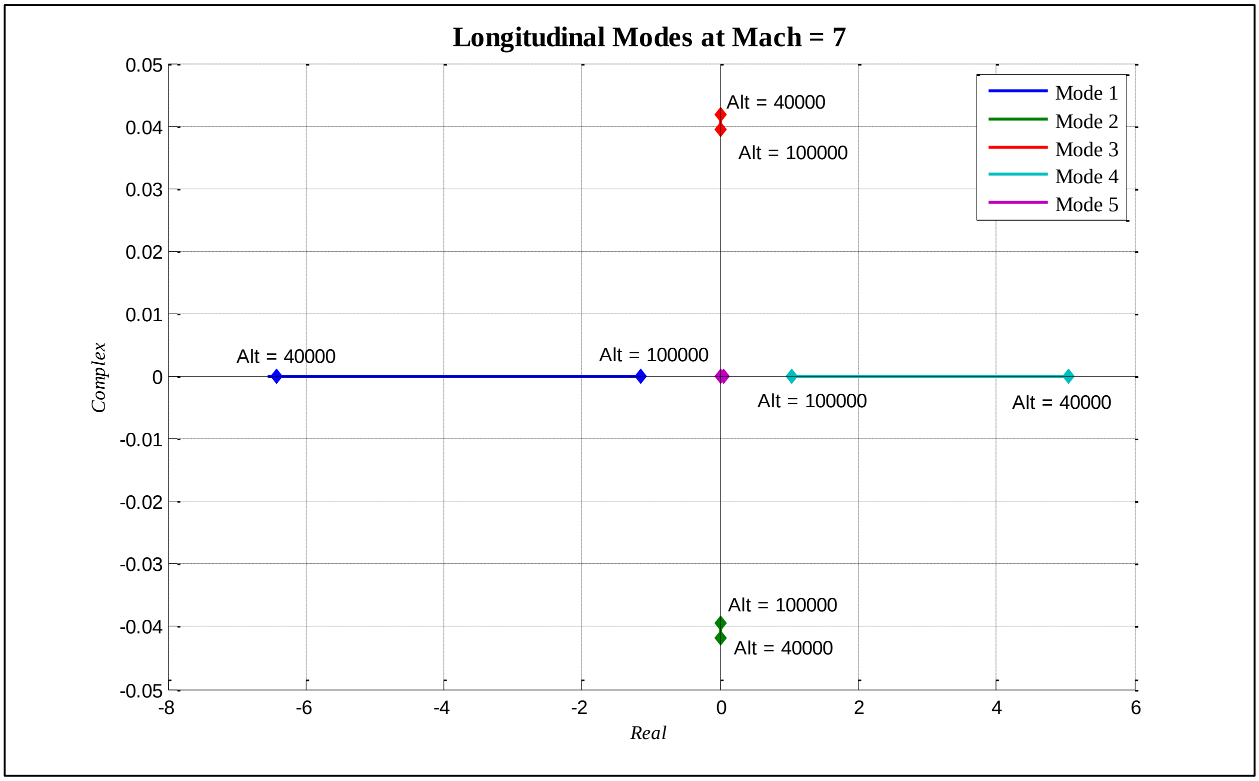

$C_{n,P}$ Interpolation Surface
0.2
0.15
0.1
0.05
0
$C_{n,p}$
20
15
10
5
Mach
0
2
4
6
8
10
12
β (deg)

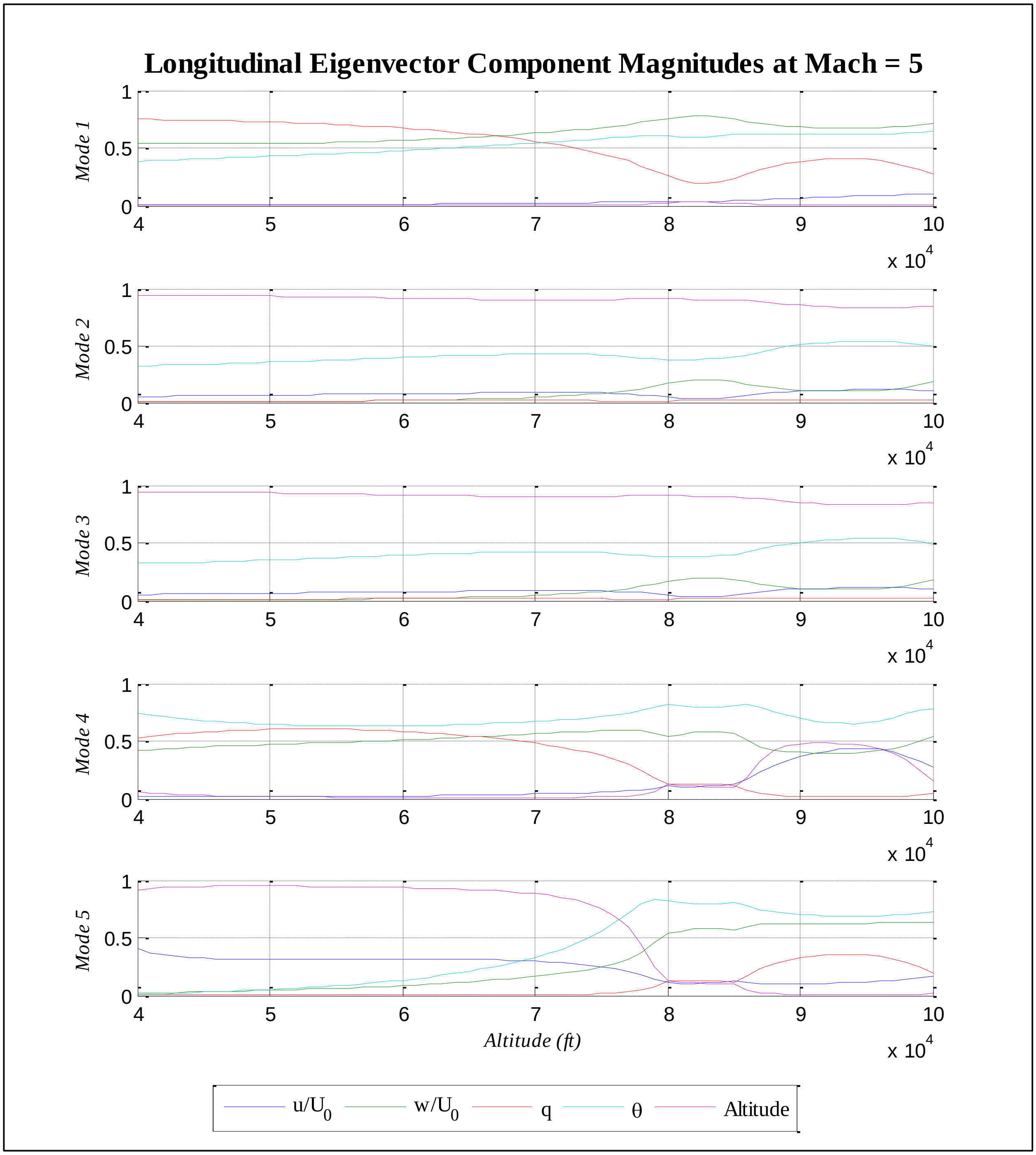

$C_{n,R}$ Interpolation Surface
0
-0.5
-1
-1.5
$C_{n,R}$
20
15
10
5
Mach
0
2
4
6
8
10
12
β (deg)

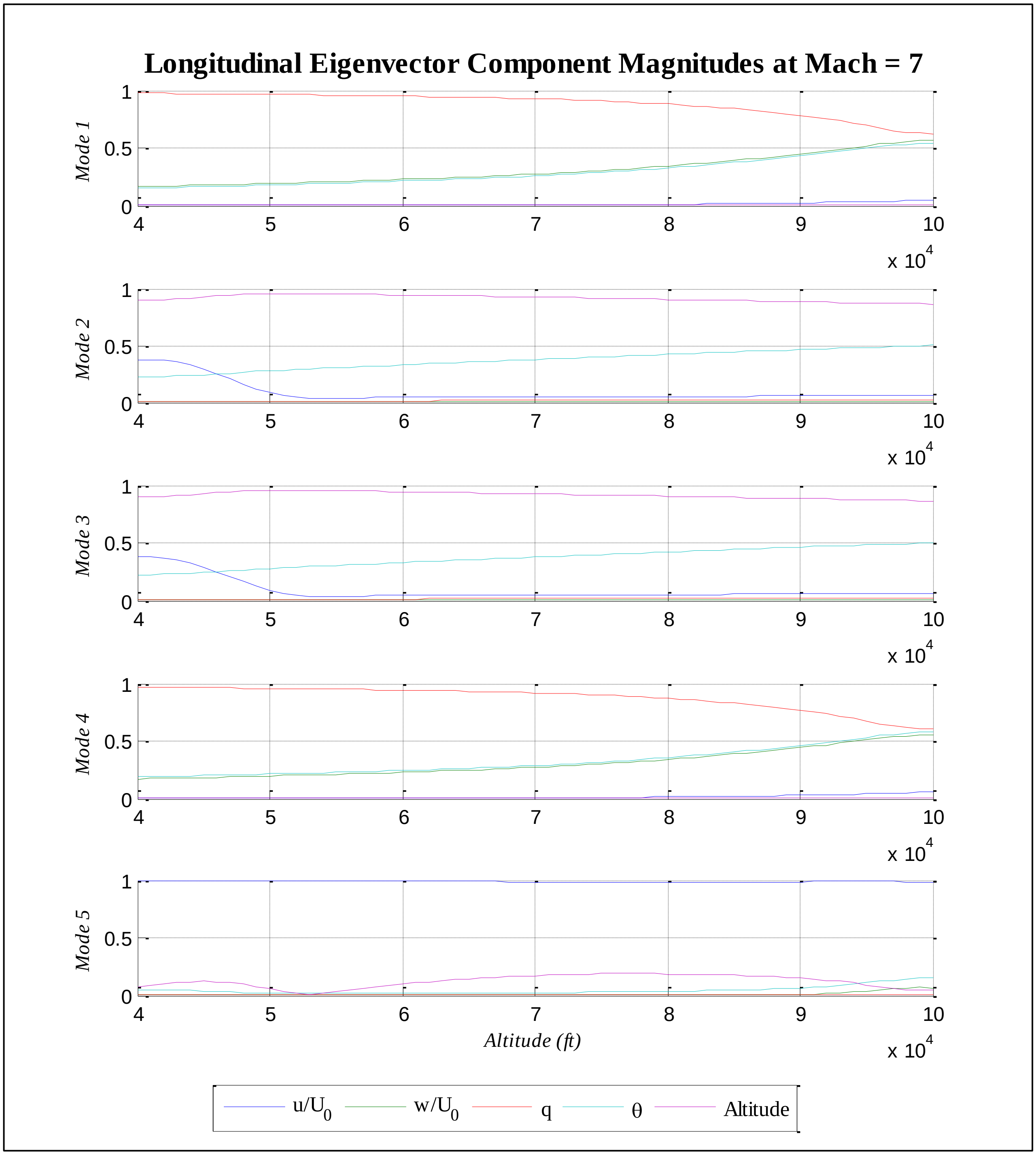
Appendix 2 Additional Trim Dependent Mode Analysis

Longitudinal Modes Changing with Altitude

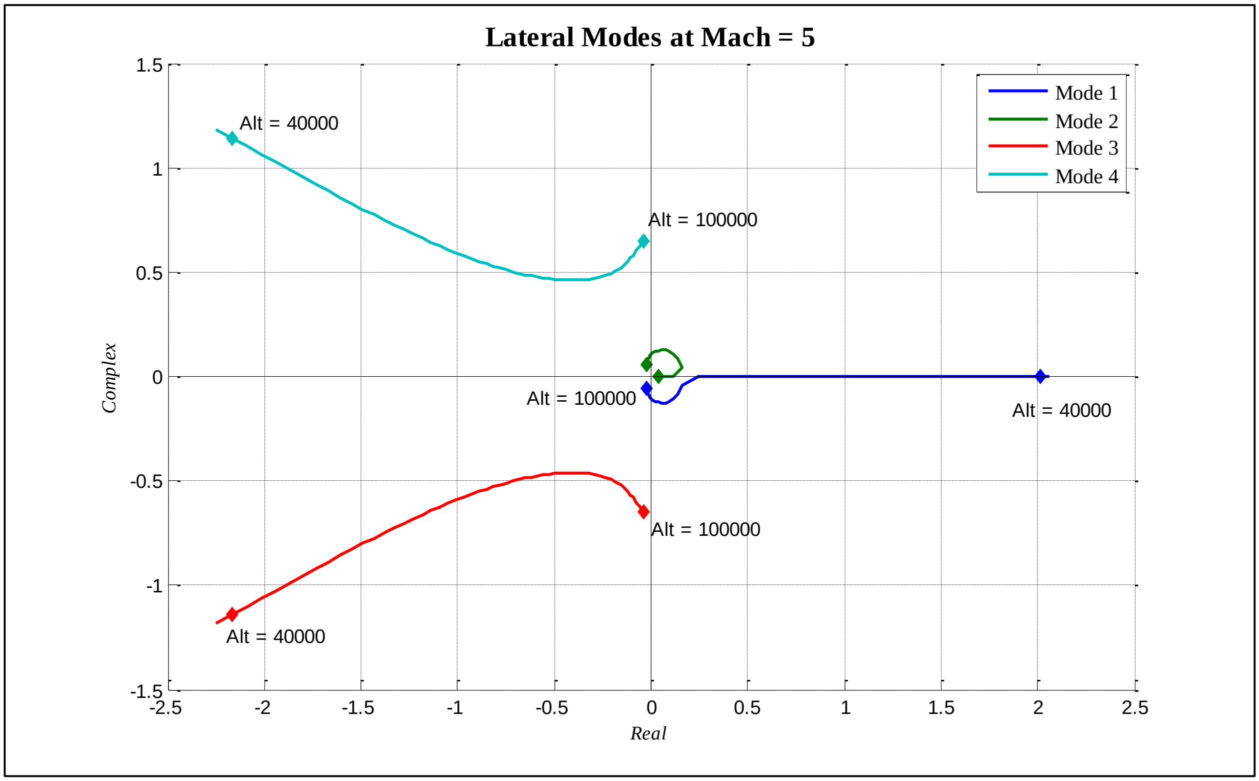


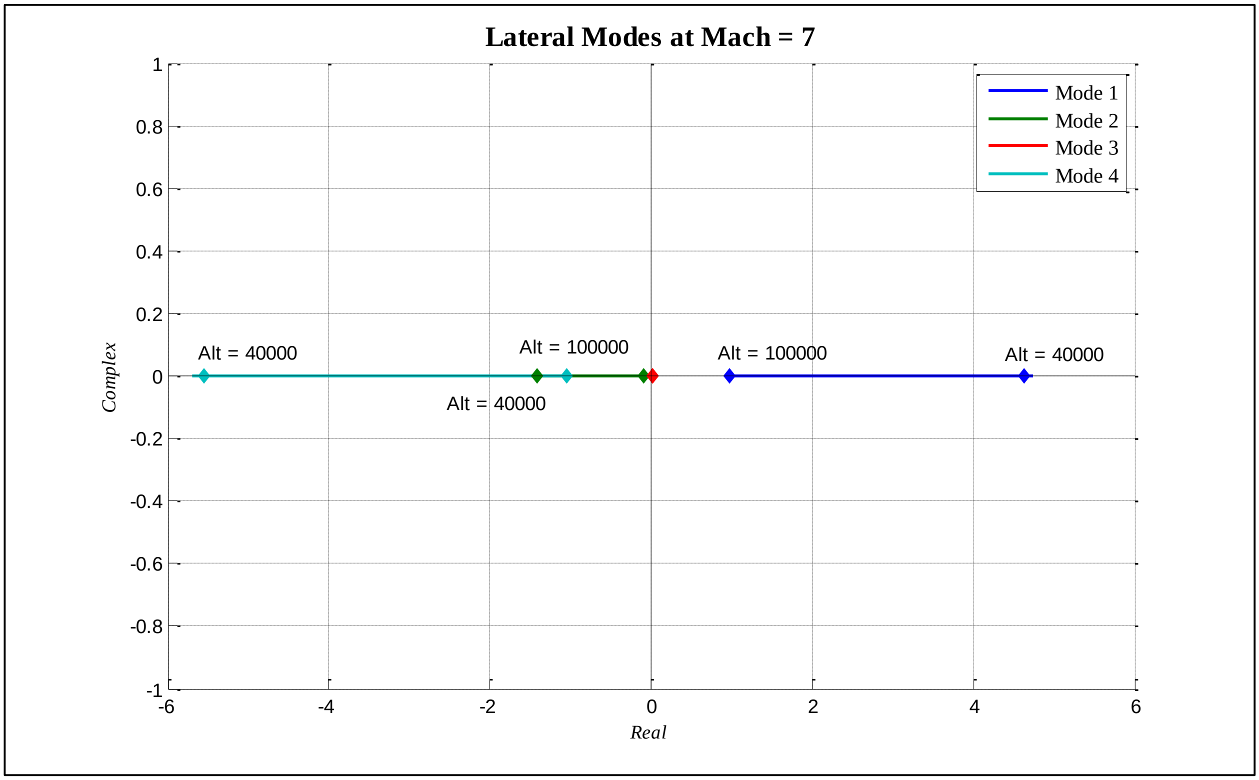


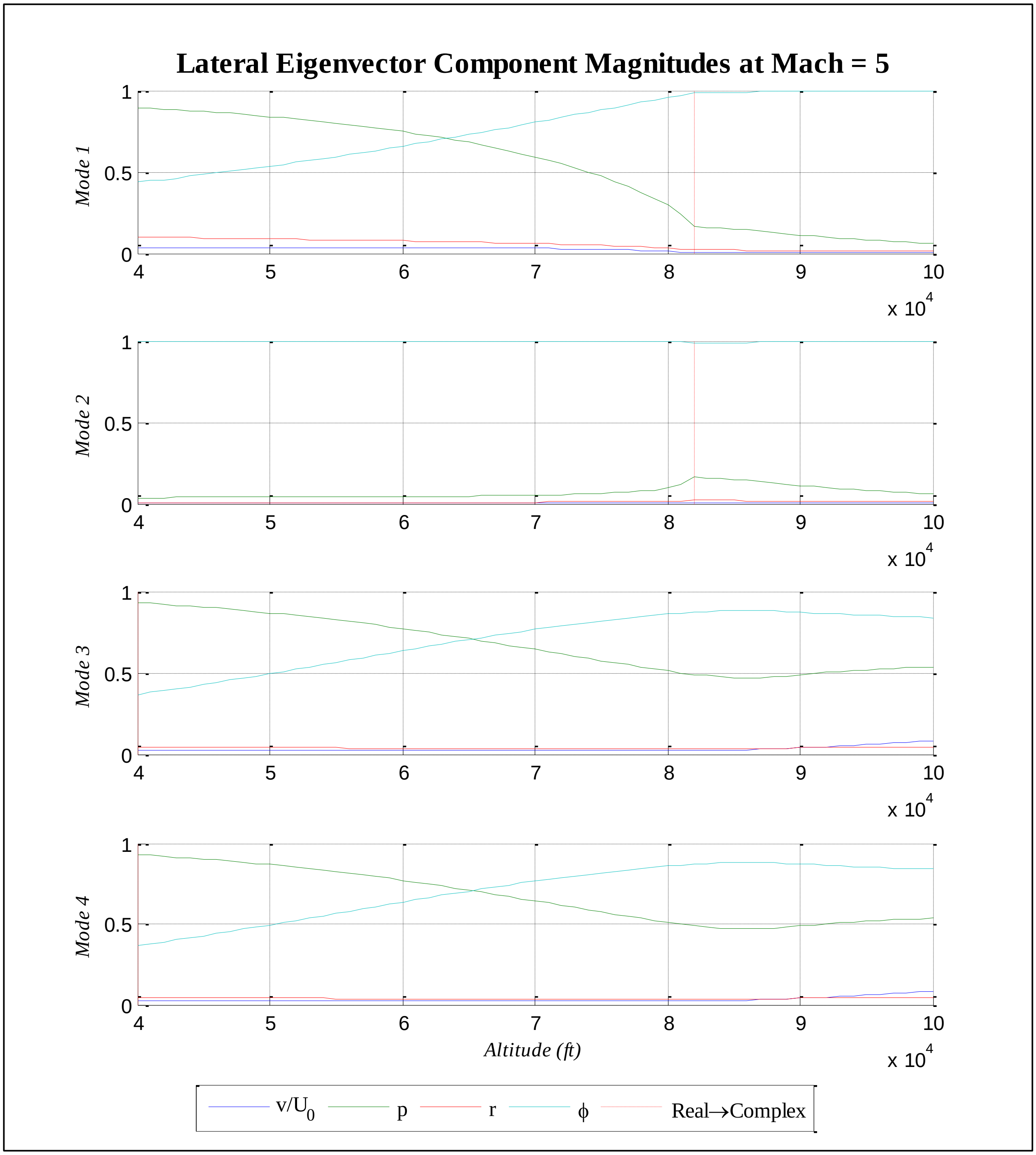

Longitudinal Eigenvector Component Magnitudes at Mach = 5
Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
0
0.5
1
4
5
6
7
8
9
10
x 10^4
Altitude (ft)
u/U0
w/U0
q
θ
Altitude

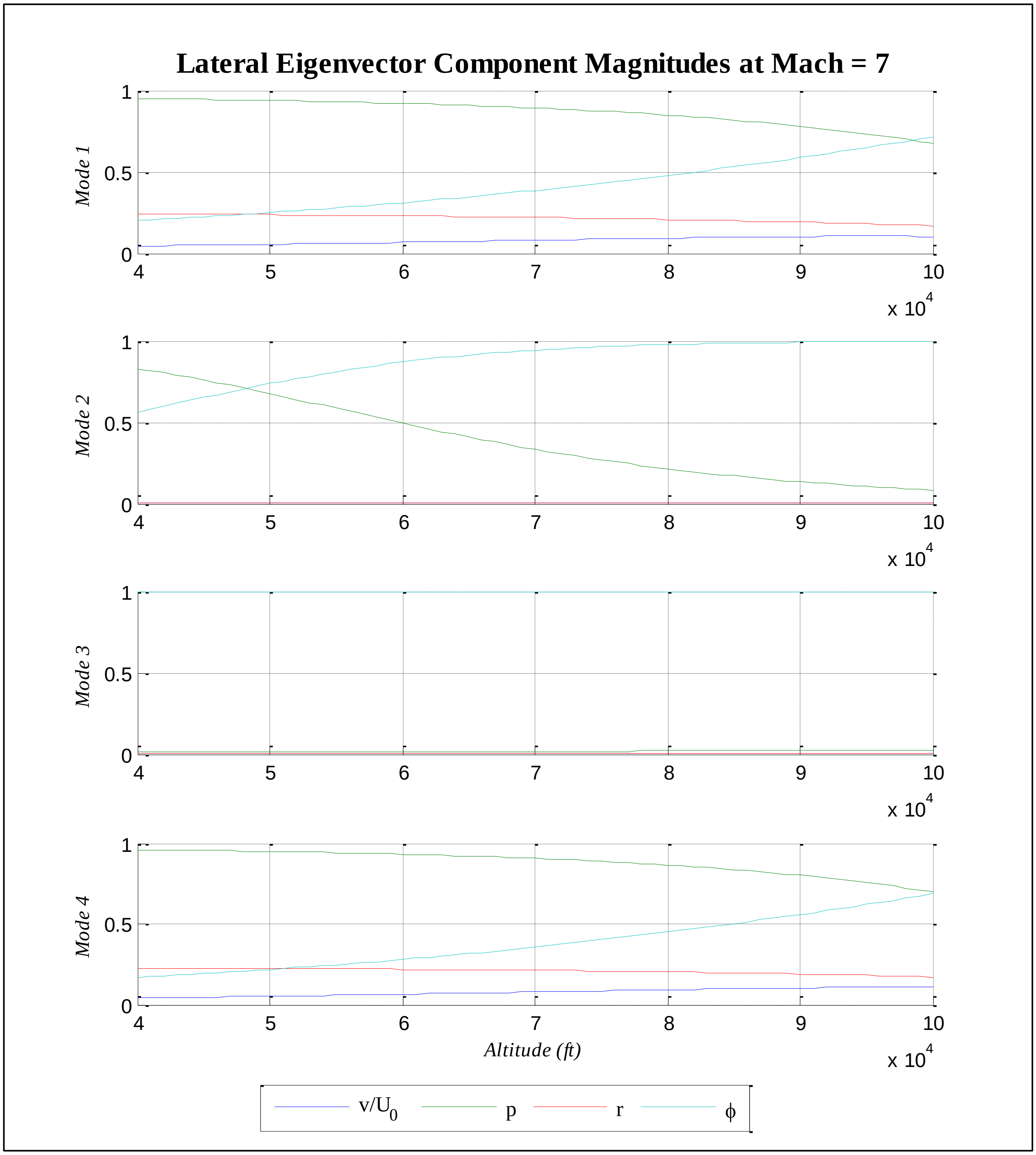

Longitudinal Eigenvector Component Magnitudes at Mach = 7
Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
1
0.5
0
4
5
6
7
8
9
10
x 10^4
Altitude (ft)
u/U0
w/U0
q
θ
Altitude

Lateral-Directional Modes Changing with Altitude



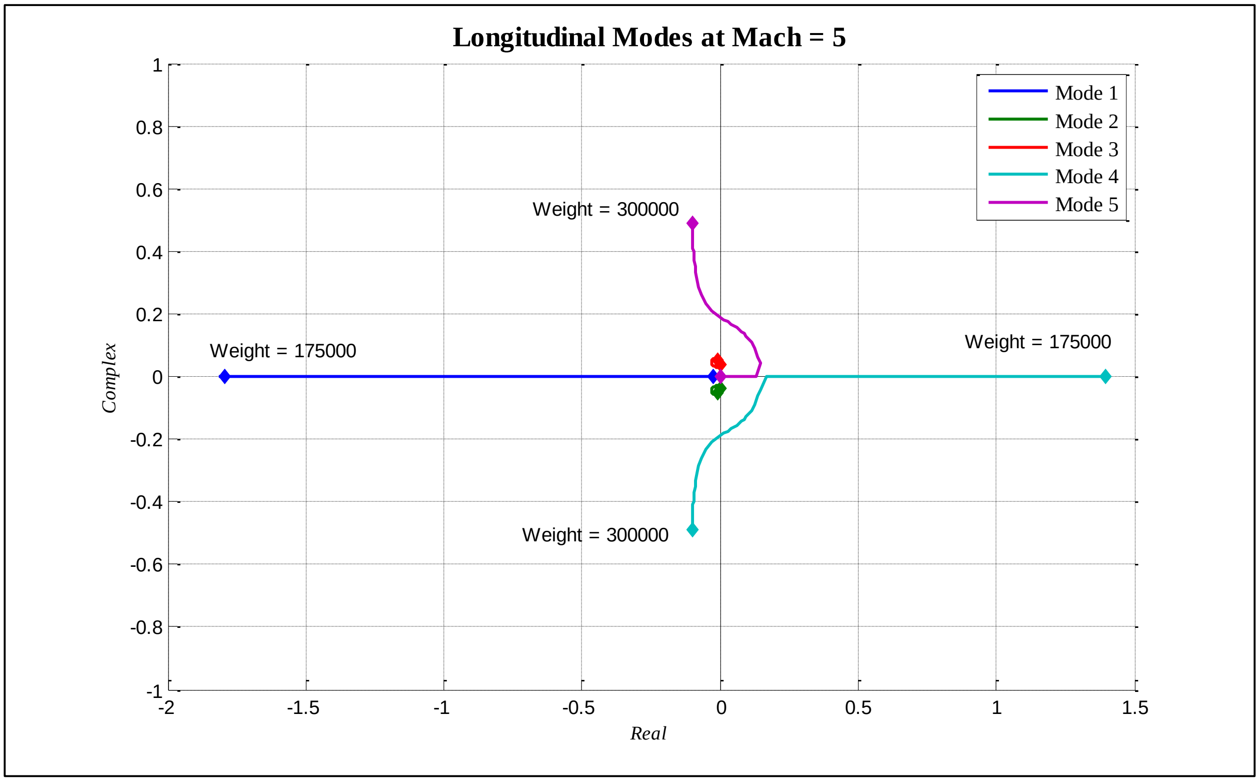


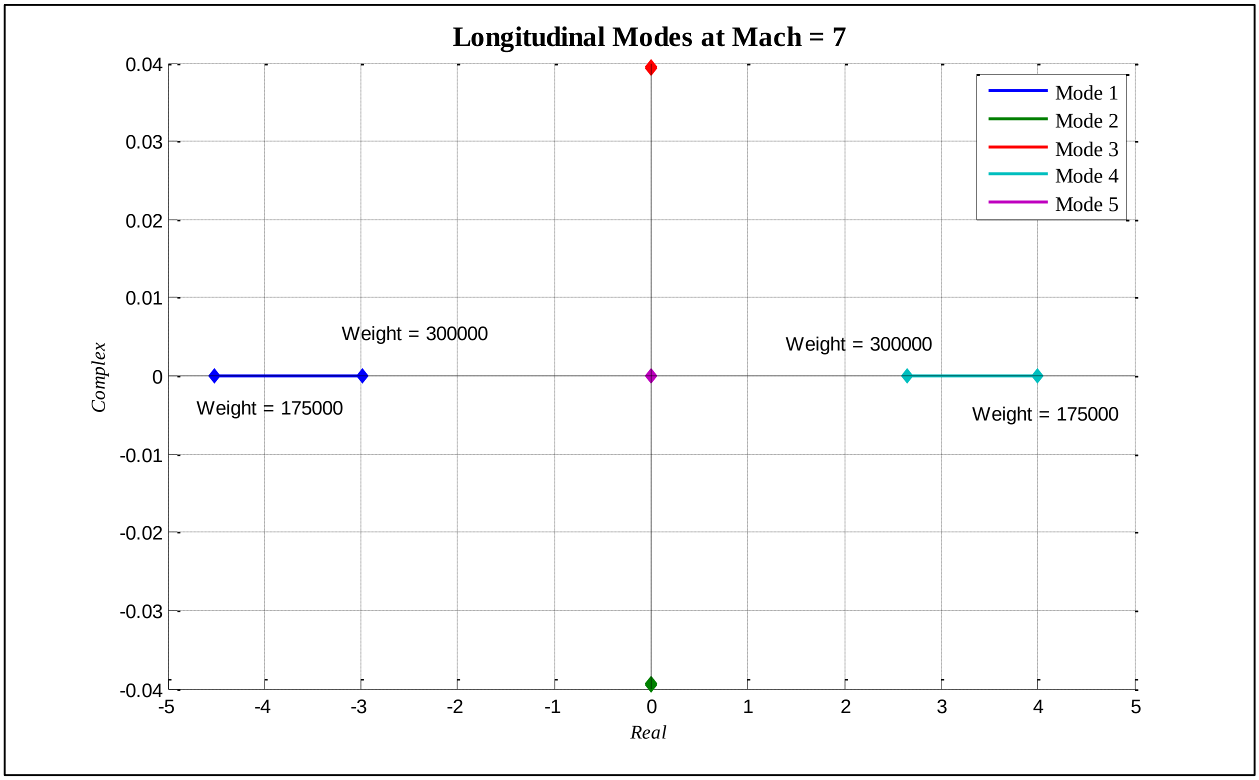

Lateral Eigenvector Component Magnitudes at Mach = 5
Mode 1
Mode 2
Mode 3
Mode 4
0
0.5
1
4
5
6
7
8
9
10
x 10^4
Altitude (ft)
v/U0
p
r
ϕ
Real→Complex

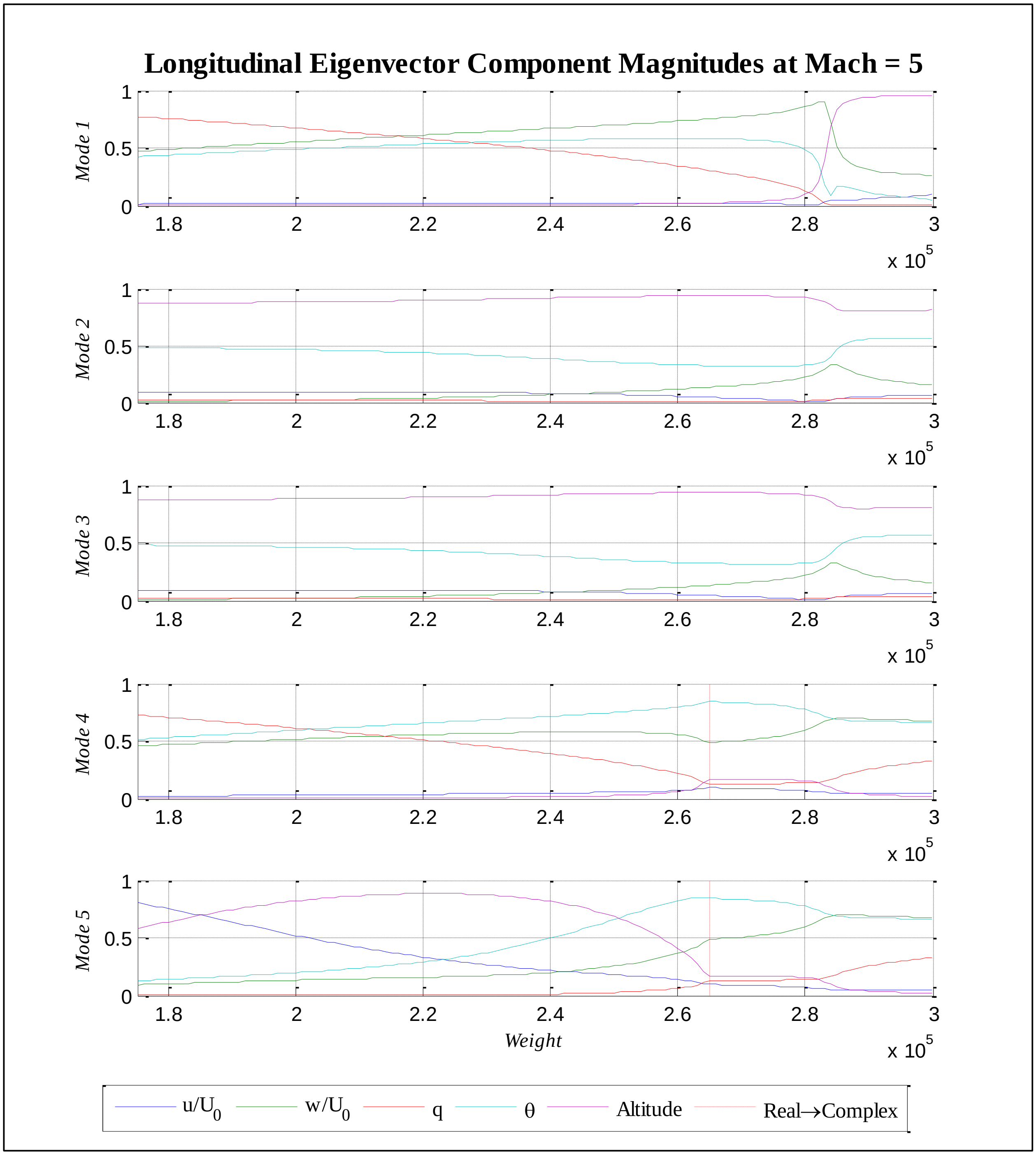


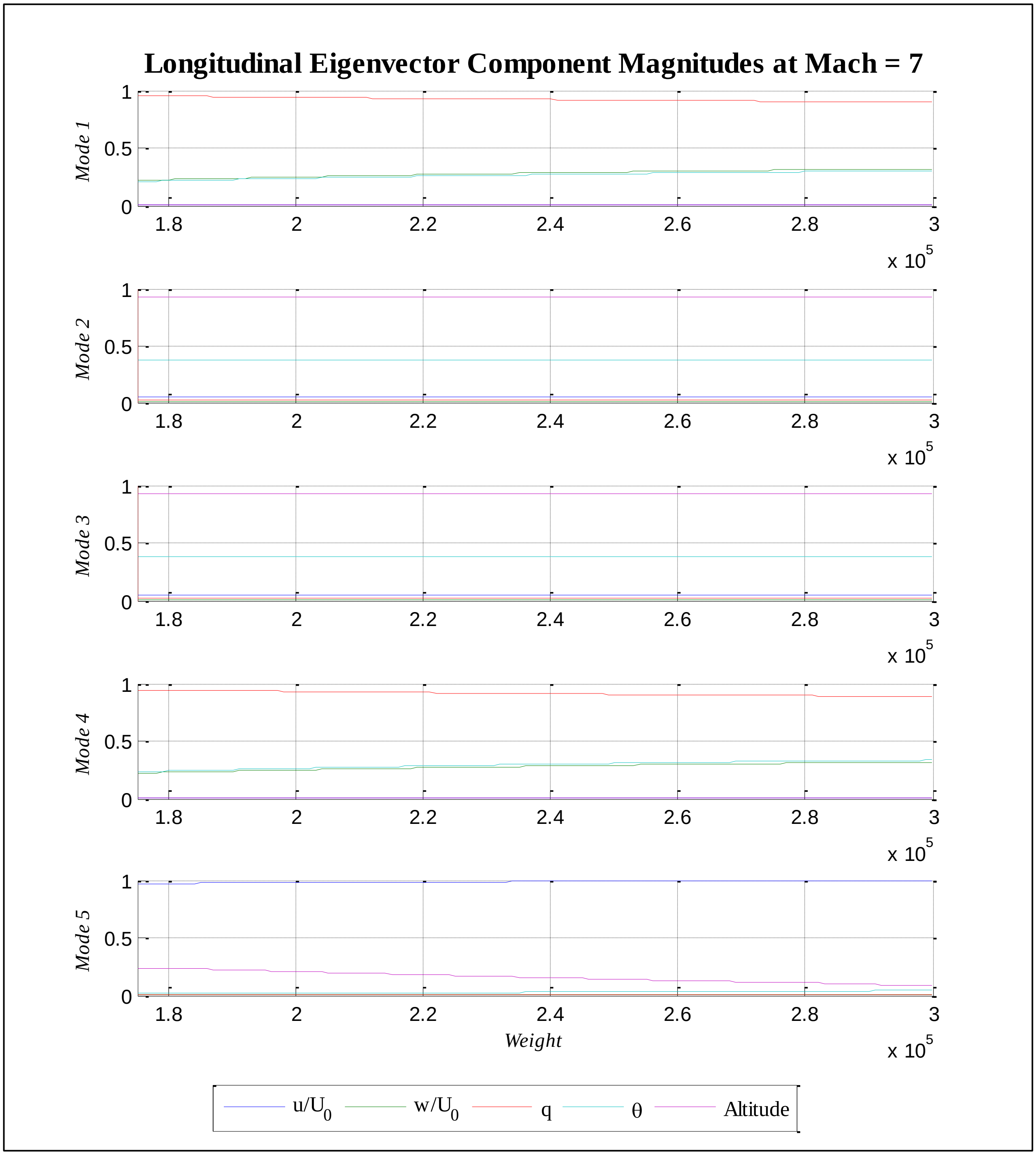
Lateral Eigenvector Component Magnitudes at Mach = 7

Longitudinal Modes Changing with Weight

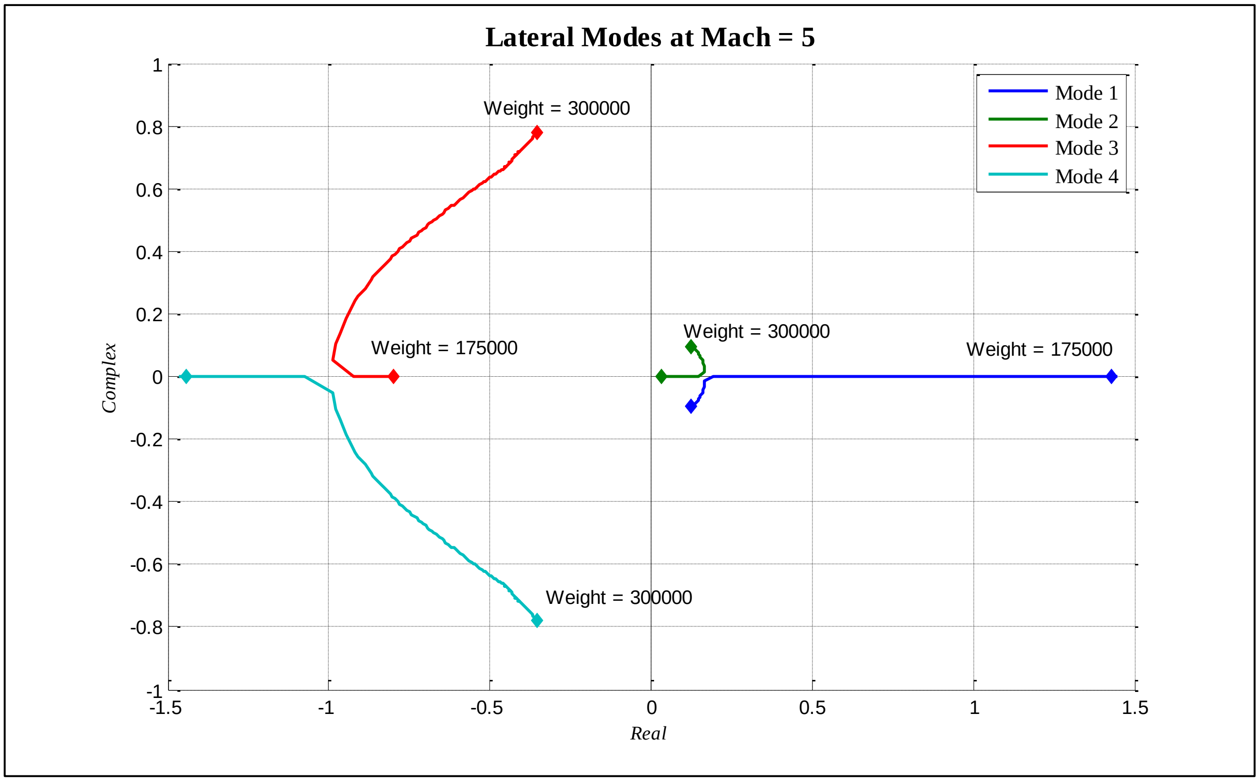


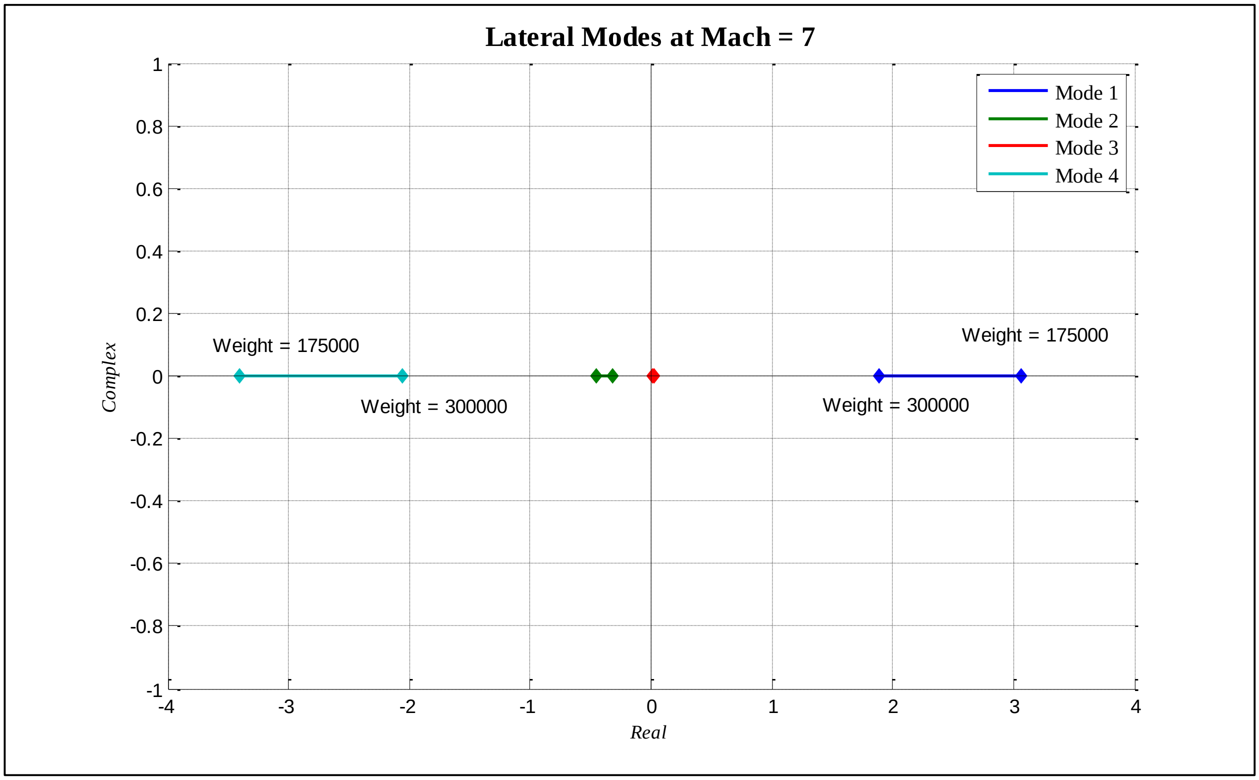


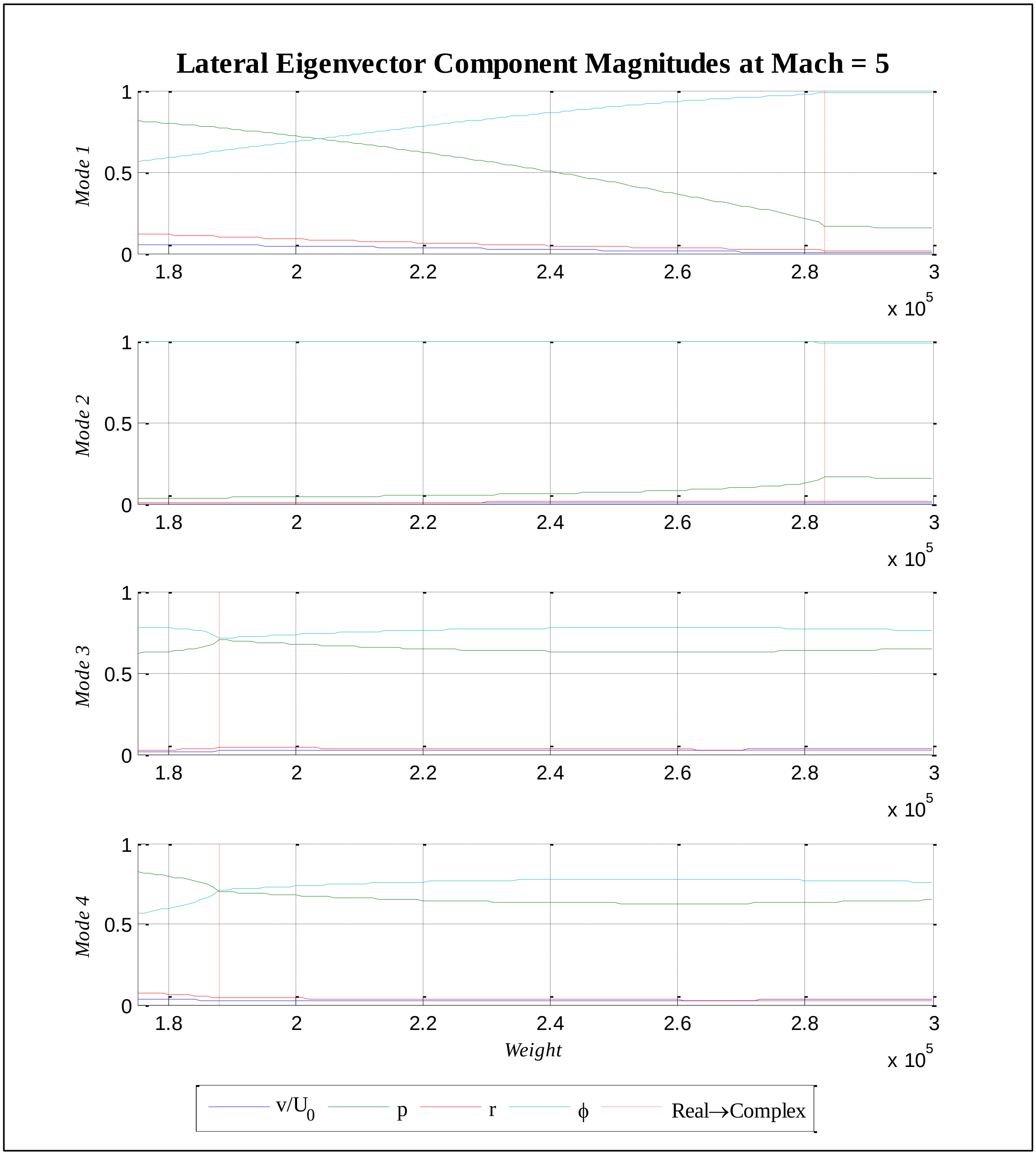

Longitudinal Eigenvector Component Magnitudes at Mach = 5
Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
0
0.5
1
1.8
2
2.2
2.4
2.6
2.8
3
x 10^5
Weight
u/U$_0$
w/U$_0$
q
θ
Altitude
Real→Complex

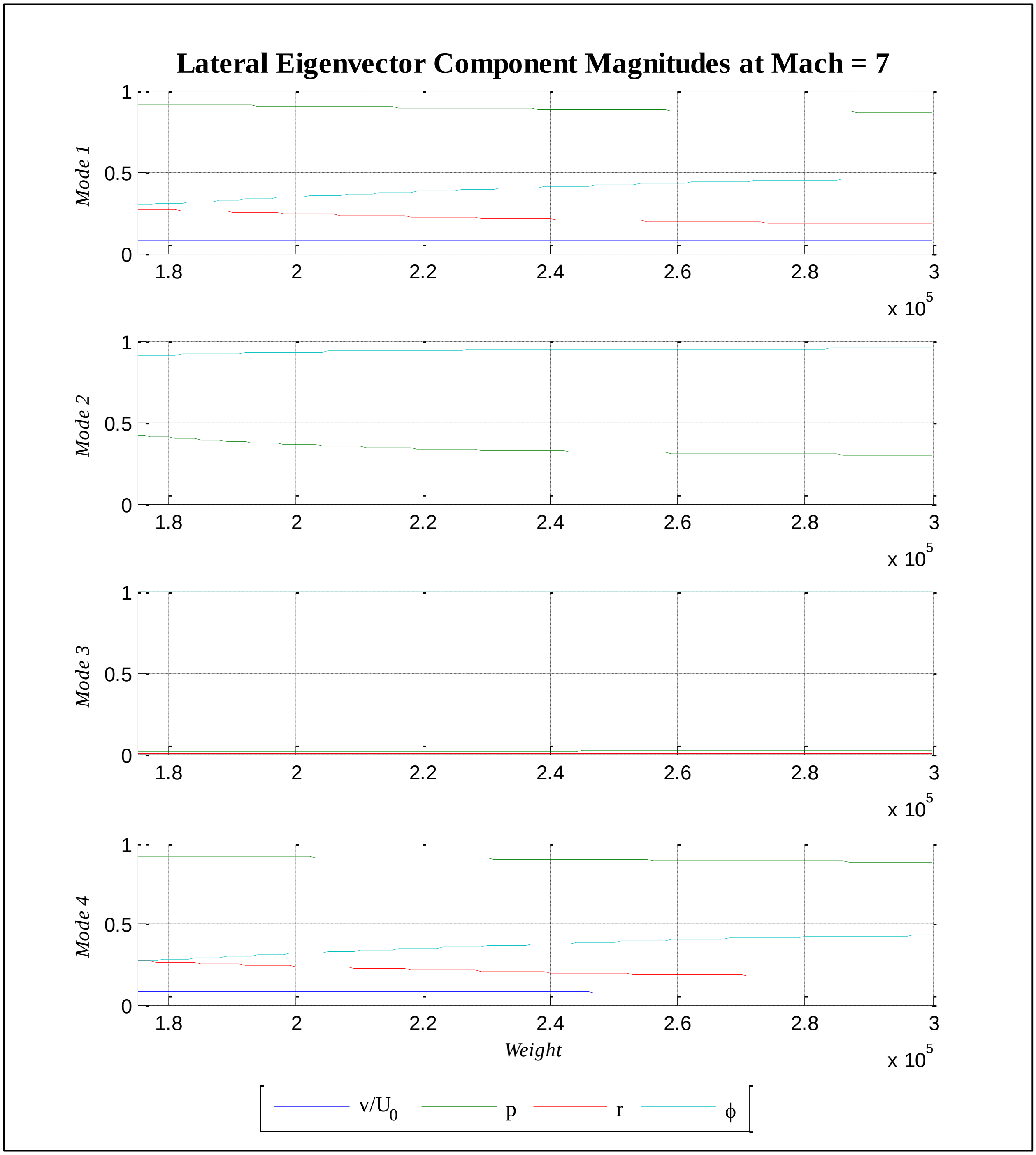

Longitudinal Eigenvector Component Magnitudes at Mach = 7
Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
1
0.5
0
1.8
2
2.2
2.4
2.6
2.8
3
x 10^5
Weight
u/U0
w/U0
q
θ
Altitude

Lateral-Directional Modes Changing with Weight

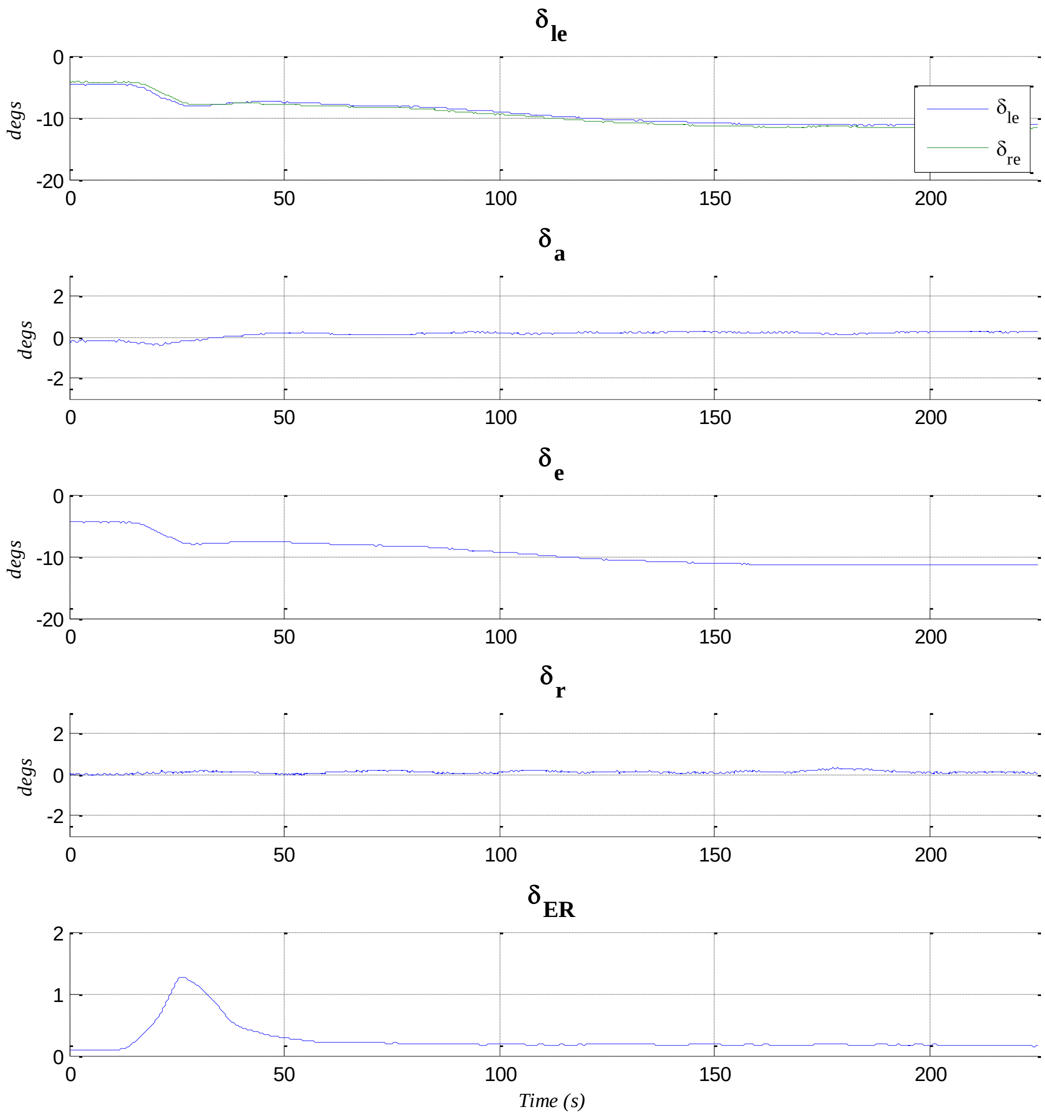


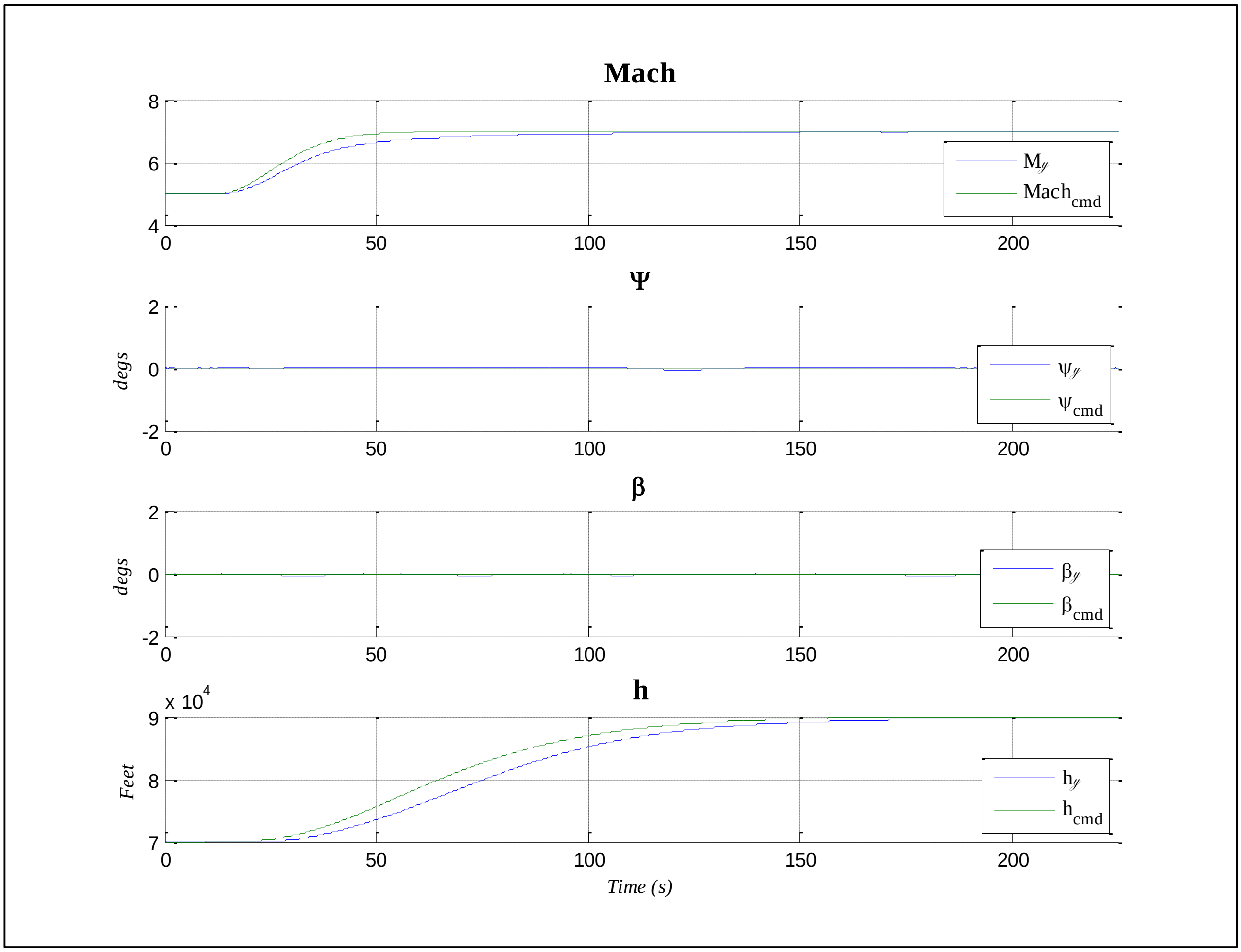


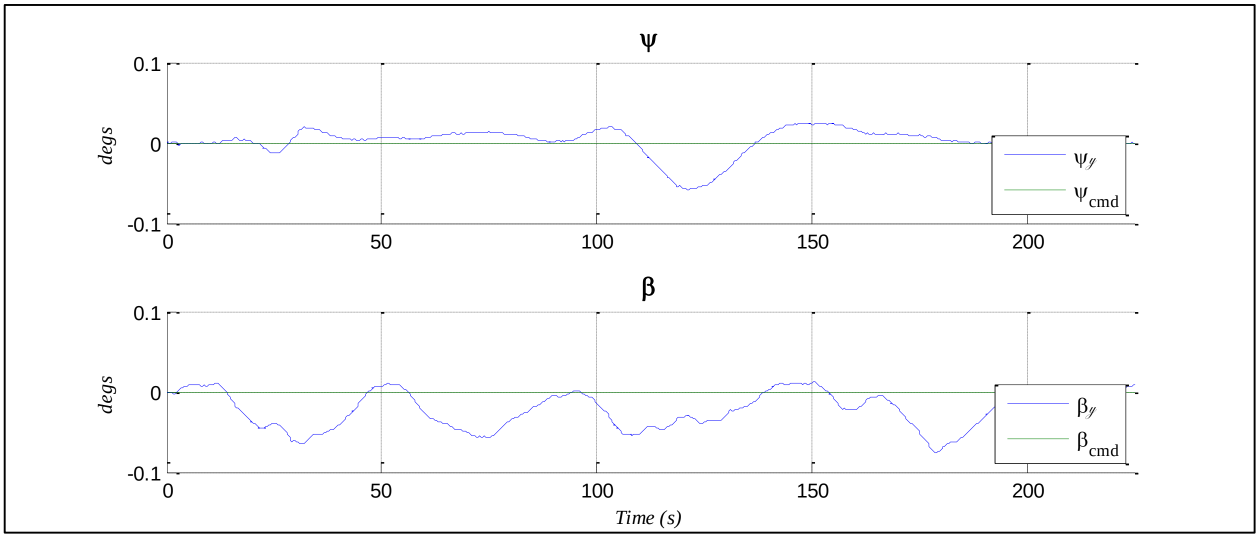

Lateral Eigenvector Component Magnitudes at Mach = 5


Lateral Eigenvector Component Magnitudes at Mach = 7
Mode 1
Mode 2
Mode 3
Mode 4
1
0.5
0
1.8
2
2.2
2.4
2.6
2.8
3
x 10^5
Weight
v/U0
p
r
ϕ

Appendix 3 Noisy Simulation Time History




Mach
8
6
4
0
50
100
150
200
My
Machcmd
Ψ
2
0
-2
degs
ψy
ψcmd
β
2
0
-2
degs
βy
βcmd
h
x 10^4
9
8
7
Feet
hy
hcmd
Time (s)


ψ
0.1
0
-0.1
degs
0
50
100
150
200
ψy
ψcmd
β
0.1
0
-0.1
degs
βy
βcmd
Time (s)