Control System Development for Autonomous Aerobatic Maneuvering with a Fixed-Wing Aircraft

by

Brian C. Reitz

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 13, 2014

Keywords: flight dynamics, unmanned, autopilot, autonomy, optimal control, aerobatics, guidance

Advisory Committee

Gilbert L. Crouse, Jr., Chair, Adjunct Associate Professor of Aerospace Engineering
John E. Cochran, Jr., Professor Emeritus of Aerospace Engineering
Winfred A. Foster, Jr., Professor of Aerospace Engineering
Andrew J. Sinclair, Associate Professor of Aerospace Engineering

Abstract

The increasing popularity of unmanned aircraft today has led to the development of more sophisticated vehicles with greater operating capabilities. The multirole nature of many civil and military unmanned aircraft along with the desire to increase the amount of automation requires the capability of the onboard control systems to handle a wide range of flight maneuvers. With the expanding flight envelope of unmanned aircraft, it is desirable to be able to quickly tune control gains for both conventional maneuvers as well as maneuvers involving unusual attitudes and airspeeds below stall speed.

The focus of this research is on designing and simulating an autopilot system capable of executing aerobatic maneuvers in addition to conventional unmanned aircraft control. Aerobatic maneuvering is used to gauge the autopilot's effectiveness in handling unusual attitudes as well as airspeeds close to and below stall speed. Modern control theory is applied to the development of a pilot-inspired control strategy based on linearized models of multiple flight conditions. A comparison between using linear aircraft models based on stability derivatives from a linear vortex lattice method and using linear models based on linearized wind tunnel data to determine the control gains is performed. The effects of sensor noise and control actuators on the performance of the autopilot is explored. Testing is also performed to determine control improvement when the control gains are scheduled based on airspeed.

Simulation shows that the pilot-inspired control strategy, together with the developed gain tuning method, can control the aircraft during both normal flight and while performing aerobatic maneuvers. While the developed autopilot is intended for use on unmanned aircraft, the same control method could also be applied to manned aircraft which could benefit from a lightweight control system capable of autonomous stall and spin recovery.

Acknowledgments

First, I would like to sincerely thank Dr. Gilbert Crouse, Jr., for his guidance, encouragement, and support throughout my graduate education as well as for his advice while completing this dissertation. I would like to thank the following members of my committee for their time, suggestions, and criticisms of this work: Dr. John Cochran, Jr., Dr. Winfred Foster, Jr., and Dr. Andrew Sinclair. Also, I would like to recognize the faculty of the Aerospace Engineering Department at Auburn University, namely Dr. Roy Hartfield, Jr. and Dr. R. Steven Gross for their advice and encouragement, thought-provoking conversations, and the opportunity to teach. I also acknowledge and appreciate the financial support offered by the department.

I would also like to thank the great friends I met while at Auburn. I will forever cherish our friendships and I appreciate the much needed advice and diversion you all provided from the mental strain of graduate school.

I owe infinite gratitude to my family, especially my father Steven and late grandfather George, for their steadfast encouragement and unwavering support throughout my studies.

Finally, to my beautiful wife Rachel, I am perpetually thankful for you and your support, love, and understanding. I truly could not have accomplished this without you.

Table of Contents

List of Figures

List of Tables

Nomenclature

α	Angle of attack
β	Sideslip angle
δ_a	Aileron deflection
δ_e	Elevator deflection
δ_r	Rudder deflection
δ_T	Throttle position
γ	Flight-path angle
$\mathbf{A}$	System state matrix
$\mathbf{B}$	System input matrix
$\mathbf{C}$	Observation matrix
$\mathbf{D}$	Observation input matrix
$\mathbf{F}$	Observation reference-command matrix
$\mathbf{G}$	System reference-command matrix
$\mathbf{h}$	Angular momentum vector
$\mathbf{K}$	Gain matrix
$\mathbf{r}$	Vector of reference commands
$\mathbf{u}$	Vector of system inputs

$\mathbf{x}$	Vector of state variables
$\mathbf{y}$	Vector of measurable outputs
ϕ	Roll angle
ψ	Yaw (heading) angle
ρ	Density
θ	Pitch angle
a_y	Lateral acceleration
b	Wing span
c	Mean chord
C_D	Drag coefficient
C_L	Lift coefficient
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient
C_n	Yawing moment coefficient
C_P	Power coefficient
C_T	Thrust coefficient
C_Y	Side-force coefficient
C_{D_0}	Parasite drag coefficient
D	Propeller diameter
g	Gravitational acceleration

h	Altitude
I	Mass moment of inertia
J_{prop}	Propeller advance ratio
m	Mass
n	Propeller rotation speed, rev/s
p	Roll rate
p_D	Position Down
p_E	Position East
p_N	Position North
q	Pitch rate
q_∞	Dynamic Pressure
r	Yaw rate
r_w	Washed-out yaw rate
S	Wing area
u	X-body velocity component
v	Y-body velocity component
V_w	Airspeed
w	Z-body velocity component
x_w	Washout filter state

Chapter 1

Introduction

1.1 Motivation

Unmanned aircraft (UA) have been employed to perform numerous tasks related to both civilian and military missions. Most of the missions in which UA are used involve flight around a steady-state trimmed flight condition to or from a specific waypoint. These missions have lead to autopilots that were designed only to maintain vehicle states such as speed, altitude, and heading as well as fairly gentle maneuvers to transition between steady-state conditions. Most conventional autopilots are based on a limited set of linearized aircraft models which are usually derived based on the assumptions that the aircraft attitude angles are reasonably small and that the aircraft longitudinal and lateral states are separable. This control methodology leads to complications when currently-flying UA are adapted to perform multiple types of missions. For UA that are intended to fly military combat missions, or missions where an evasive maneuver may be needed to avoid enemy fire, the limited control authority offered by the conventional autopilot will result in a vehicle that is more susceptible to enemy countermeasures. With the recent push to integrate UA into the National Airspace System (NAS), both military and civilian UA need to be maneuverable to be able to avoid conflicting air traffic and obstacles caused by structures or terrain [1]. The limits imposed on the operational capabilities of aircraft equipped with conventional autopilots leads to the development of a control system capable of complex flight maneuvers without airspeed constraints.

In addition to the UA application, work in the field of maneuvering autopilots could also be applied to manned aircraft and used to increase the safety of manned aviation. Data from the General Aviation Manufacturers Association (GAMA) shows that deliveries

of new aircraft manufactured in the U.S. have significantly declined since 1978 [2]. There are numerous reasons speculated for the cause of the decline, but one of the major reasons is the perception of general aviation safety. In regard to manned aviation in aircraft weighing less than 12,500 pounds, an Air Safety Institute study found that stall/spin accidents are the most deadly, accounting for almost 10 percent of all accidents and 13.7 percent of fatal accidents [3]. With accident reports showing that a major cause of general aviation accidents is a stall or a stall followed by a spin, it is believed that a light-weight autopilot system capable of stall and spin recovery could decrease the number of accidents each year. This reduction in accident rate would mean fewer pilot fatalities and hopefully an increased positive perception of general aviation safety.

1.2 Background

The increasing demand for automation from the Department of Defense (DoD) stems from decreases in the defense budget, insufficient manpower to support complex missions, operational environments that are hazardous to human activity, and new mission requirements with increased capabilities for adaptive control of vehicles in unpredictable environments [4–6]. The economics of UA have been studied extensively and Friedman [7] discussed the advantages of unmanned combat air systems when compared to a fleet of fighter or attack aircraft. The idea for a fighter or attack type UA leads to the study of control during maneuvers with large pitch and roll angles as well as rapid accelerations.

1.2.1 Maneuvering Autopilots

Multiple examples of maneuvering flight control systems are presented in the literature and are reviewed to provide an overview of the methods that have been employed in the past. Recently, Mattei and Monaco [8] designed an autopilot for a highly maneuverable asymmetric missile using a backstepping approach to control both bank-to-turn and skid-to-turn maneuvers. Jang et al. [9] demonstrated through simulation the performance of a

flight control system capable of vertical takeoff and landing with a test bed vehicle despite the presence of bending modes and sloshing propellant. A control system for the aerial rendezvous of two small UA was developed by Nichols et al. [10] which used a passively towed drogue by a mothership UA to guide the following UA through the use of vision-based sensors.

An autopilot system that incorporated a nonlinear path-following guidance law was developed in [11–13] and used to perform basic aerobatic maneuvers as well as a mid-air rendezvous of two UA. An acceleration command was generated to align the velocity vector of the aircraft with a reference point along the desired path. The control gains to generate the acceleration command were determined through the use of a linear quadratic regulator along with a single linear aircraft model. A similar strategy was used by Peddle and Jones [14], who developed an attitude independent acceleration controller in which the specific force vector was controlled. The specific force vector was determined by subtracting the gravitational force component from the total force vector and was done to allow control of the aircraft that was orientation independent.

Vertical hovering maneuvers have been a research area of interest in recent years and have been accomplished indoors with a small electric-powered UA through the use of an optical motion capture system to determine position and attitude [15, 16]. Matsumoto et al. [17] developed a control strategy for the turnaround of a high thrust-to-weight ratio UA flying in an indoor space using vertical hovering maneuvers to minimize the vertical space required to reverse heading. A Lyapunov-based backstepping controller, which is useful when system states are controlled by other states, was used in combination with a quaternion based attitude description in [18, 19] to transition to and from stationary hover with a UA and in [20] to develop a longitudinal autopilot for a missile.

1.2.2 Modern Control Design

Eigenstructure assignment is an extension of the pole placement technique for single-input/single-output (SISO) systems which allows the designer the ability to assign the closed-loop eigenvalues as well as the eigenvectors for multi-input/multi-output (MIMO) systems within limits [21]. This method of control design has been used extensively in the past [22–24] and is relevant in aircraft design because the time response of an aircraft control system depends on both the eigenvalues (poles) and the eigenvectors (zeros). Specification MIL-F-8785 [25] provides flying qualities specifications for piloted aircraft in terms of natural frequencies and damping ratios and can be used along with the eigenstructure assignment technique to design flight control systems for both military and civilian aircraft. Alag and Duke [26] successfully demonstrated the use of the eigenstructure technique to develop an autopilot that would provide precise, repeatable control of an F-15 during flight test maneuvers. The specifications in MIL-F-8785-C were used during the design process and their approach was able to control maneuvers such as pushover pullups, excess-thrust windup turns, and thrust-limited turns.

A modern control approach commonly used in aircraft, missile, and spacecraft control design is the linear quadratic regulator (LQR) [27–29]. A nice benefit of the LQR technique is that, unlike SISO design methods, all of the control gains are solved for simultaneously so that all the control loops can be closed at the same time. There are different approaches to the use of the LQR method with some incorporating full-state feedback while others use output feedback [21]. In some instances the quadratic cost used to determine the control gains is minimized over a finite horizon (fixed time interval) while in other instances the cost is minimized over an infinite horizon. The idea behind the LQR approach is that control gains are selected in order to minimize the energy in the states without using too much control energy. The control design engineer selects the weighting matrices in the cost function to trade off requirements between a fast response (making the state perturbations go to zero more quickly) and the amount of control effort employed. A modified version

of linear quadratic control theory was introduced in [30] for application to the servodesign problem where instead of regulating the states to zero, the states are commanded to follow a reference command. This method is referred to as a linear quadratic tracker (LQT) and allows compensator dynamics of any desired structure to be incorporated into the system model. This is a nice benefit because the knowledge of classical aircraft control design can be used to form the control structure based on the application being investigated.

While the previously described methods were based on linear aircraft models, studies involving alternative techniques of handling the known nonlinearities associated with aircraft dynamics have also been performed. Dynamic inversion is a popular method to control a nonlinear system and is based on the technique of feedback linearization [21]. The method requires that the controller have a full model of nonlinearities of the aircraft. Johnson et al. [31] combined the use of dynamic inversion with a neural network to account for deficiencies in the aircraft reference model to develop and flight test an adaptive controller to transition a fixed-wing UA to and from vertical hover. A similar approach was used by Bruner [32] to separate the coupled modes of a general aviation aircraft to allow direct control of airspeed and flight-path angle.

Model predictive control, or receding-horizon control, is a robust control technique that has been used to perform autonomous loops and rolls [33] as well as autonomously control a UA after the loss of an aerodynamic surface [34]. This strategy uses a model of the aircraft to predict the behavior of certain states by regulating the aircraft about a predefined, time-varying trajectory. The aircraft response is then used to calculate an optimal control input onboard the aircraft based on the minimization of a cost function over a future horizon where the horizon is the amount of foreseeable future taken into account. Since the cycle of prediction and control optimization happens onboard the aircraft, the method is robust to unmodeled aircraft dynamics such as flexible bending modes as well as unmodeled atmospheric disturbances. Another robust control technique known as loop-shaping was presented by AlSwailem [35] to control a catapult-launched UA. Loop-shaping combines

the traditional intuition of classical control methods with optimization techniques to achieve stable controllers despite differences in the modeled and actual aircraft.

Another robust control design method is the linear quadratic Gaussian/loop-transfer recovery (LQG/LTR) technique. This method aims to take advantage of the robust stability properties of the LQR with full state feedback which are an infinite gain margin and no less than 60 degree phase margin [21]. If all of the states are available (and the structure of the compensator is of no concern), the state-feedback LQR is beneficial. If, however, all of the states are not available and some or all of the measured states are affected by noise (which is commonly the case in aircraft control) an observer is needed to estimate the unavailable/actual states from the measured outputs [36]. This is exactly what the LQG/LTR technique does. Even though all of the states may not be available for feedback and the measured states may contain noise, the observer is able to estimate all of the states so that the guaranteed robustness properties of the LQR with full-state feedback can be recovered [21]. The procedure of combining the LQR technique with a Kalman filter [37] to design a dynamic regulator is referred to as linear quadratic Gaussian (LQG) design [21]. Due to the presence of the observer in LQG design, the robustness properties of the LQR are no longer guaranteed and need to be recovered through the loop-transfer recovery (LTR) technique [38]. One downside of the LQG/LTR approach is that it is not suitable for plants that are non-minimum-phase or have stable zeros with large time constants [21]. In some applications, even if the plant is non-minimum-phase, satisfactory results can still be achieved [38].

1.2.3 Gain Scheduling

To extend linear control methods over the entire flight envelope, including changes in velocity and altitude, gain scheduling has been successfully implemented since the 1960's [39]. Gain scheduling involves the decomposition of a nonlinear system into multiple linear subsystems. Leith and Leithead [40] present theoretical results relating nonlinear system dynamics to a group of linear systems in addition to describing the classical gain scheduling design

procedure. Smith [41] used gain scheduling to design an autopilot for bank-to-turn missiles. Myrand-Lapierre et al. [42] used a systematic approach based on a logic-based switching supervisor to switch between two separate linear control methods which together defined the motion of a nonlinear system. The technique was used to switch between level flight and hover for a fixed-wing aircraft. With a similar method, Krogh [43] switched between two linear flight regimes to allow for transitions from hover to level flight. Gavrilets [44] presented the gain-scheduled design of an autopilot for aerobatic maneuvering of a miniature helicopter based on human-inspired logic for switching between flight control methods.

1.2.4 Aerodynamic Modeling

Computational aerodynamics uses numerical solutions of the governing equations of fluid mechanics to determine the forces and moments on a body [45]. The aerodynamic modeling method selected for this work is a vortex lattice method called Athena Vortex Lattice (AVL). Vortex lattice methods are similar to panel methods in that singularities of unknown strength are placed on a surface and a system of linear algebraic equations is solved to determine the strengths [45]. AVL was originally written by Harold Youngren in 1988 and is based on the work of Lamar [46–48], Lan [49], and Miranda [50] [51]. The code is open-source, user friendly, and is continuously updated by Youngren and Mark Drela of MIT. The input and output files for the code are also easily understood and the documentation in [51] provides helpful tips and examples on the use of the program.

Since the control system developed in this work is intended to control both normal flight operations as well as post-stall maneuvers and initial testing is performed using simulation, it is beneficial to review aerodynamic modeling methods of flight conditions involving speeds below stall speed. The aerodynamic modeling of post-stall conditions has been a topic of concern for many years and has been modeled for a number of different fighter aircraft [52–56]. An increase in stall/spin accidents with general aviation aircraft in the 1980's led to modeling research by NASA [57]. More recently, research into the post-stall dynamics of transport

category aircraft has been performed in an effort to better understand the loss of control motions that lead to spin accidents as well as improve simulations for pilot training and control system testing [58,59]. Murch and Foster [60] explained the development of new wind tunnel motion rigs for data acquisition as well as multiple blending methods for combining forced oscillation data with rotary balance data based on the characteristics of aircraft motion in order to more accurately simulate the highly dynamic post-stall flight regime.

1.3 Objective

This work focuses on gaining a better understanding of the control system requirements to accomplish aerobatic maneuvers. The approach described herein has the potential to increase the safety of general aviation flight and contribute to the development of more advanced UA capable of handling the problems associated with modern warfare. Federal aviation regulation 91.303 defines aerobatic flight as "any maneuver involving an abrupt change in aircraft attitude, an abnormal attitude, or abnormal acceleration not necessary for normal flight" [61]. Loops and rolls are two basic aerobatic figures and are selected to initially test the maneuvering functionality of the autopilot. Advanced aerobatic maneuvers such as knife-edge flight, spins, stall turns, and tail slides are used to test the autopilot's ability to maintain control during maneuvers that involve unusual attitudes as well as airspeeds close to and below stall speed.

Figure 1.1 outlines the process followed to develop the control system. Once a test aircraft is selected, two sets of linear aircraft models are derived for use during the tuning of the control gains. Two sets of models are used to demonstrate the effects of inaccurate aerodynamic data on the overall performance of the system. The first set of models uses linear aerodynamic data from a vortex lattice model combined with the linearized equations of motion to generate state-space models of the aircraft. The second set of models is created by linearizing the wind tunnel data from the nonlinear simulation. Chapter 2 describes the aircraft model as well as the linearization process. Pilot-controlled simulation data is

presented in Chapter 3 along with an explanation of the control approach for each maneuver. Chapter 4 discusses the infinite-horizon optimal control approach used to calculate the control gains and Chapter 5 presents the nonlinear simulation results.

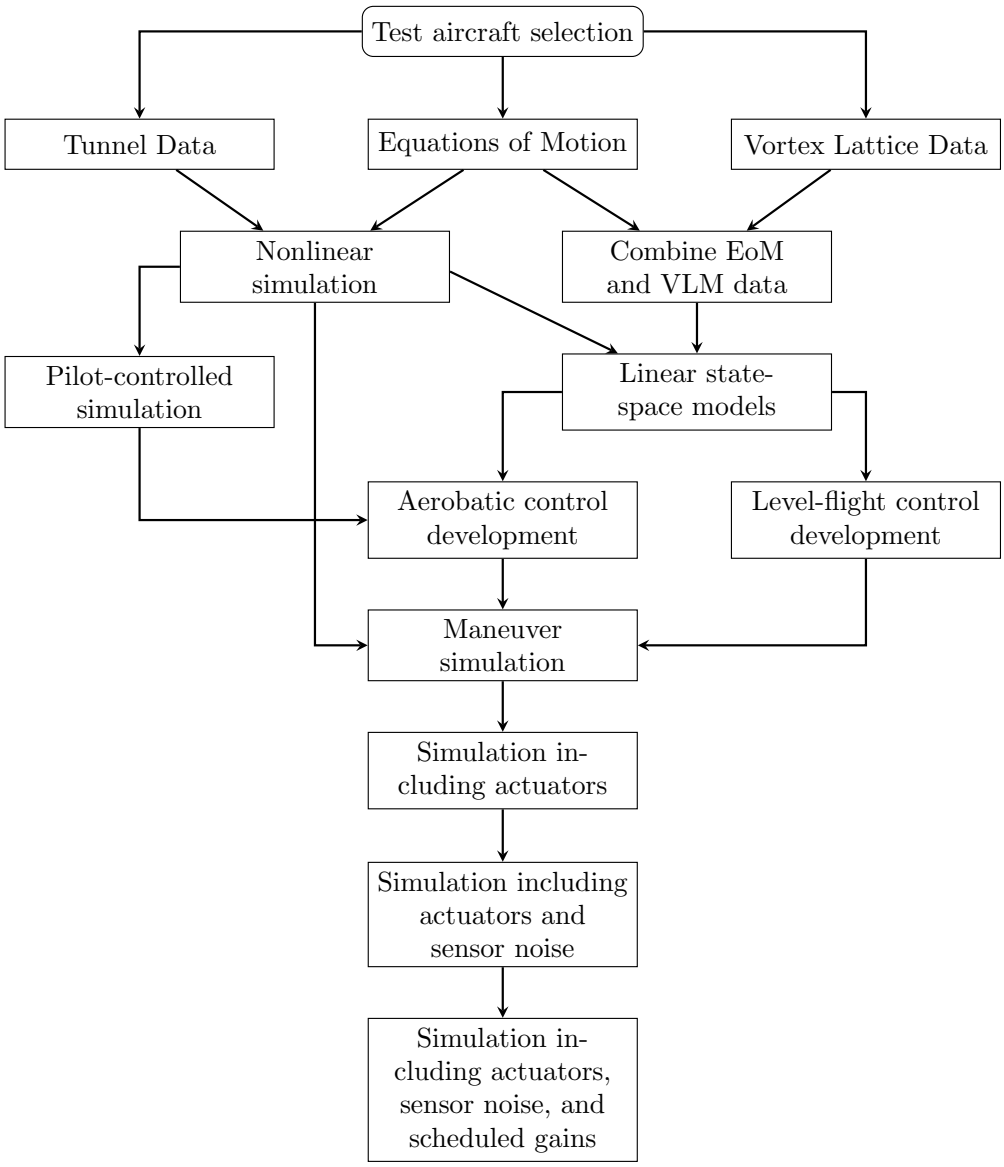


Figure 1.1: Work-flow block diagram.

The author's specific contributions and the location of their corresponding discussion in this dissertation are listed as follows:

- Section 2.2.3 presents a comparison of vortex lattice modeling techniques for propeller driven airplanes.
- A pilot-inspired approach to control-loop development for maneuvering a fixed-wing aircraft based on pilot-controlled simulation data is described in Chapter 3.
- Section 4.2.5 discusses how to apply an optimal gain tuning technique, based on linear quadratic methods, to maneuvering flight control design while using coupled dynamics and actuator models.
- Chapter 5 shows the comparison results of using vortex lattice models in the place of wind tunnel data to design control systems using an infinite-horizon optimal control approach.
- Also in Chapter 5 are the results of a comparison between constant gains and scheduled gains for maneuvering flight control.

Chapter 2

Aircraft Model

The test aircraft selection was based on wind tunnel data availability, the capability of the aircraft to perform the desired test maneuvers, and the airframe being a low-cost, commercially available model aircraft to allow for future development and testing. Owens et al. [62] provide some details of the model used by NASA to conduct flight test experiments of advanced control design concepts. Figure 2.1 shows a picture of the aircraft from [62], which is a Hangar 9 Ultra-Stick^{TM} 120. Tables 2.1 and 2.2 list the geometric properties and the control surface sizes of the test aircraft. The mass characteristics are given in Table 2.3 where the vertical center of gravity (CG) position is measured above the fuselage centerline. It is important to point out that because of the aircraft symmetry, the only nonzero product of inertia is the one associated with the XZ plane.

Figure 2.1: Aircraft for initial control system testing.

Table 2.1: Geometric properties of the test aircraft.

	Main Wing	Horizontal Stabilizer	Vertical Stabilizer
Area, ft^2	8.28	2.38	0.5
Span, ft	6.29	2.44	0.5
Mean Chord, ft	1.42	0.88	1.0
Dihedral, deg	0.7	0	90
Leading Edge Sweep, deg	0	0	59
Incidence, deg	0	0	0
Twist, deg	0	0	0

Table 2.2: Test aircraft control surface sizes.

	Aileron	Elevator	Rudder
Span, ft	1.34	0.85	0.5
Root Chord, ft	0.261	0.347	0.565
Tip Chord, ft	0.193	0.347	0.255
Spanwise Root Location, ft	1.68	0.37	0

Table 2.3: Test aircraft mass characteristics.

Weight, lb	x_{cg}/c	y_{cg}/b	z_{cg}/c	I_X, slug-ft^2	I_Y, slug-ft^2	I_Z, slug-ft^2	I_{XZ}, slug-ft^2
16.35	0.22	0	0.166	0.6319	0.7446	1.2542	-0.14

To test the developed control system, a nonlinear model of the test aircraft with six degrees of freedom (DoF) was developed using MATLAB's Simulink environment. The equations of motion used in the simulation are derived in the next section, followed by an explanation of the aerodynamic and thrust models used to represent the forces and moments applied to the airplane. The simulation also includes actuator and sensor models to represent the equipment onboard the aircraft. The chapter concludes with the derivation of linear models of the airplane which are used in the design of the control system.

2.1 Equations of Motion

The nonlinear equations of motion (EoM) are derived using methods presented in [21, 63–65]. The Earth is assumed flat and considered to be an inertial frame. The aircraft

is assumed to be a rigid body with constant mass and constant mass distribution. Euler orientation angles ϕ, θ, and ψ are used in the feedback path of the control system to describe the orientation of the aircraft-based body frame, located at the CG, because of their intuitive nature. Equation 2.1 is Newton's second law applied to the center of mass of the aircraft and is used to describe the translational motion [21, 63, 65].

$$m\left(\dot{\mathbf{V}} + \boldsymbol{\omega} \times \mathbf{V}\right) = \mathbf{F}_{\mathbf{gravity}} + \mathbf{F}_{\mathbf{aero}} + \mathbf{F}_{\mathbf{thrust}} \tag{2.1}$$

The vector expressions for velocity, acceleration, and angular velocity in the above equation are expressed in the body frame as

$$\mathbf{V} = u\,\hat{i} + v\,\hat{j} + w\,\hat{k} \tag{2.2}$$

$$\dot{\mathbf{V}} = \dot{u}\,\hat{i} + \dot{v}\,\hat{j} + \dot{w}\,\hat{k} \tag{2.3}$$

$$\boldsymbol{\omega} = p\,\hat{i} + q\,\hat{j} + r\,\hat{k} \tag{2.4}$$

With the aerodynamic force coefficients in the stability axes, the vector expressions for the forces applied to the aircraft from [21] are

$$\mathbf{F}_{\mathbf{gravity}} = m\,g\left(-\sin(\theta)\;\hat{i} + \sin(\phi)\;\cos(\theta)\;\hat{j} + \cos(\phi)\;\cos(\theta)\;\hat{k}\right) \tag{2.5}$$

$$\mathbf{F}_{\mathbf{aero}} = S\,q_\infty\left((C_L\;\sin(\alpha) - C_D\;\cos(\alpha))\;\hat{i} + C_Y\,\hat{j} - (C_L\;\cos(\alpha) + C_D\;\sin(\alpha))\;\hat{k}\right) \tag{2.6}$$

$$\mathbf{F}_{\mathbf{thrust}} = S\,q_\infty\left(C_{T_X}\,\hat{i} + C_{T_Y}\,\hat{j} + C_{T_Z}\,\hat{k}\right) \tag{2.7}$$

After substituting Eqs. 2.2 - 2.7 into Eq. 2.1, the conservation of linear momentum equations with respect to the body axes are given by

$$\dot{u} = r\,v - q\,w - g\,\sin(\theta) + \frac{C_L\,S\,q_\infty\,\sin(\alpha) - C_D\,S\,q_\infty\,\cos(\alpha) + C_{T_X}\,S\,q_\infty}{m} \tag{2.8}$$

$$\dot{v} = p\,w - r\,u + g\,\cos(\theta)\,\sin(\phi) + \frac{C_Y\,S\,q_\infty + C_{T_Y}\,S\,q_\infty}{m} \tag{2.9}$$

$$\dot{w} = q\,u - p\,v + g\,\cos(\phi)\,\cos(\theta) - \frac{C_L\,S\,q_\infty\,\cos(\alpha) + C_D\,S\,q_\infty\,\sin(\alpha) - C_{T_Z}\,S\,q_\infty}{m} \tag{2.10}$$

The known terms of these equations are the aerodynamic and thrust forces as well as the acceleration of gravity. In order to solve this system of nonlinear differential equations, the body-axes components of the angular velocity vector are needed as well as the Euler orientation angles.

The conservation of angular momentum applied to the mass center in the body frame will be used to determine the components of the angular velocity vector. From classical mechanics texts [66], it is shown that the derivative of the angular momentum vector, taken in the inertial frame, is equal to the total moment applied about the center of mass and is analogous to Newton's second law for translational momentum. In equation form, this is expressed as

$$\sum \mathbf{M} = \dot{\mathbf{h}_i} \tag{2.11}$$

The derivative of the angular momentum vector $\mathbf{h}$ is simply the product of the inertia matrix and the angular velocity. Since the inertia of the aircraft can be estimated, and is constant in the body frame, Eq. 2.11 can be rewritten in the body frame in a similar form as in [21,63,65] as

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \mathbf{M_{aero}} + \mathbf{M_{thrust}} \tag{2.12}$$

with the inertia matrix defined as

$$\mathbf{I} = \begin{bmatrix} I_X & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_Y & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_Z \end{bmatrix} \tag{2.13}$$

and the angular acceleration written in terms of its scalar components as

$$\dot{\boldsymbol{\omega}} = \dot{p}\,\hat{i} + \dot{q}\,\hat{j} + \dot{r}\,\hat{k} \tag{2.14}$$

With the aerodynamic moment coefficients in the stability axes, the vector expressions for the moments acting on the aircraft from [63] are

$$\mathbf{M_{aero}} = S\,q_\infty \left(b\,\left(C_l\,\cos(\alpha) - C_n\,\sin(\alpha)\right)\,\hat{i} + c\,C_m\,\hat{j} + b\,\left(C_n\,\cos(\alpha) + C_l\,\sin(\alpha)\right)\hat{k} \right) \tag{2.15}$$

$$\mathbf{M_{thrust}} = S\,q_\infty \left(b\,C_{T_l}\,\hat{i} + c\,C_{T_m}\,\hat{j} + b\,C_{T_n}\,\hat{k} \right) \tag{2.16}$$

After substitution into Eq. 2.12, the explicit conservation of angular momentum equations are given by

$$\dot{p} = -\frac{I_{XZ}\ (q\ (I_X\, p - I_{XZ}\, r) - I_Y\, p\, q + C_{T_n}\, S\, b\, q_\infty + C_n\, S\, b\, q_\infty\, \cos(\alpha) + C_l\, S\, b\, q_\infty\, \sin(\alpha))}{{I_{XZ}}^2 - I_X\, I_Z}$$
$$- \frac{I_Z\ (q\ (I_{XZ}\, p - I_Z\, r) + I_Y\, q\, r + C_{T_l}\, S\, b\, q_\infty + C_l\, S\, b\, q_\infty\, \cos(\alpha) - C_n\, S\, b\, q_\infty\, \sin(\alpha))}{{I_{XZ}}^2 - I_X\, I_Z} \tag{2.17}$$

$$\dot{q} = -\frac{p\ (I_{XZ}\, p - I_Z\, r) + r\ (I_X\, p - I_{XZ}\, r) - C_{T_m}\, S\, c\, q_\infty - C_m\, S\, c\, q_\infty}{I_Y} \tag{2.18}$$

$$\dot{r} = -\frac{I_X\ (q\ (I_X\, p - I_{XZ}\, r) - I_Y\, p\, q + C_{T_n}\, S\, b\, q_\infty + C_n\, S\, b\, q_\infty\, \cos(\alpha) + C_l\, S\, b\, q_\infty\, \sin(\alpha))}{{I_{XZ}}^2 - I_X\, I_Z}$$
$$- \frac{I_{XZ}\ (q\ (I_{XZ}\, p - I_Z\, r) + I_Y\, q\, r + C_{T_l}\, S\, b\, q_\infty + C_l\, S\, b\, q_\infty\, \cos(\alpha) - C_n\, S\, b\, q_\infty\, \sin(\alpha))}{{I_{XZ}}^2 - I_X\, I_Z} \tag{2.19}$$

The kinematic equations are derived next and are needed to determine expressions for the Euler angles in terms of the aircraft angular velocity. Using a similar approach as presented in [21], starting from the inertial frame and using two intermediate frames, whose relative angular velocities are given by the Euler angle rates, allows the angular velocity in the body frame to be written as

$$\boldsymbol{\omega} = \dot{\phi}\, \hat{\imath} + \mathbf{C}_\phi \left(\dot{\theta}\, \hat{\jmath} + \mathbf{C}_\theta\, \dot{\psi}\, \hat{k} \right) \tag{2.20}$$

where

$$\mathbf{C}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.21}$$

and

$$\mathbf{C}_\theta = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2.22}$$

Combining Eqs. 2.4 and 2.20 results in

$$\boldsymbol{\omega} \equiv \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.23}$$

The kinematic equations are determined through the inverse transformation to be

$$\dot{\phi} = p + \frac{\sin(\theta)\ (r\ \cos(\phi) + q\ \sin(\phi))}{\cos(\theta)} \tag{2.24}$$

$$\dot{\theta} = q\ \cos(\phi) - r\ \sin(\phi) \tag{2.25}$$

$$\dot{\psi} = \frac{r\ \cos(\phi) + q\ \sin(\phi)}{\cos(\theta)} \tag{2.26}$$

For the nonlinear simulation in this work, the aircraft orientation was represented with a quaternion to avoid the singularity associated with Euler angles that occurs when the pitch angle is 90 degrees. The quaternion was initialized from Euler angles since the quaternion rotation axis is not evident for compound rotations (e.g., yaw, pitch, and roll combined) [21]. The transformations from Euler angles to quaternion are given in [21] and are repeated in Eqs. 2.27 - 2.30.

$$q_0 = \cos(\phi/2)\ \cos(\theta/2)\ \cos(\psi/2) + \sin(\phi/2)\ \sin(\theta/2)\ \sin(\psi/2) \tag{2.27}$$

$$q_1 = \sin(\phi/2)\ \cos(\theta/2)\ \cos(\psi/2) - \cos(\phi/2)\ \sin(\theta/2)\ \sin(\psi/2) \tag{2.28}$$

$$q_2 = \cos(\phi/2)\ \sin(\theta/2)\ \cos(\psi/2) + \sin(\phi/2)\ \cos(\theta/2)\ \sin(\psi/2) \tag{2.29}$$

$$q_3 = \cos(\phi/2)\ \cos(\theta/2)\ \sin(\psi/2) - \sin(\phi/2)\ \sin(\theta/2)\ \cos(\psi/2) \tag{2.30}$$

The Euler angles can be found from the quaternion using Eq. 2.31 presented in [21].

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_2\, q_3 + q_0\, q_1)}{1-2\left(q_1^2+q_2^2\right)}\right) \\ \arcsin(2\,(q_0\, q_2 - q_3\, q_1)) \\ \arctan\left(\frac{2(q_0\, q_3 + q_1\, q_2)}{1-2\left(q_2^2+q_3^2\right)}\right) \end{bmatrix} \tag{2.31}$$

The kinematic equations can be written in terms of the quaternion as

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.32}$$

The equations derived thus far describe the motion of the aircraft with respect to the body frame. To describe the trajectory of the airplane with respect to the Earth-based inertial frame, the navigation equations are used. The navigation equations, also referred to as the flight path equations, are obtained by performing three successive rotations of the velocity vector from the body frame to the inertial frame. Equations 2.33 - 2.35 are the navigation equations from [63] and are the final set of equations needed to fully describe the motion of the aircraft.

$$\begin{aligned} \dot{p}_N = {} & w\,(\sin(\phi)\,\sin(\psi) + \cos(\phi)\,\cos(\psi)\,\sin(\theta)) \\ & - v\,(\cos(\phi)\,\sin(\psi) - \cos(\psi)\,\sin(\phi)\,\sin(\theta)) + u\,\cos(\psi)\,\cos(\theta) \end{aligned} \tag{2.33}$$

$$\begin{aligned} \dot{p}_E = {} & v\,(\cos(\phi)\,\cos(\psi) + \sin(\phi)\,\sin(\psi)\,\sin(\theta)) \\ & - w\,(\cos(\psi)\,\sin(\phi) - \cos(\phi)\,\sin(\psi)\,\sin(\theta)) + u\,\cos(\theta)\,\sin(\psi) \end{aligned} \tag{2.34}$$

$$\dot{p}_D = w\,\cos(\phi)\,\cos(\theta) - u\,\sin(\theta) + v\,\cos(\theta)\,\sin(\phi) \tag{2.35}$$

2.2 Aircraft Forces and Moments

This work employs the use of two separate aerodynamic modeling methods to represent the aircraft. The first method is based on nonlinear wind tunnel data and is used to create the nonlinear simulation. The second method is based on linear data obtained from an open-source vortex lattice modeling method. Both methods are used in Chapter 4 to determine two different sets of control gains. The linear vortex lattice method is used in an effort to demonstrate the development process for a newly acquired aircraft when wind tunnel data is not available as well as demonstrate the robustness of the method despite aerodynamic data inaccuracies.

2.2.1 Aerodynamic Forces and Moments

The nonlinear aerodynamic model used to represent the aerodynamic forces and moments for the simulation is based on data from [67–71] acquired in the NASA Langley Research Center 12-foot low speed tunnel. Two sets of wind tunnel tests were performed. The first set was performed in 2006 and consisted of static tests. The second set, in 2011, expanded the data to include forced oscillation tests in roll and yaw. The baseline aerodynamics are modeled as nonlinear functions of angle of attack and sideslip angle for angles of attack from -3 to 42 degrees and sideslip angles from -30 to 30 degrees. Thrust effects are included as well as control surface effects modeled as nonlinear functions of angle of attack, sideslip angle, control surface deflection, and advance ratio as defined in Eq. 2.36. The advance ratio is equivalent to the wing angle of attack and is the ratio between the distance the aircraft advances with each revolution of the propeller and the diameter of the propeller [72].

$$J_{prop} = \frac{V_w}{n\,D} \tag{2.36}$$

The dynamic data contains the effects of roll and yaw rate on the sideforce, rolling moment, and yawing moment for angles of attack from -5 to 40 degrees with dimensionless roll rates

from 0 to 0.0976 and dimensionless yaw rates from 0 to 0.1368 given by Eqs. 2.37 and 2.39, respectively. The pitch damping derivative used in the simulation is a linear estimate from the United States Air Force Stability and Control Digital Data Compendium*(DATCOM) based on the dimensionless pitch rate given by Eq. 2.38.

$$\hat{p} = \frac{p\,b}{2\,V_w} \tag{2.37}$$

$$\hat{q} = \frac{q\,c}{2\,V_w} \tag{2.38}$$

$$\hat{r} = \frac{r\,b}{2\,V_w} \tag{2.39}$$

The control surface and angular rate effects are only modeled for the coefficients where a significant effect was observed in the wind tunnel data. Asymmetries have been removed so that the data is symmetric about zero sideslip. The aerodynamic model is implemented in the simulation through the use of lookup tables with linear interpolation between the data points as well as linear extrapolation for points outside the available data. Eqs. 2.40 - 2.45 represent the aerodynamic model used in the nonlinear simulation. The baseline

*The USAF Digital DATCOM is public domain software obtainable from `http://www.pdas.com/datcomdownload.html`. A list of references including the user's manual and the original documents can be found at `http://www.pdas.com/datcomrefs.html`.

aerodynamic data is depicted in Fig. 2.2.

$$C_D = C_D\left(\alpha,\, \beta\right) + C_{D_{\delta_e}}\left(\alpha,\, \delta_e,\, J_{prop}\right) + C_{D_{\delta_T}}\left(\alpha,\, \beta,\, J_{prop}\right) \tag{2.40}$$

$$C_Y = C_Y\left(\alpha,\, \beta\right) + C_{Y_p}\left(\alpha,\, \hat{p}\right) + C_{Y_r}\left(\alpha,\, \hat{r}\right) + C_{Y_{\delta_r}}\left(\alpha,\, \beta,\, \delta_r,\, J_{prop}\right) + C_{Y_{\delta_T}}\left(\alpha,\, \beta,\, J_{prop}\right) \tag{2.41}$$

$$C_L = C_L\left(\alpha,\, \beta\right) + C_{L_{\delta_e}}\left(\alpha,\, \delta_e,\, J_{prop}\right) + C_{L_{\delta_T}}\left(\alpha,\, \beta,\, J_{prop}\right) \tag{2.42}$$

$$C_l = C_l\left(\alpha,\, \beta\right) + C_{l_p}\left(\alpha,\, \hat{p}\right) + C_{l_r}\left(\alpha,\, \hat{r}\right) + C_{l_{\delta_a}}\left(\alpha,\, \beta,\, \delta_a,\, J_{prop}\right) + C_{l_{\delta_T}}\left(\alpha,\, \beta,\, J_{prop}\right) \tag{2.43}$$

$$C_m = C_m\left(\alpha,\, \beta\right) + C_{m_q}\,\hat{q} + C_{m_{\delta_e}}\left(\alpha,\, \delta_e,\, J_{prop}\right) + C_{m_{\delta_T}}\left(\alpha,\, \beta,\, J_{prop}\right) \tag{2.44}$$

$$C_n = C_n\left(\alpha,\, \beta\right) + C_{n_p}\left(\alpha,\, \hat{p}\right) + C_{n_r}\left(\alpha,\, \hat{r}\right) + C_{n_{\delta_r}}\left(\alpha,\, \beta,\, \delta_r,\, J_{prop}\right) + C_{n_{\delta_T}}\left(\alpha,\, \beta,\, J_{prop}\right) \tag{2.45}$$

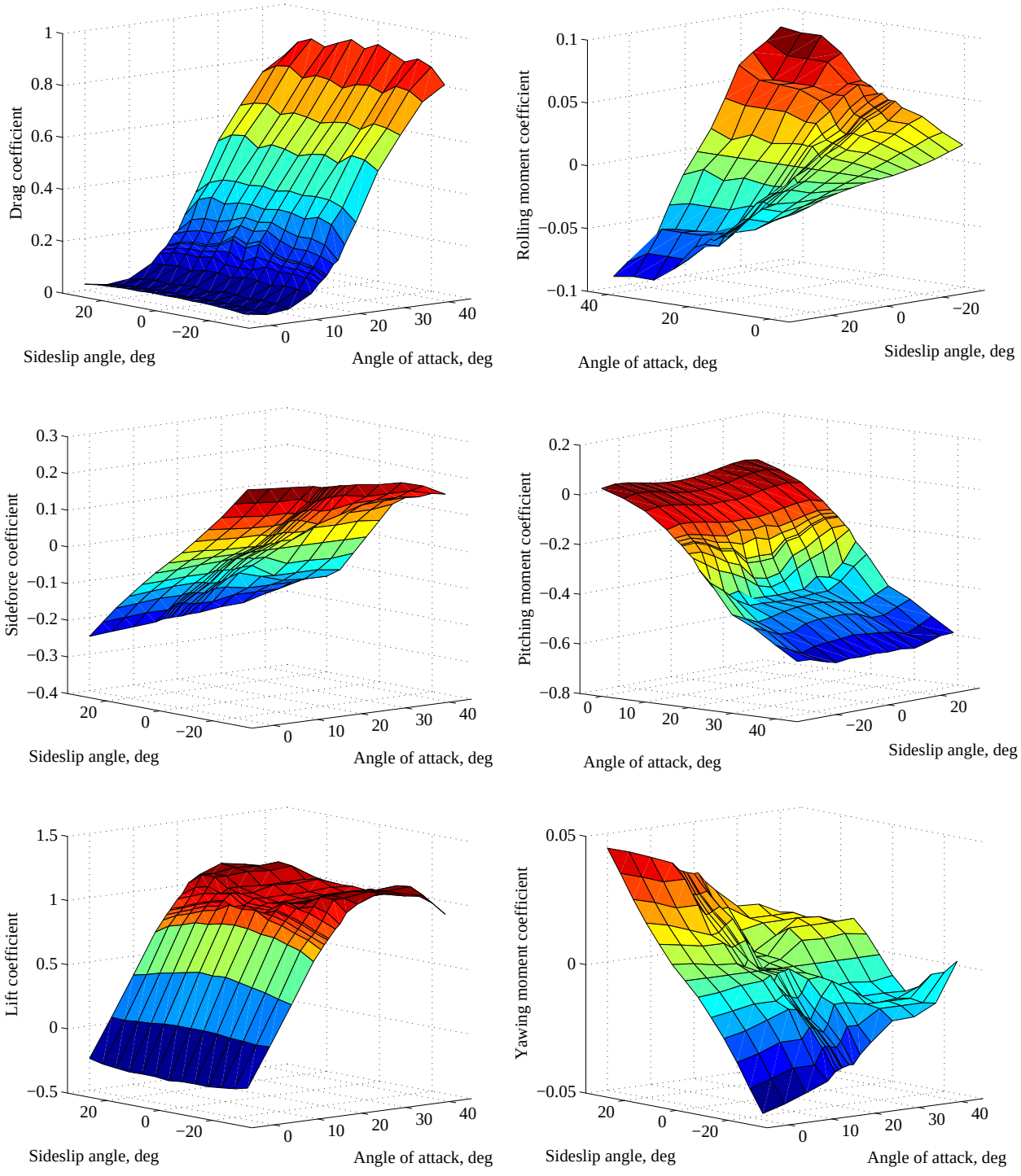


Figure 2.2: Baseline wind tunnel data.

2.2.2 Thrust Force and Moments

The aircraft used in the simulation is powered by a 1,440 watt electric motor with a propeller that has a diameter of 14 inches and a pitch of 10 inches per revolution. The nonlinear engine model used for this work is based on the model of a similar engine presented in [73] which gives polynomial representations of the thrust and power coefficients as functions of advance ratio. The thrust force was assumed to be aligned with the body axes in the $\hat{i}$ direction and is determined using the thrust coefficient model in Eq. 2.46.

$$C_T = -0.4314\, J_{prop}^4 + 1.08\, J_{prop}^3 - 0.896\, J_{prop}^2 + 0.1089\, J_{prop} + 0.0604 \tag{2.46}$$

Using the definition of the thrust coefficient, and the assumption that the thrust is aligned with the body axes, the thrust force is then found by

$$\mathbf{F_{thrust}} = C_T\, \rho\, n^2\, D^4\, \hat{i} \tag{2.47}$$

The three contributors to the total thrust moment are the engine torque, the gyroscopic effects associated with the rotating propeller and engine, and the moment due to the position of the thrust line relative to the CG. In equation form, this is expressed as

$$\mathbf{M_{thrust}} = \mathbf{M_{engine}} + \mathbf{M_{rotor}} + \mathbf{M_{thrust\,line}} \tag{2.48}$$

The moment from the engine is found by dividing the engine power by the current motor angular velocity. The engine power output is based on throttle position and is determined using a scaled version of the model presented in [73] so that the peak power output (when $\delta_T = 1$) is 1,063 ft-lb/s. The engine power is given by

$$P = 772.05\, \delta_T^2 + 310.37\, \delta_T - 19.43 \tag{2.49}$$

and the torque from the engine in the body frame is then

$$\mathbf{M_{engine}} = \frac{P}{2\,\pi\,n}\,\hat{i} \tag{2.50}$$

The rotor effects of the propulsion system are found by separating the effects from the engine and propeller. Starting with Eq. 2.11 repeated as

$$\sum \mathbf{M} = \dot{\mathbf{h}}$$

and writing the angular momentum as the product of the moment of inertia and the angular velocity of the combined motor and propeller gives

$$\mathbf{M_{engine}} + \mathbf{M_{prop}} = I_{mp}\,\dot{\boldsymbol{\omega}}_{\mathbf{engine}} \tag{2.51}$$

where

$$I_{mp} = 3.093 \times 10^{-4}\,\text{slug-ft}^2 \tag{2.52}$$

The propeller moment is determined by dividing the power produced by the propeller by the current motor angular velocity. The power produced by the propeller is determined using the power coefficient model given in Eq. 2.53.

$$C_P = 0.5054\,J_{prop}^4 - 0.5304\,J_{prop}^3 + 0.0412\,J_{prop}^2 + 0.0166\,J_{prop} + 0.0223 \tag{2.53}$$

Using the definition of the power coefficient, the moment produced by the propeller in the body frame is then found by

$$\mathbf{M_{prop}} = -\frac{C_P\,\rho\,n^2\,D^5}{2\,\pi}\,\hat{i} \tag{2.54}$$

Combining the moments from both the engine and propeller allows the calculation of the current motor speed using Eq. 2.51 as

$$\boldsymbol{\omega}_{\mathbf{engine}} = \int \frac{\mathrm{M}_{\mathrm{engine}} + \mathrm{M}_{\mathrm{prop}}}{I_{mp}}\, dt = 2\,\pi\, n\, \hat{i} \tag{2.55}$$

where the initial condition for the integrator is based on the trim condition at the start of the simulation. After accounting for taking the angular momentum derivative in the body frame, the gyroscopic moment becomes

$$\mathbf{M}_{\mathbf{rotor}} = I_{mp}\, \dot{\boldsymbol{\omega}}_{\mathbf{engine}} + \boldsymbol{\omega} \times\ I_{mp}\, \boldsymbol{\omega}_{\mathbf{engine}} \tag{2.56}$$

which, after substitution of Eq. 2.51, can be rewritten as

$$\mathbf{M}_{\mathbf{rotor}} = \mathbf{M}_{\mathbf{engine}} + \mathbf{M}_{\mathbf{prop}} + \boldsymbol{\omega} \times\ I_{mp}\, \boldsymbol{\omega}_{\mathbf{engine}} \tag{2.57}$$

where $\boldsymbol{\omega}$ is from Eq. 2.4.

Finally, the moment caused by the thrust line being offset from the CG position is found using the thrust from Eq. 2.47 as well as the CG and propeller position vectors.

$$\mathbf{M}_{\mathbf{thrust\,line}} = \mathbf{r}_{\mathbf{prop/CG}} \times \mathbf{F}_{\mathbf{thrust}} \tag{2.58}$$

The total thrust moment can then be written as

$$\mathbf{M}_{\mathbf{thrust}} = 2\,\mathbf{M}_{\mathbf{engine}} + \mathbf{M}_{\mathbf{prop}} + \boldsymbol{\omega} \times\ I_{mp}\, \boldsymbol{\omega}_{\mathbf{engine}} + \mathbf{r}_{\mathbf{prop/CG}} \times \mathbf{F}_{\mathbf{thrust}} \tag{2.59}$$

2.2.3 Vortex Lattice Modeling

Vortex lattice methods were first formulated in the late 1930's and are based on solutions to Laplace's Equation [45]. A vortex lattice potential flow method models the lifting surfaces of an aircraft as infinitely thin lattices of horseshoe vortices to estimate the aerodynamic

forces and moments [74]. The main limitations, however, are the inability of the method to estimate viscous drag or model nonlifting surfaces (e.g., fuselage and appendages). Vortex lattice modeling is used in this work to determine if the aerodynamic data from the method is accurate enough to use in the development of an autopilot for aerobatic maneuvering. AVL was chosen to implement the method due to the code's ability to quickly estimate the stability derivatives of the test aircraft by combining a full linearization of the aerodynamic model about each flight state with the specified mass properties [51]. The code is also computationally efficient, user friendly, and open source[†] making it readily available.

Since vortex lattice solutions only provide an estimate of the lift-induced drag, a parasite drag breakdown methodology from [72] was used in addition to the induced drag calculated by the vortex lattice method to determine the overall drag of the aircraft. The drag breakdown estimates the parasite drag of each aircraft component using a flat-plate skin-friction coefficient as well as a "Form Factor" that estimates the pressure drag due to viscous separation. An interference factor is included to model the drag increase due to interference between components. The parasite drag for each component was estimated as

$$C_{D_{0c}} = \frac{C_{f_c} S_{wet_c} Q_c F_c}{S} \tag{2.60}$$

Assuming turbulent flow, the skin friction coefficient was calculated from

$$C_{f_c} = \frac{0.455}{(\log Re)^{2.58}} \tag{2.61}$$

where the Reynolds number is defined as

$$Re = \frac{\rho V l}{\mu} \tag{2.62}$$

[†]`http://web.mit.edu/drela/Public/web/avl/`

The interference factor for the wing, tail, and fuselage was estimated as 1.3 based on information in [72]. The form factor for the wing and tail surfaces was given by

$$F_c = \left(1.34\, M^{0.18}\right)\left(100\,(t/c)_c^4 + 2\,(t/c)_c + 1\right) \tag{2.63}$$

with the Mach number defined as

$$M = V/a \tag{2.64}$$

The thickness ratios "t/c" used for the wing and tail surfaces were 0.12 and 0.03 respectively. The form factor for the fuselage was calculated from

$$F_{\text{fuse}} = 1.4\left(1 + \frac{60}{f^3} + \frac{f}{400}\right) \tag{2.65}$$

where the fineness ratio is based on the fuselage length and diameter as

$$f = l/d \tag{2.66}$$

Landing-gear drag was estimated as a summation of the drags of the wheels and struts based on their respective frontal areas with a 20 percent increase to account for interference between the components.

$$C_{D_{0\text{gear}}} = 1.2\,(C_{D_{0\text{wheels}}} + C_{D_{0\text{struts}}}) \tag{2.67}$$

with

$$C_{D_{0\text{wheels}}} = 0.25\, A_{\text{wheels}}/S \tag{2.68}$$

$$C_{D_{0\text{struts}}} = 1.4\, A_{\text{struts}}/S \tag{2.69}$$

The total parasite drag was estimated from Eq. 2.70 as 0.034 and Table 2.4 lists the estimated parasite drag coefficients for each component. The estimate is based on the summation of the drag contribution from each component with a 25 percent increase added to account for miscellaneous drag, leakage drag, and protuberance drag.

$$C_{D_0} = 1.25\left(C_{D_{0\text{wing}}} + C_{D_{0\text{vt}}} + C_{D_{0\text{ht}}} + C_{D_{0\text{fuse}}} + C_{D_{0\text{gear}}}\right) \tag{2.70}$$

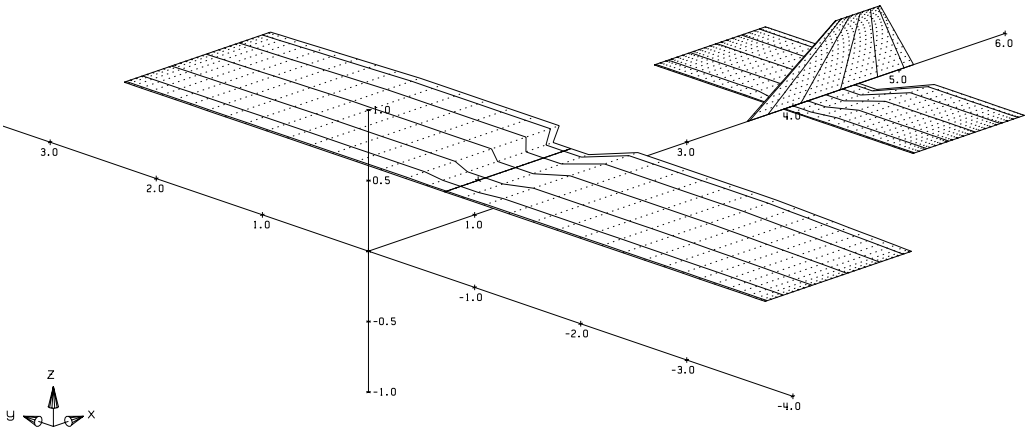
Table 2.4: Parasite drag breakdown.

Component	Parasite Drag Coefficient
Wing	0.0066
Vertical Tail	0.0004
Horizontal Tail	0.002
Fuselage	0.0058
Landing Gear	0.0122
Leakage and Protuberance	0.00675
Total	0.034

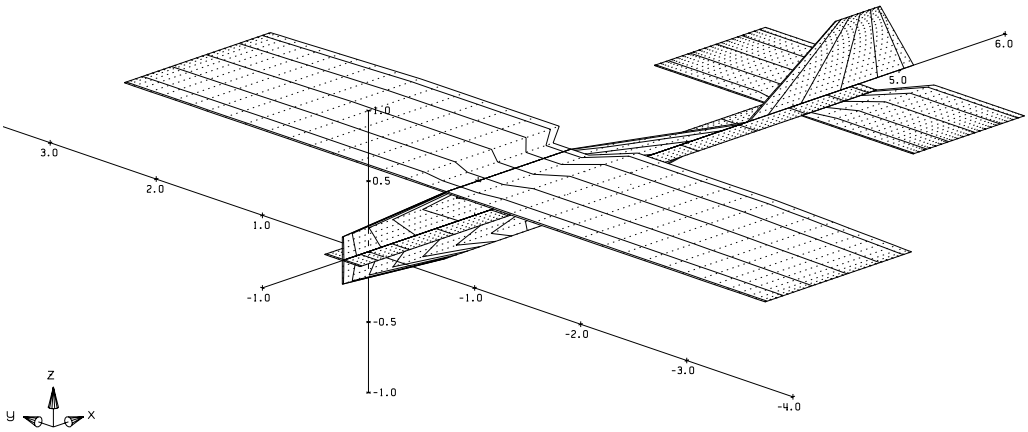
Since vortex lattice methods represent lifting surfaces as infinitely thin lattices of horseshoe vortices, the method is oriented toward combinations of thin lifting surfaces [45]. AVL has the capability to model bodies with a source+doublet line along its axis in accordance with slender-body theory [51], but tests in [75] revealed that the approach over predicted the lift-curve slope. Due to the lack of modeling ability in regards to propeller and fuselage aerodynamic effects with vortex lattice methods, an effort was made to improve the model used in this research by modeling both components as combinations of horizontal and vertical lifting surfaces. The modeling approach was inspired by previous work from [75], which proved the viability of modeling the fuselage effects on lift and pitching moment by using a horizontal lifting surface with the same cross-sectional area as the fuselage in the X-Y plane. Figure 2.3 depicts the multiple models used to determine the best method for modeling the

aerodynamic effects of the fuselage and the propeller. The level of accuracy when using the fuselage model, adapted from [75], and modeling the destabilizing effect of the propeller was examined and compared to the wind tunnel data discussed in Sec. 2.2.1. The conventional method of vortex lattice modeling, which only includes the wing and tail, was also examined.

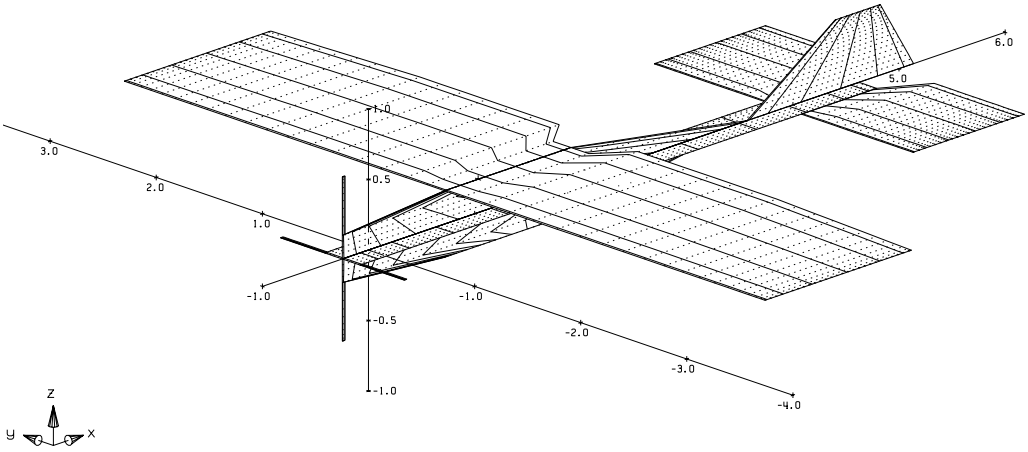
Six horseshoe vortices in a cosine distribution in the chordwise direction and twenty horseshoe vortices in a negative sine distribution in the spanwise direction were used to model the main wing and tail surfaces. The cosine chordwise distribution places the horseshoe vortex nodes closer together at the leading and trailing edges of the surface while the negative sine distribution places the nodes closer together at the wingtips. Using the negative sine distribution for the semispan is equivalent to a cosine distribution across the whole span. The particular distribution was chosen based on recommendations in [51] as well as success with previous work in [75]. The fuselage was approximated by a rectangular lifting surface in the horizontal direction and a tapered lifting surface in the vertical direction. For the horizontal surface, the cross-sectional area was matched to the actual fuselage area. To better represent the chordwise distribution of fuselage side area, a tapered vertical surface was used that closely matches the outer mold line of the actual fuselage. Both fuselage surfaces were modeled with twelve chordwise and five spanwise vortices. The chordwise vortices were in a cosine distribution and the spanwise vortices were equally spaced.



(a) Wing and tail only



(b) Including the fuselage



(c) Including the fuselage and propeller

Figure 2.3: Test aircraft as modeled in AVL.

The propeller was represented by vertical and horizontal surfaces, where the areas were based on actual prop size. The horizontal surface was used to approximate the destabilizing pitching moment due to the propeller normal force. This force is produced by the momentum change caused by the turning of the airstream and acts perpendicular to the thrust axis in the plane of the propeller disk as discussed in [72, 76]. The vertical surface was used to approximate the yawing moment caused by the propeller when the propeller disk is at an angle to the free stream flow. The blade moving downward has a higher angle of attack than the upward moving blade resulting in more thrust being produced by the blade advancing into the relative flow [72]. The difference in thrust creates a moment that tends to yaw the nose of the aircraft to the left for a clockwise propeller rotation [72]. The span of the surfaces was set based on the propeller diameter and the chord was set based on the actual chord of the propeller. Two chordwise vortices and ten spanwise vortices were used for the propeller surfaces with the chordwise in a cosine distribution and the spanwise in a negative sine distribution.

For all of the AVL runs in this comparison, the airspeed was set at 80 ft/s and the density was set at the standard atmospheric value for an altitude of 2,700 feet. Since vortex lattice methods are based on inviscid flow theory, which cannot model flow separation, the angles of attack were limited to the range from -3 to 10 degrees during the AVL runs in an attempt to avoid the highly nonlinear stall regime. Two angles of attack (two and seven degrees) were used for each sideslip-angle sweep to see if the data trends were similar with changes in angle of attack. Only the induced drag is shown in the data plots as this comparison effort is not to prove the accuracy of the overall drag estimate, but to show trends predicted by AVL when the fuselage and propeller models are included.

Figures 2.4 and 2.5 show how the addition of the fuselage surfaces in AVL compare to the wind tunnel data. Looking at Fig. 2.4 for the force coefficient data, it is observed that the fuselage model has a negligible effect on the induced drag. The slopes of both sideforce coefficient plots, however, suggest that the fuselage model helps to more accurately estimate

the changes in sideforce with changes in sideslip angle. The slope of the lift coefficient curve also improves when the fuselage model is added.

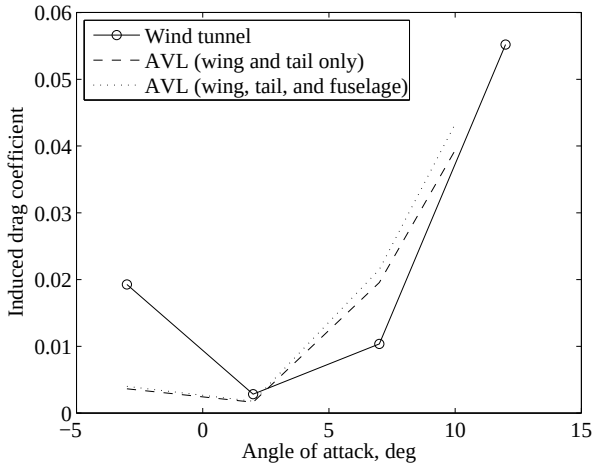
Examination of Fig. 2.5, for the moment coefficient data, reveals that the fuselage model has little impact on the rolling moment. The slope of the pitching moment curve changes dramatically with the addition of the fuselage model and, for angles of attack greater than five degrees, closely resembles the slope of the wind tunnel data. The slope of the yawing moment curve increases when the fuselage model is included. With only the wing and tail, the pitching moment slope only matches for sideslip angles from -5 to 5 degrees. Once the fuselage is included, the slope matches that of the wind tunnel data from -10 to 10 degrees of sideslip when the angle of attack is seven degrees. Tables 2.5 and 2.6 summarize the fuselage model comparison by listing the longitudinal and lateral force and moment derivatives based on the AVL data as well as the linearly approximated wind tunnel data.

Table 2.5: Longitudinal derivatives when using the fuselage model in AVL.

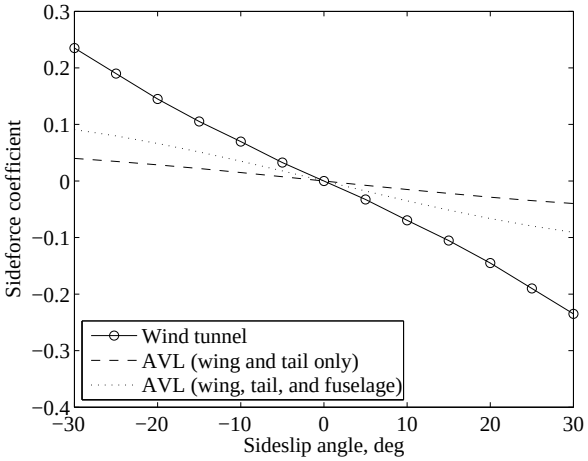
	Linearized Tunnel Data	Wing and Tail Only in AVL	Including the Fuselage in AVL
C_{L_α}, /deg	0.0764	0.0732	0.0769
C_{m_α}, /deg	-0.0080	-0.0065	-0.0129

Table 2.6: Lateral derivatives when using the fuselage model in AVL.

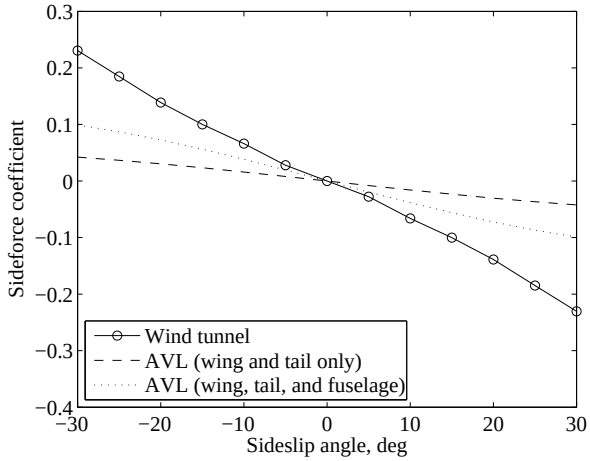
	Linearized Tunnel Data		Wing and Tail Only in AVL		Including the Fuselage in AVL	
$\alpha =$	2°	7°	2°	7°	2°	7°
C_{Y_β}, /deg	-0.0075	-0.0073	-0.0014	-0.0015	-0.0032	-0.0035
C_{l_β}, /deg	-0.0008	-0.0012	-0.0004	-0.0013	-0.0004	-0.0014
C_{n_β}, /deg	0.0013	0.0012	0.0004	0.0005	0.0006	0.0007



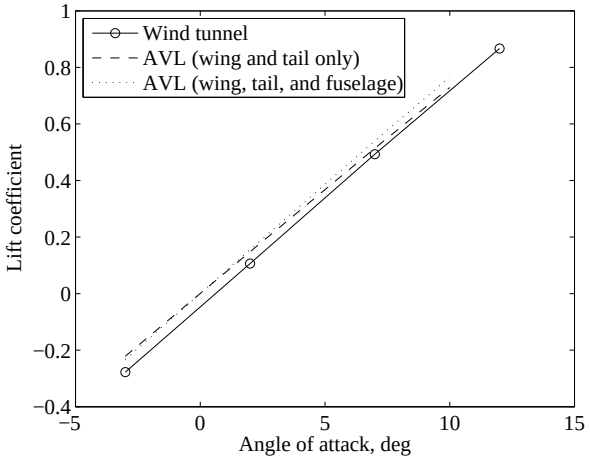
(a) No sideslip



(b) 2° Angle of attack

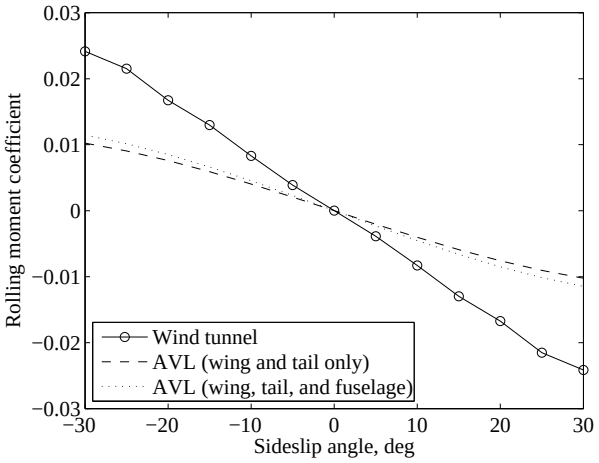


(c) 7° Angle of attack

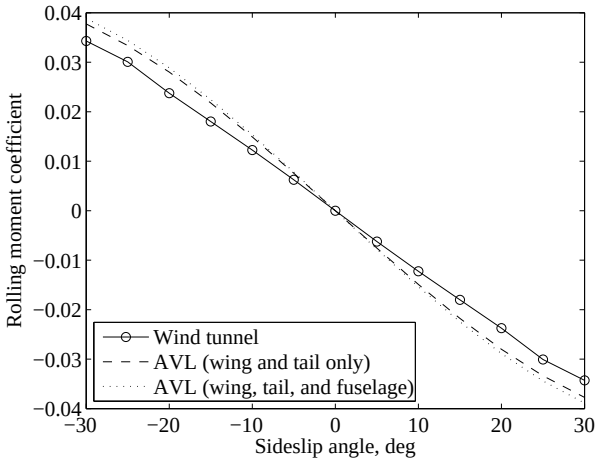


(d) No sideslip

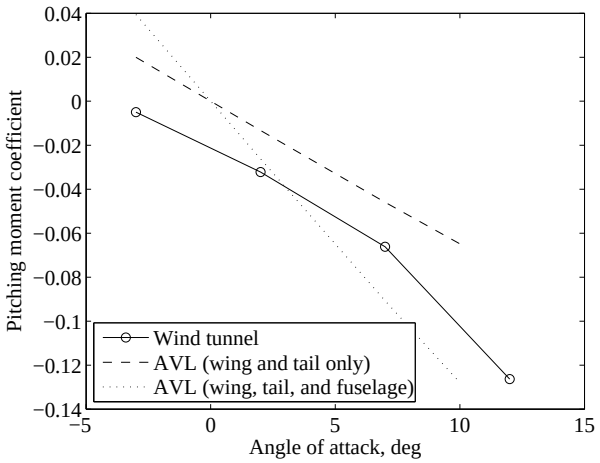
Figure 2.4: Comparison of wind tunnel data and AVL data using a fuselage model for induced drag, sideforce, and lift.



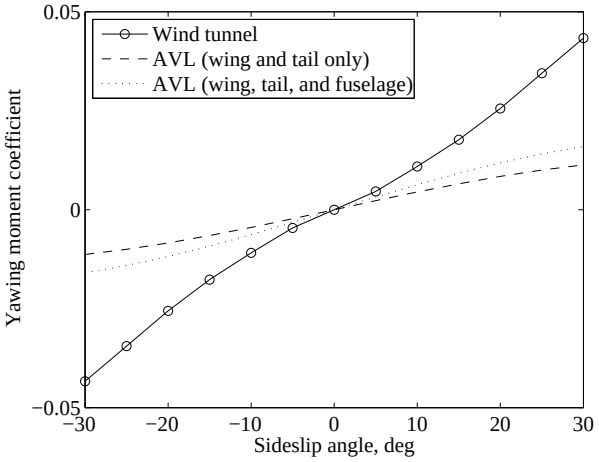
(a) 2° Angle of attack



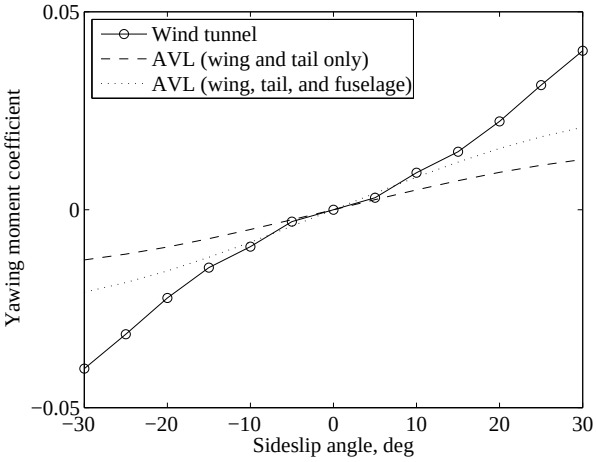
(b) 7° Angle of attack



(c) No sideslip



(d) 2° Angle of attack



(e) 7° Angle of attack

Figure 2.5: Comparison of wind tunnel data and AVL data using a fuselage model for aerodynamic moments.

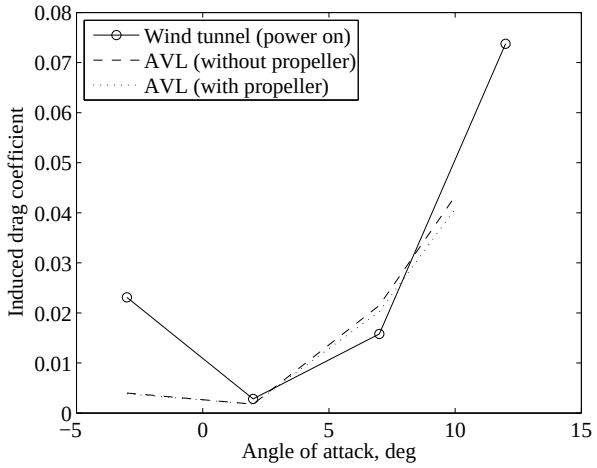
Figures 2.6 and 2.7 show the three-way comparison between the wind tunnel data including propulsion effects, the AVL model including the fuselage, and the AVL model including both the fuselage and propeller. The force coefficient data does not depict much variation between when the propeller is modeled and when it is not. The moment coefficient data in Fig. 2.7, however, shows that the pitching moment slope decreases to more closely match that of the wind tunnel data. The rolling and yawing moment slopes do not change very much when the propeller is included. Tables 2.7 and 2.8 summarize the propeller model comparison by listing the longitudinal and lateral force and moment derivatives based on the AVL data with the fuselage and propeller models as well as the linearly approximated wind tunnel data including power effects.

Table 2.7: Longitudinal derivatives when using the fuselage and propeller in AVL.

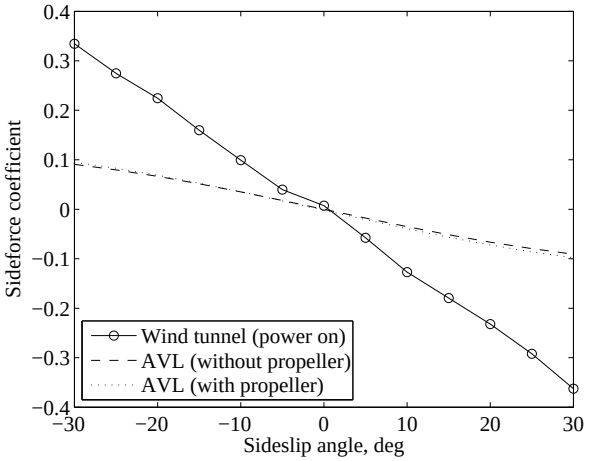
	Linearized Tunnel Data	Wing, Tail, and Fuselage in AVL	Wing, Tail, Fuselage, and Propeller in AVL
C_{L_α}, /deg	0.0891	0.0769	0.0652
C_{m_α}, /deg	-0.0037	-0.0129	-0.0059

Table 2.8: Lateral derivatives when using the fuselage and propeller in AVL.

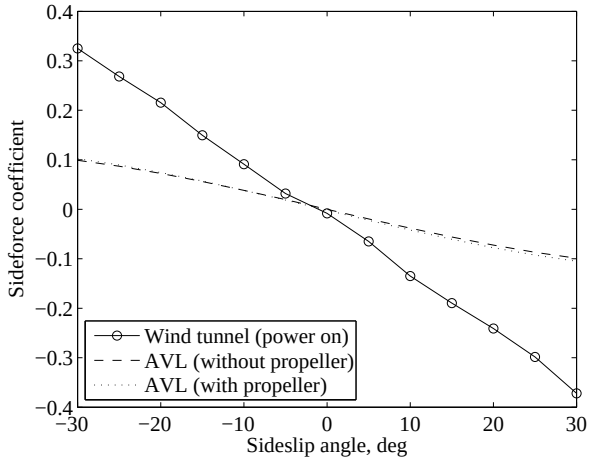
	Linearized Tunnel Data		Wing, Tail, and Fuselage in AVL		Wing, Tail, Fuselage, and Propeller in AVL	
$\alpha =$	2^o	7^o	2^o	7^o	2^o	7^o
C_{Y_β}, /deg	-0.0114	-0.0114	-0.0032	-0.0035	-0.0034	-0.0036
C_{l_β}, /deg	-0.0009	-0.0009	-0.0004	-0.0014	-0.0004	-0.0013
C_{n_β}, /deg	0.0011	0.0011	0.0006	0.0007	0.0004	0.0006



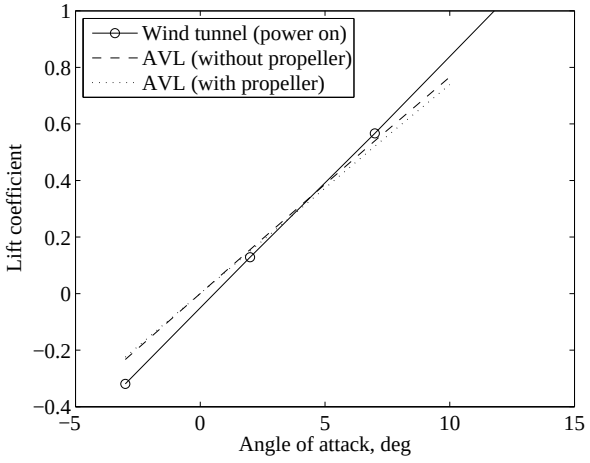
(a) No sideslip



(b) 2° Angle of attack

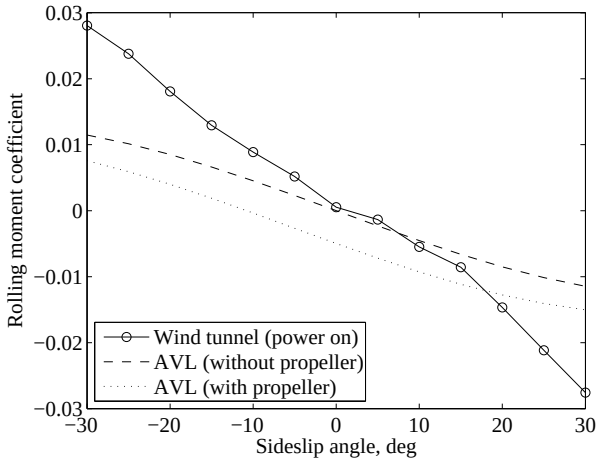


(c) 7° Angle of attack

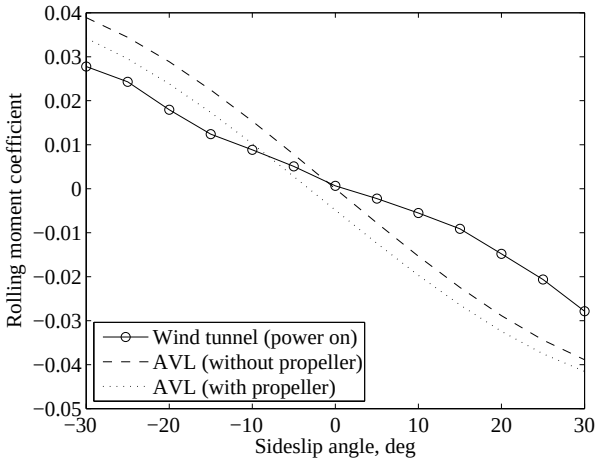


(d) No sideslip

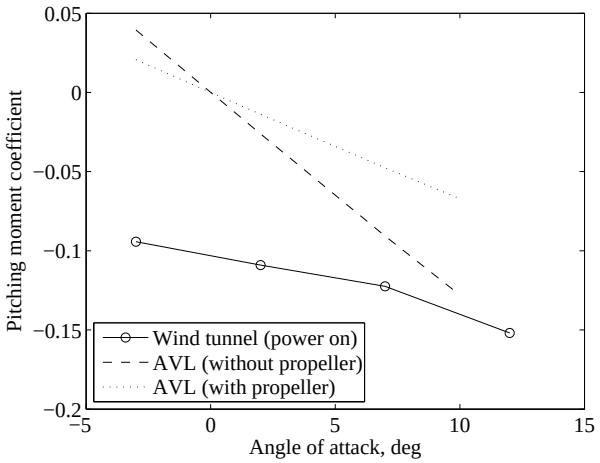
Figure 2.6: Comparison of wind tunnel data and AVL data using fuselage and propeller models for induced drag, sideforce, and lift.



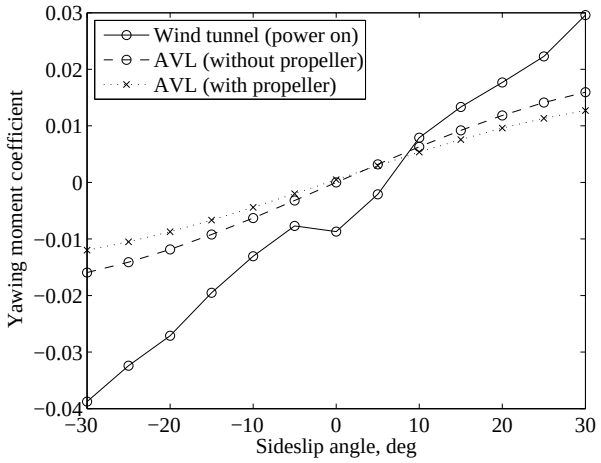
(a) 2° Angle of attack



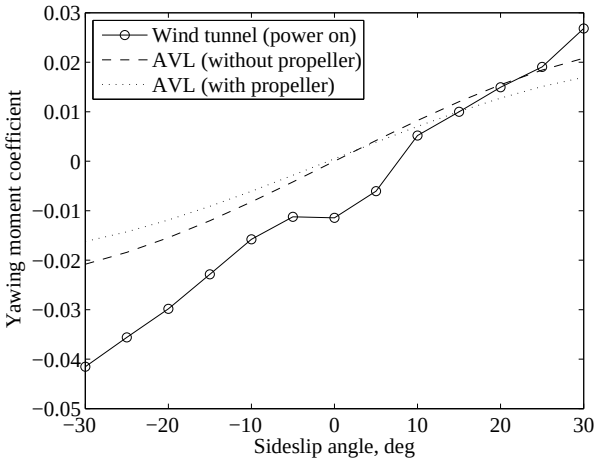
(b) 7° Angle of attack



(c) No sideslip



(d) 2° Angle of attack



(e) 7° Angle of attack

Figure 2.7: Comparison of wind tunnel data and AVL data using fuselage and propeller models for aerodynamic moments.

Based on the results from the comparison, it was decided to use both the fuselage model and the propeller model in AVL. The fuselage model is included to more closely approximate the lift-curve slope as well as the yaw stiffness. The propeller model is included to improve the estimate of the pitch stiffness. The lift-curve slope and the pitch stiffness are considered to be the two most important longitudinal stability derivatives to represent the longitudinal aircraft motion [21]. The lift-curve slope is important because it determines how changes in angle of attack modify the lift produced. It affects the damping of the pitching motion due to disturbances as well as the maneuverability. The pitch stiffness determines static stability and also affects the dynamic pitching motion. The yaw stiffness is associated with the tendency of the aircraft to head into the relative wind (the weathercock stability) and plays a large role in the aircraft dutch roll mode. Since a goal of using AVL in this research is to determine if a control system can be developed based purely on vortex lattice data, the effort is made to try to model the dynamic behavior of the airplane as closely as possible.

Appendix A gives the AVL input file and mass file used for control development in this work. Both the fuselage and propeller are included. The control surface effectiveness was reduced in the AVL model through the control gain parameter based on data in [77] for control surfaces with open hinge gaps. Data from the USAF DATCOM and Ref. [78] was used in [77] to show that the theoretical lift effectiveness of a deflected plain flap cannot be achieved in practice. The reduction occurs because the effectiveness of the control surface is sensitive to the boundary layer at the hinge and viscous effects lead to flow separation when deflection angles are greater than 10 degrees [77].

Using the vortex lattice model developed in this section, the stability derivatives for cruise were estimated assuming level, trimmed flight at an altitude of 2,700 feet and an airspeed of 80 ft/s. The steady-state angle of attack was calculated as 3.9 degrees with a lift coefficient of 0.281 and a drag coefficient of 0.04. Table 2.9 lists the stability-axis derivatives as calculated by AVL for cruise flight which will be used in Sec. 2.4.1 to determine linear state-space aircraft models.

Table 2.9: Test aircraft stability derivatives from AVL in cruise flight.

α, /rad		β, /rad	
C_{L_α}	4.270	C_{L_β}	0.0015
C_{Y_α}	0.001	C_{Y_β}	-0.224
C_{l_α}	0.002	C_{l_β}	-0.046
C_{m_α}	-0.393	C_{m_β}	0.003
C_{n_α}	0.005	C_{n_β}	0.035

p		q		r	
C_{L_p}	-0.001	C_{L_q}	8.774	C_{L_r}	-0.024
C_{Y_p}	0.100	C_{Y_q}	-0.010	C_{Y_r}	0.137
C_{l_p}	-0.347	C_{l_q}	0.001	C_{l_r}	0.054
C_{m_p}	0.001	C_{m_q}	-9.075	C_{m_r}	0.008
C_{n_p}	-0.031	C_{n_q}	0.006	C_{n_r}	-0.069

δ_e, /deg		δ_a, /deg		δ_r, /deg	
$C_{L_{\delta_e}}$	0.005	$C_{L_{\delta_a}}$	0	$C_{L_{\delta_r}}$	0
$C_{Y_{\delta_e}}$	0	$C_{Y_{\delta_a}}$	0	$C_{Y_{\delta_r}}$	0.001
$C_{l_{\delta_e}}$	0	$C_{l_{\delta_a}}$	-0.002	$C_{l_{\delta_r}}$	0
$C_{m_{\delta_e}}$	-0.011	$C_{m_{\delta_a}}$	0	$C_{m_{\delta_r}}$	0
$C_{n_{\delta_e}}$	0	$C_{n_{\delta_a}}$	0	$C_{n_{\delta_r}}$	-0.001

2.3 Actuator and Sensor Models

Control actuation is assumed to be performed with commercially available electric servo motors made by Hitec. The engine speed is assumed to be controlled using a commercial electric motor speed control made by Castle Creations. First-order low-pass linear filters were used to model the dynamics from the control surface deflection command to the actual deflection. The time constant for each filter was based on the slew-rate limit of the actuator as well as the linkage ratio between the servo arm and the control surface. The output of the actuator model was limited based on the deflection limits of the surfaces. Table 2.10 lists the control deflection limits and actuator slew rates for each control surface.

Table 2.10: Control actuator specifications.

Control	Minimum Deflection Limit	Maximum Deflection Limit	Actuator	Slew Rate, deg/s (@ 6.0 V)
Elevator	-25°	25°	HS5625MG	428.6
Aileron	-25°	25°	HS5625MG	428.6
Rudder	-30°	30°	HS5625MG	428.6
Throttle	0	1	Phoenix Edge HV40	7200

The actuators were modeled as simple lags because of the similarity of the actual servo's performance to that of the low-pass filter. An ideal low-pass filter passes all signals without attenuation up to a frequency commonly referred to as the cutoff frequency. A real low-pass filter, however, has some attenuation leading up to the cutoff frequency and introduces phase lag as well. This is similar to the behavior of a servo in that there is a time delay between when a command input is received and the output is achieved. With constant amplitude inputs given at increasing frequencies, the amplitude of the servo deflection will be attenuated as the slew-rate limit of the servo is reached. Figure 2.8 shows the Bode plots for a simple-lag filter with the cutoff frequency shown as a dotted vertical line. The magnitude plot shows that as the cutoff frequency is reached, the signal is attenuated and results in a 20 dB/decade reduction at high frequency. The phase plot asymptotically approaches zero degrees at low

frequencies and 90 degrees at high frequencies, and passes through 45 degrees at the cutoff frequency. The simple-lag filter is used in this work to model the delay in the actuators between input command and output.

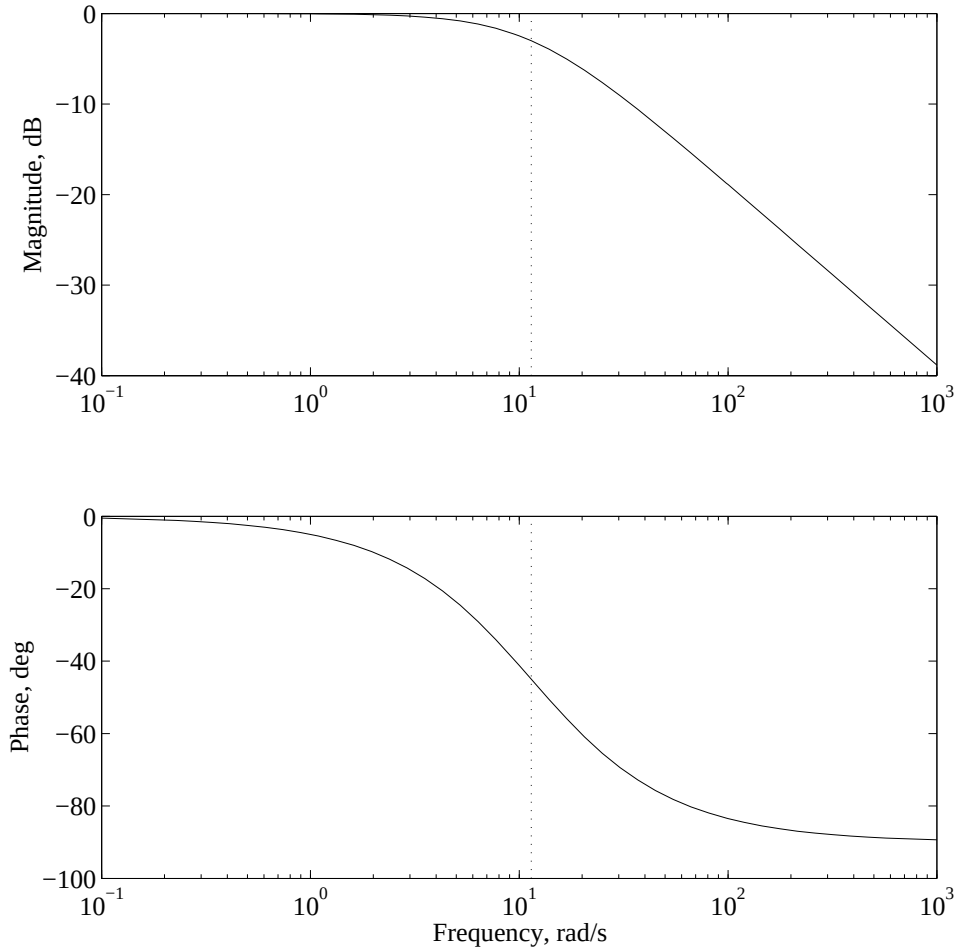


Figure 2.8: Bode plots for a simple lag.

The state equations for a simple-lag filter are

$$\dot{x} = \frac{u - x}{\tau} \tag{2.71}$$

$$y = x \tag{2.72}$$

where τ is the time constant expressed in terms of the cutoff frequency as

$$\tau = 1/\omega_c \tag{2.73}$$

For frequencies less than the cutoff frequency, the sinusoidal deflection of a servo arm can be described by Eqs. 2.74 and 2.75.

$$\delta = A \sin(\omega t) \tag{2.74}$$

$$\dot{\delta} = A\,\omega \cos(\omega t) \tag{2.75}$$

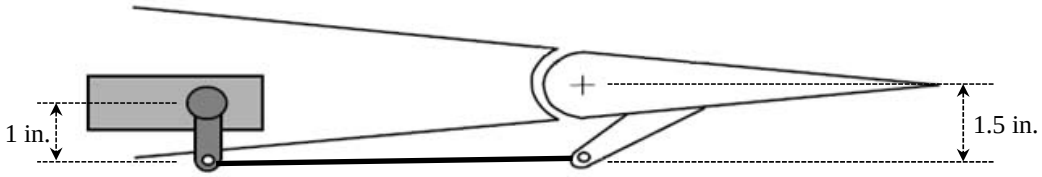
As the cutoff frequency is reached, the sinusoidal motion of the servo arm will be distorted. Since the maximum of $\dot{\delta}$ is limited, Park [13] explains that the cutoff frequency can be found from Eq. 2.75 as

$$\omega_c = \frac{\dot{\delta}_{\text{limit}}}{A} \tag{2.76}$$

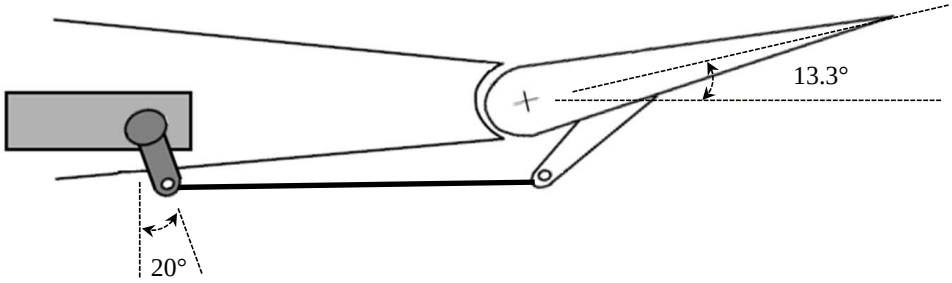
with the amplitude of the servo arm given by the product of the linkage ratio and the deflection limit.

$$A = \frac{\delta_{\text{servo}}}{\delta_{\text{surface}}}\,\delta_{\text{limit}} \tag{2.77}$$

Figure 2.9 shows how the linkage ratio is determined by the length of the servo arm and the length of the control arm. When the servo arm rotates through 20 degrees, assuming the servo arm is one inch in length and the control arm is 1.5 inches, the control surface will deflect 13 degrees. Table 2.11 lists the linkage ratios and corresponding critical frequencies for each actuator. The servo amplitude for the rudder is greater than for the aileron and elevator because the rudder has a larger deflection limit. Figure 2.10 shows the simulation diagram of the actuator based on the cutoff frequency, the rate limit, and the deflection limits.



(a) No servo deflection



(b) 20° servo deflection

Figure 2.9: Servo linkage geometry.

Table 2.11: Actuator linkage ratios and cutoff frequencies.

Control	Linkage Ratio	Servo Amplitude	ω_c, rad/s
Elevator	1.5	37.5°	11.43
Aileron	1.5	37.5°	11.43
Rudder	1.5	45°	9.52
Throttle	1	1	20

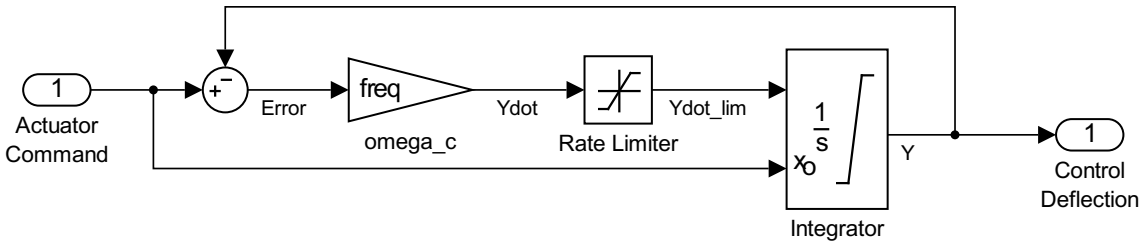


Figure 2.10: Simulation diagram of the actuator model.

The sensors are assumed to be commercial off-the-shelf components to enable cost and time-effective future development. Sensors are included in the simulation to determine the effects of having imperfect state measurements. For this work, the addition of measurement noise is a theoretical experiment and so the dynamics of the sensors and any associated delays in the data were not modeled. Airspeed, altitude, angle of attack, and sideslip angle are assumed to come from a multi-port probe similar to the one in Fig. 2.11. Accelerations, orientation angles, and angular rates are assumed to come from a commercially available inertial measurement unit (IMU) and Table 2.12 lists the specific sensors that make up the device. Zero-mean, Gaussian pseudo-random noise was added to the aircraft states based on the manufacturer's published data for the assumed equipment.[‡] Table 2.13 lists the standard deviations used to create the measurement noise.

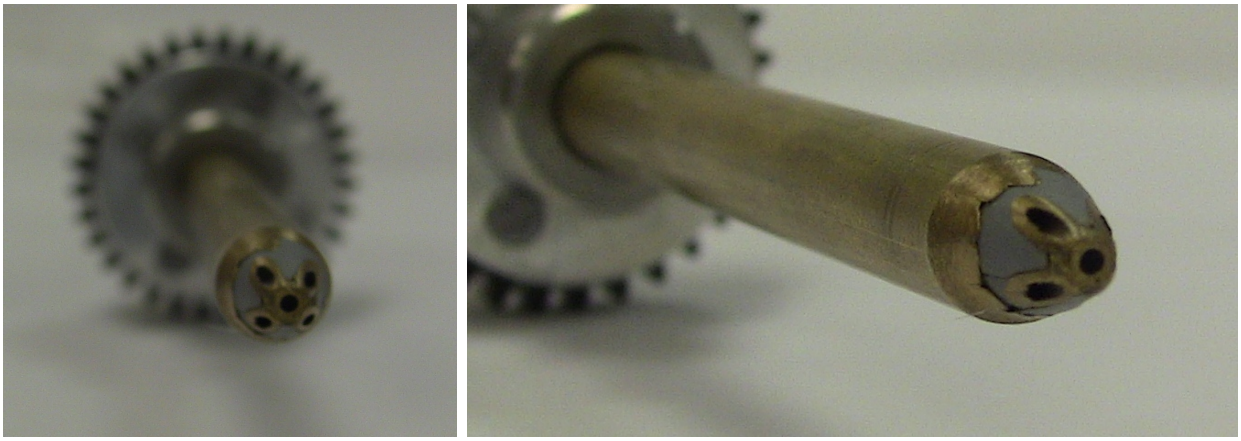


Figure 2.11: Multi-port probe for airspeed, altitude, and flow angle measurement.

Table 2.12: Sensors incorporated in the IMU.

Sensor	Make	Model
Triple-Axis Gyroscope	InvenSense	ITG-3200
Triple-Axis Accelerometer	Analog Devices	ADXL345
Triple-Axis Magnetometer	Honeywell	HMC5883L

[‡]The multi-port probe data is based on a five-hole probe manufactured by Aeroprobe Corporation. More information can be found at http://www.aeroprobe.com/uploads/Conventional.pdf. More information regarding the Razor IMU can be found at https://www.sparkfun.com/products/10736.

Table 2.13: Measurement noise standard deviations from manufacturer's data.

	Sensor	Measurement Noise Standard Deviation
Airspeed		2 ft/s
Altitude	5-hole probe	4 ft
Flow Angles		1°
Accelerations		0.004 g
Orientation Angles	9-DOF Razor IMU	2°
Angular Rates		0.38 deg/s

2.4 Linear Time-Invariant Systems

In order to use the linear quadratic control approach, a linear state-space model of the aircraft was needed. Two methods for determining the linear models were used. The first method involved linearizing the EoM presented in Sec. 2.1 around a steady-state unaccelerated flight condition. Taylor expansions of the aerodynamic and thrust force and moment coefficients were used along with small-perturbation theory to determine the state and control matrices of the aircraft. The second approach used a numerical linearization of the nonlinear simulation, which is based on the nonlinear wind tunnel data, around a defined trim condition.

2.4.1 Linear Equations of Motion

The EoM can be linearized by first using small-perturbation theory to rewrite the variables in the equations as the sum of a reference value (denoted with a bar accent) and a perturbation value. The small-perturbation assumption implies that the squares and products of the perturbations are negligible compared to first-order quantities. Additionally, for perturbations of the Euler angles

$$\cos(\phi) \approx 1, \quad \cos(\theta) \approx 1, \quad \cos(\psi) \approx 1$$

$$\sin(\phi) \approx \phi, \quad \sin(\theta) \approx \theta, \quad \sin(\psi) \approx \psi$$

The equations are then separated into a reference set (which define the trim condition when all perturbations are zero) and a perturbation set (which are linear in the perturbation variables). The perturbation equations are then used to determine the state and control matrix after substitution of the Taylor expansions of both the aerodynamic force and moment coefficients and the thrust force and moment coefficients. Equations 2.78 - 2.83 give the expansions of the aerodynamic coefficients [21, 79, 80].

$$C_D = \bar{C}_D \tag{2.78}$$

$$\begin{aligned} C_Y = \bar{C}_Y &+ C_{Y_\beta}\left(\beta - \bar{\beta}\right) + \frac{b}{2\bar{V}} C_{Y_p}\left(p - \bar{p}\right) \\ &+ \frac{b}{2\bar{V}} C_{Y_r}\left(r - \bar{r}\right) + C_{Y_{\delta_r}}\delta_r + C_{Y_{\delta_a}}\delta_a \end{aligned} \tag{2.79}$$

$$C_L = \bar{C}_L + C_{L_\alpha}\left(\alpha - \bar{\alpha}\right) + \frac{c}{2\bar{V}} C_{L_q}\left(q - \bar{q}\right) + C_{L_{\delta_e}}\delta_e \tag{2.80}$$

$$\begin{aligned} C_l = \bar{C}_l &+ C_{l_\beta}\left(\beta - \bar{\beta}\right) + \frac{b}{2\bar{V}} C_{l_p}\left(p - \bar{p}\right) \\ &+ \frac{b}{2\bar{V}} C_{l_r}\left(r - \bar{r}\right) + C_{l_{\delta_r}}\delta_r + C_{l_{\delta_a}}\delta_a \end{aligned} \tag{2.81}$$

$$C_m = \bar{C}_m + C_{m_\alpha}\left(\alpha - \bar{\alpha}\right) + \frac{c}{2\bar{V}} C_{m_q}\left(q - \bar{q}\right) + C_{m_{\delta_e}}\delta_e \tag{2.82}$$

$$\begin{aligned} C_n = \bar{C}_n &+ C_{n_\beta}\left(\beta - \bar{\beta}\right) + \frac{b}{2\bar{V}} C_{n_p}\left(p - \bar{p}\right) \\ &+ \frac{b}{2\bar{V}} C_{n_r}\left(r - \bar{r}\right) + C_{n_{\delta_r}}\delta_r + C_{n_{\delta_a}}\delta_a \end{aligned} \tag{2.83}$$

The expansions of the thrust coefficients are

$$C_{T_x} = \bar{C}_{T_x} + \frac{1}{\bar{V}} C_{T_{x_u}} \ (u - \bar{u}) \ (1 + \delta_T) \tag{2.84}$$

$$C_{T_y} = \bar{C}_{T_y} \tag{2.85}$$

$$C_{T_z} = \bar{C}_{T_z} \tag{2.86}$$

$$C_{T_l} = \bar{C}_{T_l} \tag{2.87}$$

$$C_{T_m} = \bar{C}_{T_m} \tag{2.88}$$

$$C_{T_n} = \bar{C}_{T_n} \tag{2.89}$$

where the reference values are determined by the steady-state assumption. If the aircraft is in a steady state, by definition, the thrust forces and moments must be balanced by the combination of the aerodynamic forces and moments and the gravity terms so that the linear and angular accelerations are zero. This allows the reference values of the thrust coefficients to be written in terms of the aerodynamic coefficients and the vehicle mass as

$$\bar{C}_{T_x} = \bar{C}_D \ \cos(\bar{\alpha}) - \bar{C}_L \ \sin(\bar{\alpha}) + \frac{m \, g \ \sin\left(\bar{\theta}\right)}{S \, q_\infty} \tag{2.90}$$

$$\bar{C}_{T_y} = -\frac{m \, g \ \cos\left(\bar{\theta}\right) \ \sin\left(\bar{\phi}\right)}{S \, q_\infty} \tag{2.91}$$

$$\bar{C}_{T_z} = \bar{C}_L \ \cos(\bar{\alpha}) + \bar{C}_D \ \sin(\bar{\alpha}) - \frac{m \, g \ \cos\left(\bar{\phi}\right) \ \cos\left(\bar{\theta}\right)}{S \, q_\infty} \tag{2.92}$$

$$\bar{C}_{T_l} = \bar{C}_n \ \sin(\bar{\alpha}) - \bar{C}_l \ \cos(\bar{\alpha}) \tag{2.93}$$

$$\bar{C}_{T_m} = -\bar{C}_m \tag{2.94}$$

$$\bar{C}_{T_n} = -\bar{C}_n \ \cos(\bar{\alpha}) - \bar{C}_l \ \sin(\bar{\alpha}) \tag{2.95}$$

The Taylor expansions of the force and moment coefficients are substituted into the EoM and the equations are rewritten so that all nonzero terms are on the left-hand side.

The equations will then be in the form of

$$\begin{gathered} f_1\ (\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0 \\ \vdots \qquad \vdots \\ f_{10}\ (\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0 \end{gathered}$$

where the equations for position North and East (p_N, p_E) are excluded because they are not needed for control design. The state vector is

$$\mathbf{x} = \begin{bmatrix} \phi & \theta & \psi & p & q & r & u & v & w & p_D \end{bmatrix}^T \tag{2.96}$$

and the control vector is

$$\mathbf{u} = \begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_T \end{bmatrix}^T \tag{2.97}$$

The perturbations in the state derivative, state, and control vectors must satisfy

$$\begin{gathered} \nabla_{\dot{\mathbf{x}}} f_1\, \partial\dot{\mathbf{x}} + \nabla_{\mathbf{x}} f_1\, \partial\mathbf{x} + \nabla_{\mathbf{u}} f_1\, \partial\mathbf{u} = 0 \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \nabla_{\dot{\mathbf{x}}} f_{10}\, \partial\dot{\mathbf{x}} + \nabla_{\mathbf{x}} f_{10}\, \partial\mathbf{x} + \nabla_{\mathbf{u}} f_{10}\, \partial\mathbf{u} = 0 \end{gathered} \tag{2.98}$$

where

$$\nabla_{\mathbf{x}} f_i \equiv \begin{bmatrix} \frac{\partial f_i}{\partial x_1} & \frac{\partial f_i}{\partial x_2} & \cdots & \frac{\partial f_i}{\partial x_{10}} \end{bmatrix}$$

In explicit state-space form, Eqs. 2.98 are written as

$$\dot{\mathbf{x}} = \mathbf{E}^{-1} \left(\widehat{\mathbf{A}}\, \mathbf{x} + \widehat{\mathbf{B}}\, \mathbf{u} \right) \tag{2.99}$$

where

$$\mathbf{E} = -\begin{bmatrix} \nabla_{\dot{\mathbf{x}}} f_1 \\ \vdots \\ \nabla_{\dot{\mathbf{x}}} f_{10} \end{bmatrix} \qquad \widehat{\mathbf{A}} = \begin{bmatrix} \nabla_{\mathbf{x}} f_1 \\ \vdots \\ \nabla_{\mathbf{x}} f_{10} \end{bmatrix} \qquad \widehat{\mathbf{B}} = \begin{bmatrix} \nabla_{\mathbf{u}} f_1 \\ \vdots \\ \nabla_{\mathbf{u}} f_{10} \end{bmatrix} \tag{2.100}$$

For the remainder of this dissertation, Eq. 2.99 will be written as

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} \tag{2.101}$$

where

$$\mathbf{A} = \mathbf{E}^{-1}\,\widehat{\mathbf{A}} \qquad \mathbf{B} = \mathbf{E}^{-1}\,\widehat{\mathbf{B}} \tag{2.102}$$

2.4.2 Estimating the $C_{T_{x_u}}$ Derivative

The derivative $C_{T_{x_u}}$ in Eq. 2.84 is the speed damping derivative and by definition

$$C_{T_{x_u}} = \bar{u}\,\bar{C}_{T_{x_u}} \tag{2.103}$$

Since $C_T = T/\frac{1}{2}\,\rho\,V^2\,S$ and the thrust line is in the X-body direction, the speed damping derivative can be written as

$$\frac{\partial\,C_T}{\partial\,u} = \frac{\partial\,T/\partial\,u}{\frac{1}{2}\,\rho\,V^2\,S} - \frac{2\,T}{\frac{1}{2}\,\rho\,V^3\,S}\,\frac{\partial\,V}{\partial\,u} \tag{2.104}$$

In the unperturbed reference condition (assuming small α), $V = \bar{u}$ and $\partial\,V/\partial\,u = 1$ so that the speed derivative evaluated at the reference condition is

$$\bar{C}_{T_{x_u}} = \frac{\partial\,\bar{T}/\partial\,\bar{u}}{\frac{1}{2}\,\rho\,\bar{u}^2\,S} - \frac{2}{\bar{u}}\,\bar{C}_{T_x} \tag{2.105}$$

with $\bar{C}_{T_x}$ given in Eq. 2.90. Substitution into Eq. 2.103 gives the speed damping derivative as

$$C_{T_{x_u}} = \frac{\partial \bar{T}/\partial \bar{u}}{\frac{1}{2}\rho \bar{u} S} - 2\bar{C}_{T_x} \tag{2.106}$$

The derivative of the thrust with forward speed was determined from the engine model presented in Sec. 2.2.2. Figure 2.12 shows the plot of thrust as a function of speed used to determine that $\partial \bar{T}/\partial \bar{u} = -0.12$ at a trim speed of 80 ft/s.

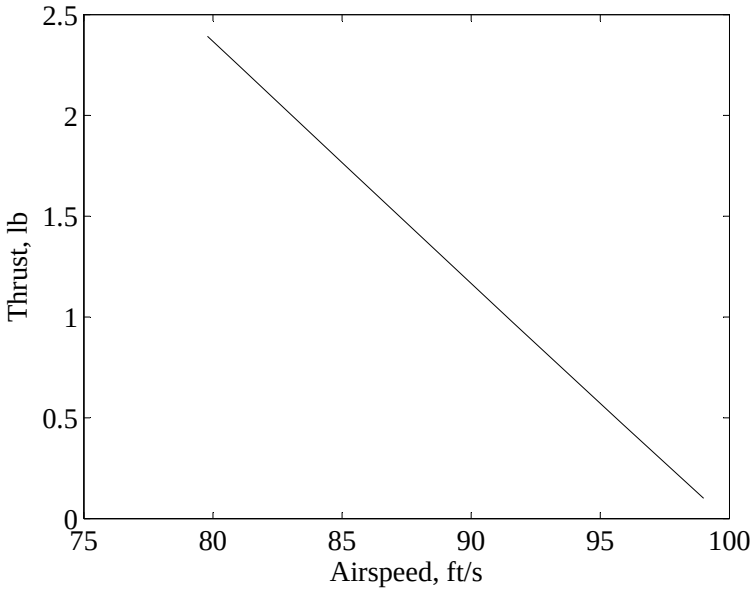


Figure 2.12: Engine thrust as a function of forward speed.

2.4.3 Including the Engine Speed

As described in Sec. 2.2.2 the time-derivative of the engine speed is a function of airspeed, altitude, current engine speed, and throttle position. With this engine model, in order to be able to design a control system that will be able to control the throttle position, the engine speed needs to be included in the state equations. With the engine speed added, the new state vector becomes

$$\mathbf{x} = \begin{bmatrix} \phi & \theta & \psi & p & q & r & u & v & w & p_D & \omega_{\text{engine}} \end{bmatrix}^T \tag{2.107}$$

The engine speed derivative is written as a function of the perturbation states as

$$\dot{\omega}_{\text{engine}} = \frac{\partial \dot{\omega}_{\text{engine}}}{\partial u} u + \frac{\partial \dot{\omega}_{\text{engine}}}{\partial w} w + \frac{\partial \dot{\omega}_{\text{engine}}}{\partial p_D} p_D + \frac{\partial \dot{\omega}_{\text{engine}}}{\partial \omega_{\text{engine}}} \omega_{\text{engine}} + \frac{\partial \dot{\omega}_{\text{engine}}}{\partial \delta_T} \delta_T \qquad (2.108)$$

The effect of the engine speed on the forward speed is also added to the state equations and is given by

$$\dot{u}_{\text{engine}} = \frac{\partial \dot{u}}{\partial \omega_{\text{engine}}} \omega_{\text{engine}} \qquad (2.109)$$

where

$$\frac{\partial \dot{u}}{\partial \omega_{\text{engine}}} = \frac{\partial \dot{u}}{\partial T} \frac{\partial T}{\partial \omega_{\text{engine}}} \approx \frac{1}{m} \frac{\partial T}{\partial \omega_{\text{engine}}}$$

The coefficients in Eq. 2.108 were found by linearizing the engine model around the trim condition and are listed in Table 2.14. Figure 2.13 shows the variations in the engine angular acceleration with deviations in the trim states as well as the variation in $\dot{u}$ with engine speed for the cruise flight condition. From the linear approximation, the forward acceleration coefficient ($\partial \dot{u}/\partial \omega_{\text{engine}}$) was found to be 0.026.

Table 2.14: Engine model coefficients.

$\partial \dot{\omega}_{\text{engine}}/\partial u$	$\partial \dot{\omega}_{\text{engine}}/\partial w$	$\partial \dot{\omega}_{\text{engine}}/\partial p_D$	$\partial \dot{\omega}_{\text{engine}}/\partial \omega_{\text{engine}}$	$\partial \dot{\omega}_{\text{engine}}/\partial \delta_T$
25.99	1.85	-0.033	-5.0	$3.38\text{x}10^3$

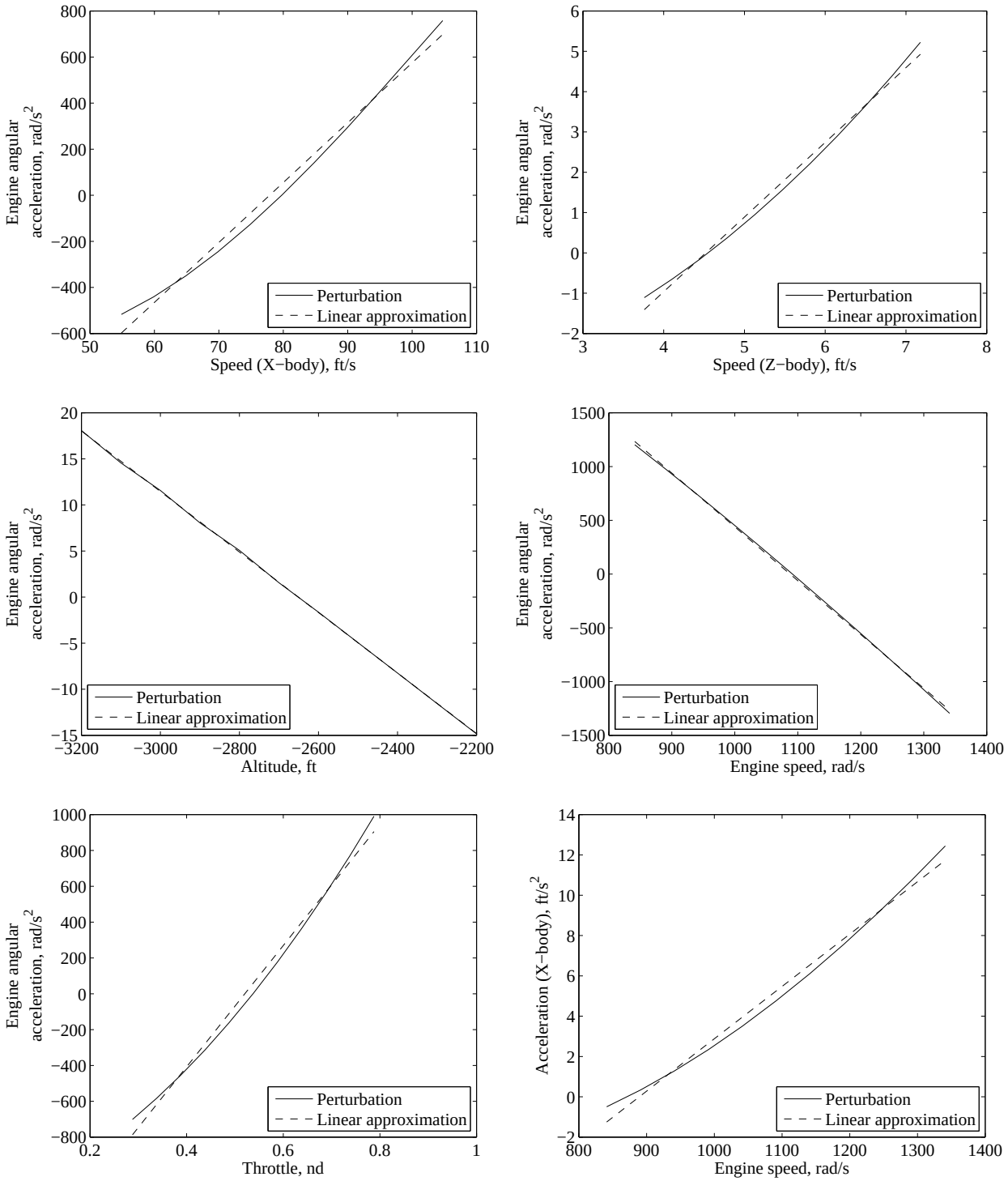


Figure 2.13: Engine perturbations with linear approximations for cruise.

2.4.4 Linearizing the Simulation

The nonlinear simulation is numerically linearized by first specifying a trim condition using MATLAB's `operspec` function. Once the trim condition is specified, the `findop` function is used to determine the inputs required to trim based on specified initial conditions. A linear model around the trim condition is then found using the `linearize` function. For level flight, the flight condition is specified as

$$V = 80\,\text{ft/s}, \quad \beta = 0^\circ, \quad h = 2,700\,\text{ft}, \quad \psi = 330^\circ$$
$$\theta = \alpha, \qquad p = 0, \qquad q = 0, \qquad r = 0$$

The trimmed states are found to be

$$V = 80\,\text{ft/s}, \quad \alpha = 4.13^\circ, \quad \beta \approx 0^\circ$$
$$h = 2,700\,\text{ft}, \quad \psi = 330^\circ, \quad \gamma \approx 0^\circ$$
$$\omega_{\text{engine}} = 1,091\,\text{rad/s}$$

with control inputs of

$$\delta_e = -4.8^\circ, \quad \delta_a = 0.6^\circ$$
$$\delta_r = -3.1^\circ, \quad \delta_T = 0.5$$

2.4.5 Validation and Comparison

Published stability derivatives in [63] for a Cessna 182 were used in conjunction with AVL to validate the linearized EoM. First, the eigenvalues of the state matrix from the EoM, using the published stability derivatives, were compared to the published eigenvalues in [63]. The comparison, shown in Fig. 2.14 with an expanded scale near the origin depicted in Fig. 2.15, showed that the eigenvalues of the state matrix matched the published values.

Table 2.15 lists the published eigenvalues as well as the values calculated by the EoM using the published data. Second, the eigenvalues of the state matrix from the EoM, using stability derivatives from AVL, were compared to the eigenvalues calculated by AVL directly. Figure 2.16 shows the second eigenanalysis comparison and Table 2.16 provides a list of the eigenvalues. Figure 2.17 depicts an expanded scale near the origin of Fig. 2.16 showing the comparison of the spiral and phugoid modes. As can be seen in the figures, the eigenvalues calculated from the EoM match the values calculated by AVL quite well. The validation effort was split into two comparisons because the effort was not to show that the AVL results matched the published values, but to show that when the stability derivatives are constant, the dynamic modes predicted using the linear EoM match existing methods. With these two comparisons both giving good results, the linear EoM were considered to be accurate enough to model the aircraft for control development.

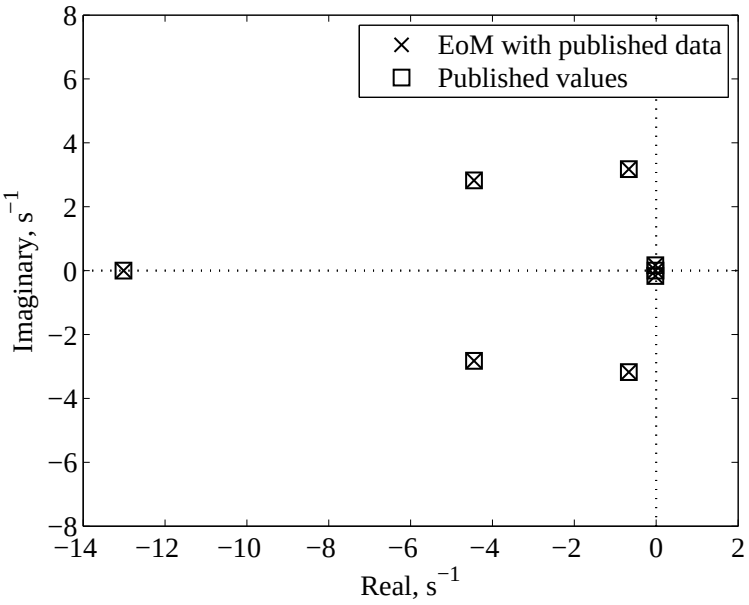


Figure 2.14: Eigenanalysis comparison of published values to the EoM using published data for a Cessna 182.

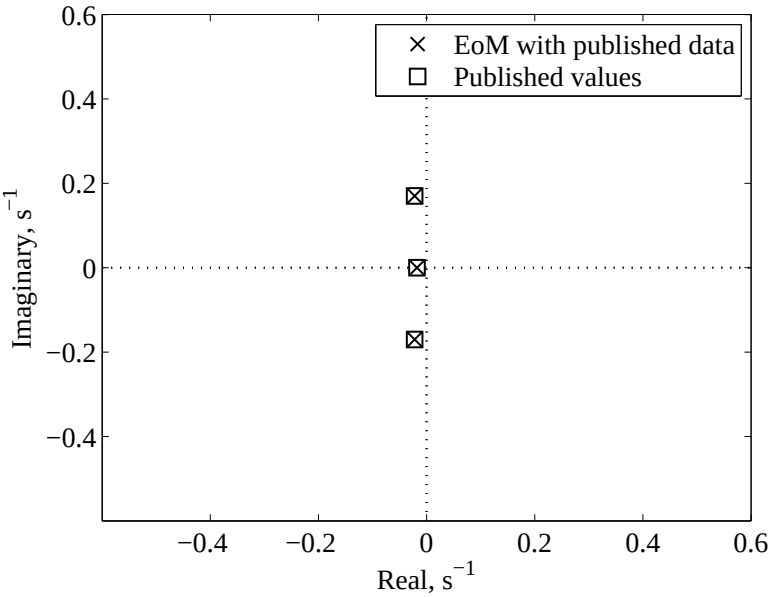


Figure 2.15: Expanded scale of eigenanalysis comparison of published values to the EoM using published data for a Cessna 182.

Table 2.15: Eigenvalues for a Cessna 182 from published data.

Dynamic Mode	Published Eigenvalues	Eigenvalues from EoM
Spiral	-0.0179	-0.01792
Phugoid	-0.022 $\pm j0.17$	-0.0221 $\pm j0.17$
Dutch Roll	-0.6703 $\pm j3.175$	-0.6703 $\pm j3.175$
Short-Period	4.45 $\pm j2.825$	-4.452 $\pm j2.825$
Roll	-13.01	-13.01

Table 2.16: Eigenvalues for a Cessna 182 using AVL.

Dynamic Mode	Eigenvalues from AVL	Eigenvalues from EoM using AVL data
Spiral	-0.0106	-0.002879
Phugoid	-0.01533 $\pm j0.1727$	-0.02293 $\pm j0.1712$
Dutch Roll	-0.7548 $\pm j3.003$	-0.761 $\pm j3.019$
Short-Period	-3.173 $\pm j4.188$	-3.174 $\pm j4.19$
Roll	-11.9	-11.81

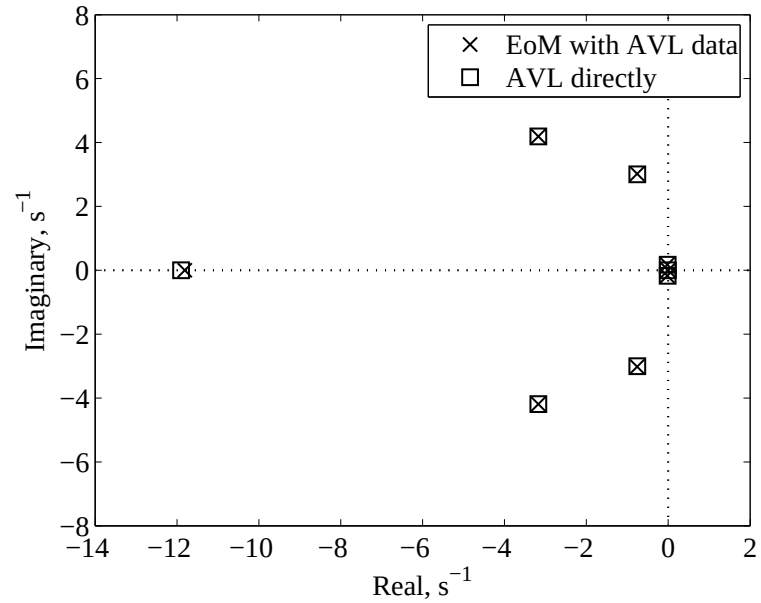


Figure 2.16: Eigenanalysis comparison of AVL results to the EoM using AVL data for a Cessna 182.

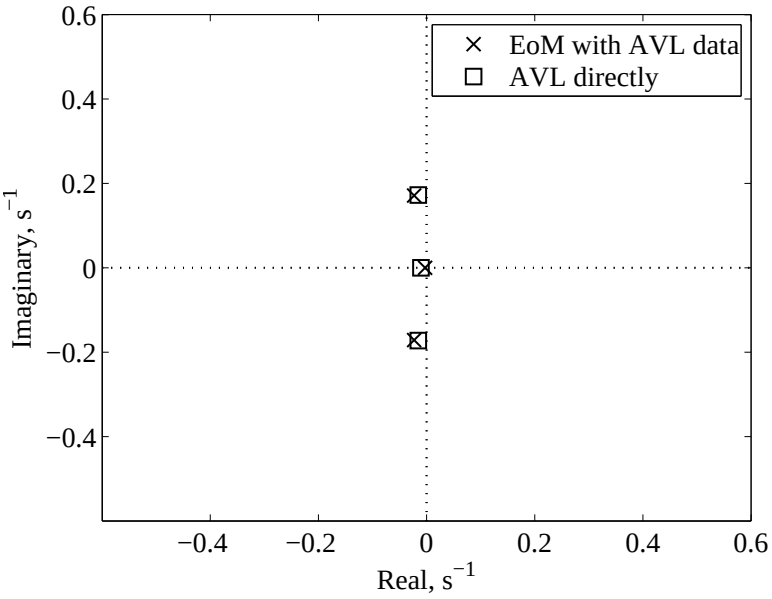


Figure 2.17: Expanded scale of eigenanalysis comparison of AVL results to the EoM using AVL data for a Cessna 182.

In addition to the validation effort, a comparison using the test aircraft was performed between the dynamic modes predicted by the EoM using data from AVL, the dynamic modes from AVL directly, and the modes predicted by the linear model of the simulation. Four separate flight conditions were compared and Table 2.17 lists each flight phase and the associated trim conditions.

Table 2.17: Flight conditions used for comparing the dynamic modes from the EoM to the linearized simulation.

Flight Phase	Trim Conditions			
Cruise	$V = 80$ ft/s	$h = 2,700$ ft	$\psi = 330°$	$\beta = \gamma = 0°$
Full Throttle	$\delta_T = 1$	$h = 2,700$ ft	$\psi = 330°$	$\beta = \gamma = 0°$
Knife-edge	$\delta_T = 1$	$h = 2,700$ ft	$\psi = 330°$	$\phi = 90°$
pre-Stall	$V = 44$ ft/s	$h = 3,000$ ft	$\psi = 330°$	$\beta = \gamma = 0°$

Figure 2.18 depicts the dynamic mode comparison for each flight condition. For most of the flight conditions, the eigenvalues from the EoM using AVL data are nearly identical to those from AVL directly which is to be expected. In the case of the cruise flight condition, the modes predicted by the EoM match those of the linear simulation pretty well. When at full throttle, however, the EoM predict a short-period mode with a higher damped frequency and a dutch roll mode with a lower natural frequency than that of the linearized simulation. The periods and damping ratios of each mode are provided in Tables 2.18 - 2.20 for the cruise, full throttle, and pre-stall flight conditions. The linear models developed in this chapter for the four flight conditions will be used in Chapter 4 to determine the autopilot control gains.

Table 2.18: Dynamic modes for cruise.

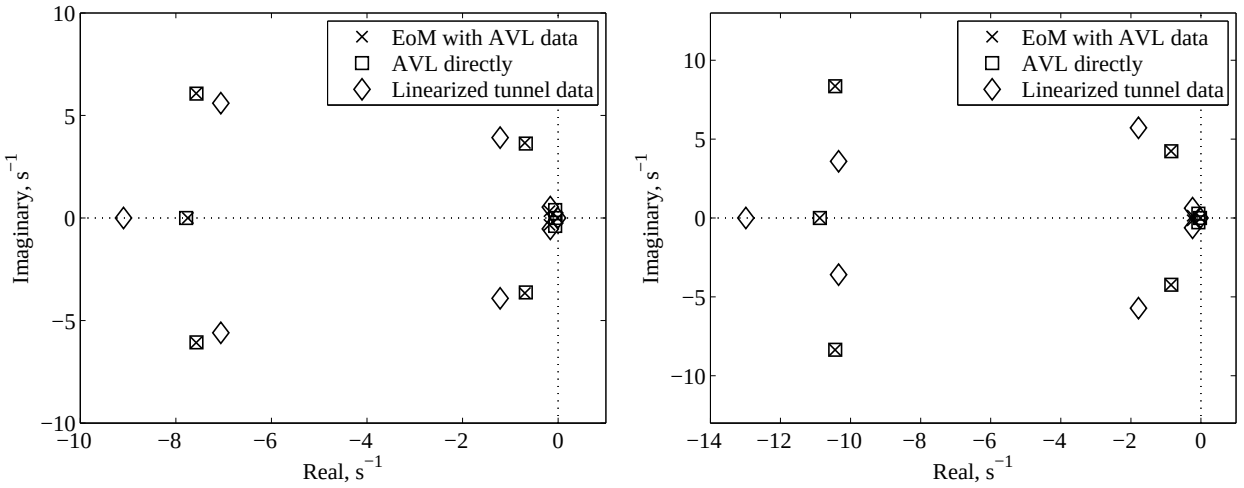
	EoM with AVL Data		Linearized Tunnel Data	
	Period or Time Constant, s	Damping Ratio	Period or Time Constant, s	Damping Ratio
Short-Period	1.03	0.780	1.12	0.783
Phugoid	18.61	0.470	11.88	0.294
Dutch Roll	1.72	0.186	1.60	0.295
Roll	0.129	-	0.11	-
Spiral	152	-	93.5	-

Table 2.19: Dynamic modes for level flight with full throttle.

	EoM with AVL Data		Linearized Tunnel Data	
	Period or Time Constant, s	Damping Ratio	Period or Time Constant, s	Damping Ratio
Short-Period	0.752	0.781	1.75	0.945
Phugoid	42.23	0.845	10.14	0.360
Dutch Roll	1.48	0.197	1.10	0.298
Roll	0.092	-	0.077	-
Spiral	91	-	36.6	-

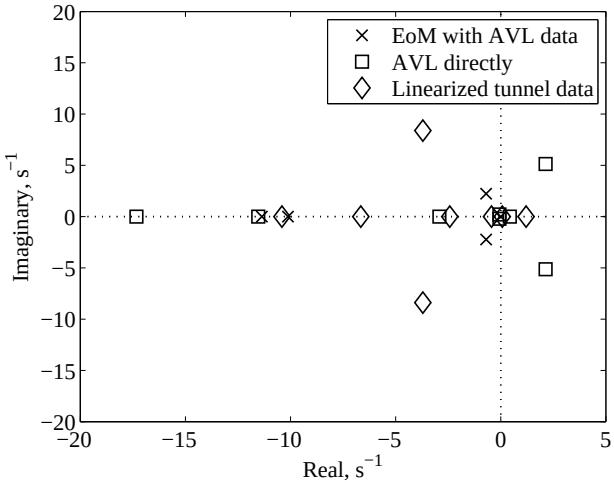
Table 2.20: Dynamic modes for pre-stall.

	EoM with AVL Data		Linearized Tunnel Data	
	Period or Time Constant, s	Damping Ratio	Period or Time Constant, s	Damping Ratio
Short-Period	1.83	0.772	0.986	0.536
Phugoid	9.48	0.223	7.50	0.1
Dutch Roll	1.77	0.101	3.90	0.325
Roll	0.237	-	0.189	-
Spiral	$T_2 = 8.6$	-	3.17	-

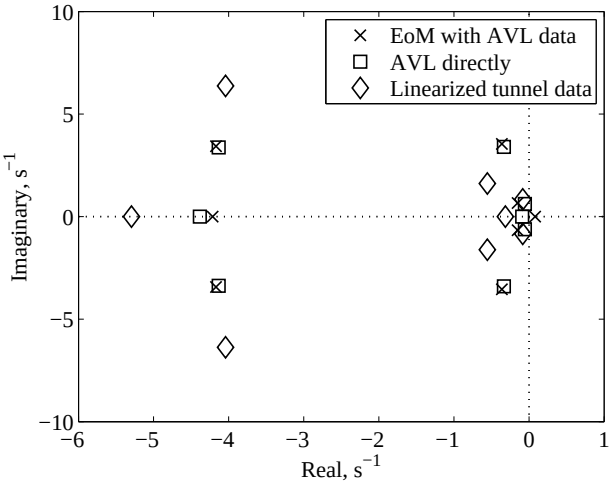


(a) Cruise

(b) Full throttle



(c) Knife-edge



(d) pre-Stall

Figure 2.18: Eigenanalysis comparison of the EoM using AVL data, AVL directly, and the linearized simulation for the test aircraft.

Chapter 3

Control Design

This work focuses on aircraft control during two main phases of flight. For cruise flight and transitions between maneuvers, a conventional autopilot was developed to maintain a desired airspeed, altitude, and heading using proportional-integral-derivative (PID) controllers. Maneuvering flight control was based on a pilot-inspired approach, where aerobatics were performed through the use of angular-rate commands based on pilot-controlled data. This differs from previous researchers' methods in [11–13, 81, 82], which center around commanding the airplane to track a specific path in space. The pilot-inspired approach to maneuvering relaxes the constraint on the exact position of the airplane and does not require the generation of a flight path.

3.1 Cruise Flight

The PID control structure was used for cruise flight and to transition between maneuvers because of its simplicity and proven performance. PID control is one of the most commonly used feedback control strategies because it combines the advantages of proportional, integral, and derivative control while eliminating the disadvantages associated with each [83–85]. The disadvantage with purely proportional control comes from the possibility of steady-state error which can be eliminated by the addition of an integral term. The disadvantage of the integral controller is that due to the response from accumulated error from the past, the stability of the system can be decreased. The decreased stability can be regained, though, by adding a derivative term. Derivative control works well because it is based on the rate-of-change of the error with time and can provide large corrections before the error becomes large. The disadvantage of derivative control, however, is that differentiation amplifies noise and can

destabilize a system. It is common to cascade the differentiator with a low-pass filter to limit the high frequency gain and noise, but care must be taken in the selection of the filter time constant to prevent the low-pass filter from cancelling the derivative action [86]. If the filter is included, the transfer function of the derivative term transforms from

$$G_D(s) = K_D\, s$$

to

$$G'_D(s) = \frac{K_D\, s}{1 + s\tau}$$

where K_D is the derivative gain and τ is the filter time constant. The filter was excluded for this work, but further implementation may require that decision to be reconsidered.

When designing the airspeed and altitude loops of the cruise flight controller, it was decided to use throttle to control altitude and elevator to control airspeed. This strategy accounts for the semi-parabolic shape of the power required versus airspeed curve of aircraft. Figure 3.1 is from [87] and depicts the increase in power required to maintain speeds below the maximum endurance speed.

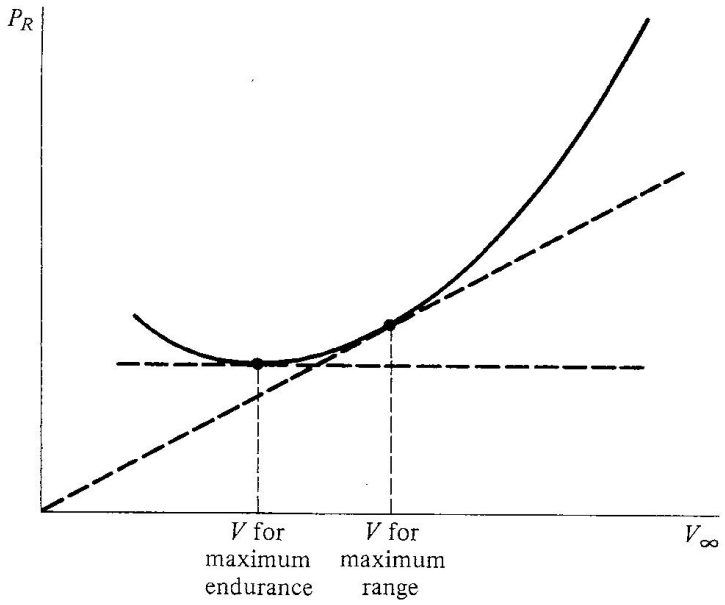


Figure 3.1: Power required for a propeller driven airplane.

If the control loops were swapped, with throttle used to track airspeed and elevator used to track altitude, speeds below the minimum power required speed would result in a constant pitch-up due to the elevator trying to maintain altitude. This occurs because the airspeed can be constant while having a constant rate of descent. If the speed is constant and the airplane is descending, the throttle would remain constant and the elevator would continue to pitch the nose up to account for the difference between the power available and the power required. The higher angles of attack would then result in a higher sink rate due to the increase in induced drag while holding power constant. This scenario would lead to a flight condition that is uncontrollable by the autopilot when throttle is used to track airspeed. To maintain control at all airspeeds, including speeds below the best endurance speed, the autopilot needs to adjust the power to maintain the desired altitude. Elevator can then be used to change speed by modifying the drag on the airplane. Figure 3.2 shows the block diagram of the airspeed and altitude control loops.

The rudder maintains zero washed-out yaw rate which ensures coordinated turns by preventing slipping or skidding and generates a yawing moment to oppose any yaw rate

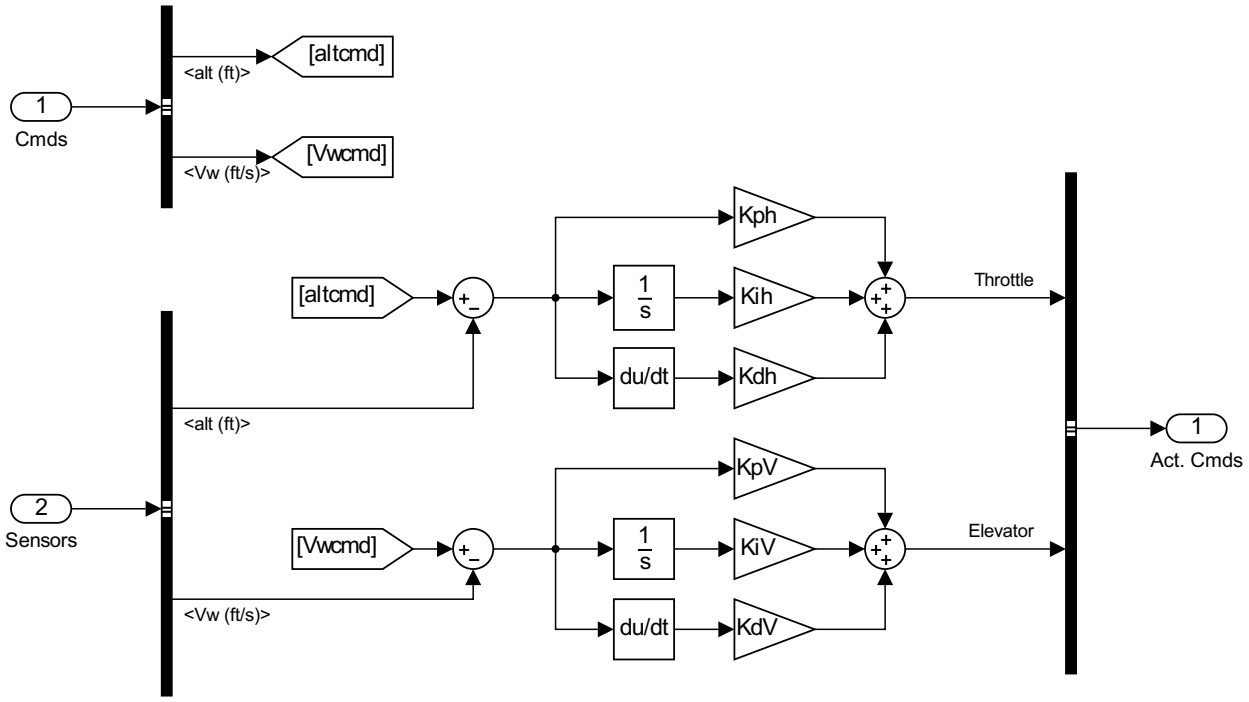


Figure 3.2: Airspeed and altitude flight control block diagram.

from the dutch roll mode. The yaw rate is "washed-out" through the use of a simple first-order high-pass filter known in this application as a washout filter. The washout filter is used because in a steady turn, the yaw rate is constant and nonzero. If yaw rate alone were used, the rudder would try to oppose the turn. The approximate differentiation of the yaw rate provided by the washout filter allows the constant yaw rate in a steady turn to be attenuated, allowing for coordinated turns. The transfer function for the washout filter is the same as for a simple lead and is given by

$$\frac{s\,\tau_w}{1 + s\,\tau_w} \tag{3.1}$$

where the time constant τ_w is a compromise between dutch roll damping and turn entry interference. According to [21], washout filter time constants are normally on the order of one second so τ_w was set to one second for this research. The rudder loop only consists of proportional control, so the trim value of the rudder deflection determined in Sec. 2.4.4 is

added to the output of the controller to obtain the total rudder deflection. The ailerons are used to control bank angle in order to track a desired heading based on feedback of both roll angle and roll rate. Figure 3.3 shows the block diagram of the control loops for washed-out yaw rate and heading.

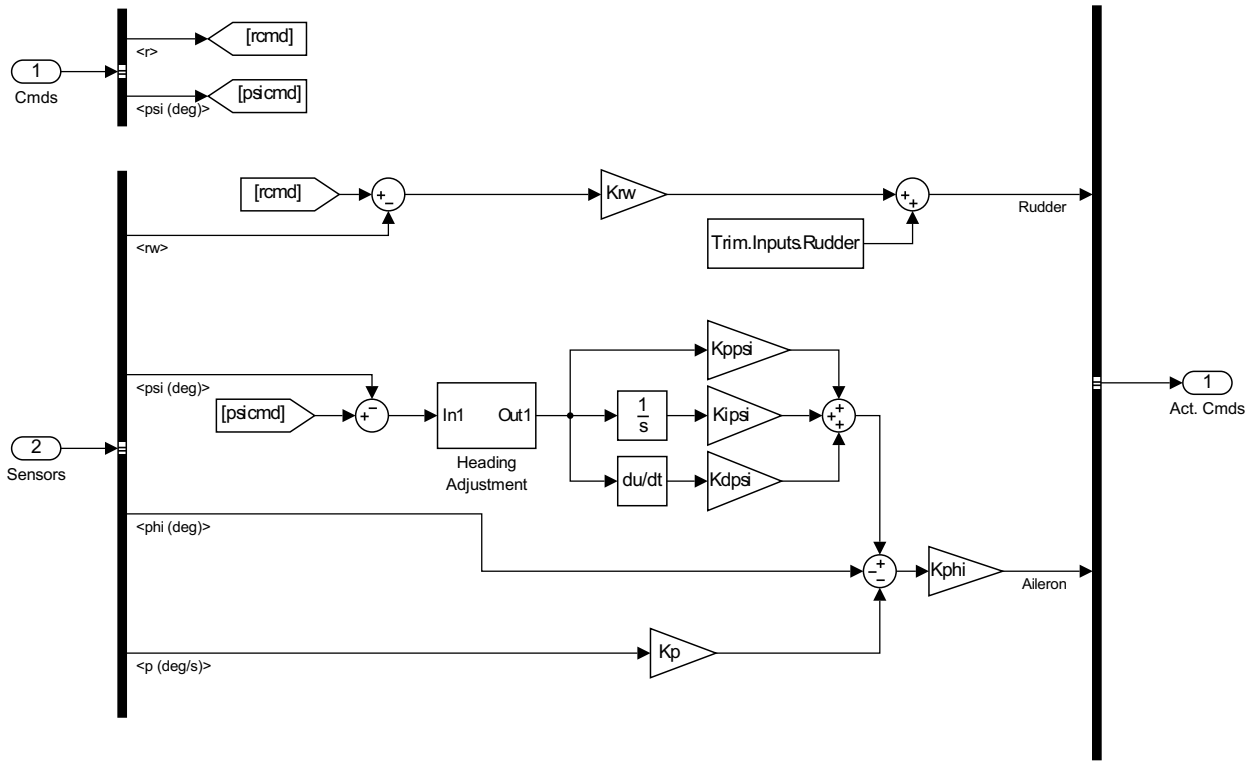


Figure 3.3: Yaw rate and heading flight control block diagram.

The heading adjustment subsystem depicted in the heading loop of Fig. 3.3 determines whether the airplane should turn left or right to reach the desired heading based on minimizing the total heading change during the turn. If the absolute value of the heading error is less than 180 degrees, the original heading error is passed through. If the heading error is larger than 180 degrees, a turn in the opposite direction is needed and the heading error is adjusted to provide such a turn. An example scenario of when the adjustment is needed would be when the aircraft is currently flying a heading of 010 degrees and the heading command is updated to 340 degrees. The heading error would be 330 degrees (340 - 010) which would result in the airplane turning right through 330 degrees. A faster and more efficient

turn-to-heading is accomplished by turning left through 030 degrees (changing the heading error to -30) which is the task performed by the heading adjustment subsystem shown in Fig. 3.4.

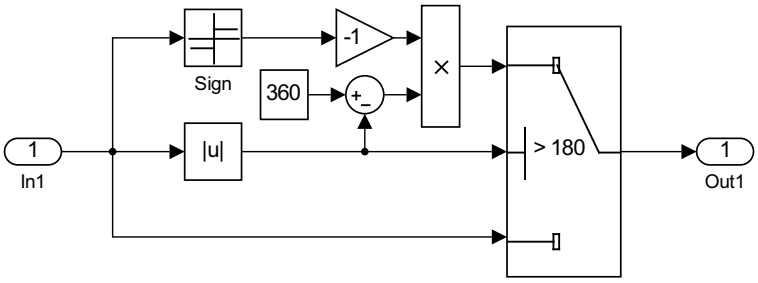


Figure 3.4: Simulation block diagram of the heading error adjustment.

A well-known problem when using integral control of an actuator with a saturation limit is integrator windup [21]. Windup occurs if the control setpoint is not reached when the actuator saturates. The integrator continues to integrate the error, taking the output of the controller further beyond the saturation limit and "winding up" the control error. When the output of the controller reverses, a considerable amount of time may pass before the excessive energy from the integrator is removed and the actuator comes out of saturation. To prevent integrator windup, a dead-zone device is used to determine if the control output will saturate the actuator. If it is determined that saturation will occur, the integral term in the controller is held at its current value until the command output comes out of saturation. Once the control output is back within the dead zone, the integral term is added back into the control loop. Figure 3.5 shows the block diagram of a single PID loop which includes the anti-windup logic used for this work and Fig. 3.6 shows the contents of the dead-zone logic block.*

*The anti-windup logic was removed from Figs. 3.2 and 3.3 as well as subsequent figures in Sec. 3.4 for clarity, but is included in the simulation for each integrator.

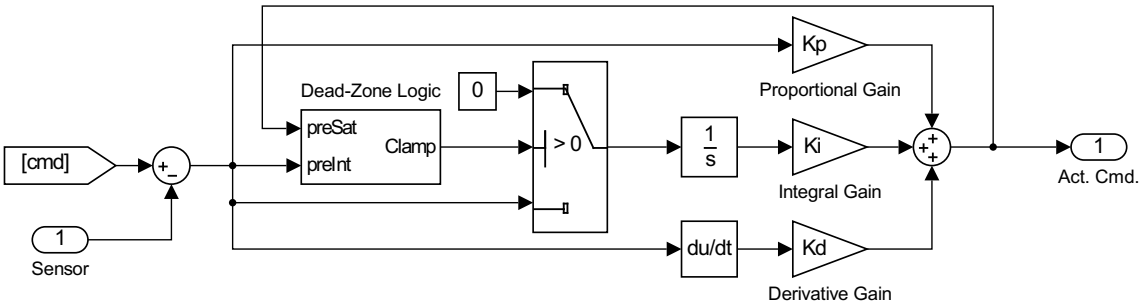


Figure 3.5: Simulation block diagram of a PID loop with anti-windup protection.

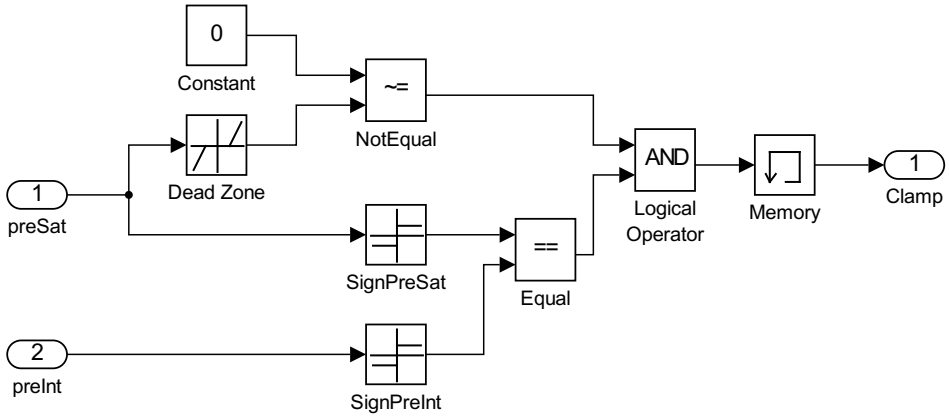


Figure 3.6: Simulation block diagram of dead-zone logic for anti-windup protection.

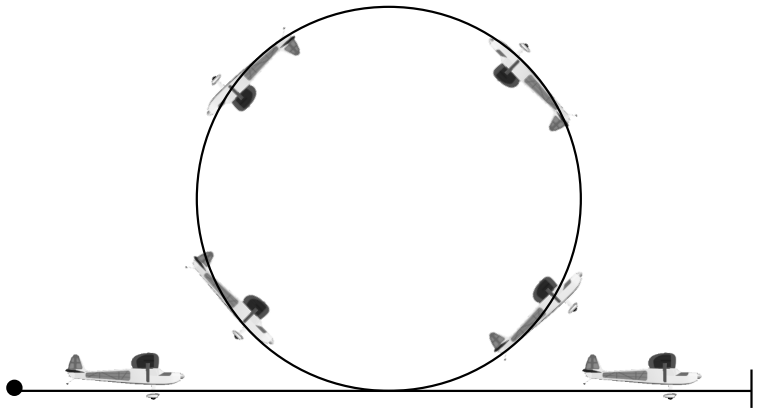
3.2 Maneuvers

Aerobatics were used to test the autopilot's functionality during maneuvers that involve unusual attitudes and airspeeds close to and below stall speed. Common aerobatic maneuvers such as loops, rolls, and knife-edge flight formed the foundation for the tests before more complicated post-stall maneuvers such as spins, stall turns, and tail slides were attempted. Some of the issues that arose while performing these maneuvers included the large variations in flight condition due to the nonlinear dynamics of the aircraft motion and the loss of control surface effectiveness at low airspeeds. The possibility for unlimited orientation angles as well as airspeeds below stall speed required the consideration of the entire flight envelope.

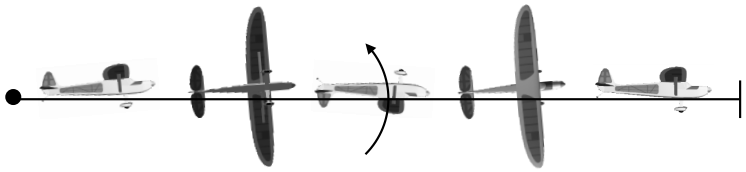
Aerobatic maneuvers are described by the Fédération Aéronautique Internationale (FAI) in the Aresti Catalog which enumerates the aerobatic maneuvers permitted in manned aircraft aerobatic competitions. Aresti diagrams were designed by Spanish aviator José Luis Aresti Aguirre[†] and are composed of geometric shapes connected by lines to represent each maneuver. In the illustrations, small circles signify the starting points of the maneuvers with short vertical lines signifying the end points. Curving arrows are used to represent rolls and numbers are used to define the number of hesitation points in the roll as well as to signify how many of the points are to be flown. Figure 3.7 shows the Aresti diagrams of the three common aerobatic maneuvers used in this work with an aircraft superimposed over the diagram to illustrate the aircraft orientation during the maneuver. The loop is entered from straight-and-level flight and is characterized by the circular path the aircraft follows during the maneuver. The roll consists of the aircraft rolling through 360 degrees of bank while maintaining constant altitude and heading. For knife-edge flight, the aircraft is first rolled through the first quarter of a four-point roll. The bank angle is then held as close to 90 degrees as possible while also maintaining constant altitude and heading. A quarter roll back to level flight signifies the end of the knife-edge maneuver.[‡]

[†] `http://www.fai.org/civa-news/personality-civa/35295-jose-luis-aresti-aguirre`

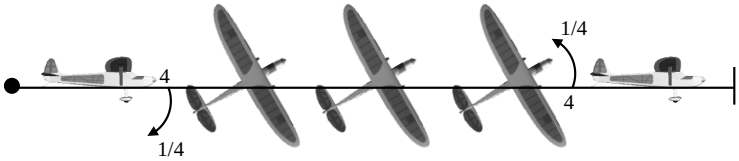
[‡] More information on the Aresti Catalog and the description of aerobatic maneuvers can be found in Section 6, Part 1 of the FAI sporting code for powered aircraft available at `http://www.fai.org/downloads/civa/SC6Part12014`. The International Aerobatic Club also maintains a list of aerobatic figures, their associated Aresti diagrams, and short descriptions of each available at `https://www.iac.org/legacy/aerobatic-figures`.



(a) Loop



(b) Roll

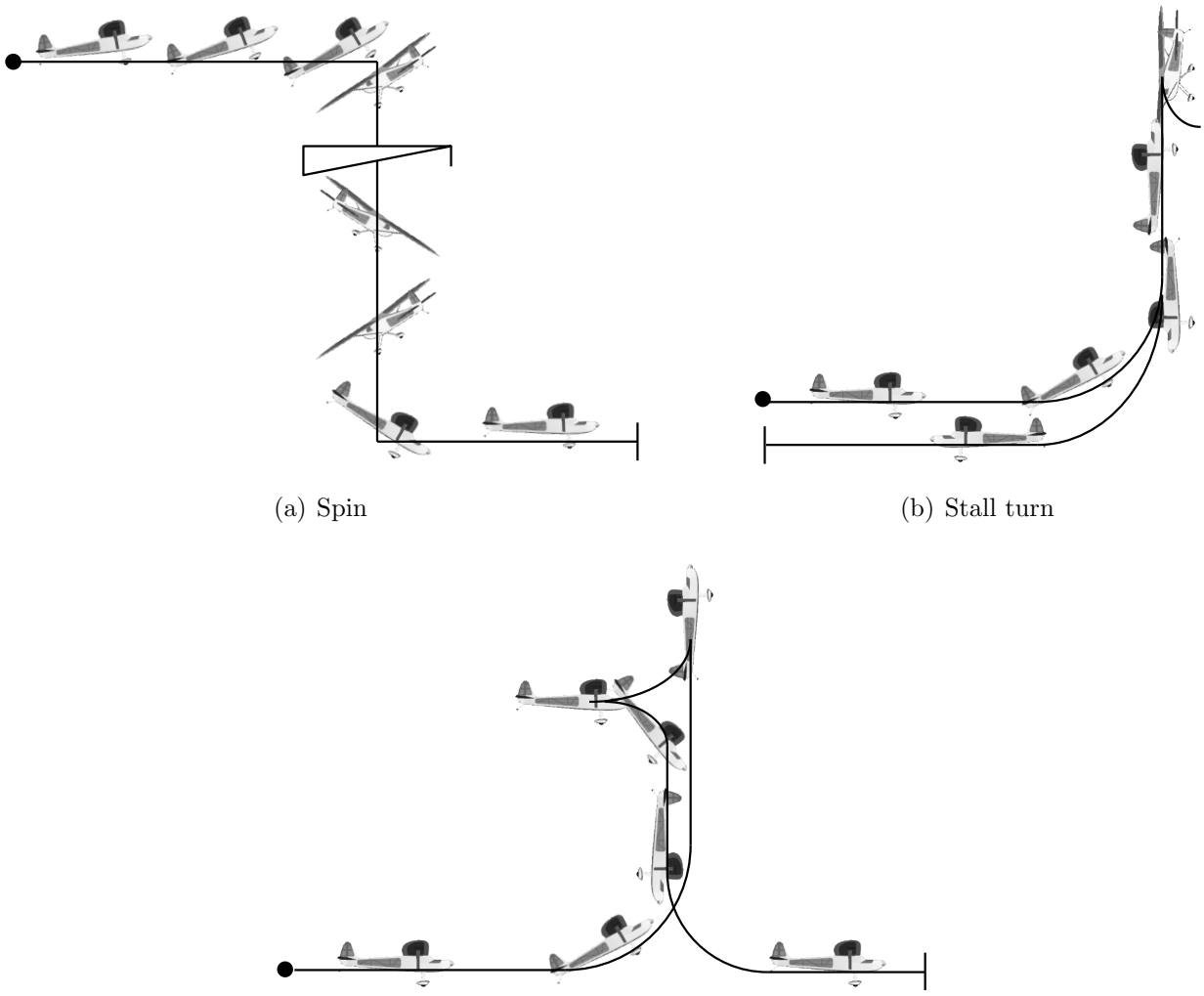


(c) Knife-edge

Figure 3.7: Diagrams of the three common aerobatic maneuvers.

Figure 3.8 illustrates the more advanced post-stall maneuvers. The spin is a special type of stall where one wing is stalled more than the other resulting in an autorotation around the vertical axis while descending. For this work, the simulated spins quickly turn into spiraling motions and never become fully-developed autorotations because the rotary derivatives are not included in the simulation. The stall turn (also known as a hammerhead turn) is a common heading-reversal maneuver and is performed by pitching the nose of the aircraft up into a vertical climb until the combination of engine thrust and momentum can no longer sustain altitude. The rudder is then used to yaw the aircraft 180 degrees into a vertical descent before recovery is initiated at the same altitude that the maneuver

began, heading in the opposite direction. Similar to the stall turn, the tail slide also begins with a vertical pitch-up and climb until the aircraft loses momentum. Once the airspeed dissipates, the aircraft "slides" backwards tail first before pitching nose-down to a vertical descent. Recovery is initiated at the same altitude as the maneuver began, heading in the same direction.



(a) Spin

(b) Stall turn

(c) Tail slide

Figure 3.8: Diagrams of the three post-stall aerobatic maneuvers.

For the rolling maneuvers, both left and right rolls are simulated to ensure the autopilot is capable of both despite propulsive moments. With full power, the propulsive moments

cause the airplane to turn to the left making a left roll easier than a roll to the right. For the same reason, knife-edge flight is simulated with the roll angle at both 90 and -90 degrees.

3.3 Pilot-Controlled Data

The pilot-inspired approach to maneuvering started with recording the state time histories during a series of manually controlled aerobatic maneuvers using the open-source flight simulator FlightGear with the JSBSim flight dynamics model.[§] Figures 3.9 and 3.10 show the flight path and recorded flight data for two consecutive pilot-controlled loops. The looping maneuvers started from level flight (on the right side of Fig. 3.9) with a speed of 100 ft/s, an altitude of 2,700 feet, and a heading of 330 degrees. The flight data shows that the aircraft states are largely repeatable with both maneuvers. It also appears that the pilot is mainly controlling the angular rates through the maneuvers, with command transitions determined by the attitude of the airplane. During the loops, the angular-rate data is either constant or changing as a linear function of time which leads to the assumption that the pilot's commands are similar to those of an angular-rate controller. Once the maneuvers begin, the roll and yaw rates are held constant at zero while the pitch rate ramps up linearly over five seconds. After the ramp, the pitch rate is held constant near 50 deg/s for almost five seconds before dropping back to zero to recover from the maneuver. The airspeed, altitude, and heading perform exactly as intuition would suggest, with the speed dropping to a minimum when the altitude is maximum. Near the top of the loop, the heading reverses almost 180 degrees while the airplane is inverted. Once the descent begins, on the back side of the loop, the speed begins to rise followed by another heading reversal before recovery from the maneuver is initiated at nearly the same altitude as it began.

[§]More information about the FlightGear project and JSBSim can be found at `http://www.flightgear.org` and `http://www.jsbsim.org`.

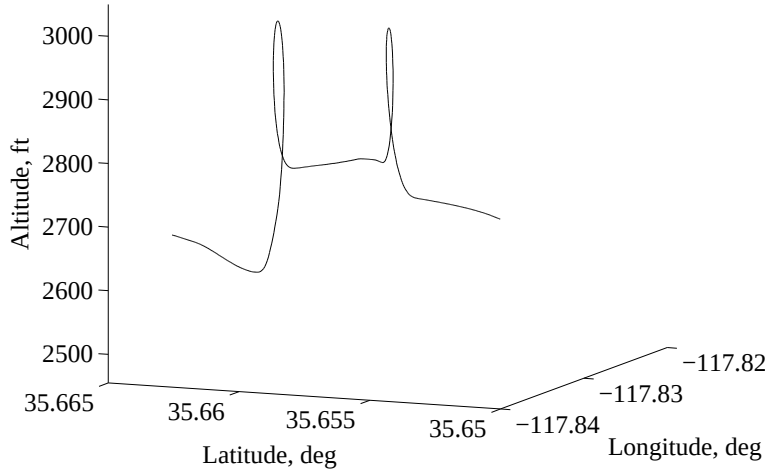


Figure 3.9: Flight path during two consecutive pilot-controlled loops.

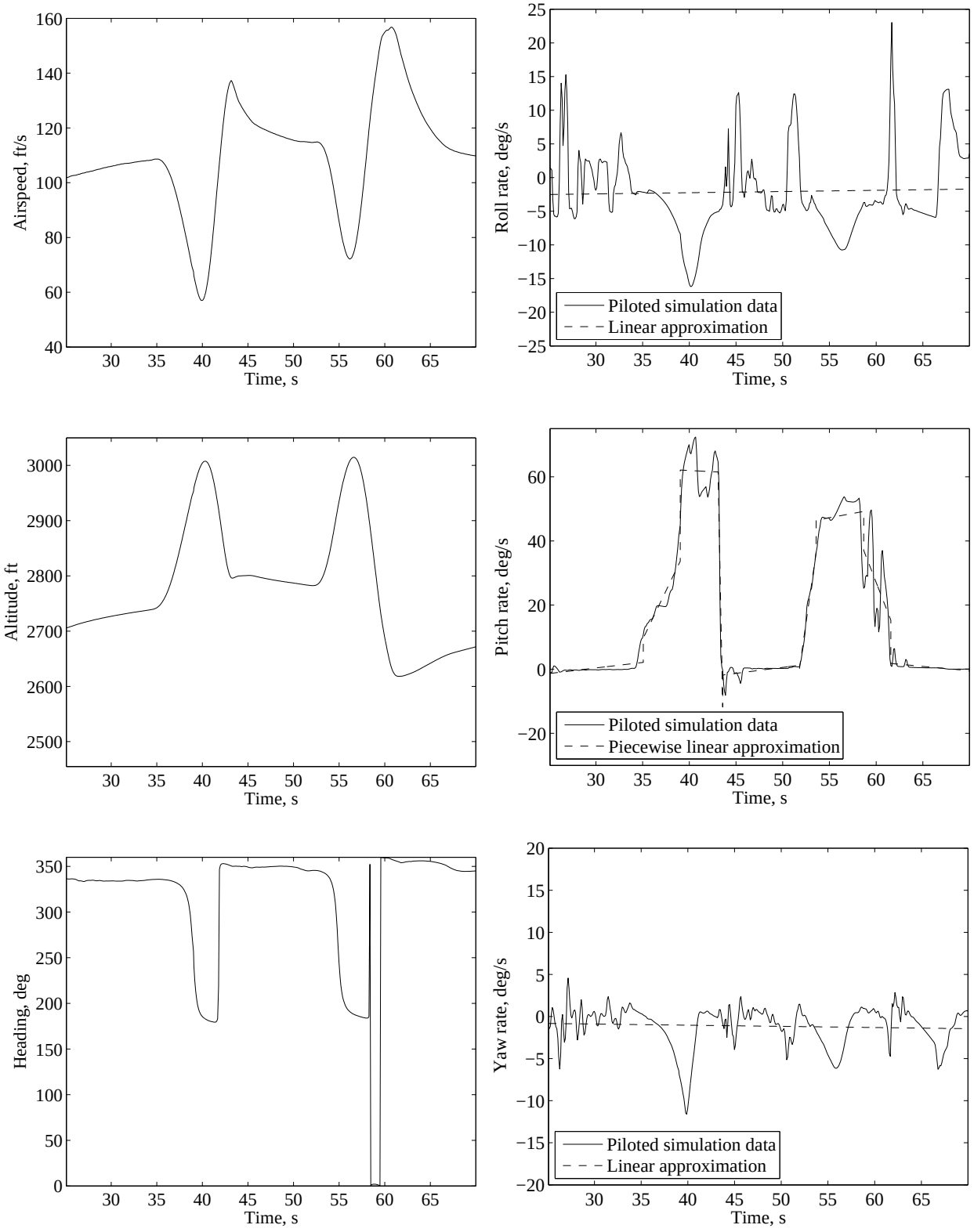


Figure 3.10: Recorded flight data during two consecutive pilot-controlled loops.

Figure 3.11 shows the recorded flight data for two consecutive rolls. Starting from level flight, a roll to the right was performed first, followed by a roll to the left. Comparing the state time histories reveals that the aircraft states during the maneuvers are similar. Each maneuver begins with a slight pitch up, which appears as an initial spike in pitch rate. The roll rate then ramps up linearly over approximately one second before being held constant at 100 deg/s for around three seconds. During the maneuvers, the yaw rate varies as an approximate sine function of roll angle. This yaw-rate command causes the yaw rate to be maximum when the aircraft is at 90 degrees of bank and is used to minimize the altitude loss associated with the reduced lift during high bank angle conditions. This pattern is most evident in the second roll (to the left) where the yaw rate is zero as the maneuver begins, increases to a maximum, then goes negative before returning to zero after the maneuver ends. The airspeed, altitude, and heading also oscillate in a sine wave manner. The airspeed decreases and the altitude increases initially due to the pitch up at the start of the maneuver. The airspeed then increases as the airplane is inverted and starts to descend before returning to the initial conditions after the roll is complete. The increase in induced drag from the aileron deflection is apparent from the heading oscillations. For the right roll, the heading initially decreases, giving the appearance of a turn to the left. This is due to the aileron deflection generating more lift on the left wing and less lift on the right wing. The increase in lift causes an increase in induced drag on the left side of the airplane, resulting in a left yaw. Once the airplane is inverted, the heading increases because now the left wing (still generating more lift) has changed sides and is causing a yaw to the right. With the aileron deflection removed, the roll stops and the heading returns to the initial condition.

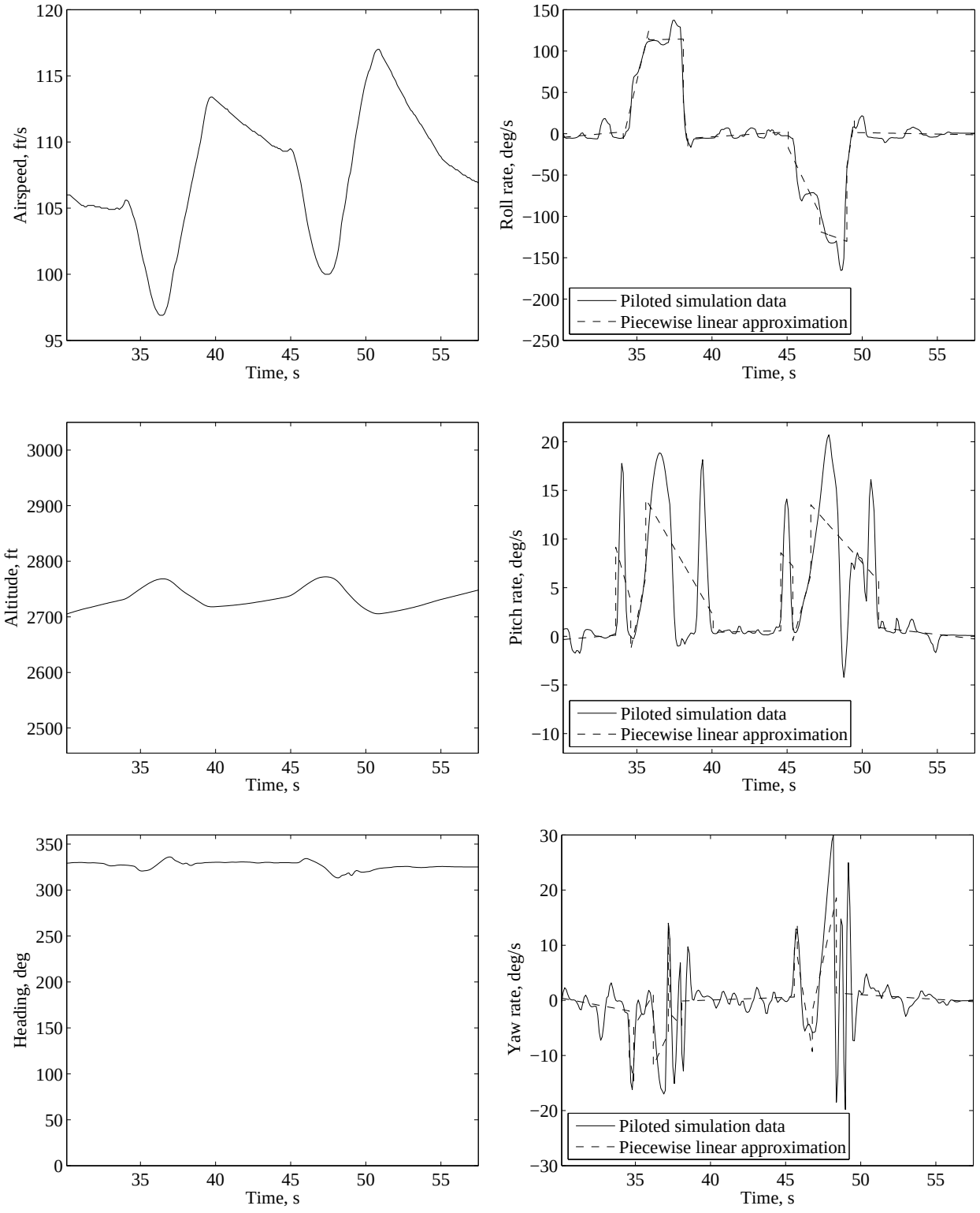


Figure 3.11: Recorded flight data during two consecutive pilot-controlled rolls in opposite directions.

Figures 3.12 and 3.13 show the state time histories during knife-edge flight to the left. During the maneuver, the roll angle, altitude, and heading are held constant. The maneuver begins from level flight before the aircraft is rolled to 90 degrees of bank. The roll rate plot shows the two spikes when the maneuver begins and ends which correspond with the ramps in roll angle. The pitch and yaw rates during the maneuver are mainly zero with the variations coming from the tracking of altitude and heading. It is important to point out that during knife-edge flight, changes in pitch produce changes in heading and changes in yaw produce changes in altitude and speed.

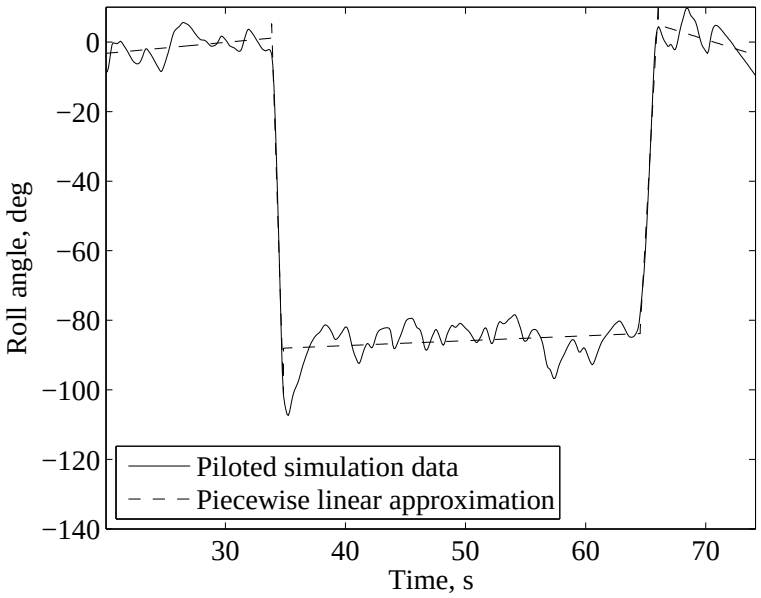


Figure 3.12: Roll angle during pilot-controlled knife-edge flight.

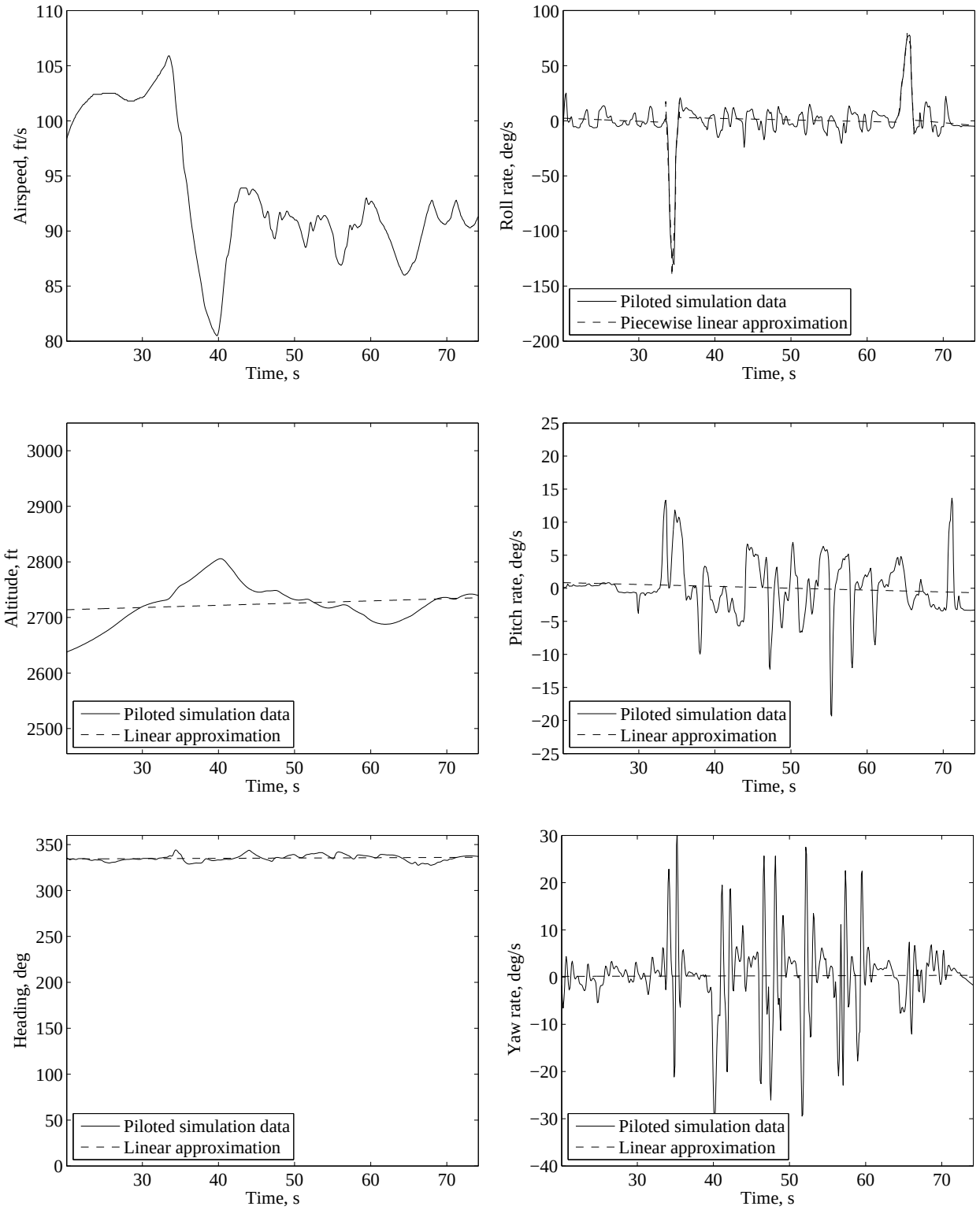


Figure 3.13: Recorded flight data during pilot-controlled knife-edge flight.

Figures 3.14 and 3.15 show the recorded flight data for a spin to the left. The spin started at an altitude of 3,000 feet to allow sufficient altitude for recovery before reaching ground level. During the initial stall portion of the maneuver, the wings are held level (zero roll angle) and the heading is held constant while the speed is decreased to induce the stall. Once the stall occurs, pro-spin inputs of aileron, elevator, and rudder result in the spinning motion of the airplane. The roll and pitch rates increase linearly throughout the spin and the changes in altitude and heading can be clearly seen as the aircraft rotates about the vertical axis. The yaw rate increases linearly over about two seconds before decreasing for around five seconds to end the maneuver. During the simulated spin, the airspeed constantly increases due to the lack of autorotation. This increase in speed is not entirely representative of an actual spin and results in a spiraling motion during the simulation. The secondary peak in the pitch rate is the pitch recovery from the maneuver which is necessary due to the nose down attitude of the airplane after the rotational motion has stopped.

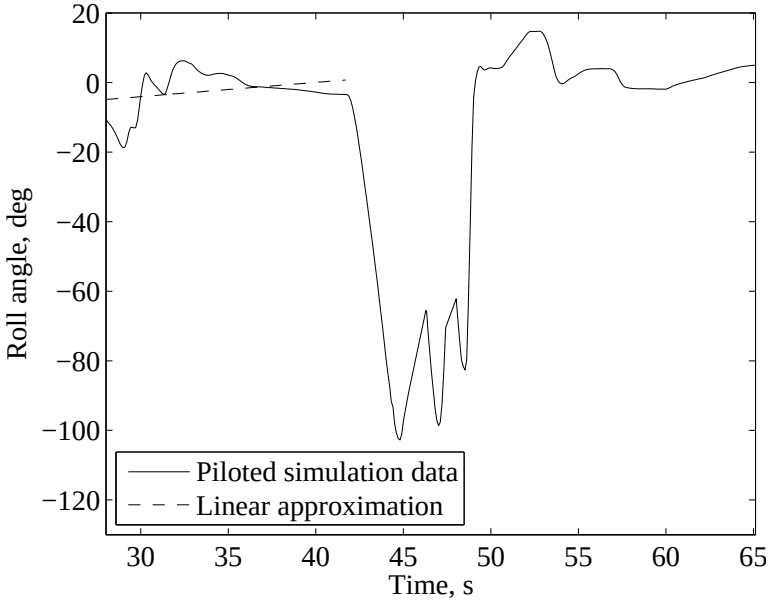


Figure 3.14: Roll angle during a pilot-controlled spin to the left.

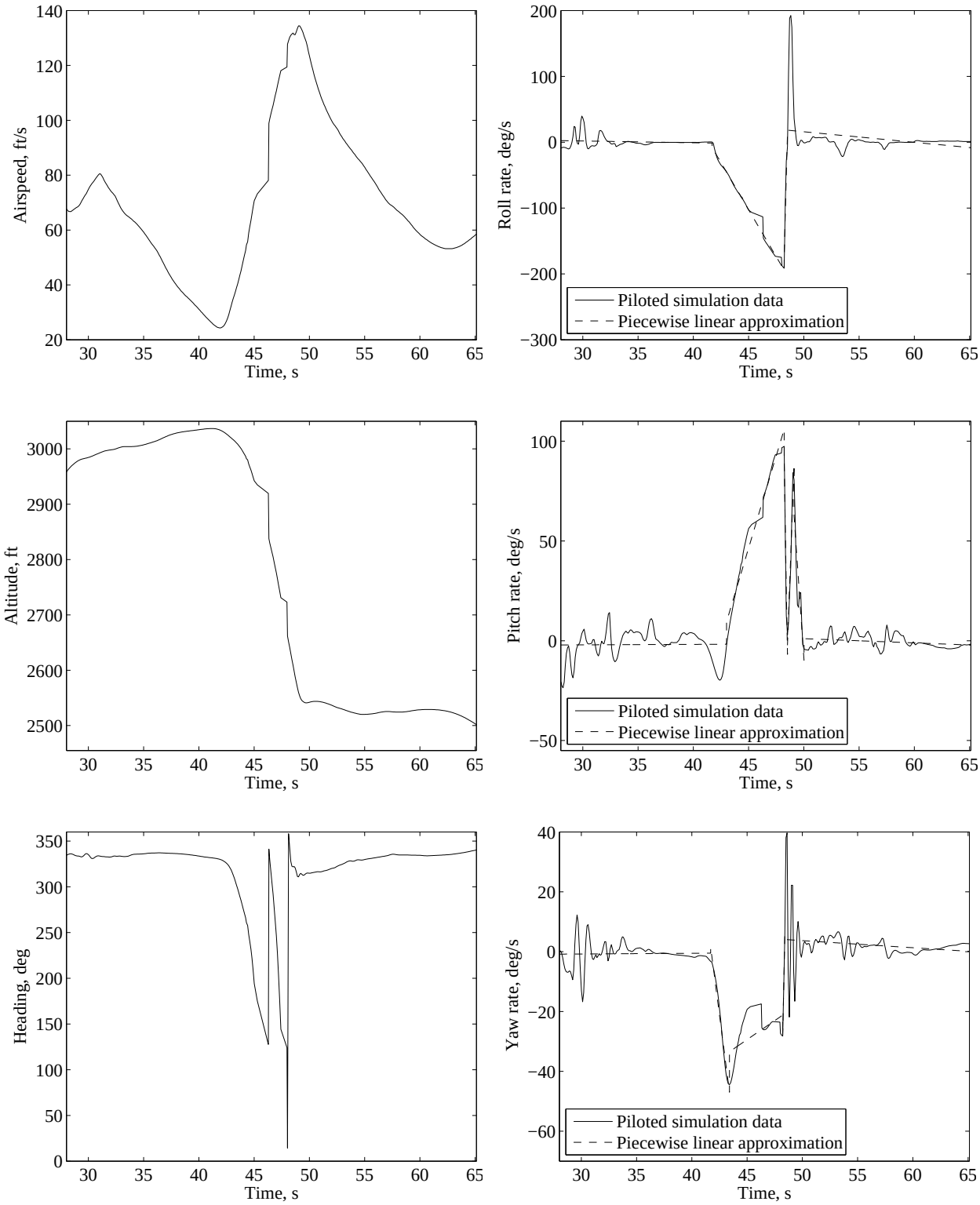


Figure 3.15: Recorded flight data during a pilot-controlled spin to the left.

The recorded data for a pilot-controlled stall turn is presented in Figs. 3.16 - 3.18. The flight path plot shows the motion of the airplane, which ultimately results in a heading reversal after the completion of the maneuver. Figure 3.17 shows the throttle position during the stall turn. For the vertical portion, the throttle is set at full power and remains there until the vertical descent begins. The throttle is closed during the vertical descent and remains closed until recovery from the maneuver is initiated. The data in Fig. 3.18 shows the increase in altitude and decrease in speed associated with the vertical portion of the maneuver. During the turn at the top of the vertical climb, the speed is well below the stall speed of the airplane, which was estimated to be 40 ft/s. For this reason, the throttle is held at full power to provide additional airflow over the rudder, resulting in adequate control authority. This effect is modeled in the nonlinear simulation by representing the control derivatives as functions of surface deflection and advance ratio as well as angle of attack and sideslip angle. Throughout the maneuver, the roll rate is maintained around zero. The pitch up to vertical as well as the pitch up to recover from the vertical descent is shown as two peaks on the pitch rate plot. The turnaround is distinguished by the peak in yaw rate which occurs around the 44^{th} second of the flight.

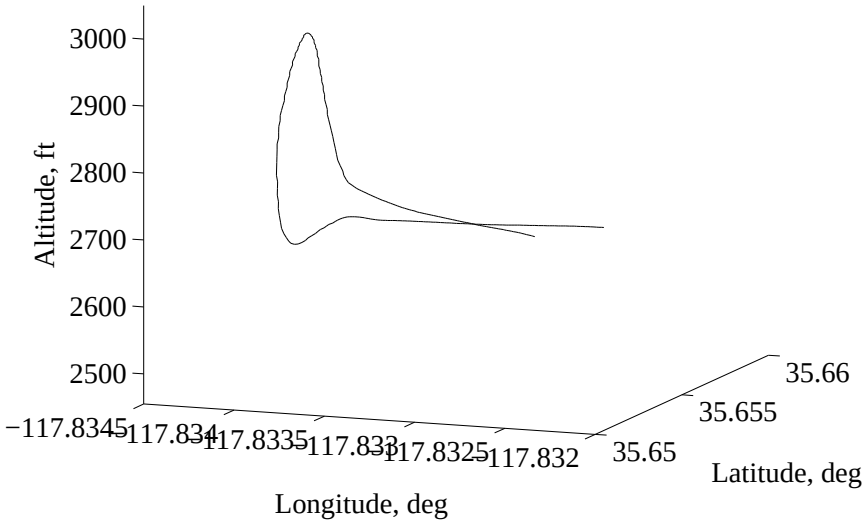


Figure 3.16: Flight path during a pilot-controlled stall turn.

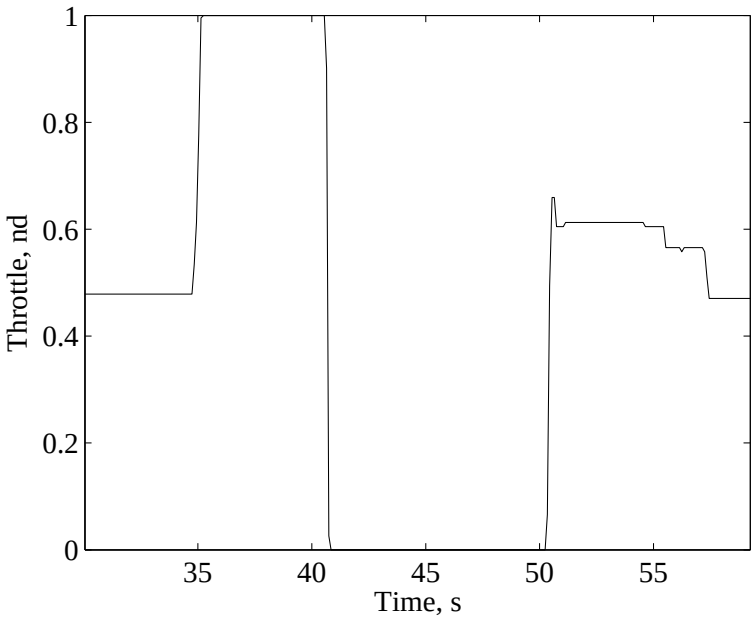


Figure 3.17: Throttle position during a pilot-controlled stall turn.

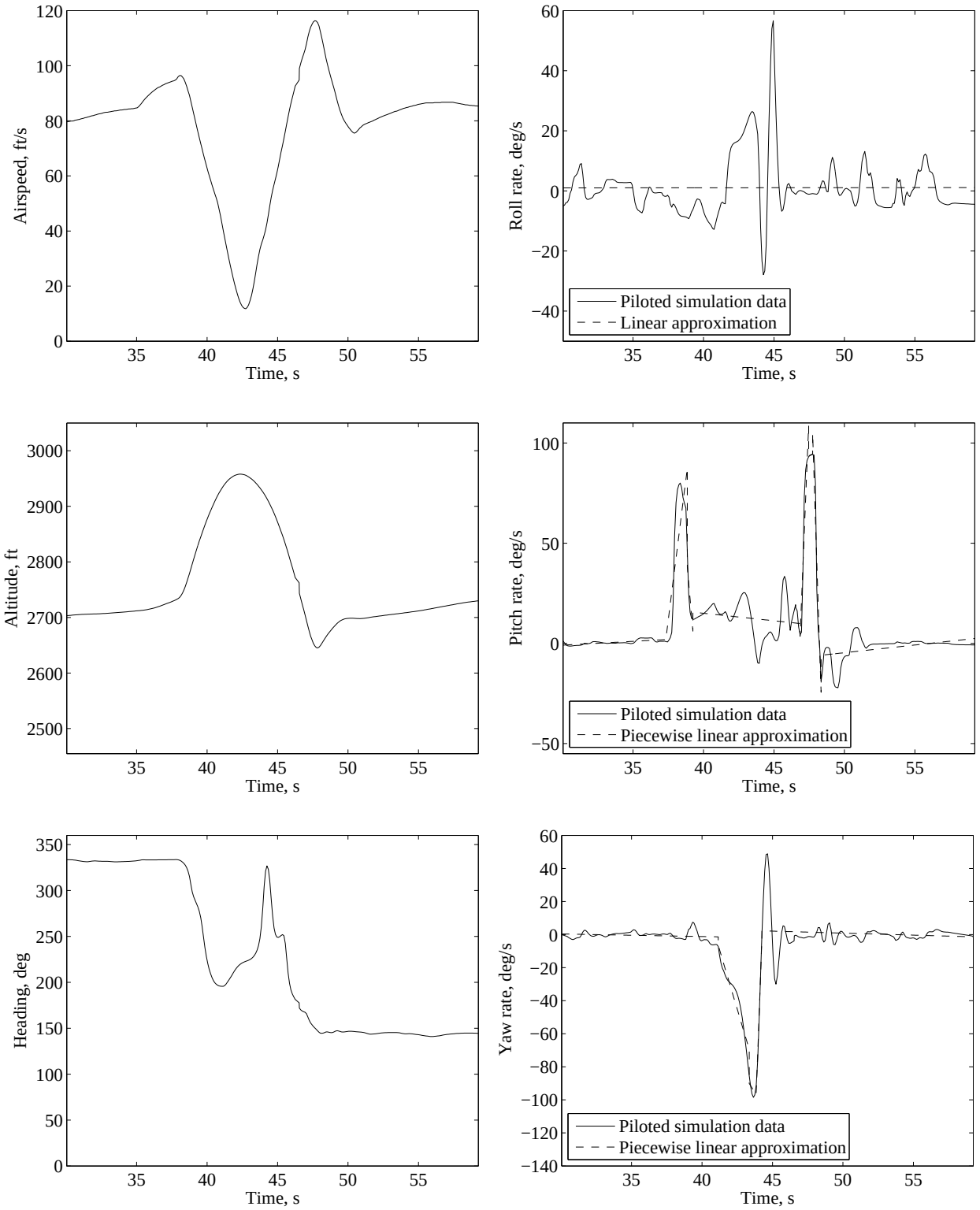


Figure 3.18: Recorded flight data during a pilot-controlled stall turn.

Figures 3.19 - 3.21 show the data from the tail slide which begins the same as the stall turn. (Motion is from right to left in Fig. 3.19.) Both maneuvers start with a vertical climb at full power. Once the momentum and engine thrust can no longer support the climb, the aircraft is held vertical as long as possible before the nose is pitched over and the vertical descent begins. The speed continues to drop throughout the vertical climb, and remains at zero momentarily during the "sliding" portion of the maneuver. The throttle is closed for the descent to prevent excess speed from building up. The heading remains relatively constant and the roll and yaw rates remain near zero during the maneuver. The pitch up to the vertical climb, the pitch down to the vertical descent, and the recovery to level is shown as three peaks on the pitch rate plot. The tail slide maneuver ends at the same speed, altitude, and heading as it began with zero angular rates.

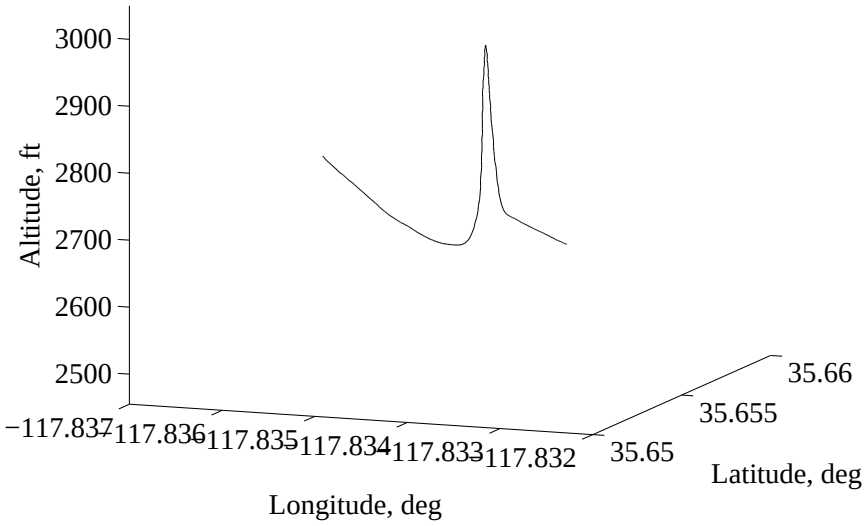


Figure 3.19: Flight path during a pilot-controlled tail slide.

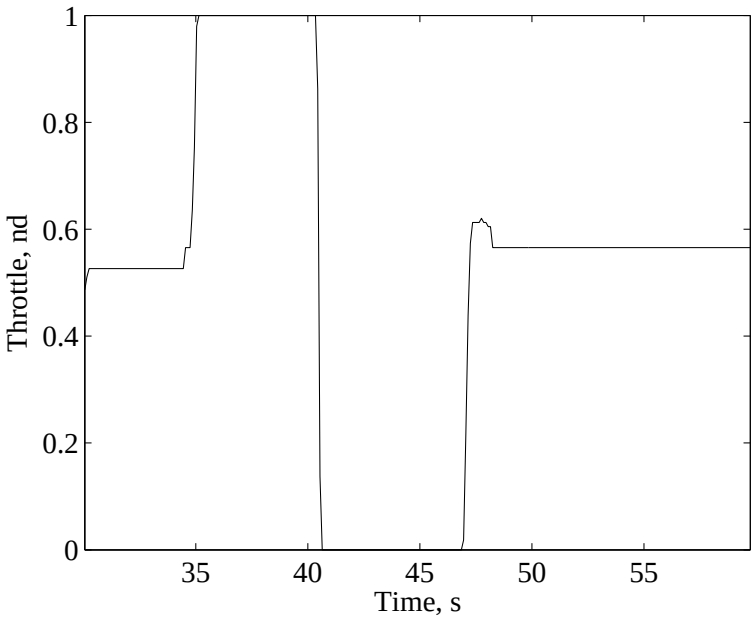


Figure 3.20: Throttle position during a pilot-controlled tail slide.

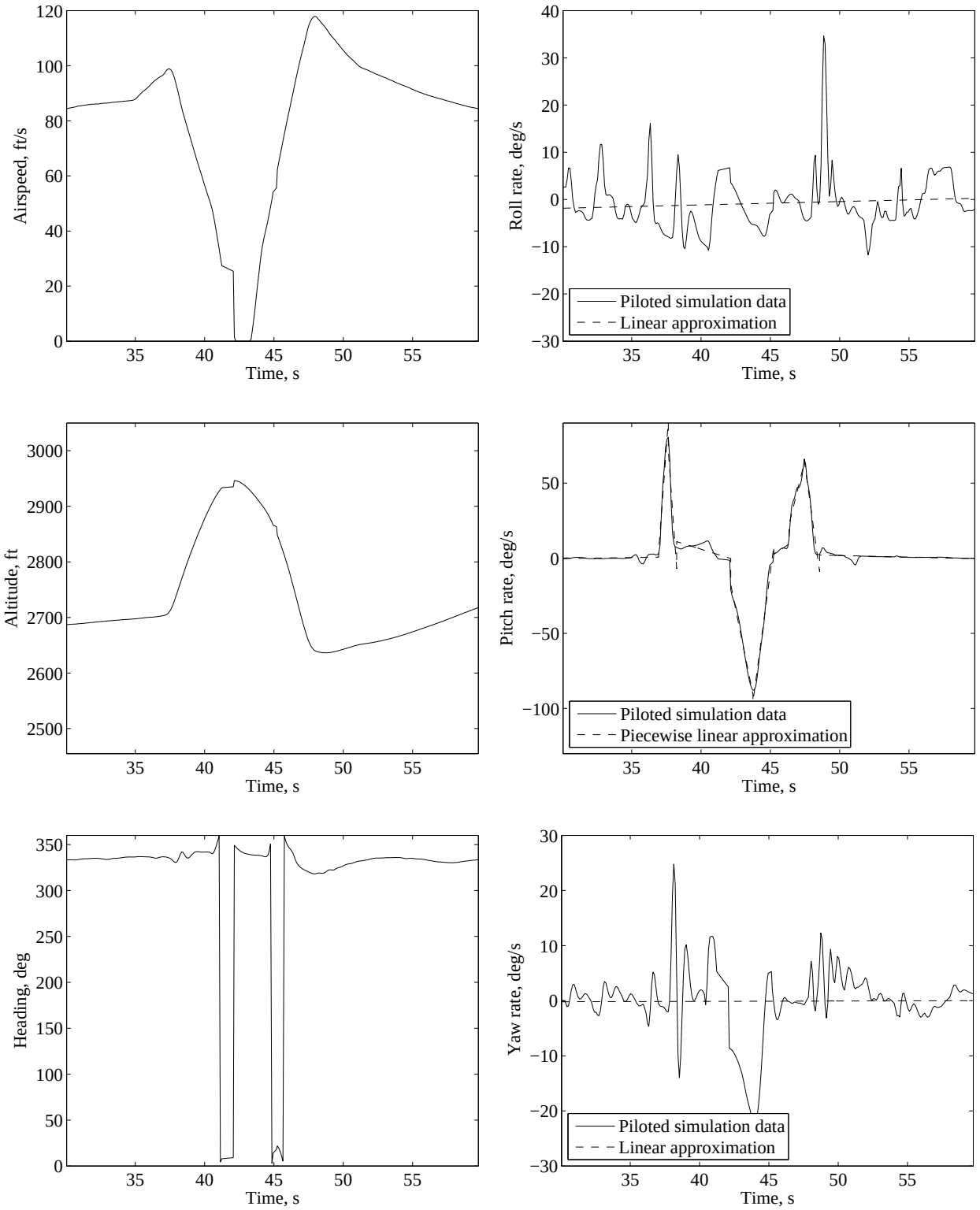


Figure 3.21: Recorded flight data during a pilot-controlled tail slide.

3.4 Maneuvering Flight Control

Using the pilot controlled data presented in the previous section, three controllers were designed to perform the aerobatic maneuvers. From the piloted data, it appeared that the pilot's inputs for the loop, roll, stall turn, and tail slide were based on the aircraft angular rates, so an angular-rate controller was developed to complete those maneuvers. Figure 3.22 shows the block diagram of the angular-rate controller. The throttle commands were passed through directly and proportional-integral (PI) control was used to control the angular rates. The integral terms were initialized at the start of the simulation based on the trim conditions

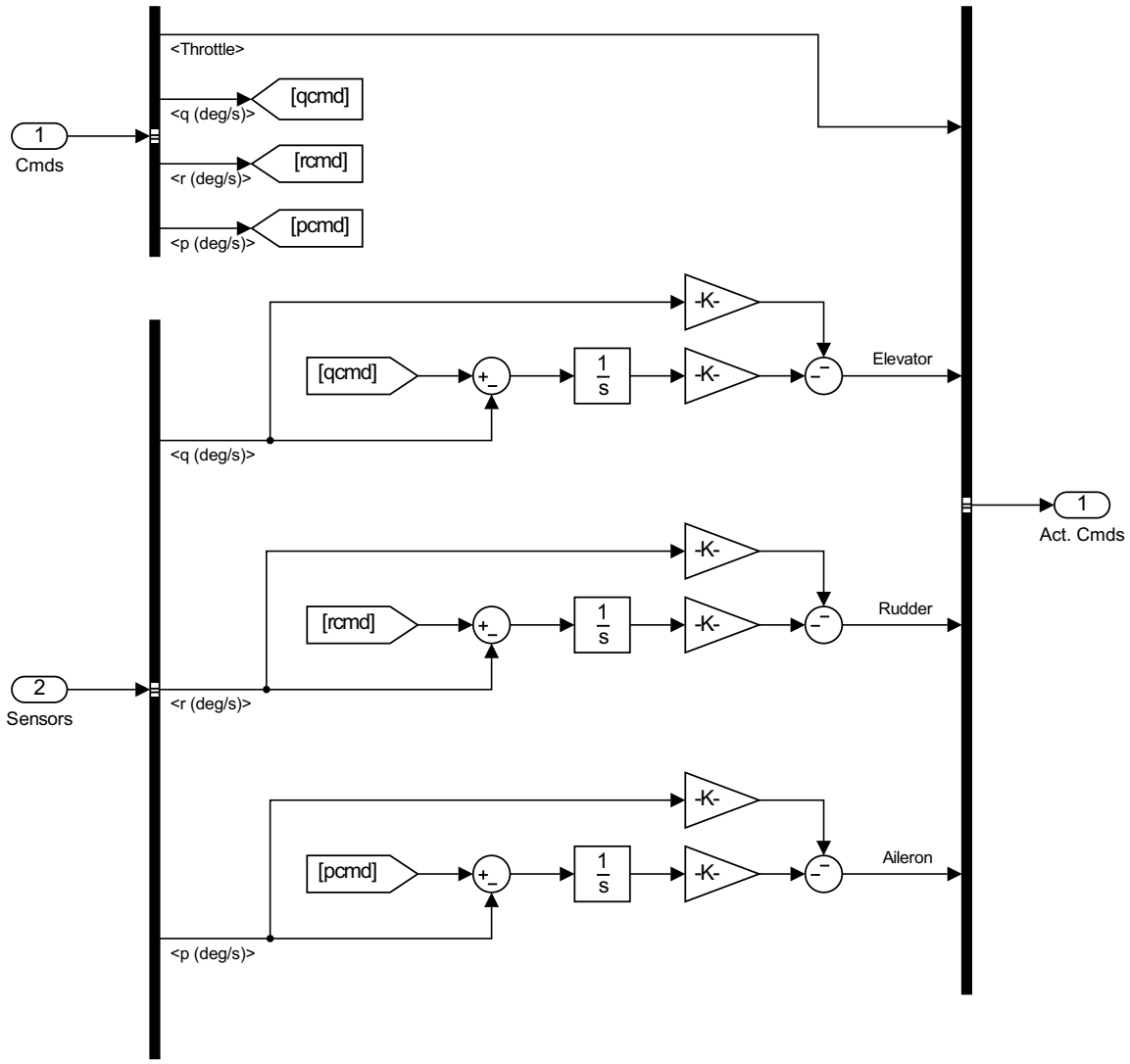


Figure 3.22: Angular-rate control block diagram.

for level flight. PI control was used to provide more precise control than a proportional-only controller while eliminating the possibility of steady-state error. Proportional control would not be adequate due to the pilot-inspired control approach being used in this work. The possibility of steady-state error could mean that the maneuvers would not be performed correctly and an effort was made to avoid that possibility by including proportional-plus-integral compensation. PI control overcomes the phase-lag problem associated with pure integral control, but in the conventional sense introduces a closed-loop zero which can cause an overshoot in the step response. Inner-loop proportional feedback, combined with pure integral control, gets rid of the closed-loop PI zero, but maintains the same characteristic equation as PI control. For the loop and roll, the throttle was commanded to remain full throughout the maneuvers. During the stall turn and tail slide, the throttle is adjusted based on the pilot-controlled data to minimize the increase in speed during the vertical descents.

Knife-edge flight is accomplished through a mode-switching mechanism due to the change in controls when the aircraft is flying level and when the aircraft is at 90 degrees of bank. While the roll angle is near 90 degrees, heading changes are made with elevator deflection and altitude changes are made with rudder deflection. The mode switch, shown in Fig. 3.23, is based on roll angle feedback so that when the roll angle is above 45 degrees (less than -45 degrees if to the left), the knife-edge control mode is engaged. The transitions from level flight to knife edge and from knife edge back to level flight, in the level to knife edge transition block of Fig. 3.23, are controlled with an angular-rate controller similar to Fig. 3.22. The throttle control for knife-edge flight is a direct feed-through of the commands, which were held at full power throughout the maneuver. Figure 3.24 shows the control loops to maintain heading, roll angle, and altitude while flying knife edge and are contained in the knife edge block of Fig. 3.23. PI control was used to control heading with the same adjustment for turn direction as in Fig. 3.4. Roll-rate feedback is combined with PI control to maintain the roll angle during the maneuver and altitude error is used to generate a lateral acceleration command for the rudder loop. The saturation block in the rudder loop is used

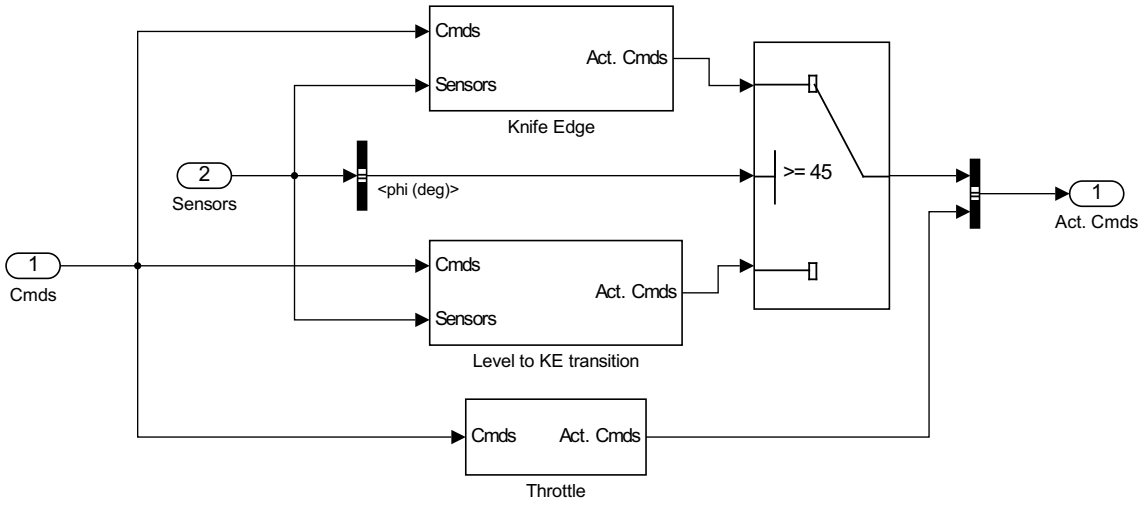


Figure 3.23: Knife-edge control mode-switching block diagram.

to compensate for the difference between the aircraft's ability to descend and its ability to climb. Due to the orientation of the airplane, a descent requires only a small fraction of the rudder deflection that is needed for a climb. The rudder deflection is limited in the down direction so as to prevent large drops in altitude followed by slow climbs when the altitude command may only be a few feet lower than the actual altitude.

The spin also uses a mode-switching mechanism to handle the change of controls between the initial stall portion of the maneuver and the actual spin itself. The current angle of attack is compared to the critical angle of attack to determine the control mode. The first time the angle of attack exceeds the critical angle, the controller switches from controlling the stall to controlling the spin. The spin controller initiates the spinning motion by commanding the pro-spin inputs of elevator, aileron, and rudder to achieve the same angular rates as recorded during the pilot-controlled spin. As with the other controllers, the throttle commands are passed directly through to the actuators. Figure 3.25 shows the spin control mode-switching diagram. The spin is controlled with an angular-rate controller similar to Fig. 3.22 with rate commands representative of the pilot-controlled data. The spin is stopped by commanding zero yaw rate before recovering to the original heading and speed. For the initial stall portion of the spin, the throttle is held constant while the elevator is used to decrease the speed.

During the stall, the ailerons attempt to keep the wings level and the rudder is used to maintain heading. The PI control loops during the spin are shown in Fig. 3.26. The aileron loop is the same as it was for knife-edge flight, where both roll angle and roll rate are fed back to generate the aileron command.

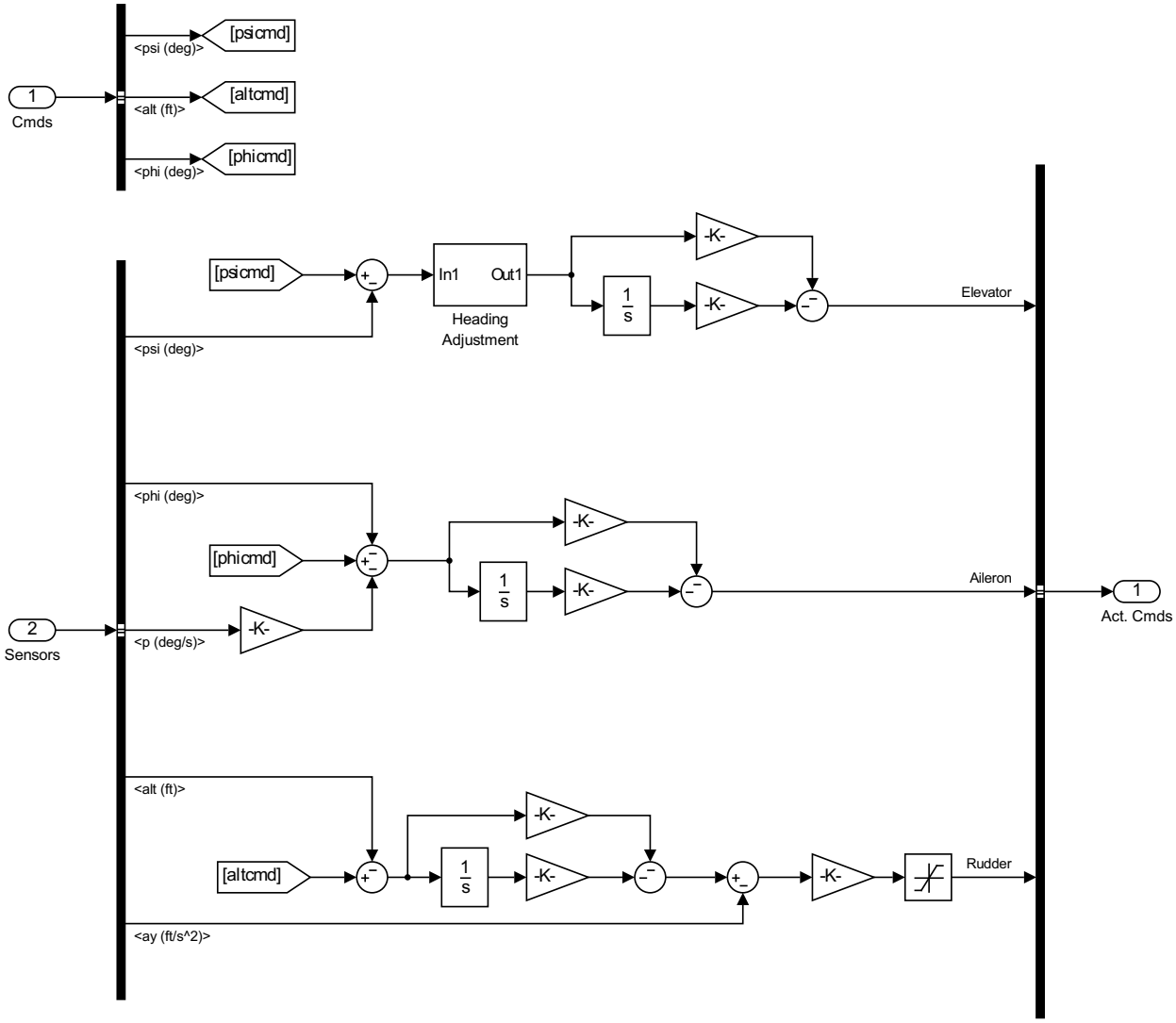


Figure 3.24: Knife-edge control block diagram.

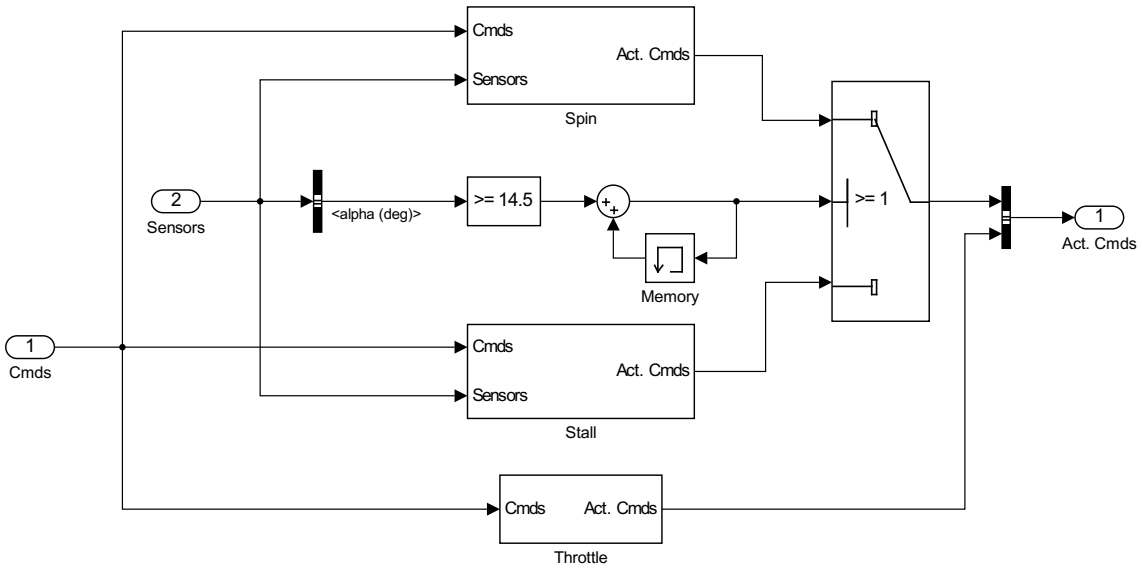


Figure 3.25: Spin control mode-switching block diagram.

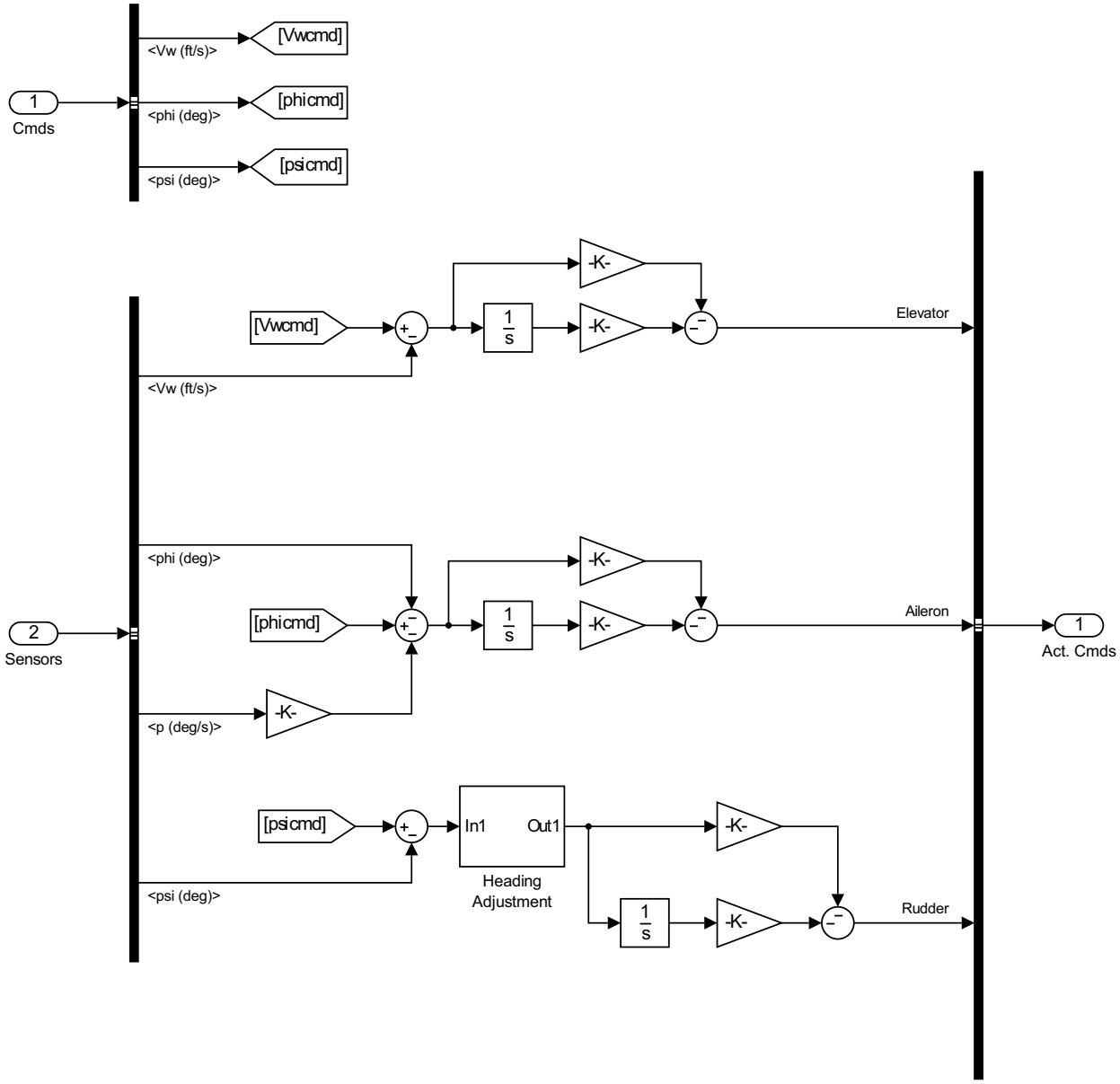


Figure 3.26: Stall control block diagram.

Chapter 4

Gain Calculation

This chapter presents the calculation of control gains for the control loop structures developed in the previous chapter. The classical root-locus method is applied to the cruise flight condition, followed by the application of the infinite-horizon optimal control method known as an LQT to cruise and maneuvering flight. The classical approach was applied to cruise flight to show the time and performance advantages that are achieved by employing well-studied optimal control techniques. The design approaches for both methods are given in the following sections while the simulation results using each method are presented in Chapter 5. This chapter concludes with the presentation of a gain scheduling approach used for maneuvering flight where the major advantage of LQT control design will be apparent.

4.1 Classical Approach

Root-locus analysis is a graphical method for examining how the roots of a polynomial change when a coefficient in the polynomial is varied and can be applied directly to the characteristic equation of a control system. The root-locus technique is used to determine when any poles become unstable or lightly damped by plotting the closed-loop transfer function as a function of a gain parameter. The root-locus method is a design technique for SISO systems which, for the aircraft control case, is being applied to a MIMO system. The application of the root-locus method to MIMO systems involves successive loop closures, essentially separating the MIMO system into a series of SISO systems. The classical approach to control design presented here involves using the root-locus plot to determine the control gains that provide adequate damping and closed-loop stability.

The control loops for cruise flight using the classical approach differ from those used with the LQT due to the SISO nature of the tool used for the design. Due to the structure of the state-space matrices in Appendix B, changes in engine speed from throttle inputs can not provide direct changes to the altitude. The only state effected by changes in engine speed is the forward speed u. Figure 4.1 shows the block diagram of the control loops for cruise flight using the classical design approach. Throttle is used to control speed through the feedback of the engine speed and elevator is used to control altitude. The altitude error

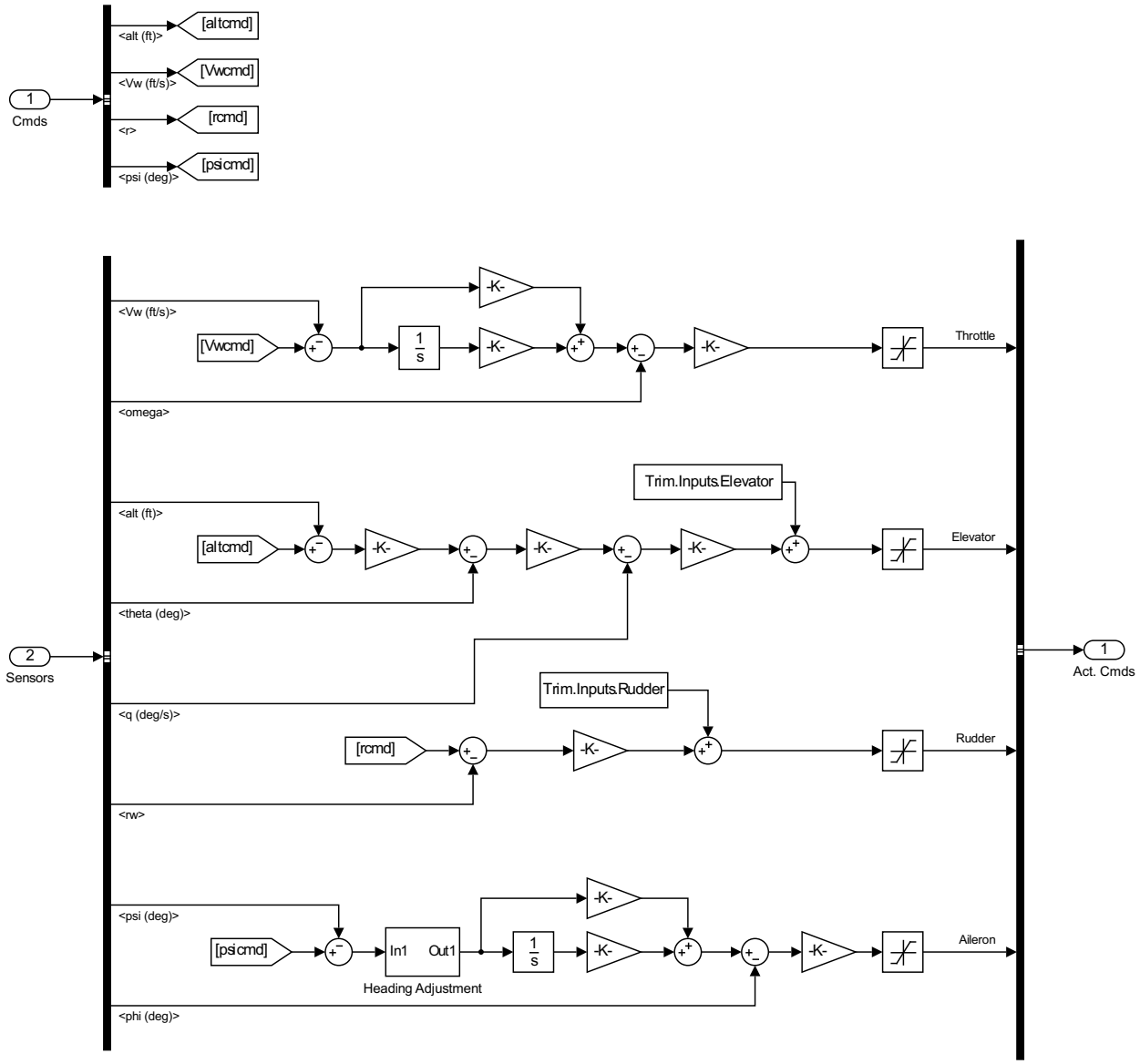


Figure 4.1: Classical-cruise control block diagram.

is used to generate a pitch angle command followed by a pitch rate command. The elevator then controls pitch rate through a proportional controller. The rudder loop is the same as in Fig. 3.3 and the aileron loop is similar with the differences being the exclusion of the roll rate feedback as well as the change to a PI controller. The trim values of elevator and rudder deflection used during the simulation of the classical approach are the same as determine in Sec. 2.4.4.

The state-space models of the aircraft in cruise are given in Appendix B and are used to determine the control gains for the cruise flight controller. In this section, the actuator models are excluded during the gain tuning process and the longitudinal and lateral states are separated. The actuator models were excluded during the classical approach to gain tuning as well as during the simulation of the classical controller to reduce the complexity of the comparison between the classical and optimal control design techniques. The longitudinal state vector is given by

$$\mathbf{x}_{long} = \begin{bmatrix} \theta & q & u & w & p_D & \omega_{\text{engine}} \end{bmatrix}^T \tag{4.1}$$

and the lateral state vector is given by

$$\mathbf{x}_{lat} = \begin{bmatrix} \phi & \psi & p & r & v & x_w \end{bmatrix}^T \tag{4.2}$$

The output vector for the longitudinal states is given by

$$\mathbf{y}_{long} = \begin{bmatrix} V_w \\ \theta \\ q \\ h \\ \omega_{\text{engine}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.997 & 0.07 & 0 & 0 \\ 57.2958 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{long} \tag{4.3}$$

and for the lateral states by

$$\mathbf{y}_{lat} = \begin{bmatrix} \phi \\ \psi \\ r_w \end{bmatrix} = \begin{bmatrix} 57.2958 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & -1 \end{bmatrix} \mathbf{x}_{lat} \tag{4.4}$$

The factors of 57.2958 convert radians to degrees.

4.1.1 Using the Linear Equations of Motion

The lateral-directional linear model determined from the EoM with AVL data is used to first determine the roll angle loop proportional gain. The open-loop transfer function from aileron deflection to roll angle was found to be

$$\frac{\phi}{\delta_a} = \frac{-82.92\,(s + 0.5557 \pm j3.2596)}{(s + 0.0084)\,(s + 0.6944 \pm j3.6629)\,(s + 7.7492)} \tag{4.5}$$

The dutch roll poles at $s = -0.6944 \pm j3.6629$ have a natural frequency of 3.73 rad/s and are lightly damped with a damping ratio of 0.19. The spiral pole is stable at $s = -0.0084$ with the roll subsidence pole at $s = -7.7492$. The root-locus plot for the roll angle loop is shown in Fig. 4.2 where the $\times$'s represent the poles and the $\circ$'s represent the zeros. The sign convention for aileron deflection means that a positive deflection leads to a negative rolling moment, so the gain value for the roll loop is negative. As the gain increases (becomes more negative) the spiral and roll subsidence poles come together to form a complex pair. The dutch roll poles move around an arc with increasing damping ratio before heading toward the complex zeros. The complex pair of poles was allowed so that the damping of the dutch roll mode could be increased as much as possible. The asterisks depict the location of the poles with $\mathrm{K}_\phi = -0.35$.

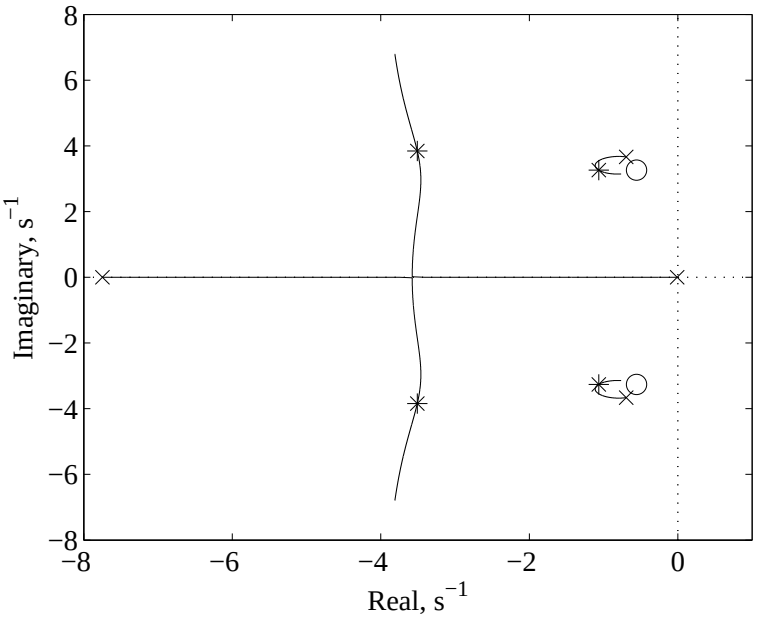


Figure 4.2: Root-locus plot for roll angle feedback.

The roll angle loop was closed and the transfer function from the reference command of roll angle to heading was found to be

$$\frac{\psi}{r_\phi} = \frac{-2.05\,(s - 3.4322)\,(s + 4.0269 \pm j1.23)}{(s + 3.508 \pm j3.8393)\,(s + 1.0652 \pm j3.2605)} \tag{4.6}$$

with a pole located at the origin corresponding to the heading mode. With the transfer function for a PI-compensator given as

$$G_c = \frac{K(s + z)}{s}$$

and with $z = 0.157$, the plant and compensator can be combined to produce the root-locus plot depicted in Fig. 4.3. The plot shows that as the heading gain is increased, the heading pole forms a complex pair with the compensator pole that circles around the compensator zero. The compensator zero was placed slightly to the left of the heading and compensator poles to prevent the complex pair formed by the two from moving toward the zero at $s = 3.4322$ resulting in an unstable complex pair. This was accomplished using MATLAB's

`sisotool` function to determine the effects on the root-locus plot as the location was varied. The natural frequency and the damping ratio of the dutch roll mode are both decreased with

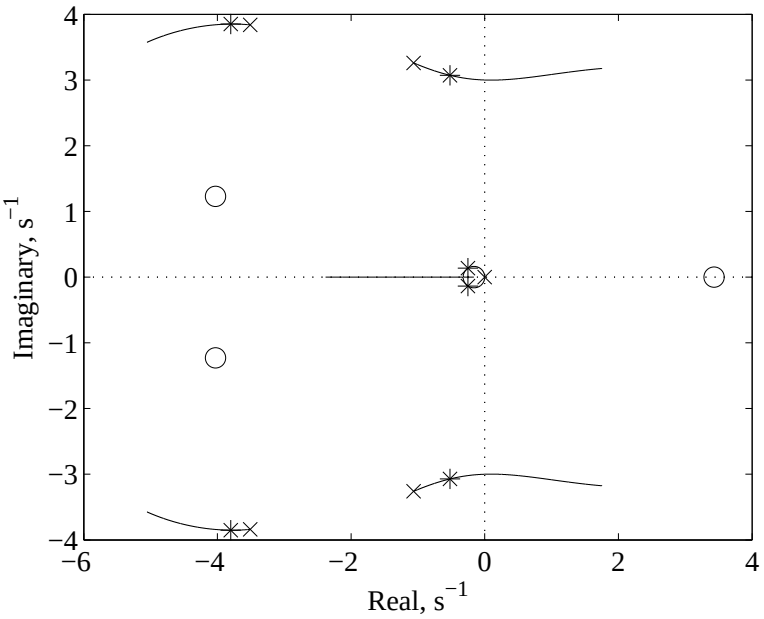


Figure 4.3: Root-locus plot for heading feedback.

increasing gain values. The asterisks show the location of the poles with $\mathrm{K}_{P_\psi} = 1.18$, with $\mathrm{K}_{I_\psi} = 0.185$ given by the location of the compensator zero.

After the heading loop was closed, the open-loop transfer function from rudder deflection to washed-out yaw rate was given by

$$\frac{r_w}{\delta_r} = \frac{-10.7\,(s + 3.7113 \pm j3.7451)\,(s + 0.6465)}{(s+1)\,(s + 3.8011 \pm j3.8514)\,(s + 0.5209 \pm j3.072)\,(s + 0.2512 \pm j0.1352)} \tag{4.7}$$

with three zeros located at the origin. Figure 4.4 shows the root-locus plot for the rudder loop. Just as with the ailerons, positive rudder deflection creates a negative yawing moment, so the rudder gain is negative. As the rudder gain is made more negative, the dutch roll poles follow an arc of approximately constant natural frequency and increasing damping ratio before heading toward the zeros at the origin. The rudder gain was chosen so that the dutch roll mode would be stable while changes in yaw rate would be quickly corrected. The asterisks again show the location of the poles with $\mathrm{K}_{r_w} = -5$.

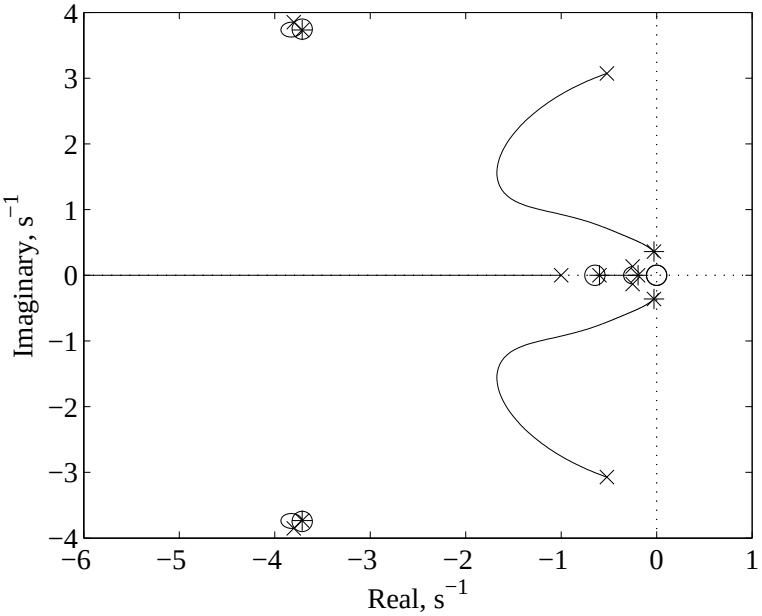


Figure 4.4: Root-locus plot for washed-out yaw rate feedback.

With the lateral-directional control loops complete, the longitudinal aircraft model was used to determine the gains for the elevator and throttle loops. The elevator to pitch rate transfer function was determined to be

$$\frac{q}{\delta_e} = \frac{-73.02\,(s+5.8512)\,(s+5.1488)\,(s+0.2681)\,(s-0.0001)}{(s+7.5745\pm j6.0777)\,(s+5.1406)\,(s+0.1095\pm j0.3619)\,(s-0.0004)} \tag{4.8}$$

The gain for the pitch rate loop is negative and the root-locus plot, Fig. 4.5, shows that as the gain is decreased the phugoid poles quickly approach the open-loop zeros. The short-period mode exhibits an increase in damping and natural frequency with increasing gains until the poles reach the real axis and become non-oscillatory. The pitch-rate gain was selected as -2.5 and the asterisks represent the pole locations of the closed-loop system.

The pitch-rate loop was closed and the closed-loop matrices were used to find the transfer function θ/r_q as

$$\frac{\theta}{r_q} = \frac{182.3\,(s+5.8512)\,(s+5.1488)\,(s+0.2681)\,(s-0.0001)}{(s+191.59)\,(s+6.0705)\,(s+5.1467)\,(s+0.2083)\,(s+0.0560)\,(s-0.0004)} \tag{4.9}$$

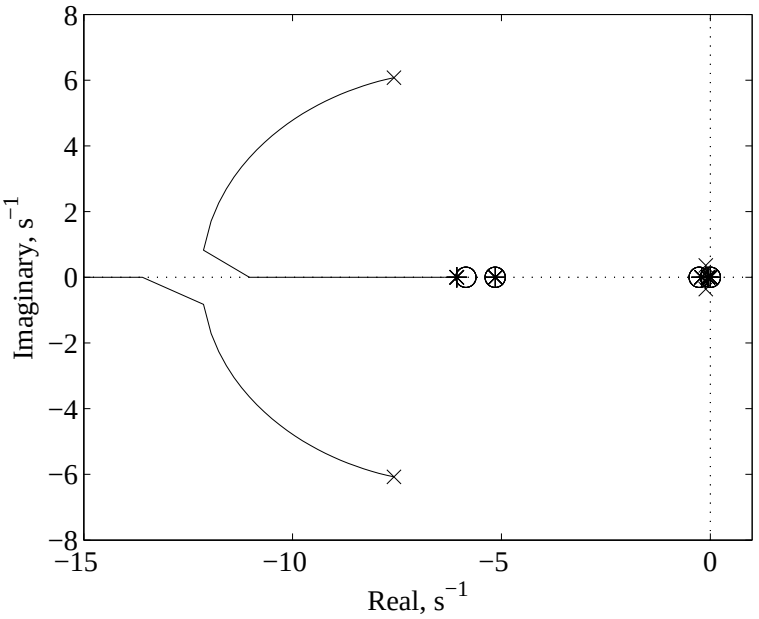


Figure 4.5: Root-locus plot for pitch-rate feedback.

Figure 4.6 shows the relatively small movement in the closed-loop poles associated with gain changes due to most of the poles being so close to the zeros.

Closing the pitch angle loop with $K_\theta = 6.5$ allowed the calculation of the altitude command to altitude transfer function given by

$$\frac{h}{r_h} = \frac{-9.668\,(s - 37.31)\,(s + 27.14)\,(s + 0.2133)\,(s + 5.1412)}{(s + 185.18)\,(s + 7.45)\,(s + 5.08 \pm j0.02)\,(s + 0.27)} \tag{4.10}$$

with the non-minimum-phase zero at $s = 37.31$ indicating that the initial response to a positive-step r_h (negative elevator deflection) will be a decrease in altitude. Figure 4.7 shows the root-locus plot used to determine the altitude gain as $K_h = 1.07$.

The final loop to be closed for the classical cruise controller is the throttle loop. Figure 4.8 shows the closed-loop step response of the engine speed with $K_{\omega_{\text{engine}}} = 0.02$. With the engine speed loop closed, the transfer function from the reference speed command to the airspeed was found to be

$$\frac{V_w}{r_{V_w}} = \frac{1.753\,(s + 8.8985)\,(s + 1.774 \pm j1.7899)\,(s + 1.1003)}{(s + 72.68)\,(s + 8.8952)\,(s + 1.7957 \pm j1.8469)\,(s + 0.3304)} \tag{4.11}$$

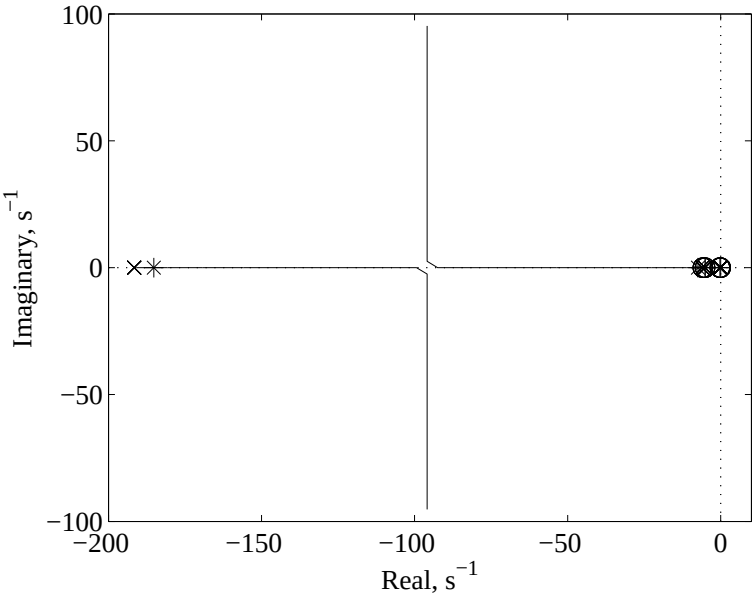


Figure 4.6: Root-locus plot for pitch angle feedback.

Combining the PI compensator transfer function with the compensator zero at $s = -1.1$ and the closed-loop engine speed model gives the speed loop for which the root-locus is plotted in Fig. 4.9. The compensator zero was placed to the left of the pole at $s = -0.3304$ to prevent the complex pair formed by it and the compensator pole from forming an unstable complex pair. The proportional speed gain was set to 7.27 which resulted in an integral gain of 8 based on the location of the compensator zero. To summarize, Table 4.1 lists the control gains determined in this section using the classical root-locus approach.

Table 4.1: Cruise gains determined using the EoM and the root-locus control design method.

Longitudinal Gain	Value	Lateral Gain	Value
K_{P_V}	7.27	K_{r_w}	-5.0
K_{I_V}	8.0	K_{P_ψ}	1.18
$K_{\omega_{\text{engine}}}$	0.02	K_{I_ψ}	0.185
K_h	1.07	K_ϕ	-0.35
K_θ	6.5		
K_q	-2.5		

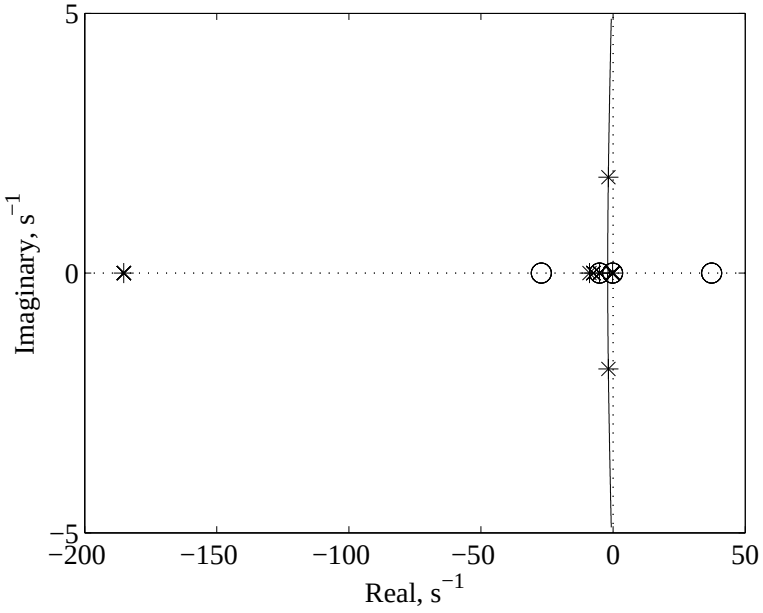


Figure 4.7: Root-locus plot for altitude feedback.

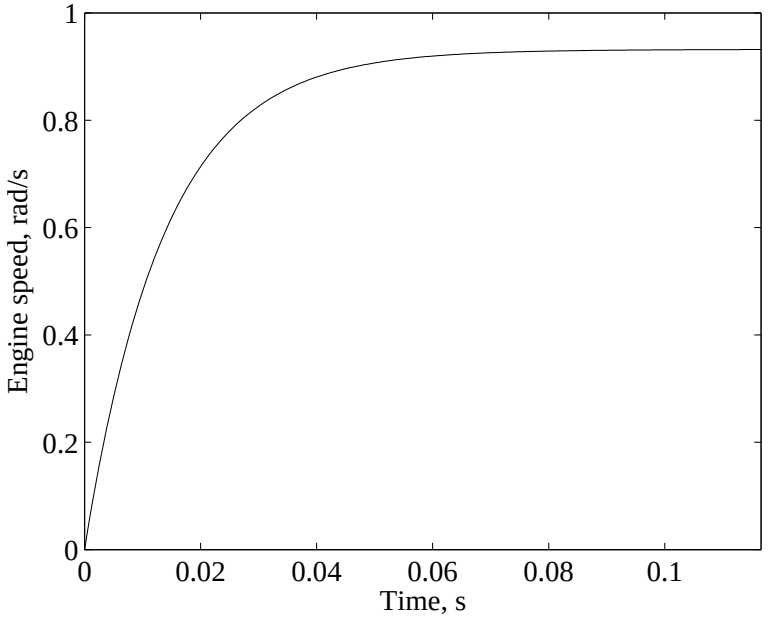


Figure 4.8: Engine speed step response.

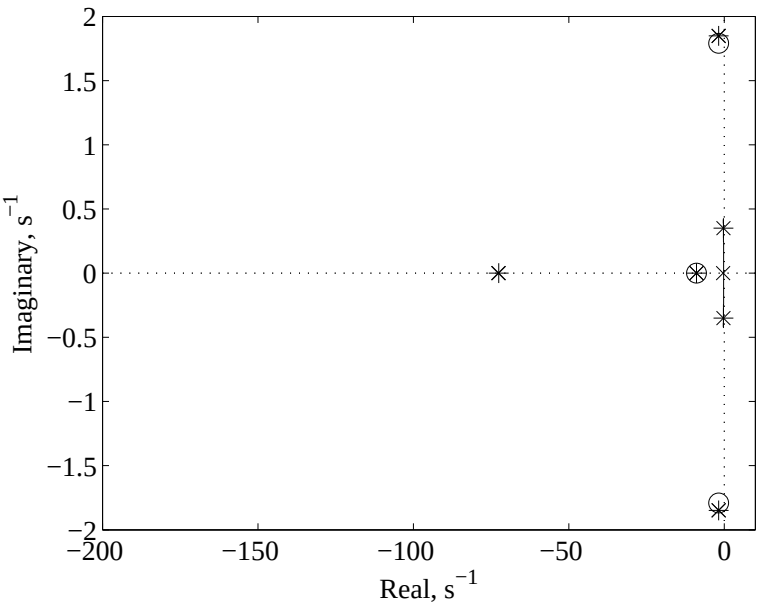
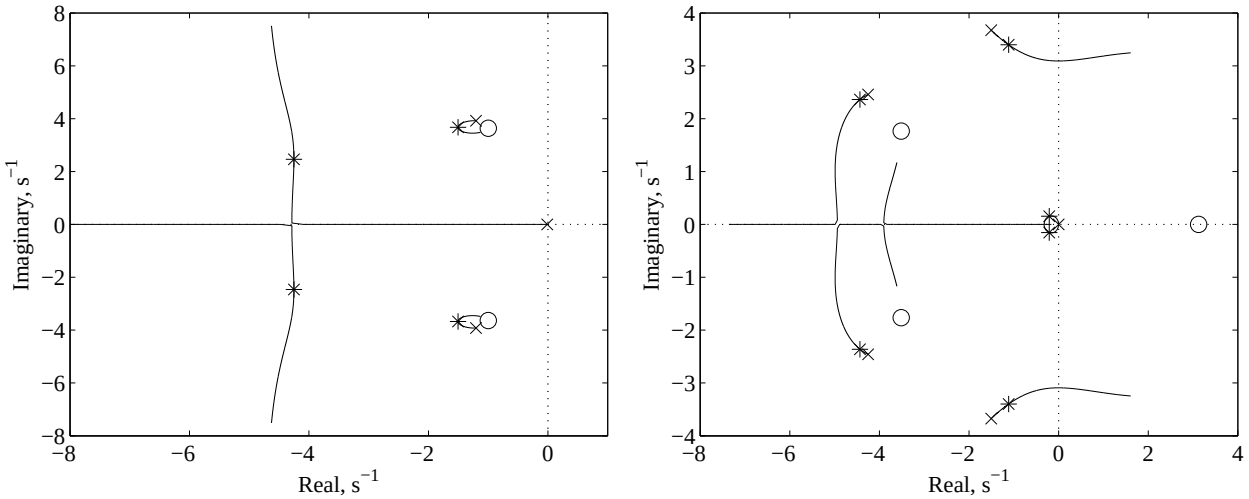


Figure 4.9: Root-locus plot for airspeed feedback.

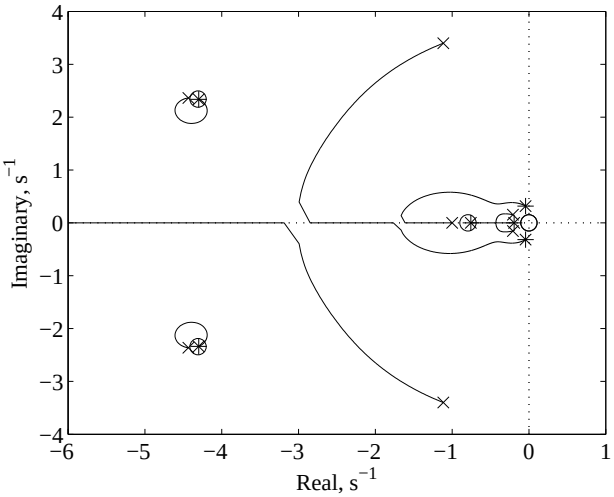
4.1.2 Using the Linearized Tunnel Data

The same root-locus approach presented in the previous subsection was used in combination with the aircraft model determined by linearizing the wind tunnel data to determine a second set of control gains for the cruise controller. Figure 4.10 presents the root-locus plots for the lateral-directional control loops. The roll angle gain was selected as -0.254 to allow for maximum damping of the dutch roll mode. The spiral and roll subsidence modes come together to form a complex pair, but are well damped. The zero in the PI compensator for



(a) Roll angle feedback

(b) Heading feedback



(c) Washed-out yaw rate feedback

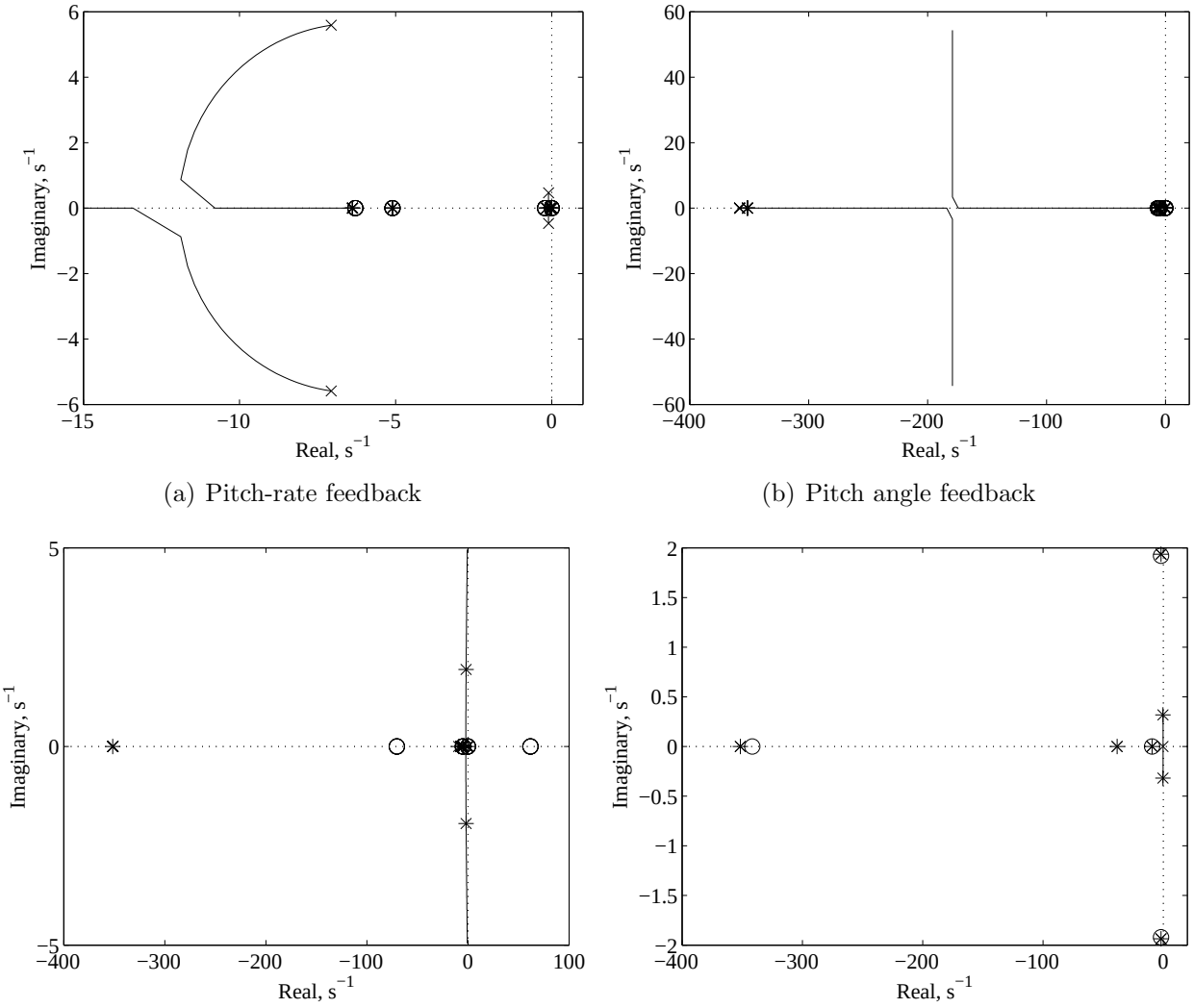
Figure 4.10: Root-locus plots for the lateral control loops.

the heading loop was located at $s = -0.152$, causing the heading pole to combine with the compensator pole to form a complex pair. The rudder gain was selected as -3.3 so that the dutch roll pole, which combined with the washout-filter pole to form a secondary complex pair, remained stable.

Figure 4.11 illustrates the root-locus plots for the longitudinal control loops and Fig. 4.12 shows the closed-loop step response of the engine speed. The short-period poles become non-oscillatory after the pitch-rate gain is increased beyond 0.11. In the case of the pitch angle loop, the closed-loop poles move very little due to the closeness of the open-loop zeros. The pitch angle gain would have to be greater than 91.5 for the poles to form a complex pair. The altitude gain was chosen to be 1.12 so that the complex poles that form would be adequately damped with a damping ratio of 0.7 and a natural frequency of 2.66 rad/s. The engine speed gain was selected so that the step response would have a fast rise time. With $\mathrm{K}_{\omega_{\mathrm{engine}}} = 0.01$, the engine speed reaches 80 percent of the commanded amplitude in only 0.07 seconds. Steady-state error was allowed in the engine speed response so that the initial response to changes in throttle position would be quick. The root-locus plot for the speed loop was found by combining the PI compensator with the closed-loop engine speed model. The compensator zero was located at $s = -0.911$ after trying a few different locations and plotting the step response. Table 4.2 summarizes the control gains determined using a model of the aircraft found by linearizing the wind tunnel data along with the classical root-locus design method.

Table 4.2: Cruise gains determined using the linearized tunnel data with the root-locus approach.

Longitudinal Gain	Value	Lateral Gain	Value
K_{P_V}	7.7	K_{r_w}	-3.3
K_{I_V}	7.0	K_{P_ψ}	0.99
$\mathrm{K}_{\omega_{\mathrm{engine}}}$	0.01	K_{I_ψ}	0.15
K_h	1.12	K_ϕ	-0.254
K_θ	6.6		
K_q	-4.0		



(a) Pitch-rate feedback

(b) Pitch angle feedback

(c) Altitude feedback

(d) Airspeed feedback

Figure 4.11: Root-locus plots for the longitudinal control loops.

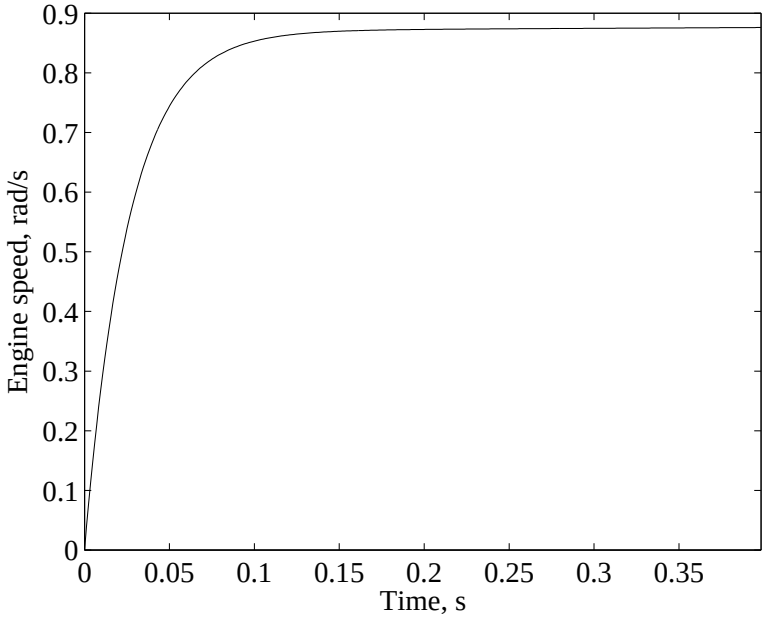


Figure 4.12: Step response of the engine speed controller.

4.2 Linear Quadratic Tracker

Unlike the classical control design approach, which is based on successive loop closure, modern techniques allow for the calculation of all the control gains simultaneously so that all the control loops can be closed at the same time. The control gain calculation is based on the state-variable model, which contains more information about the system than the SISO loops used in classical control design. This allows MIMO control systems to be designed with guaranteed closed-loop stability and good time responses in a quick and direct manner.

The modern control approach used in this work is based on the tracker problem outlined in [21,30], where it is desired for an aircraft to follow a reference command signal. This differs from the popular modern control technique known as a LQR, which attempts to minimize any non-zero perturbed states so that the system can be returned to its steady-state condition. The LQT technique regulates the states to follow the reference commands by minimizing the tracking error in Eq. 4.12, which is the difference between the reference command signals and the performance outputs of the system.

$$\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{z}(t) \tag{4.12}$$

The LQT approach allows controller dynamics of any desired structure to be incorporated into the system model. This allows the knowledge of classical aircraft control design to be used to form the compensator structure based on the application being investigated. Since the control approaches from classical aircraft control are robust to disturbances and uncertainties in the model, incorporating similar feedback structures should afford the same robustness to the LQT. Augmenting the aircraft dynamics with the compensator dynamics

results in state-space equations that are of the form

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} + \mathbf{G}\,\mathbf{r} \tag{4.13}$$

$$\mathbf{y} = \mathbf{C}\,\mathbf{x} + \mathbf{D}\,\mathbf{u} + \mathbf{F}\,\mathbf{r} \tag{4.14}$$

$$\mathbf{z} = \mathbf{H}\,\mathbf{x} \tag{4.15}$$

where $\mathbf{y}$ is the output available for feedback and $\mathbf{z}$ is the performance output used to calculate the tracking error. The control input is expressed as

$$\mathbf{u} = -\mathbf{K}\,\mathbf{y} \tag{4.16}$$

where $\mathbf{K}$ is the constant gain matrix to be determined. Using Eq. 4.16, the closed-loop output equation is found to be

$$\begin{aligned}\mathbf{y} &= (\mathbf{I} + \mathbf{D}\,\mathbf{K})^{-1}\,\mathbf{C}\,\mathbf{x} + (\mathbf{I} + \mathbf{D}\,\mathbf{K})^{-1}\,\mathbf{F}\,\mathbf{r} \\ &\equiv \mathbf{C}_c\,\mathbf{x} + \mathbf{D}_c\,\mathbf{r}\end{aligned} \tag{4.17}$$

and the closed-loop state equation is

$$\begin{aligned}\dot{\mathbf{x}} &= (\mathbf{A} - \mathbf{B}\,\mathbf{K}\,\mathbf{C}_c)\,\mathbf{x} + (\mathbf{G} - \mathbf{B}\,\mathbf{K}\,\mathbf{D}_c)\,\mathbf{r} \\ &\equiv \mathbf{A}_c\,\mathbf{x} + \mathbf{B}_c\,\mathbf{r}\end{aligned} \tag{4.18}$$

At this point, it is important to mention a few significant things about the LQT approach used in this work. First, if Eq. 4.17 is substituted into Eq. 4.16, the resulting control input will contain both feedback and feedforward terms. This means that both the closed-loop poles and compensator zeros may be affected by varying the gain matrix $\mathbf{K}$, giving better results than by placing only the poles [21]. Second, the reference commands in the reference input matrix $\mathbf{r}$ are assumed to be step commands. This assumption is based on the performance

specifications of most autopilots being in the time domain and is proven valid by Stevens and Lewis [21] for any arbitrary reference signal. Finally, the closed-loop behavior of a control system designed using a linear quadratic approach depends directly on the performance criterion used during the design. Due to this fact, the actual control engineering takes place during the determination of the control loops and while selecting the performance criterion.

4.2.1 Standard Performance Criterion

To determine the control gains, the LQT employs a quadratic performance criterion in the time domain that is easily modifiable to achieve different performance objectives. The optimal gains are determined by minimizing the performance criterion, or cost function, which integrates the deviations in both the states and inputs with respect to time over an infinite horizon. The impact of each state and input is weighted in the cost function through the use of weighting matrices. The standard performance criterion for the LQT is

$$J = \frac{1}{2}\int_0^\infty \left(\tilde{\mathbf{x}}^T\,\mathbf{Q}\,\tilde{\mathbf{x}} + \tilde{\mathbf{u}}^T\,\mathbf{R}\,\tilde{\mathbf{u}}\right) dt \tag{4.19}$$

where $\mathbf{Q}$ is positive semidefinite ($\mathbf{Q} \geq 0$) and $\mathbf{R}$ is positive definite ($\mathbf{R} > 0$) to ensure that the cost is non-negative. The tildes represent deviations from the steady-state values of the states and inputs with the closed-loop dynamics of the state deviation given by

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}_c\,\tilde{\mathbf{x}} \tag{4.20}$$

$$\tilde{\mathbf{y}} = \mathbf{C}\,\tilde{\mathbf{x}} \tag{4.21}$$

$$\tilde{\mathbf{z}} = \mathbf{H}\,\tilde{\mathbf{x}} \tag{4.22}$$

and the control deviation by

$$\tilde{\mathbf{u}} = -\mathbf{K}\,\tilde{\mathbf{y}} \tag{4.23}$$

Since the closed-loop plant will be asymptotically stable, Eq. 4.18 can be written at steady state (denoted with a bar accent) as

$$0 = \mathbf{A}_c\,\bar{\mathbf{x}} + \mathbf{B}_c\,\bar{\mathbf{r}} \tag{4.24}$$

so the steady-state response is

$$\bar{\mathbf{x}} = -\mathbf{A}_c^{-1}\,\mathbf{B}_c\,\bar{\mathbf{r}} \tag{4.25}$$

Equation 4.23 can be substituted into Eq. 4.19 to show that

$$J = \frac{1}{2}\int_0^\infty \left[\tilde{\mathbf{x}}^T\left(\mathbf{Q} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}}\right] dt \tag{4.26}$$

From Stevens and Lewis [21], if it is assumed that a constant, symmetric, positive-semidefinite matrix $\mathbf{P}$ can be found so that

$$\frac{d}{dt}\left(\tilde{\mathbf{x}}^T\,\mathbf{P}\,\tilde{\mathbf{x}}\right) = \tilde{\mathbf{x}}^T\left(\mathbf{Q} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}} \tag{4.27}$$

then J may be written as

$$J = \frac{1}{2}\,\tilde{\mathbf{x}}^T(0)\,\mathbf{P}\,\tilde{\mathbf{x}}(0) - \frac{1}{2}\lim_{t\to\infty}\,\tilde{\mathbf{x}}^T(t)\,\mathbf{P}\,\tilde{\mathbf{x}}(t) \tag{4.28}$$

Assuming the closed-loop system is asymptotically stable, the limit will vanish with time, leaving

$$J = \frac{1}{2}\,\tilde{\mathbf{x}}^T(0)\,\mathbf{P}\,\tilde{\mathbf{x}}(0) \tag{4.29}$$

Since the plant starts from rest, the deviations in the states when $t = 0$ will be equal and opposite to the steady-state values so that the optimal cost becomes

$$J = \frac{1}{2}\,\bar{\mathbf{x}}^T\,\mathbf{P}\,\bar{\mathbf{x}} = \frac{1}{2}\,\mathrm{trace}(\mathbf{P}\,\mathbf{X}) \tag{4.30}$$

where

$$\mathbf{X} \equiv \bar{\mathbf{x}}\,\bar{\mathbf{x}}^T \tag{4.31}$$

A constraint equation to determine $\mathbf{P}$ is found by substituting Eq. 4.20 into Eq. 4.27 to get

$$\begin{aligned}\tilde{\mathbf{x}}^T \left(\mathbf{Q} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}} &= \dot{\tilde{\mathbf{x}}}^T\,\mathbf{P}\,\tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T\,\mathbf{P}\,\dot{\tilde{\mathbf{x}}} \\ &= \tilde{\mathbf{x}}^T \left(\mathbf{A}_c^T\,\mathbf{P} + \mathbf{P}\,\mathbf{A}_c\right)\tilde{\mathbf{x}}\end{aligned} \tag{4.32}$$

which must hold for all $\tilde{\mathbf{x}}(t)$, leaving

$$0 = g \equiv \mathbf{A}_c^T\,\mathbf{P} + \mathbf{P}\,\mathbf{A}_c + \mathbf{Q} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C} \tag{4.33}$$

The optimal control gains are then found by minimizing the cost function in Eq. 4.30 subject to the constraint in Eq. 4.33 and Eq. 4.31. Bryson and Ho [88] give a convenient guideline for selecting the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ and for most cases $\mathbf{Q} = \mathbf{H}^T\,\mathbf{H}$ can be assumed. In order for the minimization to converge, however, all of the states have to be weighted in the cost function due to the observability condition presented in [21]. The weighting matrices then become design parameters, whose elements may not correspond directly to the performance objectives, determined in an iterative process based on the time response to yield good results.

4.2.2 Modifying the Performance Criterion

The standard performance criterion of Eq. 4.26 only lightly penalizes small errors due to a slow pole(s). This could result in a control system design whose step response is fast, with minimal overshoot, but may require a considerable amount of time to fully reach the steady-state value. By time-weighting the state deviations in the performance criterion, a heavy penalty can be placed on errors occurring late in the response. This helps to suppress the effect of a slow pole and eliminate lightly damped settling behavior [21]. The time-weighted

performance criterion is then given by

$$J = \frac{1}{2}\int_0^\infty \left[t^k\,\tilde{\mathbf{x}}^T\,\mathbf{P}\,\tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T\left(\mathbf{Q} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}}\right] dt \tag{4.34}$$

where both $\mathbf{P}$ and $\mathbf{Q}$ are $n \times n$ matrices.

To simplify the selection of $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$, and to avoid the iterative design process associated with selecting their entries, the performance criterion will be written in terms of the error deviation and the control deviation as

$$J = \frac{1}{2}\int_0^\infty \left(t^k\,\tilde{\mathbf{e}}^T\,\tilde{\mathbf{e}} + \varrho\,\tilde{\mathbf{u}}^T\,\tilde{\mathbf{u}}\right) dt \tag{4.35}$$

where the error deviation is

$$\tilde{e} = -\mathbf{H}\,\tilde{\mathbf{x}} \tag{4.36}$$

This results in a control system which will attempt to minimize the error without using a great deal of control effort. The design variables are also reduced to only two design parameters (k and ϱ) because Eq. 4.35 is the same as Eq. 4.34 with $\mathbf{P} = \mathbf{H}^T\,\mathbf{H}$, $\mathbf{Q} = 0$, and $\mathbf{R} = \varrho\,\mathbf{I}$. The control weighting variable ϱ adjusts the authority given to the control system with smaller values allowing more aggressive control and larger values placing a greater penalty on the control effort resulting in less authority.

The solution of the time-weighted performance criterion is found by successively integrating by parts. Equation 4.35 can be rewritten as

$$J = \frac{1}{2}\int_0^\infty \left[t^k\,\tilde{\mathbf{x}}^T\,\mathbf{P}\,\tilde{\mathbf{x}}\right] dt + \int_0^\infty \left[\tilde{\mathbf{x}}^T\left(\mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}}\right] dt \tag{4.37}$$

Using the method of integration by parts with $u = t^k$ and $dv = \tilde{\mathbf{x}}^T\,\mathbf{P}\,\tilde{\mathbf{x}}$ and defining

$$-\frac{d}{dt}\left(\tilde{\mathbf{x}}^T\,\mathbf{P}\,\tilde{\mathbf{x}}\right) = \tilde{\mathbf{x}}^T\,\mathbf{P}_0\,\tilde{\mathbf{x}} \tag{4.38}$$

the cost can be expressed as

$$J = \frac{1}{2}\left[-t^k\,\tilde{\mathbf{x}}^T\,\mathbf{P}_0\,\tilde{\mathbf{x}}\Big|_0^\infty + \int_0^\infty \left(k\,t^{k-1}\,\tilde{\mathbf{x}}^T\,\mathbf{P}_0\,\tilde{\mathbf{x}}\right)dt\right] + \frac{1}{2}\int_0^\infty \left[\tilde{\mathbf{x}}^T\left(\mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}}\right]dt \quad (4.39)$$

which for an asymptotically stable system simplifies to

$$J = \frac{1}{2}\int_0^\infty \left(k\,t^{k-1}\,\tilde{\mathbf{x}}^T\,\mathbf{P}_0\,\tilde{\mathbf{x}}\right)dt + \frac{1}{2}\int_0^\infty \left[\tilde{\mathbf{x}}^T\left(\mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}}\right]dt \quad (4.40)$$

As the process of integrating by parts is repeated, the cost is found to be

$$J = \frac{1}{2}\int_0^\infty \left[\prod_{n=0}^{k-1}(k-n)\,\tilde{\mathbf{x}}^T\,\mathbf{P}_{k-1}\,\tilde{\mathbf{x}}\right]dt + \frac{1}{2}\int_0^\infty \left[\tilde{\mathbf{x}}^T\left(\mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\,\tilde{\mathbf{x}}\right]dt \quad (4.41)$$

which can be rewritten as

$$J = \frac{1}{2}\int_0^\infty \left[\tilde{\mathbf{x}}^T\left(k!\,\mathbf{P}_{k-1} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\tilde{\mathbf{x}}\right]dt \quad (4.42)$$

By defining

$$-\frac{d}{dt}\left(\tilde{\mathbf{x}}^T\,\mathbf{P}_k\,\tilde{\mathbf{x}}\right) = \tilde{\mathbf{x}}^T\left(k!\,\mathbf{P}_{k-1} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C}\right)\tilde{\mathbf{x}} \quad (4.43)$$

Equation 4.42 can be evaluated at the limits of integration and for a stable system, the value of the cost can be written as

$$J = \frac{1}{2}\,\tilde{\mathbf{x}}^T(0)\,\mathbf{P}_k\,\tilde{\mathbf{x}}(0) = \frac{1}{2}\,\bar{\mathbf{x}}^T\,\mathbf{P}_k\,\bar{\mathbf{x}} = \frac{1}{2}\,\mathrm{trace}(\mathbf{P}_k\,\mathbf{X}) \quad (4.44)$$

subject to Eq. 4.31 and the constraint equations found from Eqs. 4.38 and 4.43. After substitution of Eq. 4.20 ($\dot{\tilde{\mathbf{x}}} = \mathbf{A}_c\,\tilde{\mathbf{x}}$) into Eqs. 4.38 and 4.43, the nested Lyapunov equations

for the time-weighted cost function are found to be

$$\begin{aligned} 0 = \quad g_0 \quad &\equiv \mathbf{A}_c^T\,\mathbf{P}_0 + \mathbf{P}_0\,\mathbf{A}_c + \mathbf{P} \\ 0 = \quad g_1 \quad &\equiv \mathbf{A}_c^T\,\mathbf{P}_1 + \mathbf{P}_1\,\mathbf{A}_c + \mathbf{P}_0 \\ &\vdots \\ 0 = g_{k-1} &\equiv \mathbf{A}_c^T\,\mathbf{P}_{k-1} + \mathbf{P}_{k-1}\,\mathbf{A}_c + \mathbf{P}_{k-2} \\ 0 = \quad g_k \quad &\equiv \mathbf{A}_c^T\,\mathbf{P}_k + \mathbf{P}_k\,\mathbf{A}_c + k!\,\mathbf{P}_{k-1} + \mathbf{C}^T\,\mathbf{K}^T\mathbf{R}\,\mathbf{K}\,\mathbf{C} \end{aligned} \tag{4.45}$$

which can be solved to find $\mathbf{P}_k$ in Eq. 4.44. The MATLAB function file created to compute the time-weighted cost is given in Appendix C to clarify the use of the equations in this subsection.

A nice benefit of the time-weighted performance criterion is that the observability problem may be corrected for any $k > 0$. This is because even though all of the states may not be observable in $\mathbf{P}$, they may all be observable in $k!\,\mathbf{P}_{k-1}$ [21]. If so, the last Lyapunov equation in 4.45 will have a positive definite solution $\mathbf{P}_k$, correcting the observability problem [21]. Since the maneuvering controller contains integral terms to minimize the steady-state error, the observability correction is important as the angular-rate states can never give information about the integral states in the open-loop configuration. The time-weighting exponent can be selected as any non-negative integer, but experience from working with multiple examples in [21] resulted in the decision to set $k = 2$ for this work. Experiments were performed with multiple values of ϱ, with larger values placing a greater penalty on the control effort. Good results were obtained with $\varrho = 1$, so for this research, the control weighting $\mathbf{R}$ was set equal to the identity matrix.

4.2.3 Initial Stabilizing Gain Matrix

The optimal control gains are determined using a Simplex minimization routine to evaluate the cost function for a specific value of the gain matrix $\mathbf{K}$. The minimization was

accomplished in this work through the use of MATLAB's `fminsearch` function, which is based on the method of Lagarias et al. [89]. Since the minimization is an iterative process, an initial estimate of a gain matrix that stabilizes the closed-loop system is needed. Multiple methods could be used to compute the initial gain matrix, including the classical technique of Sec. 4.1, the eigenstructure assignment technique as used in [90], and the control design functionality of MATLAB's Control System Toolbox.

The initial gain matrices for this work were determined using the `fminsearch` function to minimize the maximum real part of the eigenvalues of the closed-loop state matrix. This minimization was performed until the maximum real part of the eigenvalues was less than -0.01. While this method will always result in a gain matrix that stabilizes the closed-loop system, its dependance on the Simplex algorithm means that it also needs a starting point. Table 4.3 lists the matrices used to begin the initialization process for each controller. To obtain the starting matrices, knowledge of the signs corresponding to the static loop sensitivity of the transfer functions was used with a trial-and-error method to determine what starting matrix would yield an initial stabilizing gain matrix for the LQT minimization.

Table 4.3: Matrices used to begin the initialization process for the LQT minimization.

Controller	Starting Gain Matrix
Cruise Flight	$0.01 \begin{bmatrix} 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
Angular Rates	$\begin{bmatrix} -1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
Knife-edge Flight	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
Stall	$\begin{bmatrix} 1 & -1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

4.2.4 LQT Derivation for Cruise

Recall that the plant matrices of the aircraft in cruise flight are given in Appendix B. Using the control-loop structure defined in Sec. 3.1 and shown in Figs. 3.2 and 3.3, the

control inputs during cruise flight can be written as

$$\delta_e = K_{P_V}\, e_V + K_{I_V} \int e_V\, dt + K_{D_V}\, \dot{e}_V \tag{4.46}$$

$$\delta_a = K_\phi \left(r_\phi - \mathbf{C}_\phi\, \mathbf{x}_{plant} - K_p\, \mathbf{C}_p\, \mathbf{x}_{plant} \right) \tag{4.47}$$

$$\delta_r = K_{r_w} \left(r_{r_w} - \mathbf{C}_{r_w}\, \mathbf{x}_{plant} \right) \tag{4.48}$$

$$\delta_T = K_{P_h}\, e_h + K_{I_h} \int e_h\, dt + K_{D_h}\, \dot{e}_h \tag{4.49}$$

where

$$r_\phi = K_{P_\psi}\, e_\psi + K_{I_\psi} \int e_\psi\, dt + K_{D_\psi}\, \dot{e}_\psi \tag{4.50}$$

The state matrix is augmented with three integrators of the form

$$\dot{\boldsymbol{w}} = \begin{bmatrix} \dot{w}_V \\ \dot{w}_h \\ \dot{w}_\psi \end{bmatrix} = \begin{bmatrix} e_V \\ e_h \\ e_\psi \end{bmatrix} = \begin{bmatrix} r_V - \mathbf{C}_V\, \mathbf{x}_{plant} \\ r_h - \mathbf{C}_h\, \mathbf{x}_{plant} \\ r_\psi - \mathbf{C}_\psi\, \mathbf{x}_{plant} \end{bmatrix} \tag{4.51}$$

which allows the dynamics to be written in the form of Eqn. 4.13 as

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_{plant} & 0 & 0 & 0 \\ -\mathbf{C}_V & 0 & 0 & 0 \\ -\mathbf{C}_h & 0 & 0 & 0 \\ -\mathbf{C}_\psi & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{B}_{plant} \\ 0 \quad 0 \quad 0 \quad 0 \\ 0 \quad 0 \quad 0 \quad 0 \\ 0 \quad 0 \quad 0 \quad 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{r} \tag{4.52}$$

with

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{plant} & w_V & w_h & w_\psi \end{bmatrix}^T \tag{4.53}$$

$\mathbf{u}$ from Eq. 2.97 and

$$\boldsymbol{r} = \begin{bmatrix} r_V & r_\psi & r_{r_w} & r_h \end{bmatrix}^T \tag{4.54}$$

The reference commands will be step inputs, allowing their derivatives to be assumed constant. The tracking-error derivatives with respect to time can then be expressed as

$$\dot{e}_V = -\mathbf{C}_V \left(\mathbf{A}_{plant}\, \mathbf{x}_{plant} + \mathbf{B}_{plant}\, \mathbf{u}\right) \tag{4.55}$$

$$\dot{e}_h = -\mathbf{C}_h \left(\mathbf{A}_{plant}\, \mathbf{x}_{plant} + \mathbf{B}_{plant}\, \mathbf{u}\right) \tag{4.56}$$

$$\dot{e}_\psi = -\mathbf{C}_\psi \left(\mathbf{A}_{plant}\, \mathbf{x}_{plant} + \mathbf{B}_{plant}\, \mathbf{u}\right) \tag{4.57}$$

Rewriting the control inputs of Eqs. 4.46 - 4.49 and accounting for negative feedback (defined in Eq. 4.16) results in

$$-\delta_e = K_{P_V} \left(\mathbf{C}_V\, \mathbf{x}_{plant} - r_V\right) - K_{I_V}\, w_V + K_{D_V}\, \mathbf{C}_V \left(\mathbf{A}_{plant}\, \mathbf{x}_{plant} + \mathbf{B}_{plant}\, \mathbf{u}\right) \tag{4.58}$$

$$-\delta_a = K_\phi \left(\mathbf{C}_\phi\, \mathbf{x}_{plant} + K_p\, \mathbf{C}_p\, \mathbf{x}_{plant} - r_\phi\right) \tag{4.59}$$

$$-\delta_r = K_{r_w} \left(\mathbf{C}_{r_w}\, \mathbf{x}_{plant} - r_{r_w}\right) \tag{4.60}$$

$$-\delta_T = K_{P_h} \left(\mathbf{C}_h\, \mathbf{x}_{plant} - r_h\right) - K_{I_h}\, w_h + K_{D_h}\, \mathbf{C}_h \left(\mathbf{A}_{plant}\, \mathbf{x}_{plant} + \mathbf{B}_{plant}\, \mathbf{u}\right) \tag{4.61}$$

The output equation then becomes

$$\mathbf{y} = \begin{bmatrix} \mathbf{C}_V & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ \mathbf{C}_V\,\mathbf{A}_{plant} & 0 & 0 & 0 \\ \mathbf{C}_\phi + K_p\,\mathbf{C}_p + K_{P_\psi}\,\mathbf{C}_\psi + K_{D_\psi}\,\mathbf{C}_\psi\,\mathbf{A}_{plant} & 0 & 0 & -K_{I_\psi} \\ \mathbf{C}_{r_w} & 0 & 0 & 0 \\ \mathbf{C}_h & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \mathbf{C}_h\,\mathbf{A}_{plant} & 0 & 0 & 0 \end{bmatrix} \mathbf{x}$$
$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mathbf{C}_V\,\mathbf{B}_{plant} \\ K_{D_\psi}\,\mathbf{C}_\psi\,\mathbf{B}_{plant} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mathbf{C}_h\,\mathbf{B}_{plant} \end{bmatrix} \mathbf{u} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -K_{P_\psi} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{r} \tag{4.62}$$

with the performance output

$$\mathbf{z} = \begin{bmatrix} V \\ \psi \\ r_w \\ h \end{bmatrix} = \begin{bmatrix} \mathbf{C}_V & 0 & 0 & 0 \\ \mathbf{C}_\psi & 0 & 0 & 0 \\ \mathbf{C}_{r_w} & 0 & 0 & 0 \\ \mathbf{C}_h & 0 & 0 & 0 \end{bmatrix} \mathbf{x} \tag{4.63}$$

Using Eqs. 4.58 - 4.61, the gain matrix is determined to be

$$\mathbf{K} = \begin{bmatrix} K_{P_V} & K_{I_V} & K_{D_V} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_\phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{r_w} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{P_h} & K_{I_h} & K_{D_h} \end{bmatrix} \tag{4.64}$$

To account for the four gain terms that appear in the output equation as well as to constrain the zero entries of the gain matrix to remain exactly zero, a new gain vector containing only the non-zero terms from Eq. 4.64 is defined for use in the optimization routine. With each evaluation of the cost function, the state and output equation matrices from Eqs. 4.52 and 4.62 are recalculated based on the updated value of the optimization matrix $\widehat{\mathbf{K}}$ in Eq. 4.65. The gain matrix in Eq. 4.64 is also reformed based on the updated gain values before all the control loops are closed. Referencing the cost function file in Appendix C, the function `matrices` recalculates the state-space matrices and the function `CLmatrices` closes the control loops based on the updated values in $\widehat{\mathbf{K}}$.

$$\widehat{\mathbf{K}} = \begin{bmatrix} K_{P_V} & K_{I_V} & K_{D_V} & K_\phi & K_p & K_{P_\psi} & K_{I_\psi} & K_{D_\psi} & K_{r_w} & K_{P_h} & K_{I_h} & K_{D_h} \end{bmatrix} \tag{4.65}$$

Figures 4.13 - 4.15 show the closed-loop step response plots using the linear aircraft model with actuator models, obtained by linearizing the wind tunnel data. Similar plots for the linear model obtained using AVL data with the EoM are shown in Figures 4.16 - 4.18. Based on the linear analysis, good performance should be expected with the gains from both methods. Using the linear models from the wind tunnel data, the airspeed step response rises to 75 percent of the commanded value in around two seconds before reaching the steady-state value after five seconds. The altitude response to a step input of airspeed is just as one would expect based on the elevator pitching the nose down to increase speed. Initially, there is a loss in altitude, but only four seconds pass before the altitude error is

returned to near zero. Similar results are obtained for the airspeed step response when using the linear model from the EoM. The airspeed and altitude stabilize after five seconds with the heading and washed-out yaw rate returning to near zero after 10 seconds. The off-axis response of heading and yaw rate during step commands of airspeed is due to the changing propulsive moments experienced during the initial decrease in altitude that occurs as the speed increases. The altitude decrease is caused by the elevator pitching the nose down to increase the speed. The nose-down orientation produces a decrease in altitude before the throttle is able to return the altitude back to the trimmed value. The changing throttle settings produce varying moments in roll and yaw which result in slight heading and yaw rate oscillations.

The control gains for both linear aircraft models also produced good step responses from changes in altitude. With the model from the tunnel data, the altitude response has a small overshoot at two seconds before reaching the steady state after four seconds. Due to the altitude being controlled by the throttle, a small oscillation in airspeed occurs with the altitude change, but is damped out after three seconds. The overshoot in altitude is slightly higher when using the linear model from the EoM, but still results in a good response by reaching steady state in five seconds. As mentioned in regard to the airspeed step response, the off-axis oscillations in heading and yaw rate during step commands of altitude are due to the changing propulsive moments encountered as a result of the changing throttle settings.

With both linear models, the initial response to a change in heading is opposite the command. This natural and undesirable tendency is referred to as adverse yaw and is caused by the difference in drag between a downward deflected aileron and an upward deflected aileron [72]. The wing with the downward deflected aileron experiences an increase in lift which results in an increase in induced drag. The wing with the upward deflected aileron experiences less lift resulting in a decrease in induced drag. The difference in induced drag causes a yawing moment in the opposite direction from the rolling moment due to the aileron deflection and results in an initial response opposite the command. Using the model from the

tunnel data, the heading oscillates slightly during the response while the response from the EoM model is more direct. The response from both models results in a heading change that only takes about five seconds to complete. During the heading change, only slight changes in speed and altitude occur before returning to zero after 10 seconds.

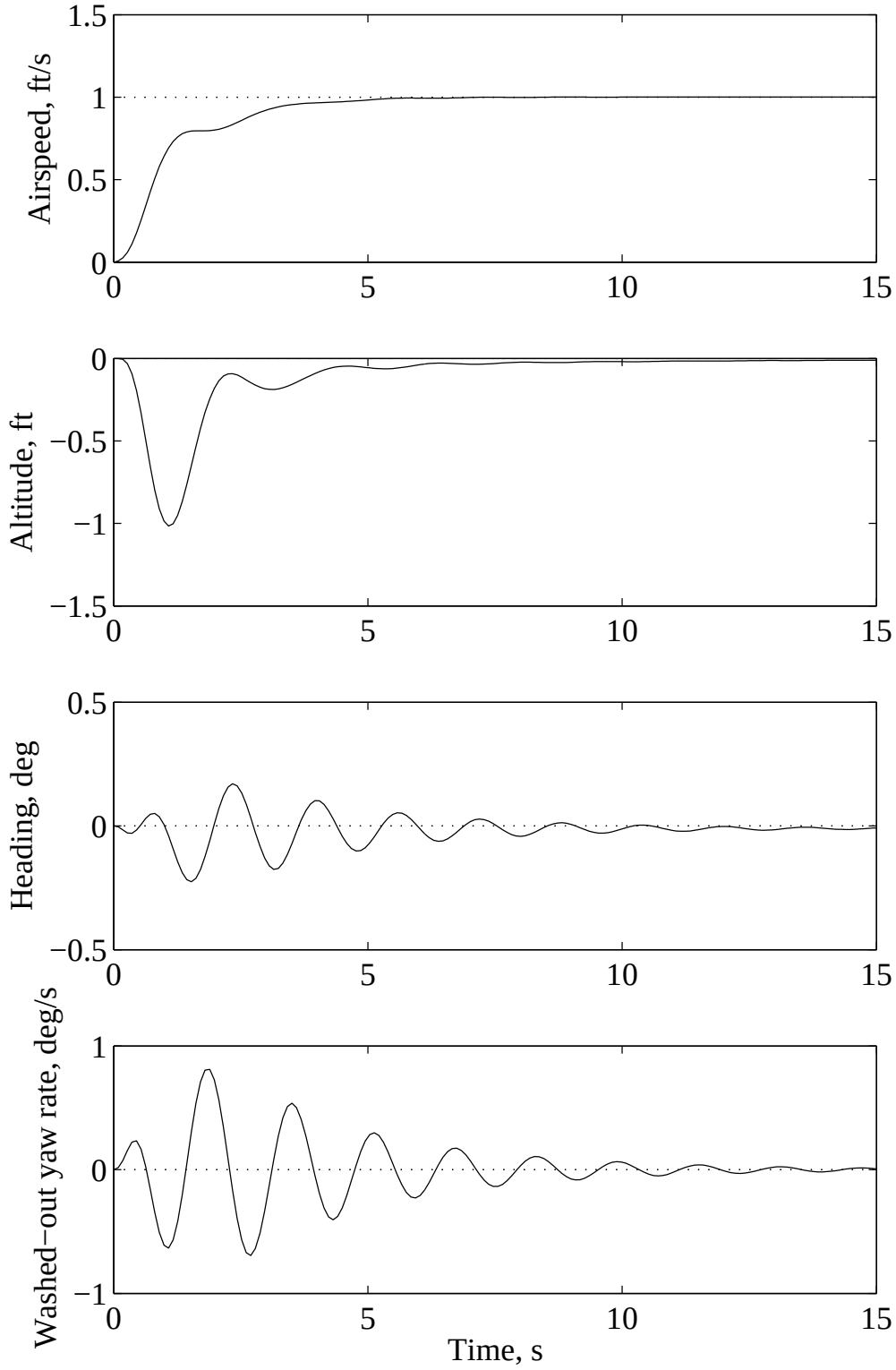


Figure 4.13: Airspeed step response during cruise using the linear model of the simulation.

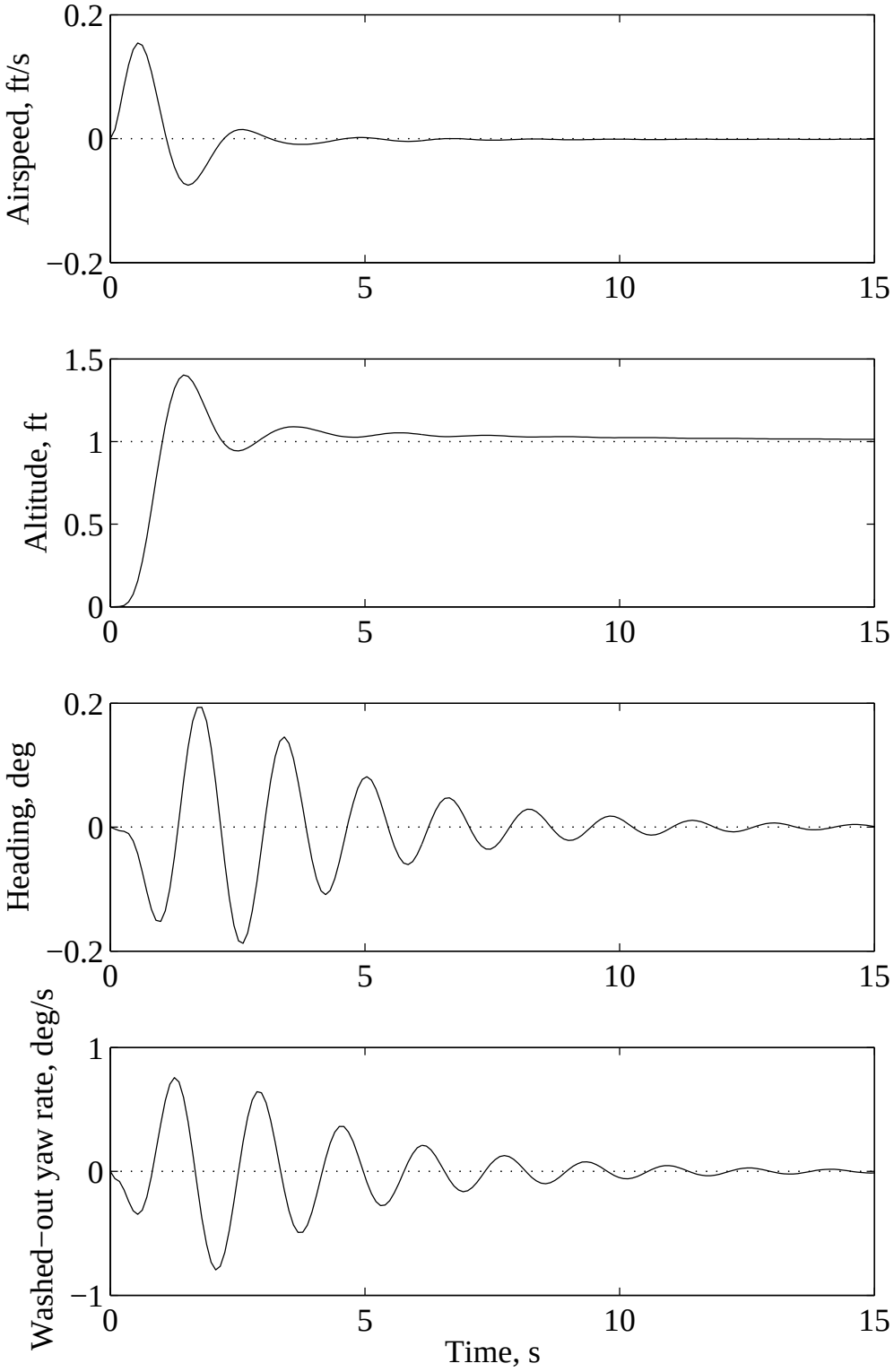


Figure 4.14: Altitude step response during cruise using the linear model of the simulation.

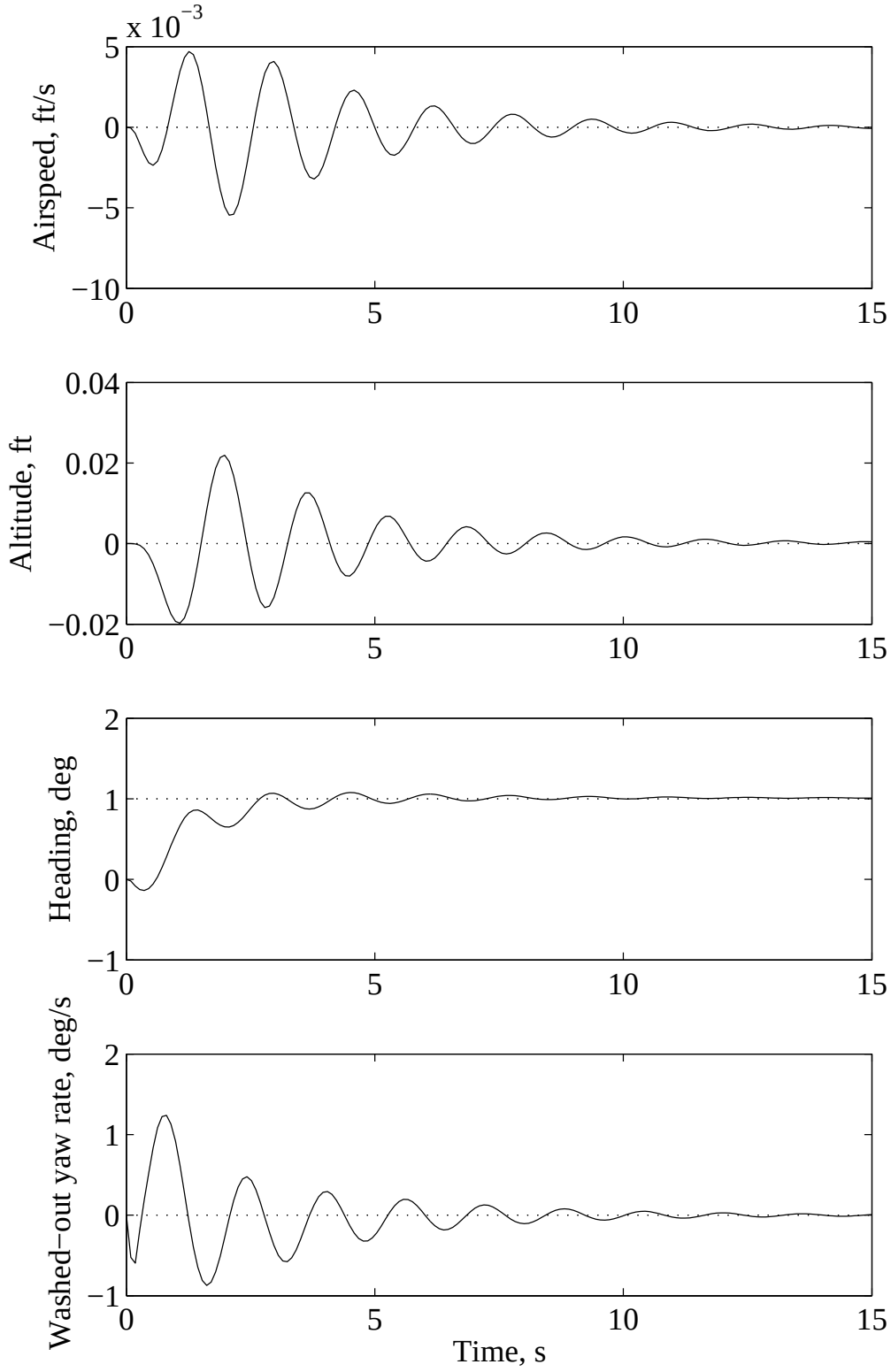


Figure 4.15: Heading step response during cruise using the linear model of the simulation.

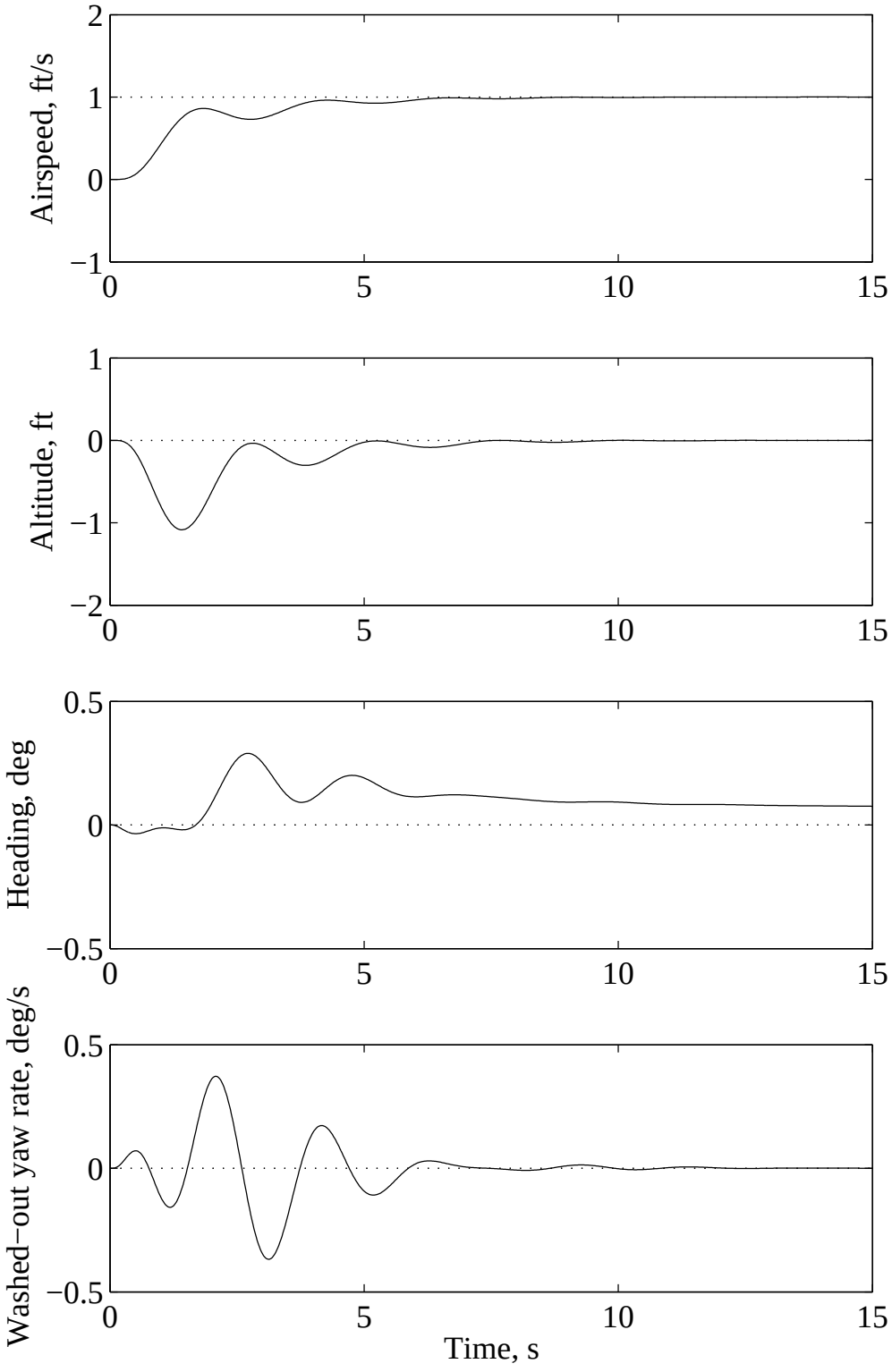


Figure 4.16: Airspeed step response during cruise using the linear model from the EoM.

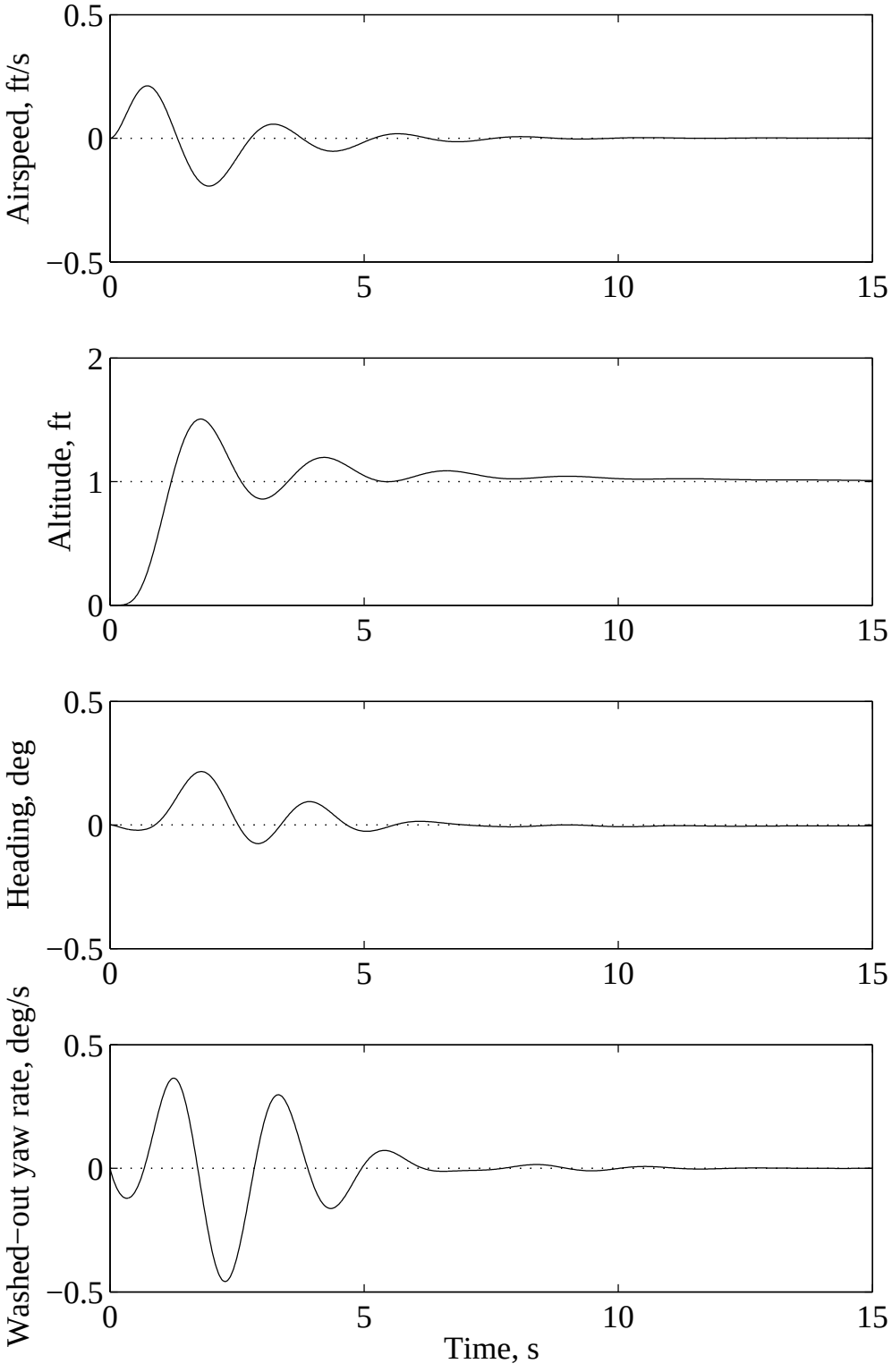


Figure 4.17: Altitude step response during cruise using the linear model from the EoM.

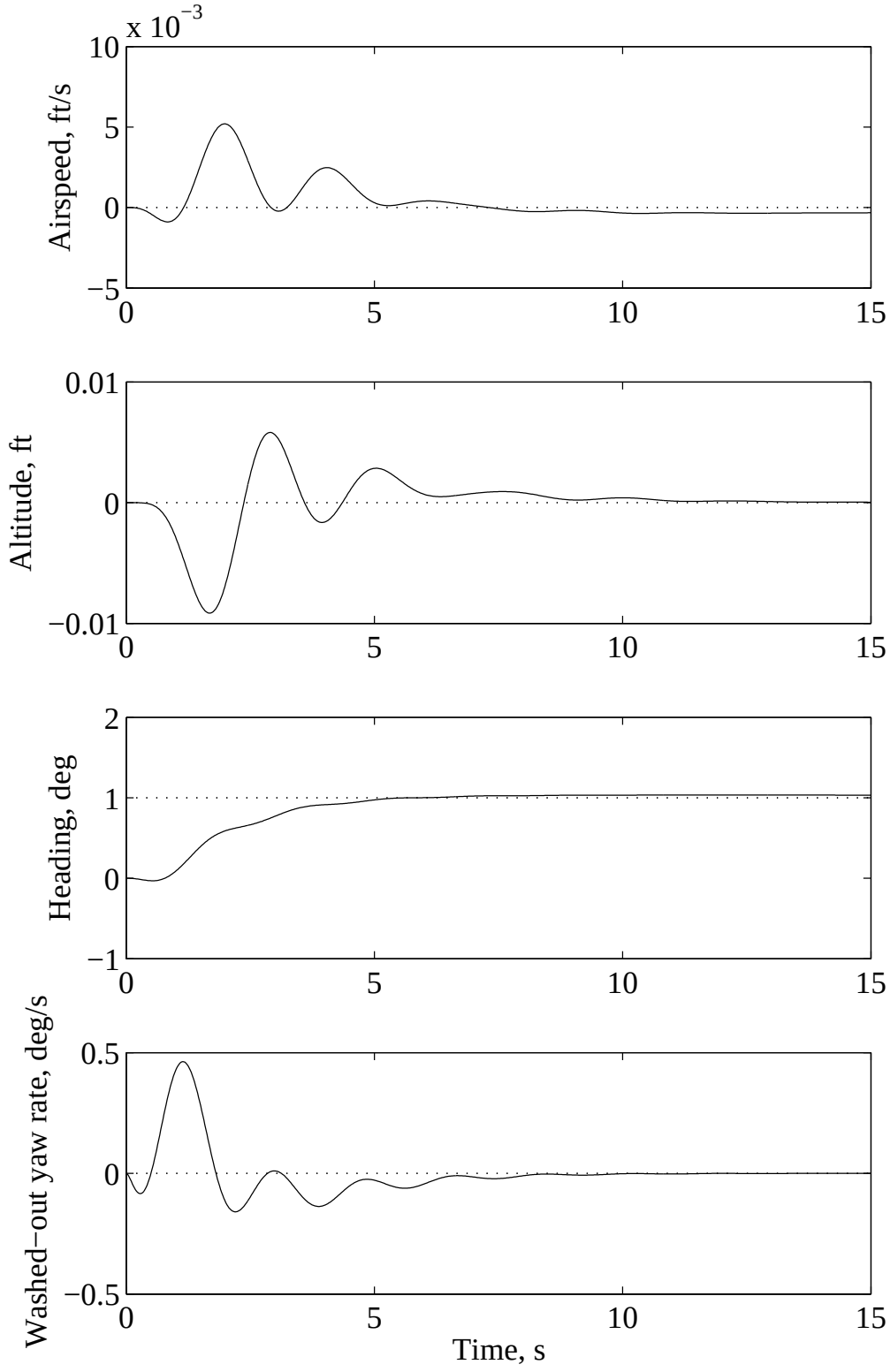


Figure 4.18: Heading step response during cruise using the linear model from the EoM.

4.2.5 Applying the LQT to Maneuvering Flight

A process similar to the one used for cruise flight is followed to determine the system matrices for the angular-rate controller, knife-edge controller, and stall controller. In an effort to avoid repetition in the context of this dissertation, the derivation of those matrices will be omitted from the text. Instead of the repetitive derivation, a discussion about the differences in the plant matrices used for each controller is offered to provide a better understanding of how to apply the LQT technique to maneuvering flight control design.

The LQT approach is inherently sensitive to the inclusion of states that are not controlled by the feedback structure. For example, if engine speed, altitude, and heading are not being controlled, then those states (and the corresponding rows and columns of the state-space matrices) should be removed before minimizing the cost function. For all of the maneuvers in this work, the throttle was controlled in an open-loop manner, so the engine speed state was removed during the LQT gain tuning process for each maneuvering controller. Since the washed-out yaw rate was only used as feedback during cruise, the state corresponding to the washout filter was also removed for each maneuvering controller. In addition to the engine speed and washout filter states, in the case of the angular-rate controller, the states corresponding to the orientation angles and the altitude were also removed. The pitch angle state and the states corresponding to the forward (X-body) and downward (Z-body) speed components were removed during the design of the knife-edge flight controller. During the gain tuning of the stall controller, only the altitude state was removed (in addition to the engine speed and washout filter states) because the roll angle, roll rate, heading, and speed were used as the feedback terms. To summarize, Table 4.4 lists the states that were removed from the system matrices during the gain tuning process for each maneuvering controller.

Table 4.4: States omitted during the application of the LQT to maneuvering flight.

Controller	Omitted States
Angular Rates	ϕ, θ, ψ, h, ω_{engine}, x_w
Knife-edge Flight	θ, u, w, ω_{engine}, x_w
Stall	h, ω_{engine}, x_w

After employing the LQT approach to tune the the angular-rate controller gains, Figs. 4.19 - 4.21 show the step plots using the linearized wind tunnel data and Figs. 4.22 - 4.24 show the step plots using AVL data with the EoM. For all of the plots shown in this subsection, the linear aircraft models with actuator models were used. The roll-rate response is fast with very little overshoot for both aircraft models. With the model from the tunnel data, the roll-rate response takes about 2.5 seconds while the response with the EoM model using data from AVL occurs in less than one second.

The pitch-rate response for the model from the tunnel data is fast with a rise time of 0.5 seconds, contains no overshoot, and has little effect on the roll and yaw rates. For the EoM model with AVL data, the pitch-rate response has a slight overshoot before reaching steady state in roughly two seconds. After the step input of pitch rate, the small deviations in roll and yaw rate for both linear models quickly return to zero.

Figure 4.21 shows that the yaw-rate response is much slower than both the roll and pitch rates when using the linear model from the tunnel data. The time to reach 90 percent of the step height is approximately 10 seconds. The yaw-rate response for the linear model from the EoM, however, is predicted to only take about five seconds but contains a small amount of initial overshoot. The roll-rate response, from the step input in yaw rate, has approximately the same shape for both models, with the EoM model response having about half the amplitude as the response from the tunnel data model. For both models, changes in yaw rate really have no effect on pitch rate. After reviewing all of the step plots, it was decided that the calculated angular-rate gains should provide the controller performance needed to execute aerobatic maneuvers.

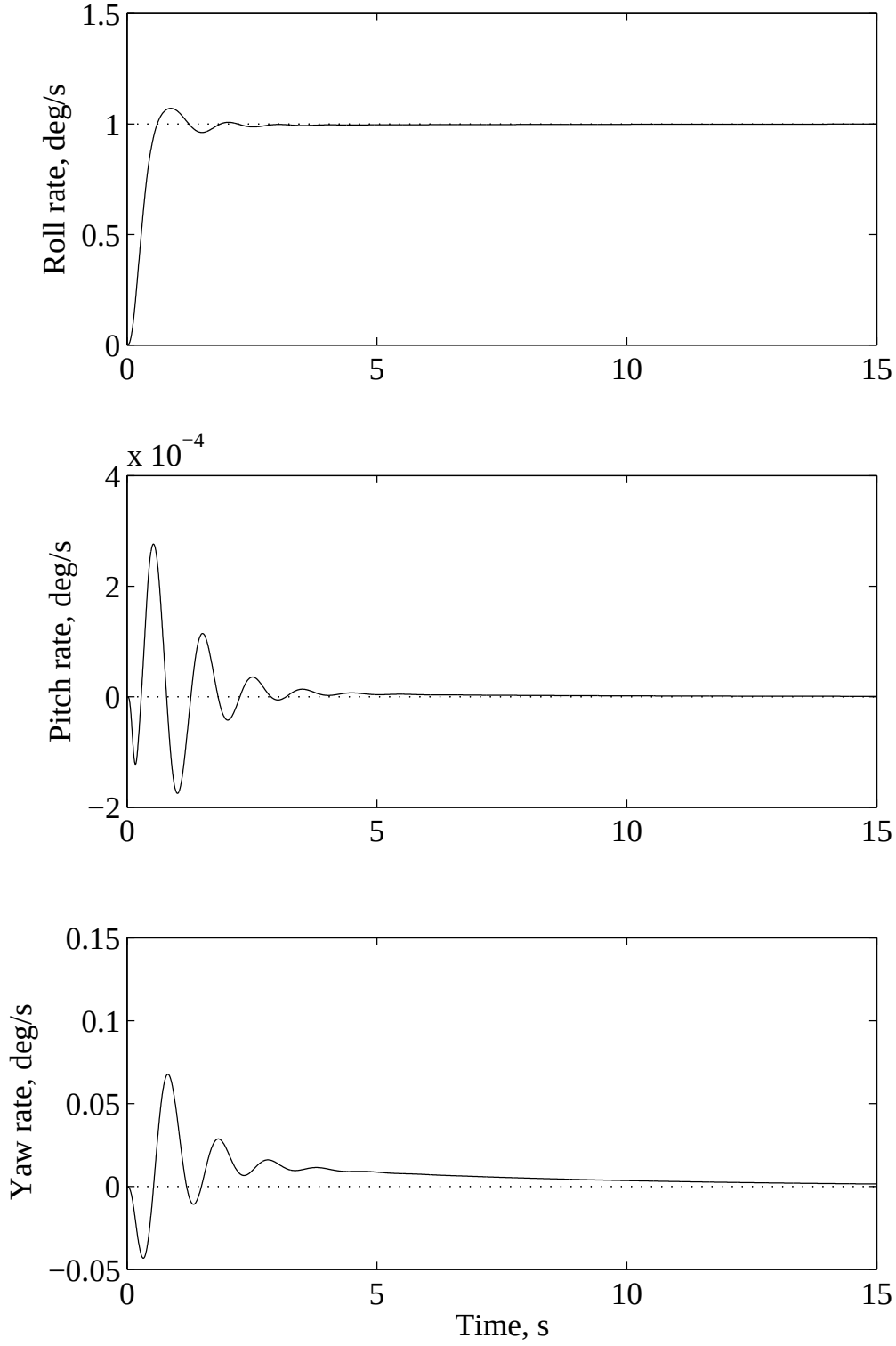


Figure 4.19: Roll-rate step response using the linear model of the simulation.

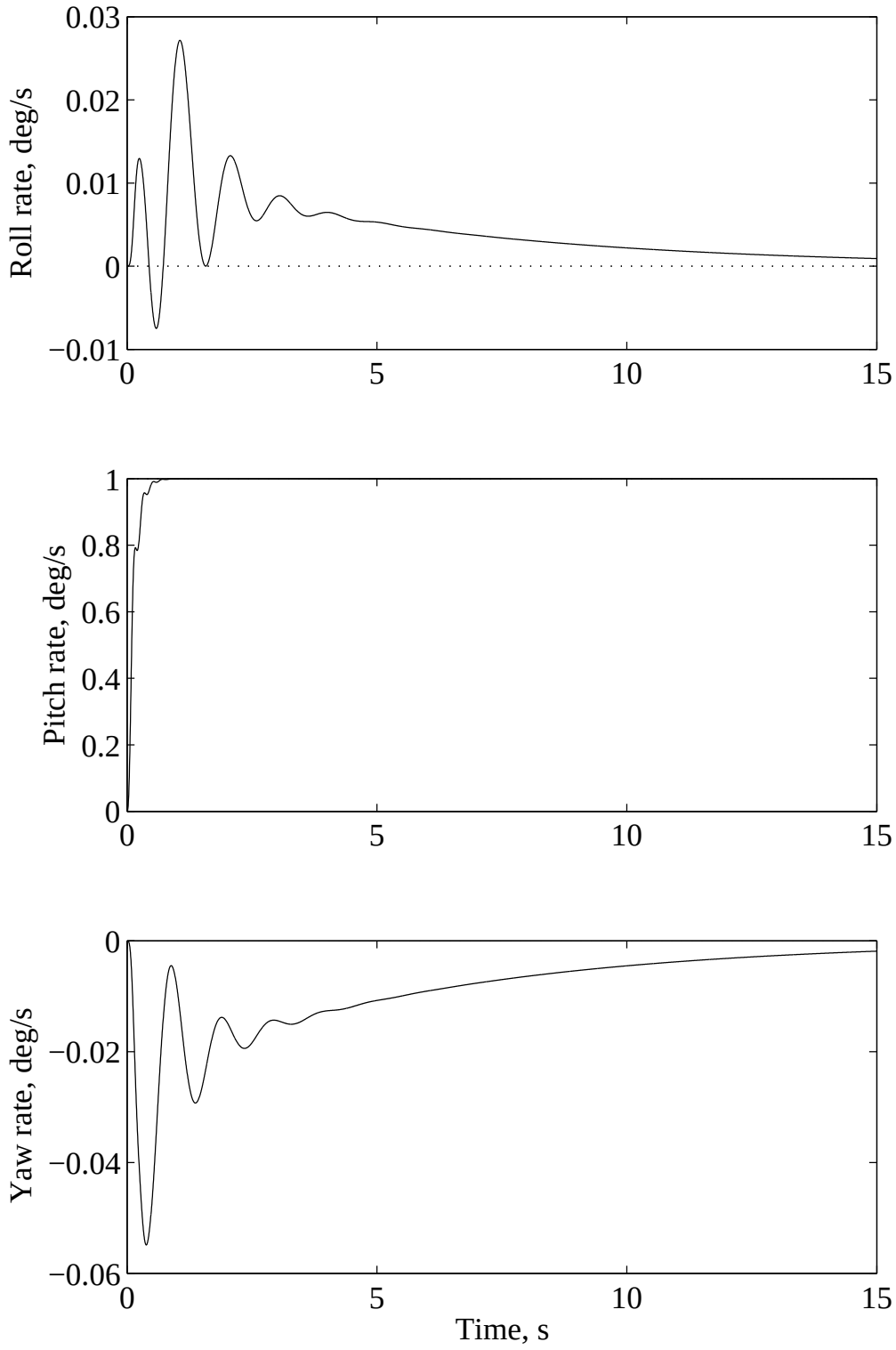


Figure 4.20: Pitch-rate step response using the linear model of the simulation.

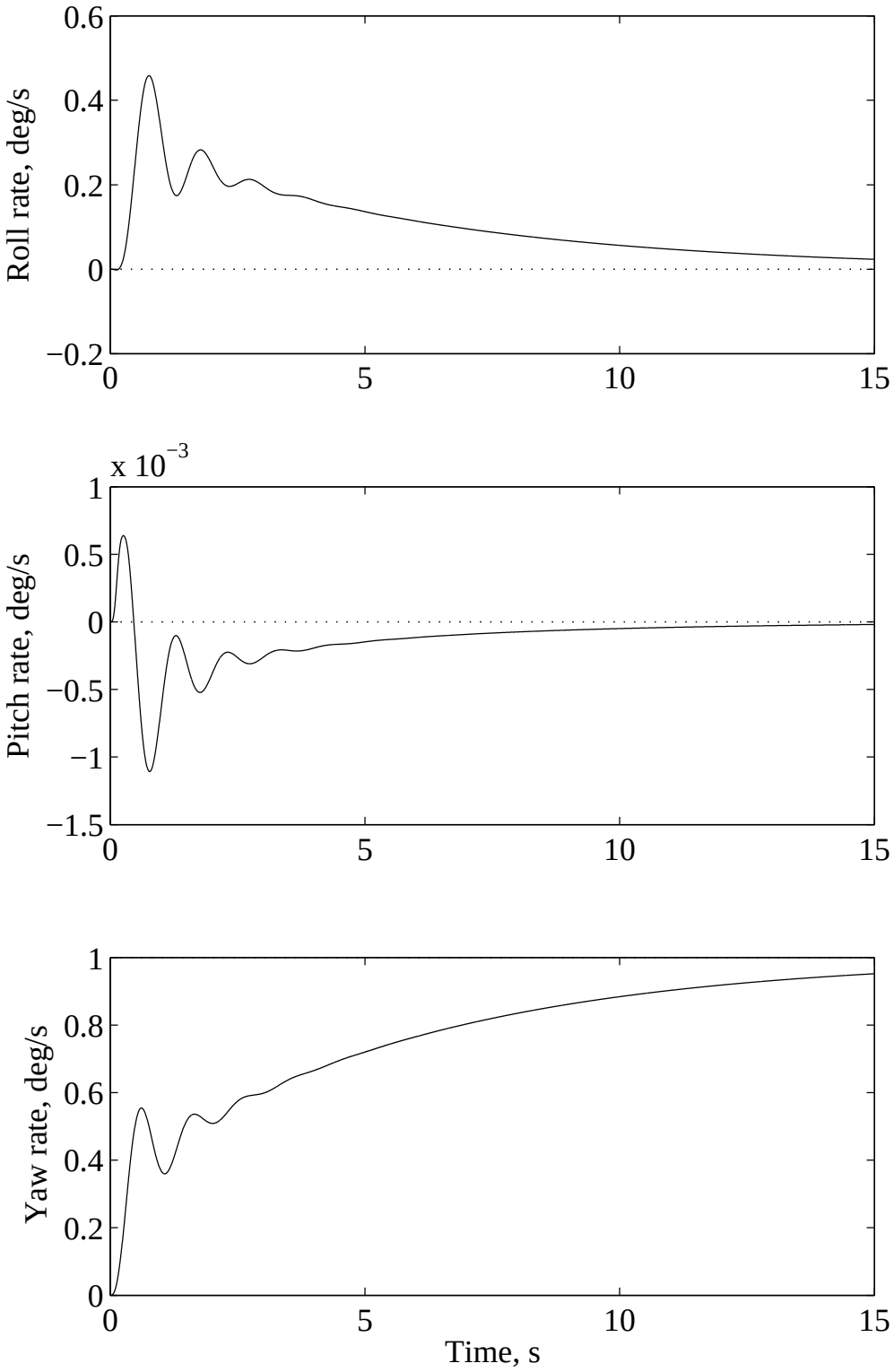


Figure 4.21: Yaw-rate step response using the linear model of the simulation.

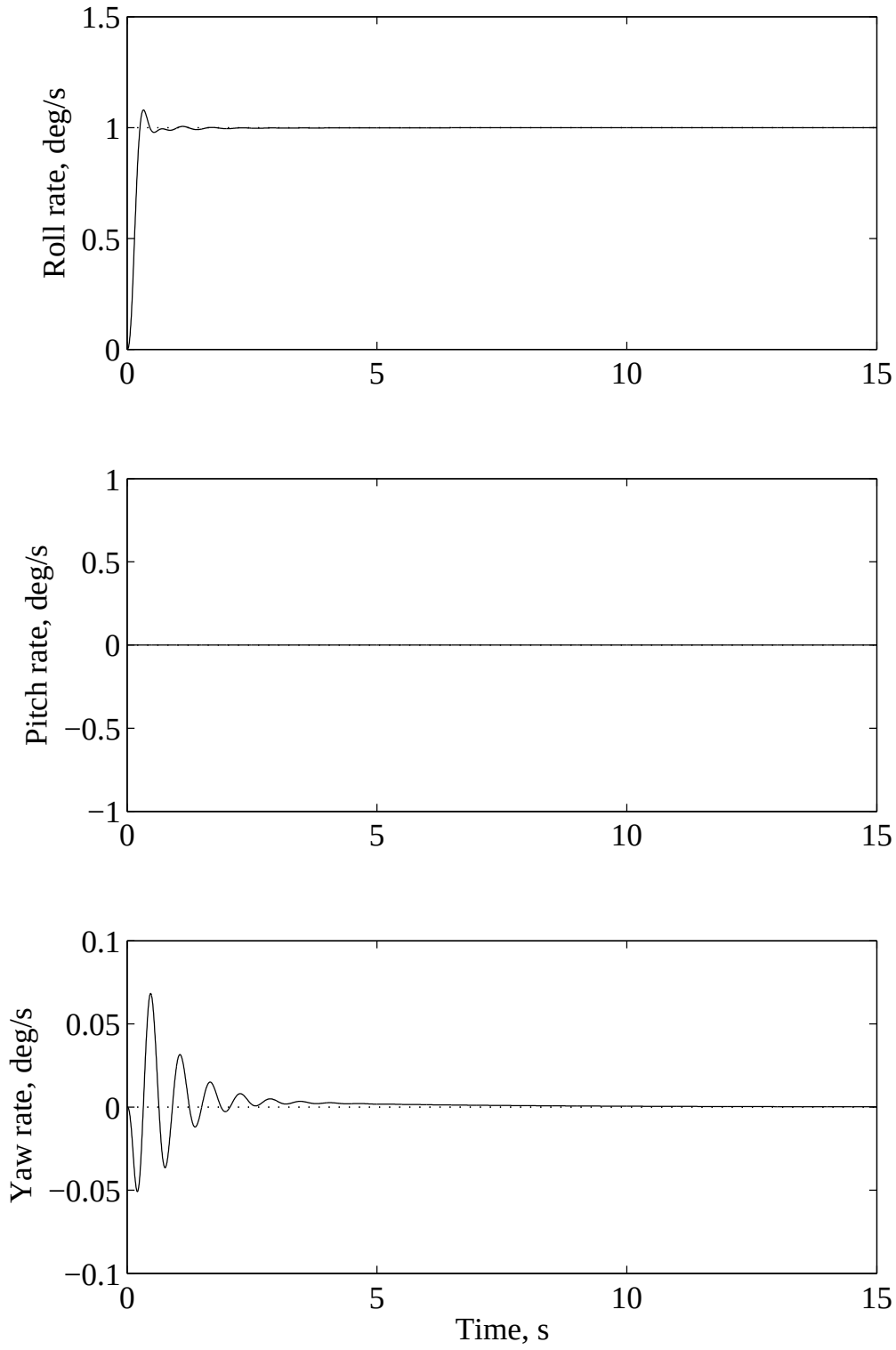


Figure 4.22: Roll-rate step response using the linear model from the EoM.

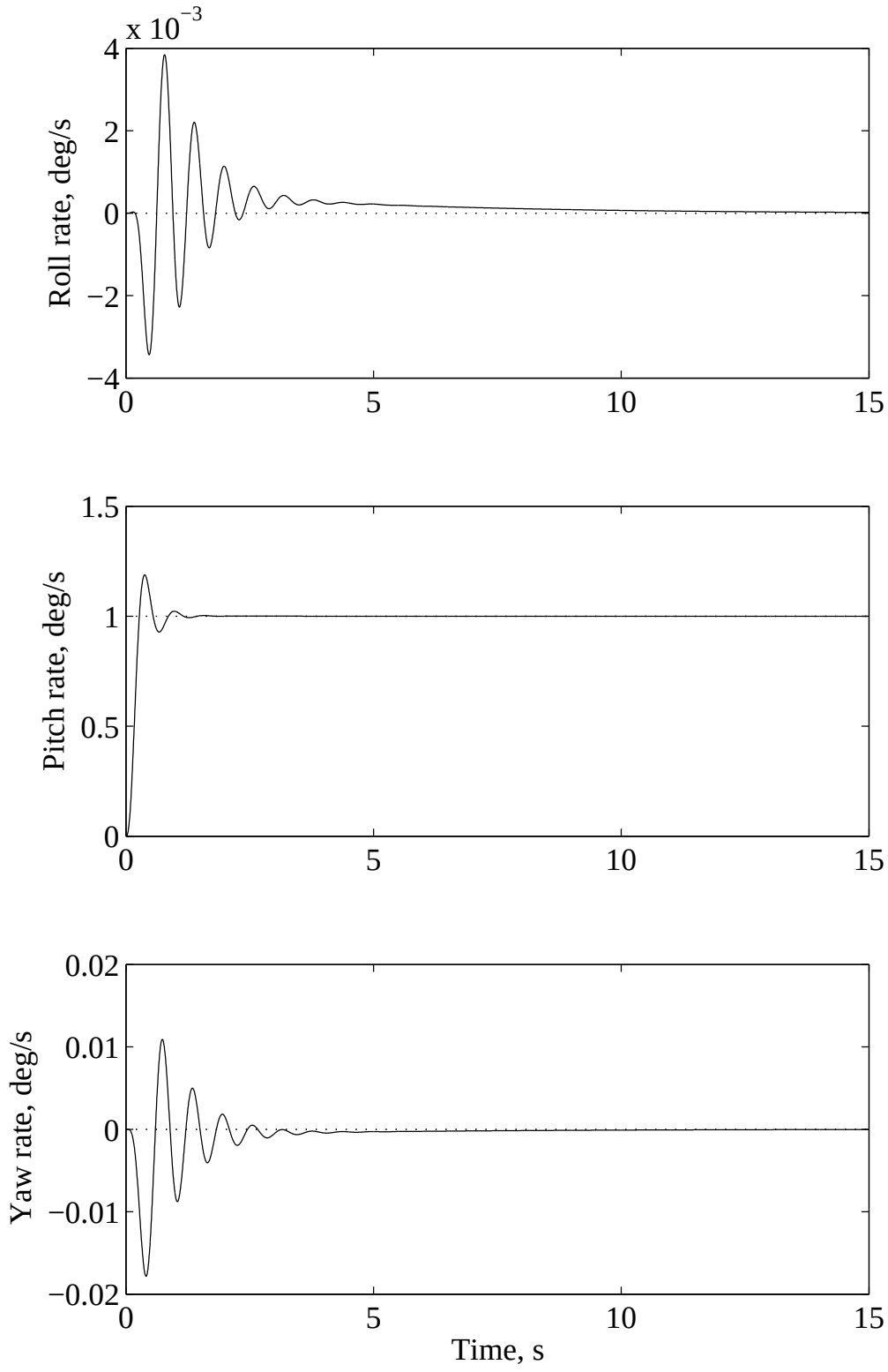


Figure 4.23: Pitch-rate step response using the linear model from the EoM.

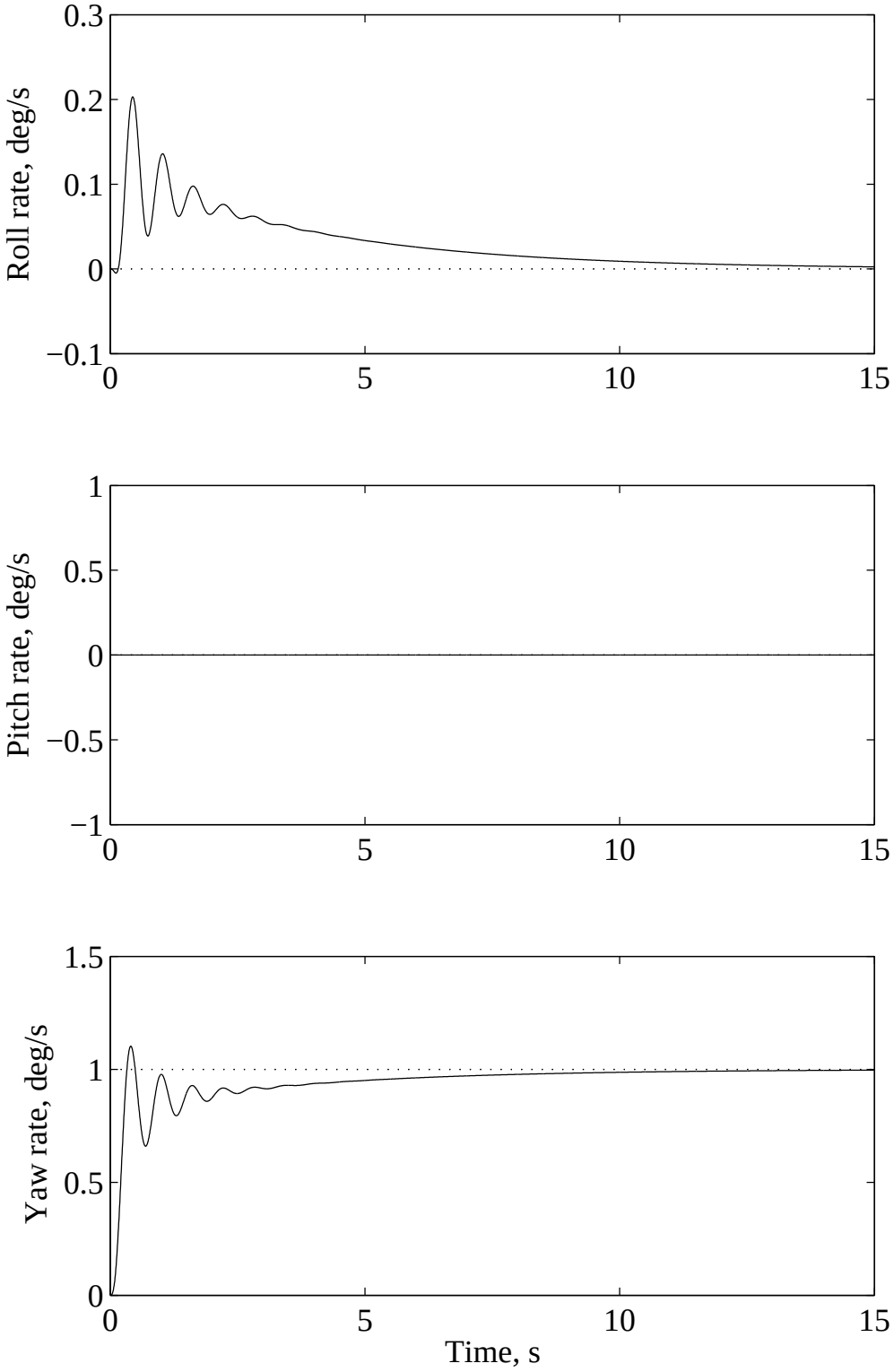


Figure 4.24: Yaw-rate step response using the linear model from the EoM.

Figures 4.25 - 4.30 show the step plots for knife-edge flight. The roll angle response is fast, having a rise time of only 2.5 seconds, and has little effect on the altitude and heading for both linear models. The altitude response appears slightly better for the linear model from the tunnel data because the overshoot and settling time are both reduced. Using the EoM model, the altitude response is still adequate, but it contains a few oscillations which last about 10 seconds before settling on the steady-state value.

The heading response using the linear model from the tunnel data contains a slight overshoot, but has a settling time of only three seconds. When using the EoM model, the heading response has an initial overshoot, but returns to steady state in nearly two seconds. The slight deviation in roll angle, due to the step in heading, is removed in under four seconds for both linear models.

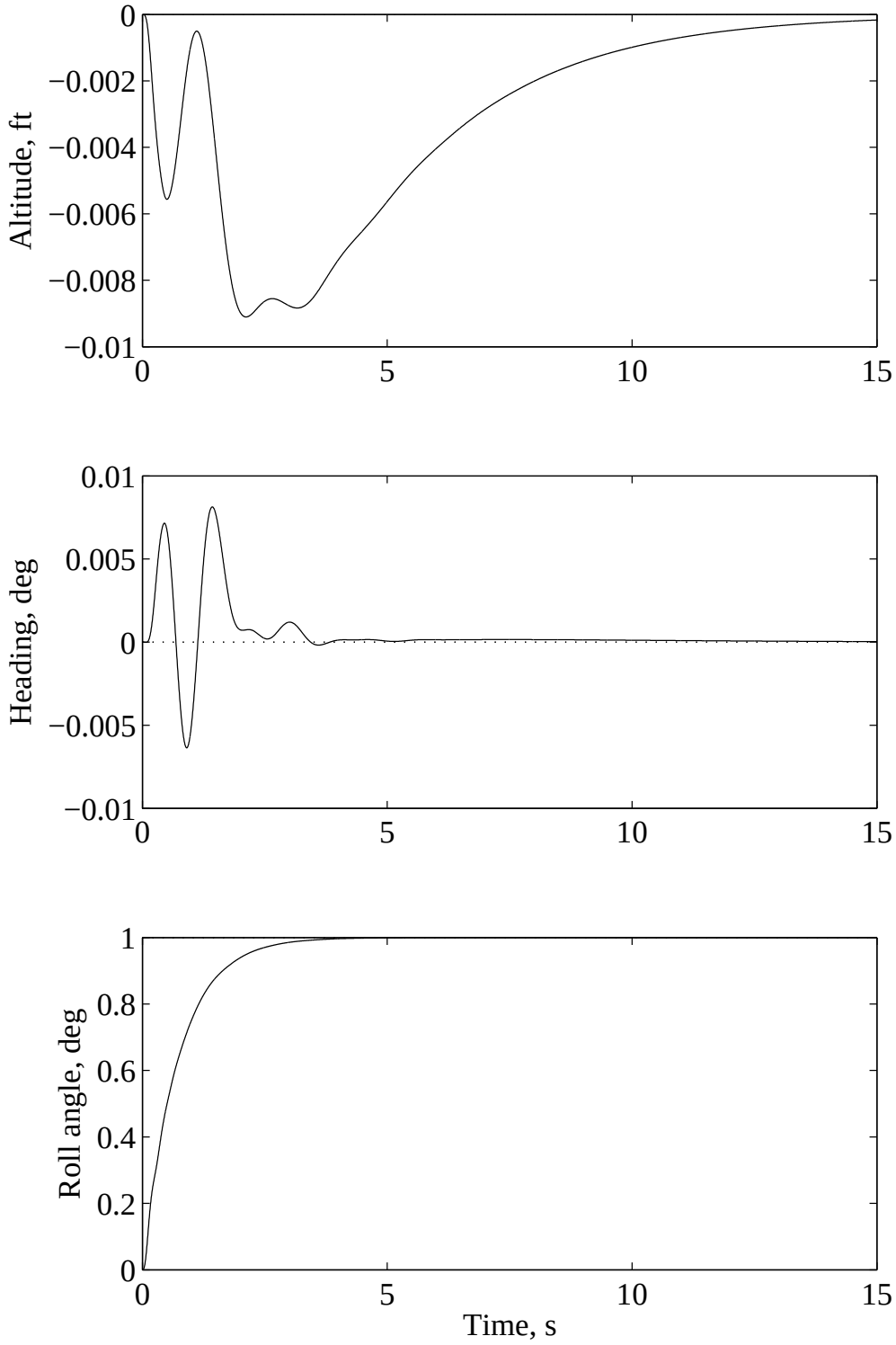


Figure 4.25: Roll angle step response for knife edge using the linear model of the simulation.

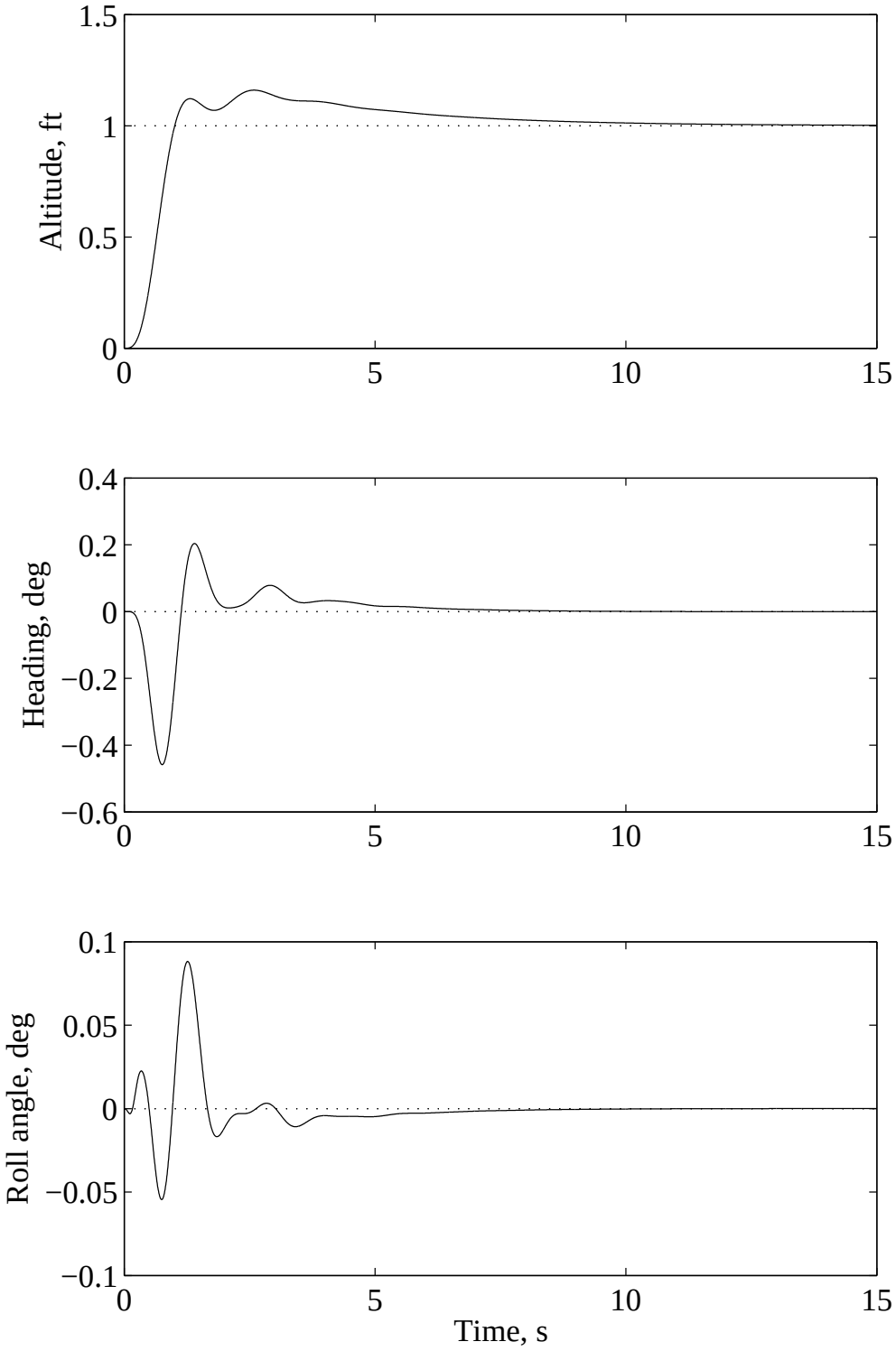


Figure 4.26: Altitude step response for knife edge using the linear model of the simulation.

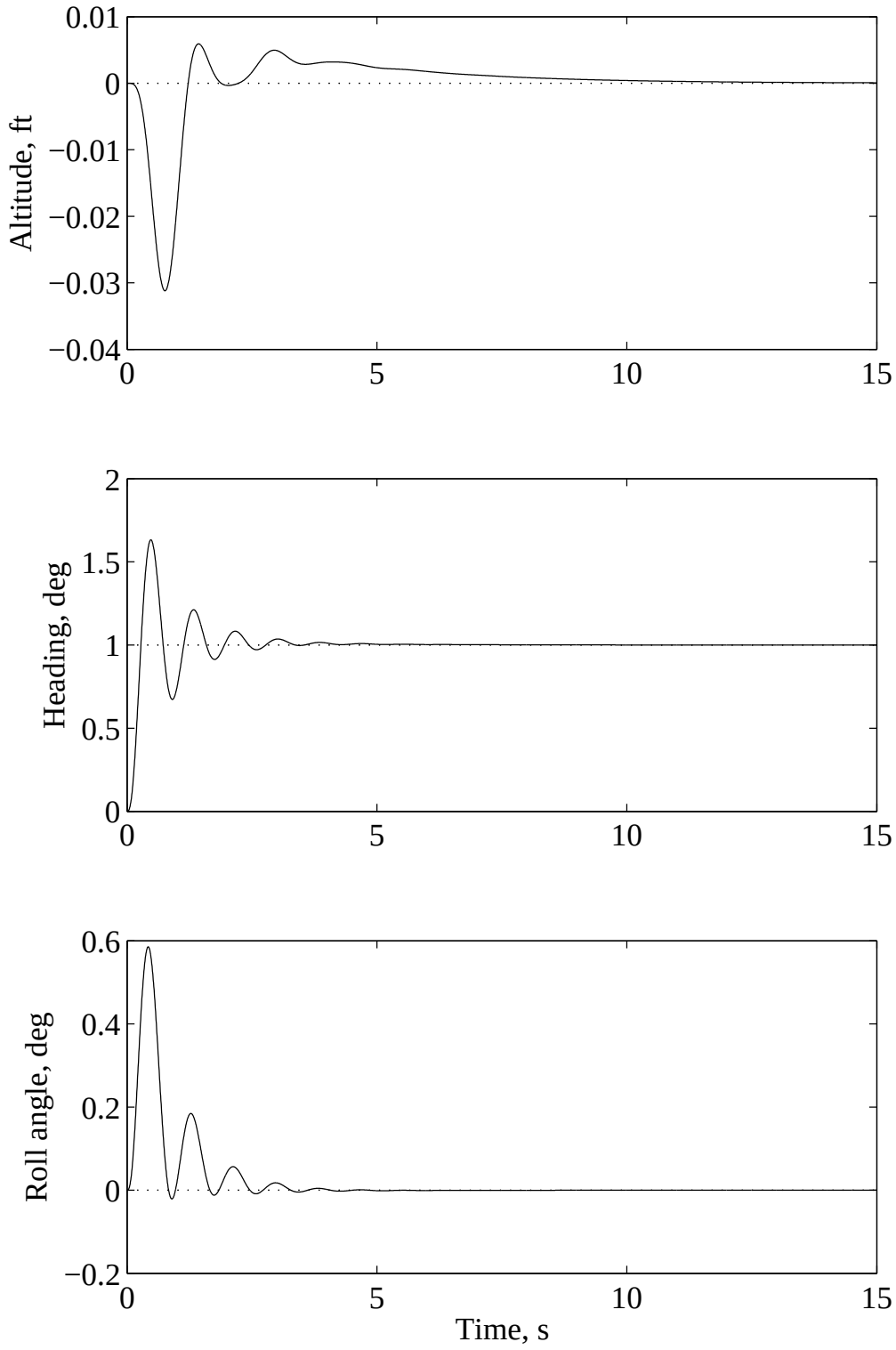


Figure 4.27: Heading step response for knife edge using the linear model of the simulation.

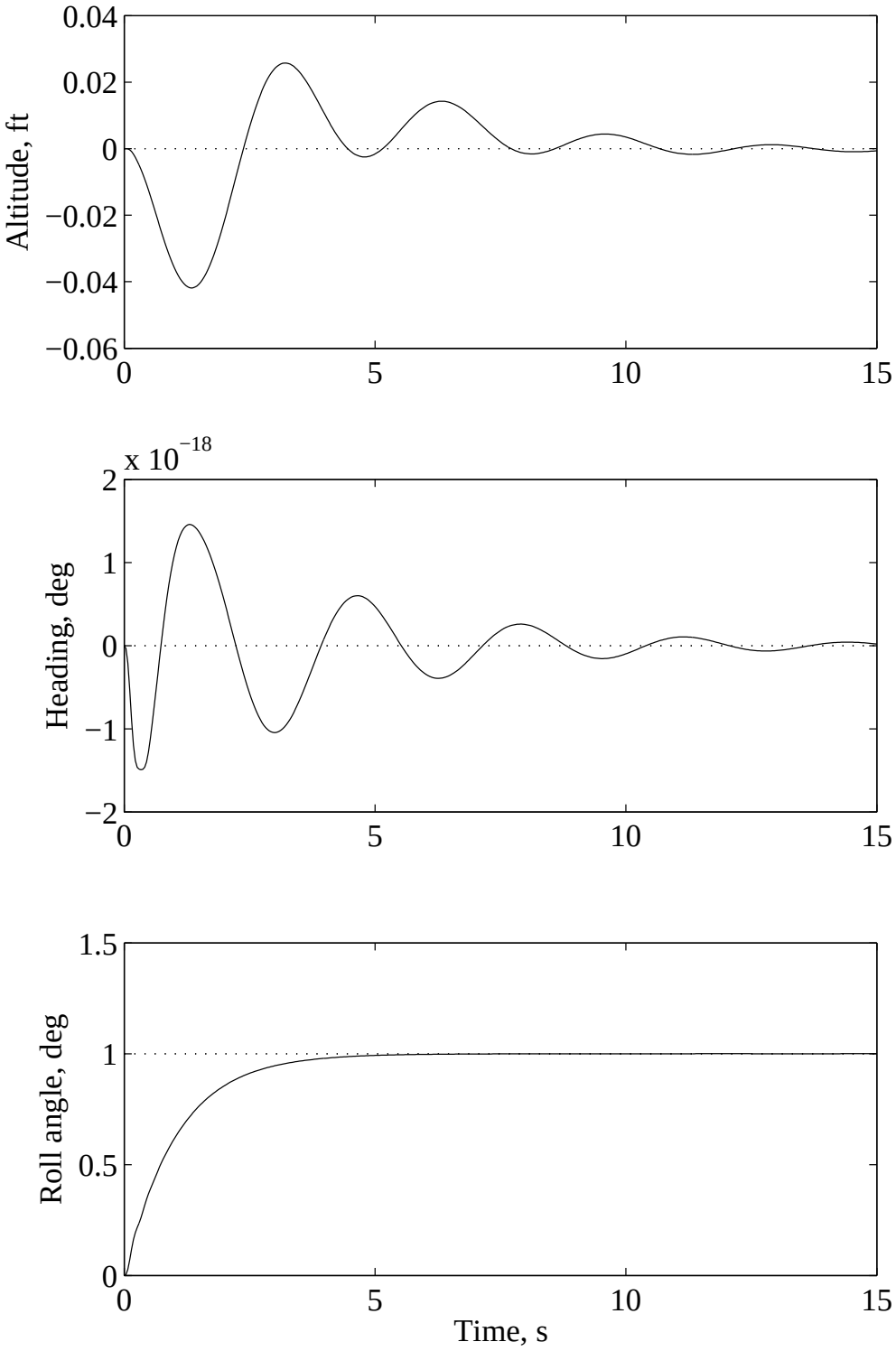


Figure 4.28: Roll angle step response for knife edge using the linear model from the EoM.

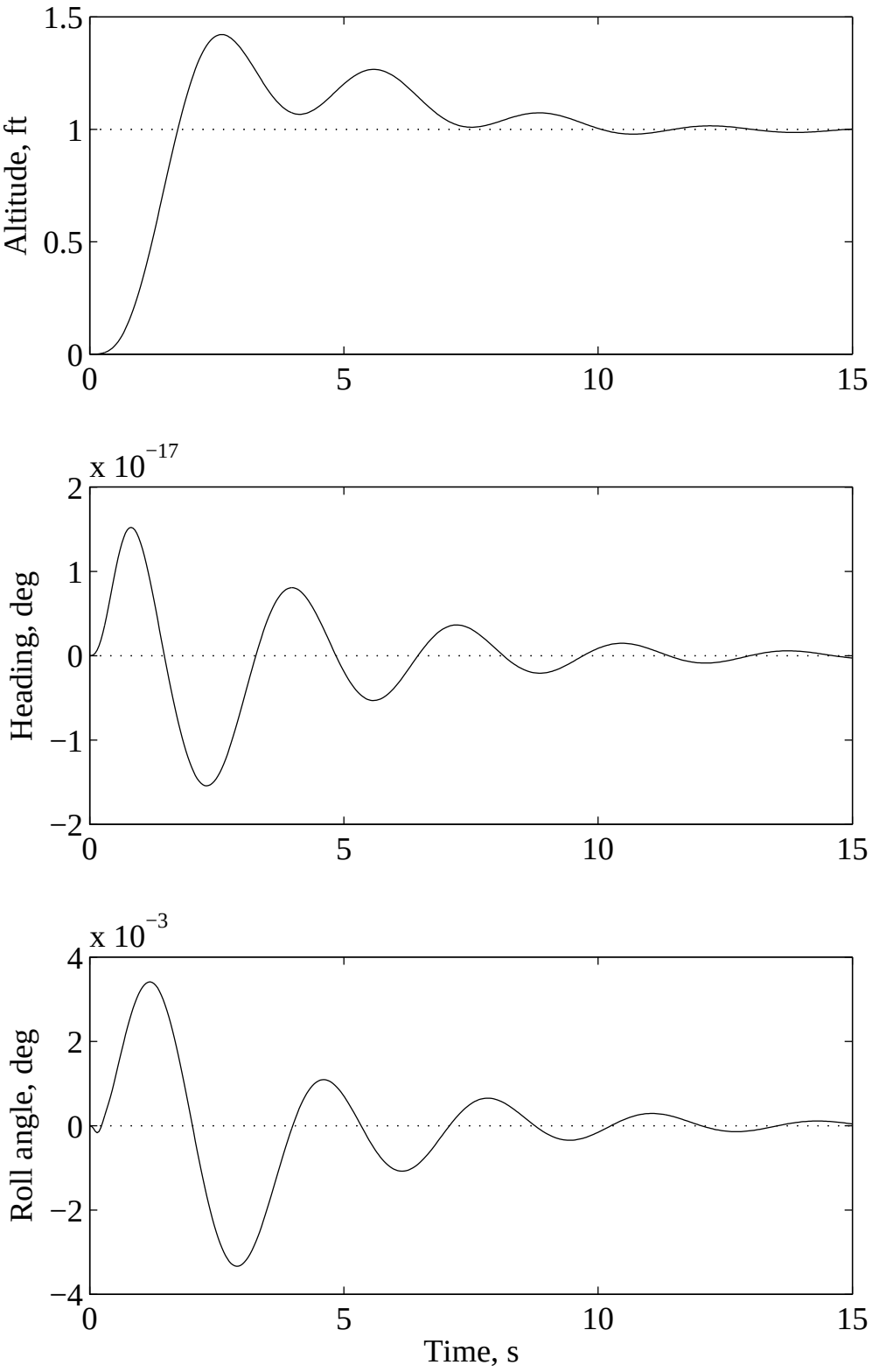


Figure 4.29: Altitude step response for knife edge using the linear model from the EoM.

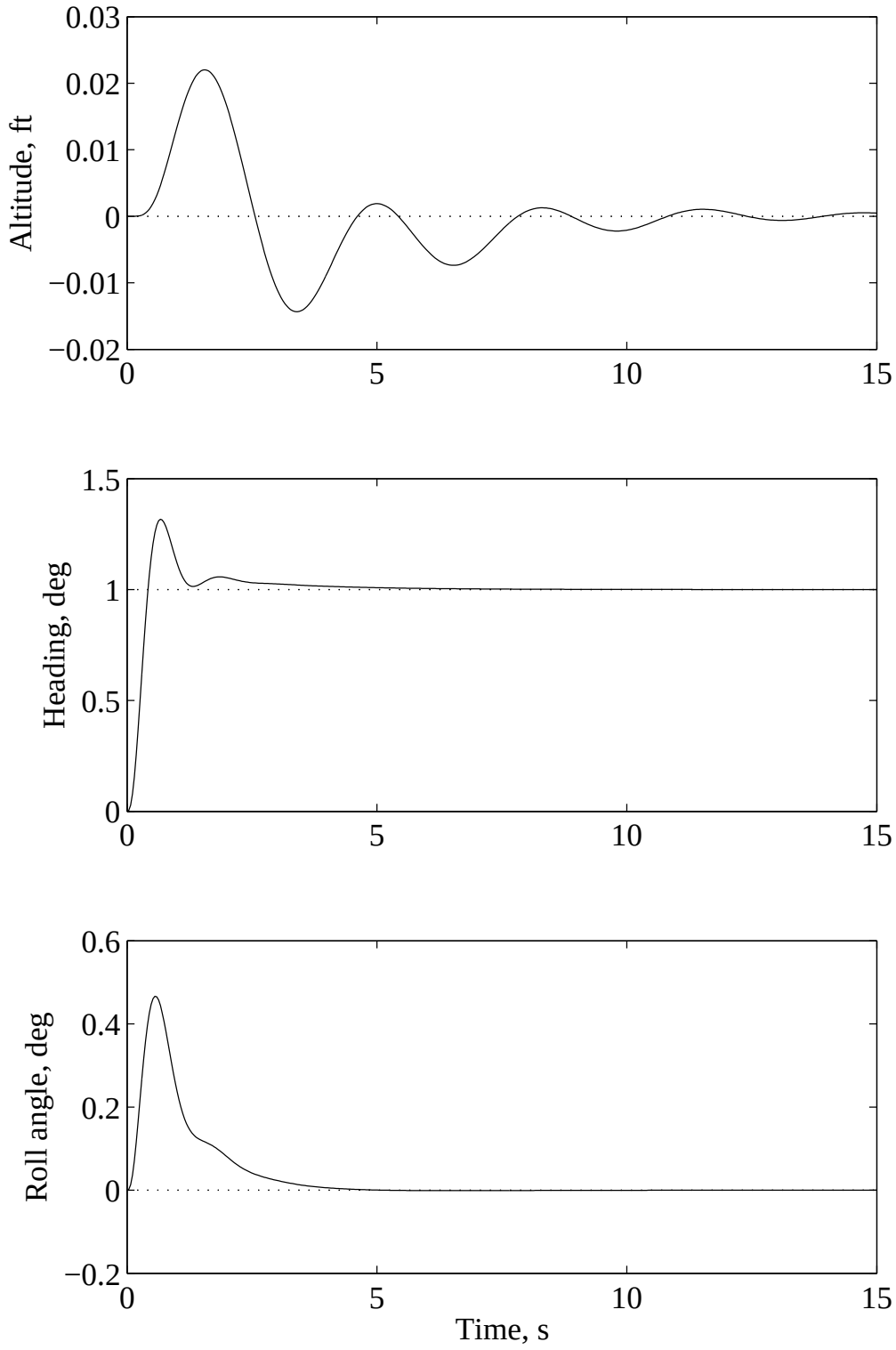


Figure 4.30: Heading step response for knife edge using the linear model from the EoM.

The step plots for the first phase of the spin, in Figs. 4.31 - 4.33, show that the control gains determined using the linear model from the tunnel data should provide adequate control to track roll angle, speed, and heading. Changes in roll angle occur within five seconds, but the response takes about 20 seconds to reach steady state. Speed and heading are both affected by the roll angle change, but both return to zero after 30 seconds. From the plots, changes in speed will take longer than changes in roll angle with a rise time for the speed response of nearly 30 seconds. The response to a change in heading command results in a few large oscillations before reaching steady state in approximately 20 seconds. These slower responses should not be a huge concern, however, because the goal leading up to the stall and spin is to simply maintain heading with the wings level. The speed tracker during the stall just needs to slow the aircraft down enough for the stall to occur while the heading and roll angle trackers try to maintain zero deviations from the steady state. Once the aircraft stalls, control switches to the angular-rate controller to accomplish the spin.

The same logic applies to the step response plots in Figs. 4.34 - 4.36 for the linear model from the EoM. The biggest concerns with the step plots for stall are that speed changes occur without too much oscillation and the heading and roll angle loops are stable. The heading and roll angle loops need to be damped enough for the stall to occur without the aircraft experiencing an abrupt change in attitude. The roll angle response shows large variations in heading and roll angle before reaching steady state, but both loops appear stable. Speed changes occur relatively quickly and without much deviation in heading and roll angle. Similar to the roll angle response, the heading response is also stable, but contains a few large oscillations resulting in a settling time of about 20 seconds. An important note is that due to the linear nature of the vortex lattice method, errors are expected to be introduced when using the linear model from AVL to control flight regimes near stall. While the control gains produce stable loops based on the linear model used to determine them, they may not produce stable loops when applied to the nonlinear aircraft.

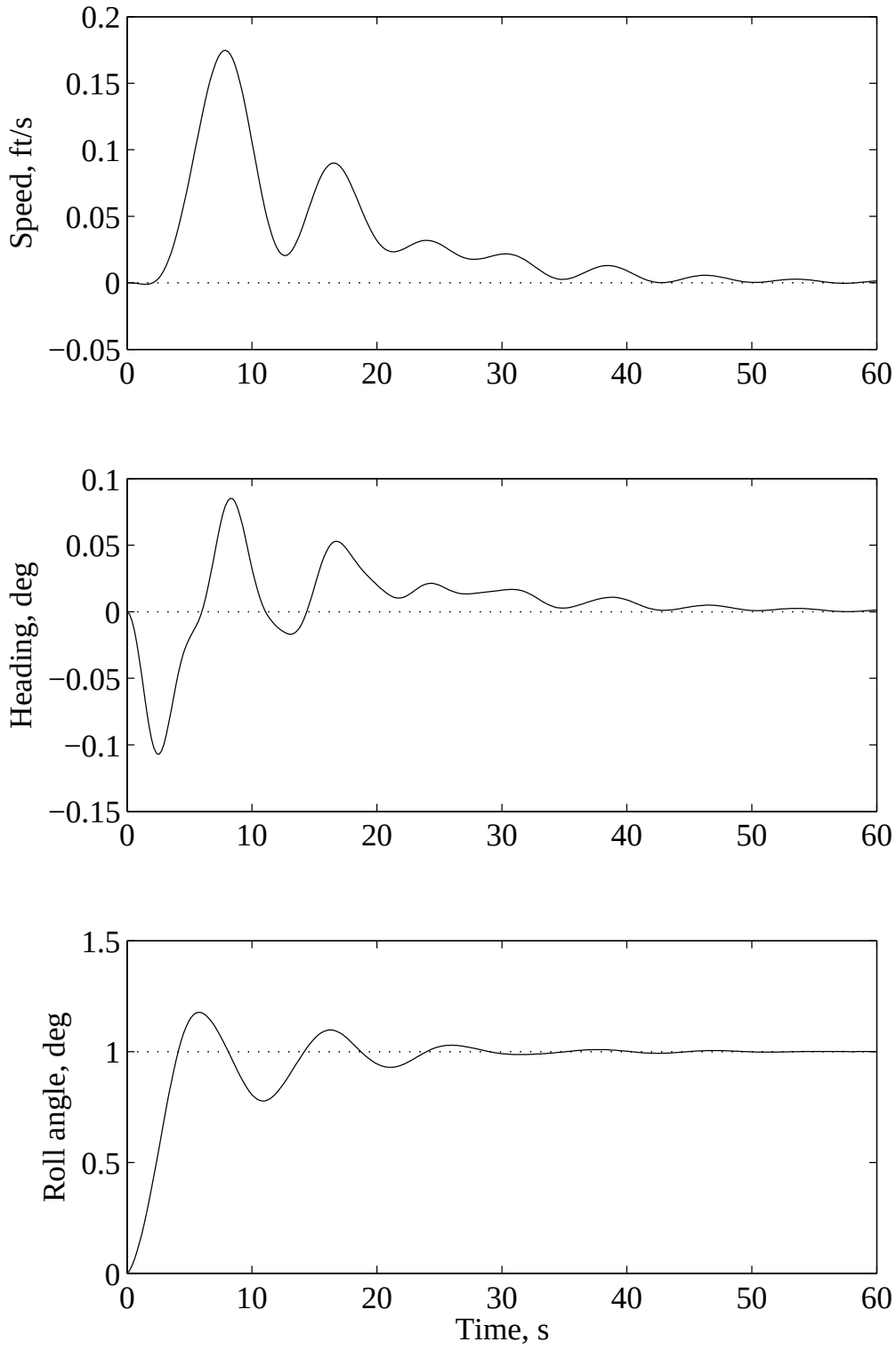


Figure 4.31: Roll angle step response during stall using the linear model of the simulation.

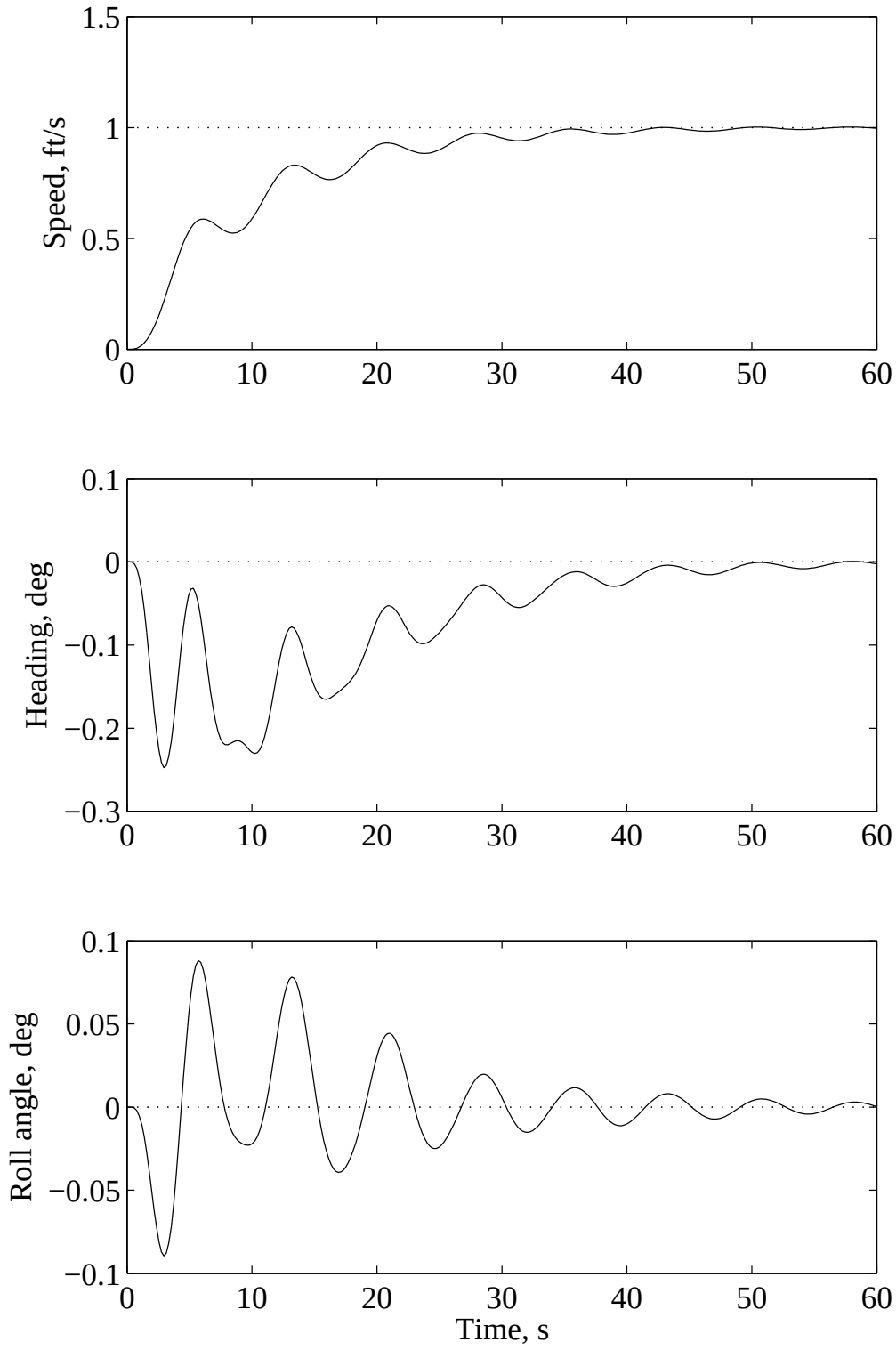


Figure 4.32: Airspeed step response during stall using the linear model of the simulation.

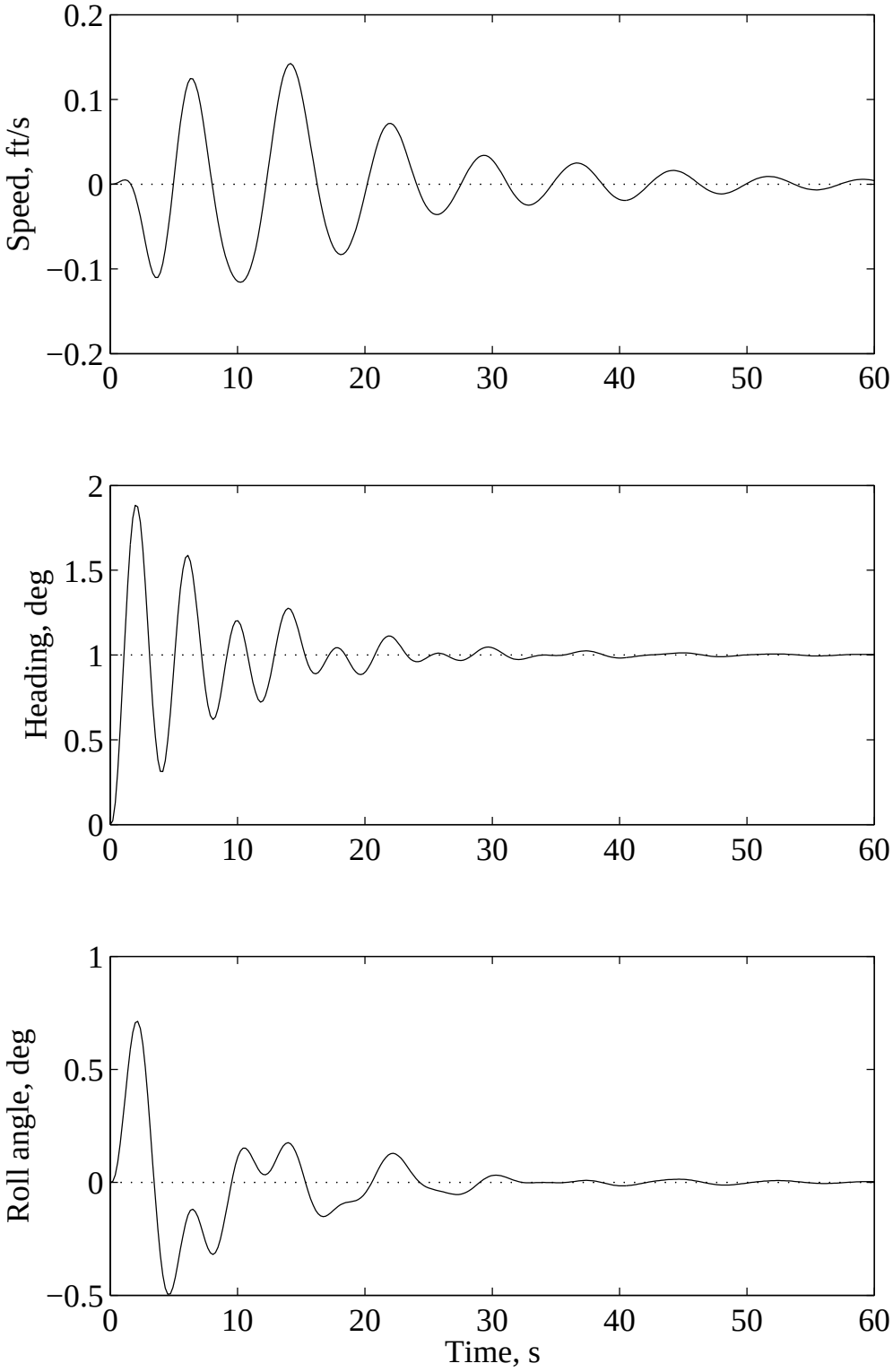


Figure 4.33: Heading step response during stall using the linear model of the simulation.

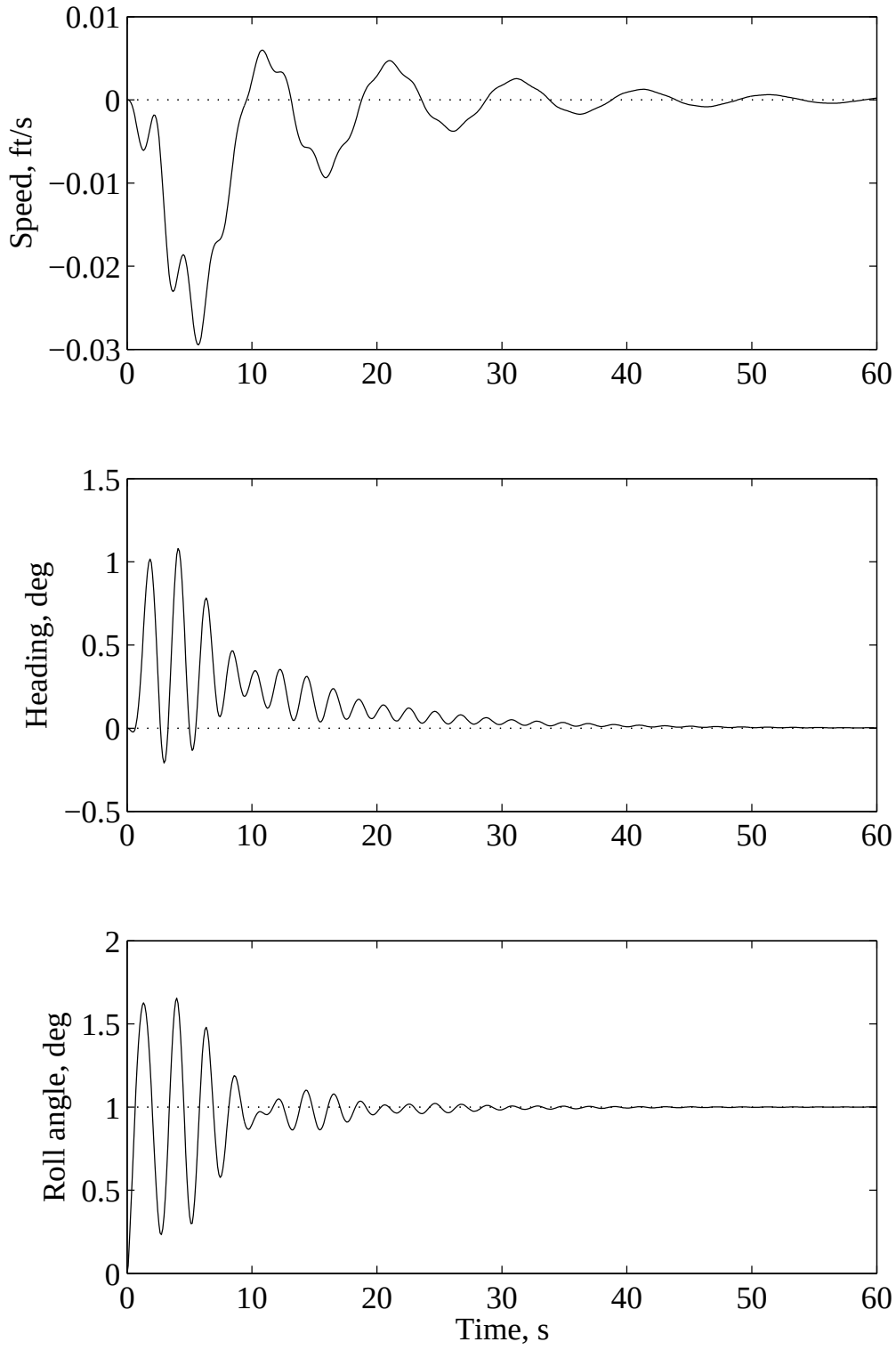


Figure 4.34: Roll angle step response during stall using the linear model from the EoM.

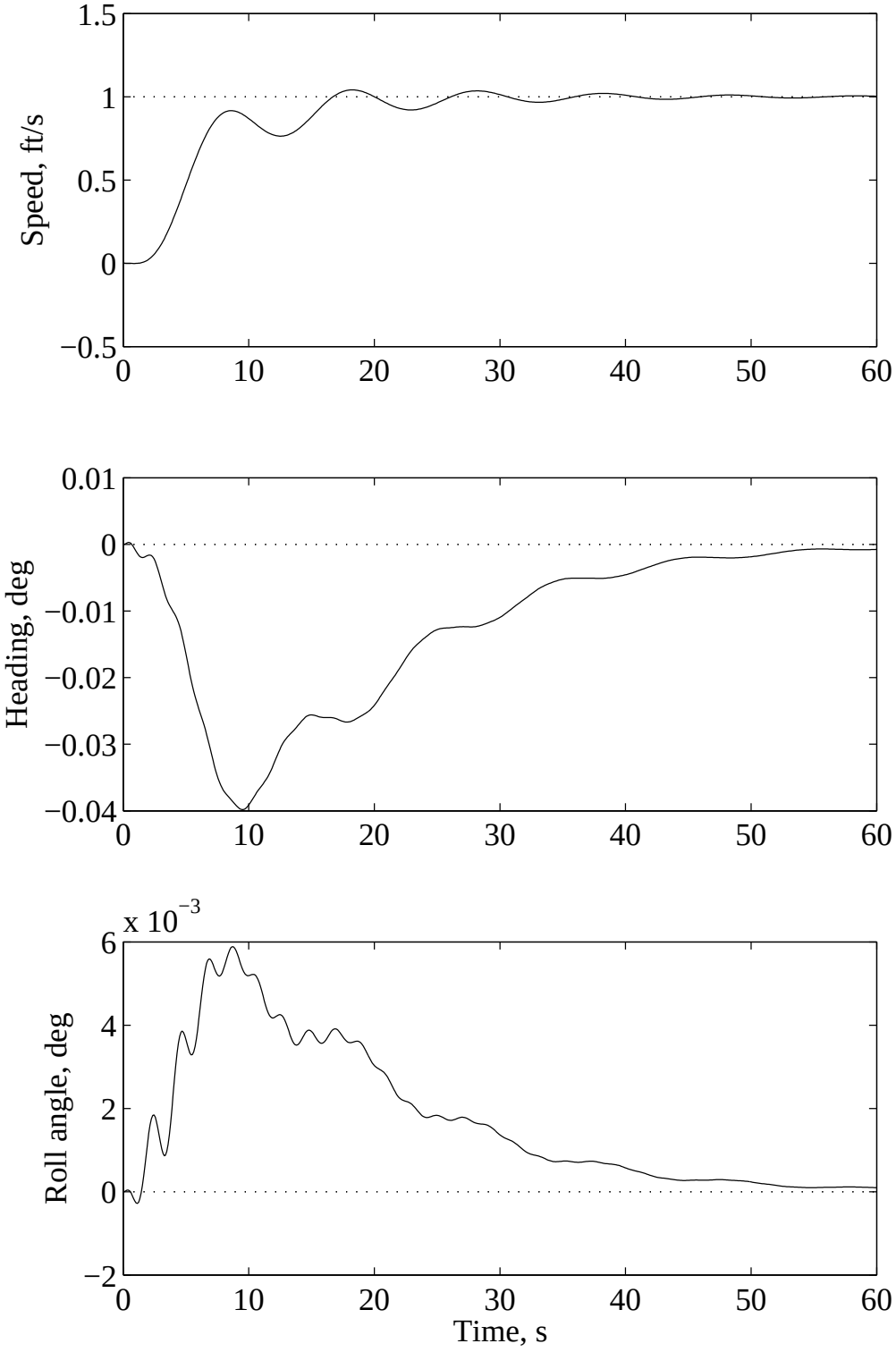


Figure 4.35: Airspeed step response during stall using the linear model from the EoM.

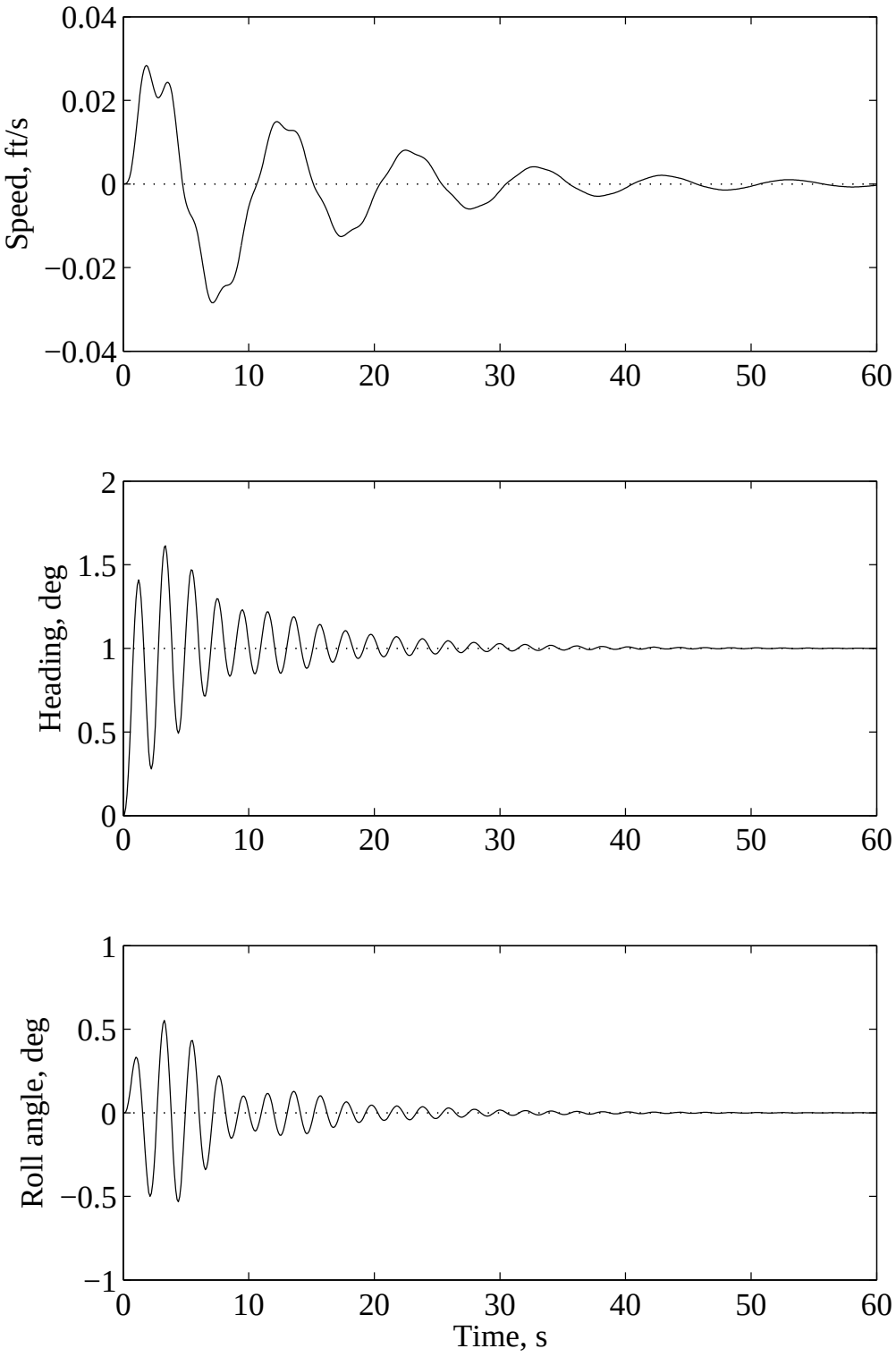


Figure 4.36: Heading step response during stall using the linear model from the EoM.

4.3 Gain Scheduling

Two different approaches were used to implement the control gains for the maneuvering autopilot. Results were first obtained using constant gains for each maneuver. Once the control gains calculated using the LQT method were proven to work for both cruise and maneuvering flight, the maneuvering gains were combined with the gains for the cruise controller to allow transitions from maneuvering flight back to straight-and-level flight. This first approach was based on the type of maneuver being flown as well as the time to complete each maneuver. The second approach built on the success of the first by scheduling multiple sets of gains for each maneuver based on airspeed. The gain scheduling technique was combined with the gain switches based on maneuver from the first approach to allow transitions back to straight-and-level flight. This second technique was used to show how using multiple linear models throughout the maneuver affects the overall performance of the autopilot.

The first approach was implemented in the simulation using matrix definitions of the gains for each controller. The gain matrix corresponding to the maneuver being flown was selected and used along with the respective control loops. Since the guidance commands during maneuvering flight are functions of time, the time for the transition back to cruise flight is known at the start of the maneuver. Once the transition time is reached, the cruise controller is used to generate the actuator commands to return the aircraft back to straight-and-level flight.

To implement the gain scheduling technique, multiple equilibrium flight conditions were used to determine linear models throughout the flight envelope of each maneuver. The linear aircraft models were determined using both AVL and the linearized simulation to show the comparison of each method. The LQT method was then used in succession for each of the linear models to obtain multiple sets of control gains. The flight conditions used to obtain the linear models were determined by analyzing the results when the gains were held constant. After reviewing the simulation data (presented in Ch. 5) consisting of airspeed,

angle of attack, and sideslip angle for each maneuver, it was decided to schedule the gains based on airspeed. This decision was based on the fact that the angle of attack and sideslip angle only slightly deviated from their trimmed values throughout the maneuvers, but the airspeed experienced changes of up to 80 ft/s. Since the relative errors of the small-angle approximation are less than one percent for angles less than 15 degrees,* it was assumed that small changes in angle of attack and sideslip would have minimal effects on the linear aircraft models. Table 4.5 lists the ranges of airspeed, angle of attack, and sideslip angle encountered during each maneuver.

Table 4.5: Airspeed, angle of attack, and sideslip angle ranges during maneuvers.

	Loop	Roll	Knife Edge	Spin	Stall Turn	Tail Slide
Airspeed, ft/s	40-130	80-145	40-140	35-120	20-110	10-110
Angle of Attack, deg	4-10	-3-4	-2-14	0-18	-5-12	-10-20
Sideslip Angle, deg	-5-0	-14-10	$\pm$40	$\pm$20	-5-25	-5-20

The airspeed schedule consists of eleven different airspeed values ranging from just over the estimated stall speed of 40 ft/s up to 144 ft/s in intervals of 10 ft/s. This results in eleven different sets of gains for each controller. Figures 4.37 and 4.38 depict the variation in the knife-edge gains with airspeed while Fig. 4.39 shows the variation in the angular-rate gains using aircraft models obtained by linearizing the nonlinear simulation. The gains for the stall controller were not scheduled because the airspeed will only decrease during that phase of the spin and linear models cannot be obtained below the stall speed. The transition from maneuvering to straight-and-level flight is also controlled with constant gains due to the small changes in speed that occur during that time. The airspeed gain schedule is implemented in the simulation through the use of lookup tables for each controller. Linear interpolation is used to determine the gains for speeds between the data points and the end-point gains are used when the speed falls outside the available data.

*http://en.wikipedia.org/wiki/Small-angle_approximation

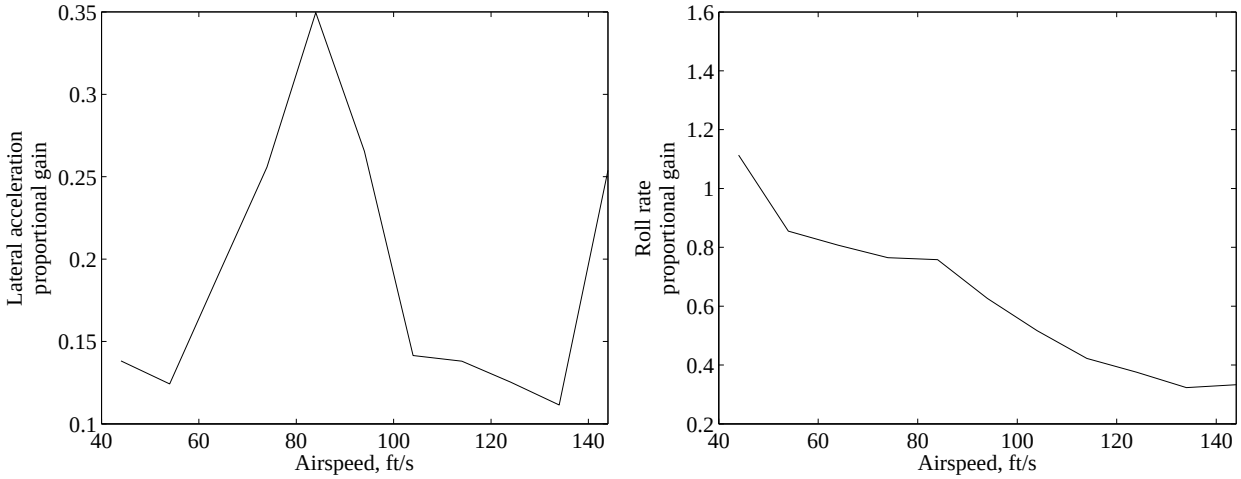


Figure 4.37: Knife-edge lateral acceleration and roll rate gain variation with airspeed.

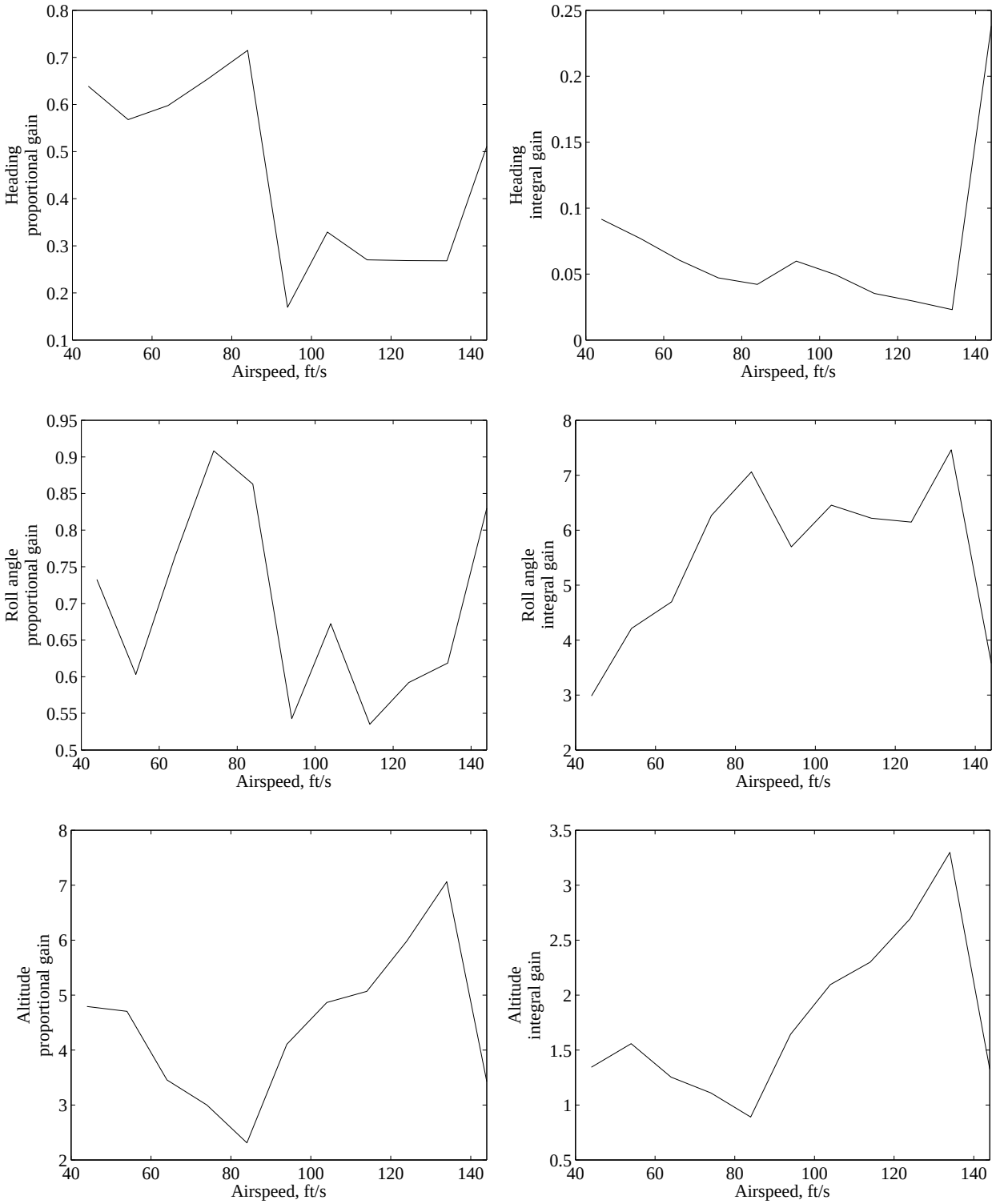


Figure 4.38: Knife-edge gain variation with airspeed.

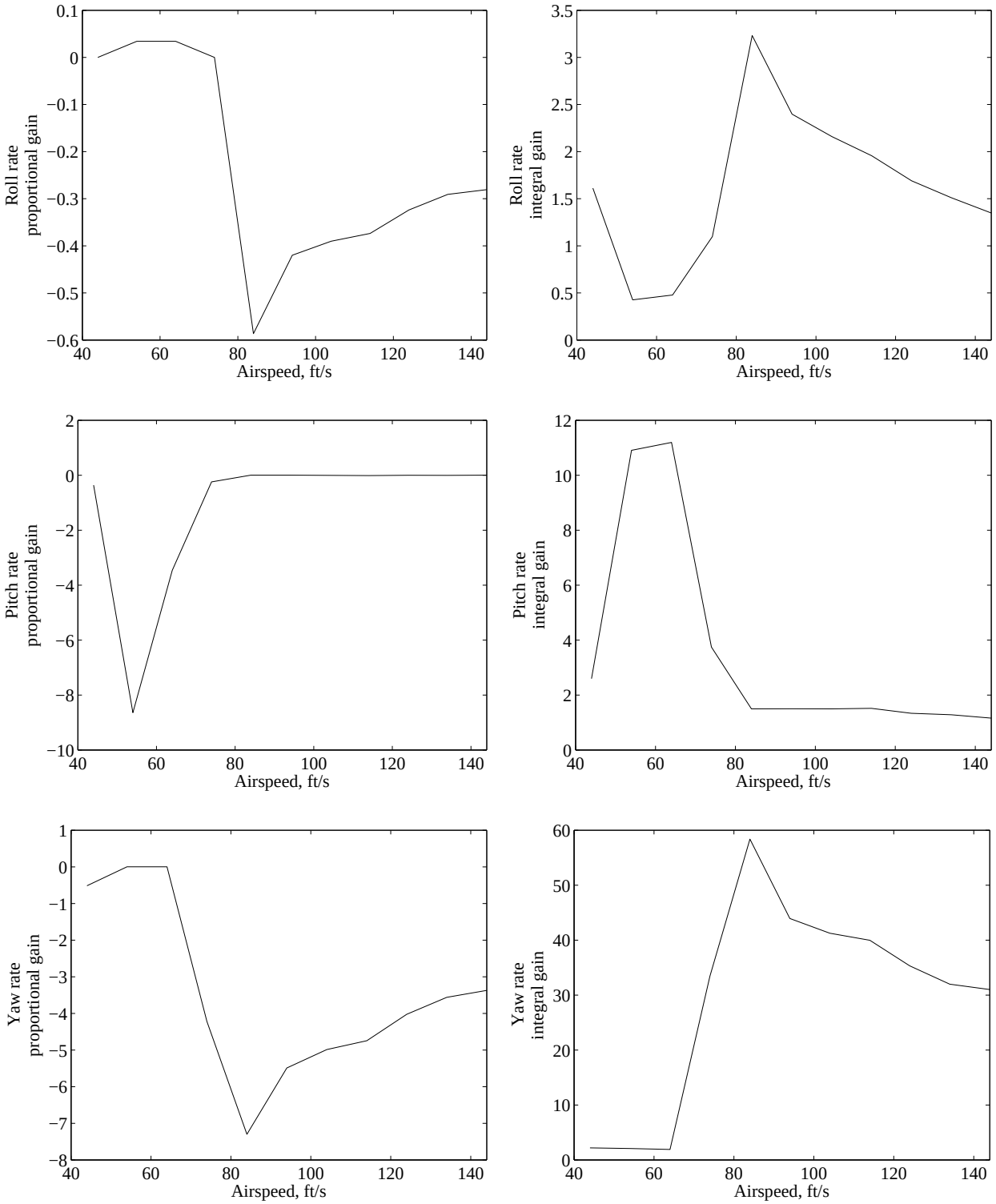


Figure 4.39: Angular-rate gain variation with airspeed.

Chapter 5

Nonlinear Simulation and Performance Comparison

Unlike the results presented in the previous chapter, which were based on linear simulations of the aircraft models, the results presented in this chapter are all based on the nonlinear aircraft model described in Chapter 2. Despite the differences in the linear models used to determine the control gains, the same nonlinear aircraft model was used to determine the performance of the control system, simulating the application of the method to an actual aircraft in flight. The two linear models are distinguished in the results by dashed lines and dotted lines. The dashed lines correspond to the gains calculated using the aircraft model obtained by linearizing the nonlinear wind tunnel data. The dotted lines correspond to the gains calculated using the aircraft model obtained from the linear EoM using data from AVL.

The altitude plots in this chapter are in terms of the height above mean sea level and the heading values in the simulation are limited based on the modulus after division such that $0° \leq \psi < 360°$. To provide physical significance to the altitude values, the vertical axis lower limit is based on the ground-level elevation of the Inyokern Airport (KIYK) located in Inyokern, California and used as the setting for the simulations. While the physical location has no real effect on the results, it allows for the visualization of the aircraft motion, which helps to verify the simulations.

5.1 Cruise

The cruise simulation began from level, trimmed flight at a speed of 80 ft/s, an altitude of 2,700 feet, and a heading of 330 degrees. Step inputs in speed, altitude, and heading were commanded at 10, 20, and 30 seconds respectively. The first simulations were performed

using the control gains found from the classical root-locus approach in Sec. 4.1. Two different linear aircraft models were used to determine the gains and Fig. 5.1 shows the comparison of the results for each model. The airspeed and altitude responses were similar for both sets of gains. The airspeed response to a command increase of 5 ft/s had a rise time of about eight seconds and a small overshoot. The altitude response to a decrease in the commanded altitude of 50 feet occurred within two seconds and resulted in an airspeed spike that was brought back to steady state in approximately 20 seconds. The heading response had the longest settling time out of the three step inputs and lasted well over one minute for both sets of gains. The gains obtained from the linearized wind tunnel data for the heading loop resulted in a response that was more damped than the gains from the EoM using AVL data. The response for the linearized tunnel data gains reached steady state after 90 seconds, while the response from the EoM gains was still oscillating with an amplitude of 10 degrees. The values in the heading plot of Fig. 5.1 were not constrained to be less than 360 degrees to more easily show that both sets of gains produced stable responses despite having long settling times and not being acceptable for a final design. The washed-out yaw rate plot shows the change in yaw rate caused by the abrupt altitude change at the 20 second mark. This response can be explained by examining the airspeed plot and visualizing what happened as the altitude was commanded to decrease. The altitude is controlled with the elevator, so a descent is achieved by pitching the aircraft nose down. The response to the nose down attitude is twofold; the altitude decreases, and the airspeed increases. The increase in speed results in a throttle response to return the airspeed back to the commanded value. This reduced power setting results in a spike in yaw rate as the trimmed value of the rudder deflection produces a yawing moment greater than the yawing moment caused by the engine at a lower-than-trimmed throttle setting. Subsequent oscillations in yaw rate coincide with the heading oscillations that occurred after 30 seconds.

The results using the root-locus method, especially the heading response, could potentially be improved with subsequent iterations of the gain values or with modifications to the

control loop structure. The reason for including the results in this dissertation, however, was not to achieve perfect aircraft performance using the root-locus method, but to highlight the time and performance benefits that can be achieved with modern control design techniques. The root-locus design approach is time intensive and the results obtained with the method greatly depend on the amount of experience the control engineer possesses. This can result in varying closed-loop performance based purely on the person performing the implementation. For this work, the results from the method were considered satisfactory and are compared to the LQT method for cruise flight.

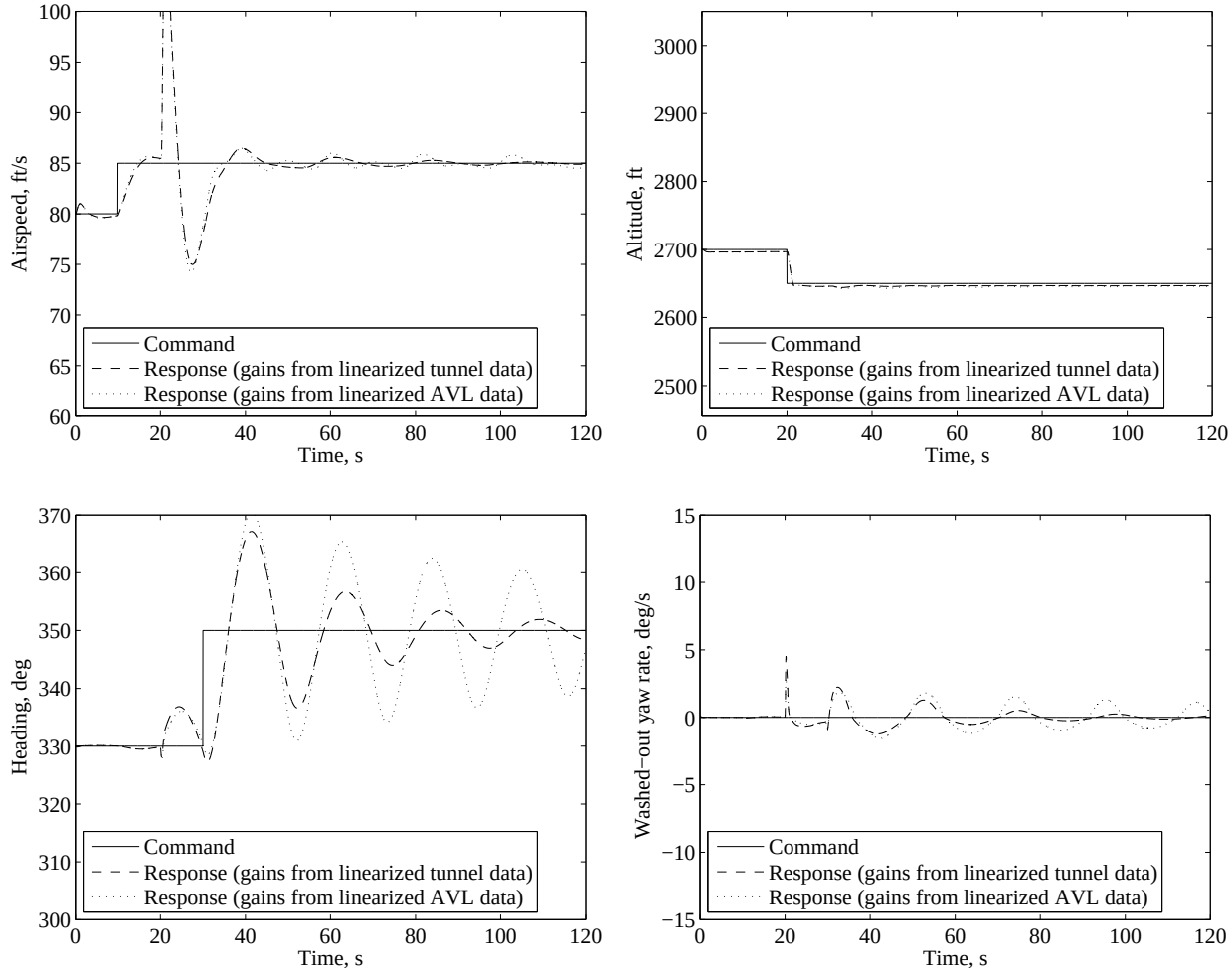


Figure 5.1: Nonlinear simulation of step inputs during cruise using gains from a classical root-locus approach.

The remainder of the results in this chapter employed the LQT method to calculate the control gains and Fig. 5.2 illustrates the first of the LQT results. The same step inputs of speed, altitude, and heading were used to test the cruise gains determined with the LQT method. Both linear aircraft models were also used to allow a comparison of the results obtained with each modeling method. The results from both sets of gains were similar for the speed and altitude responses. The 5 ft/s step in airspeed took about two seconds to complete and the 50 feet decrease in altitude required about five seconds to reach steady state. For the gains from the linearized tunnel data, the 20 degree change in heading took approximately two seconds while the gains from the EoM model resulted in a rise time of six seconds. The washed-out yaw rate plot shows the disturbances caused by the step inputs with each disturbance returning to the commanded zero value in around seven seconds for both sets of gains.

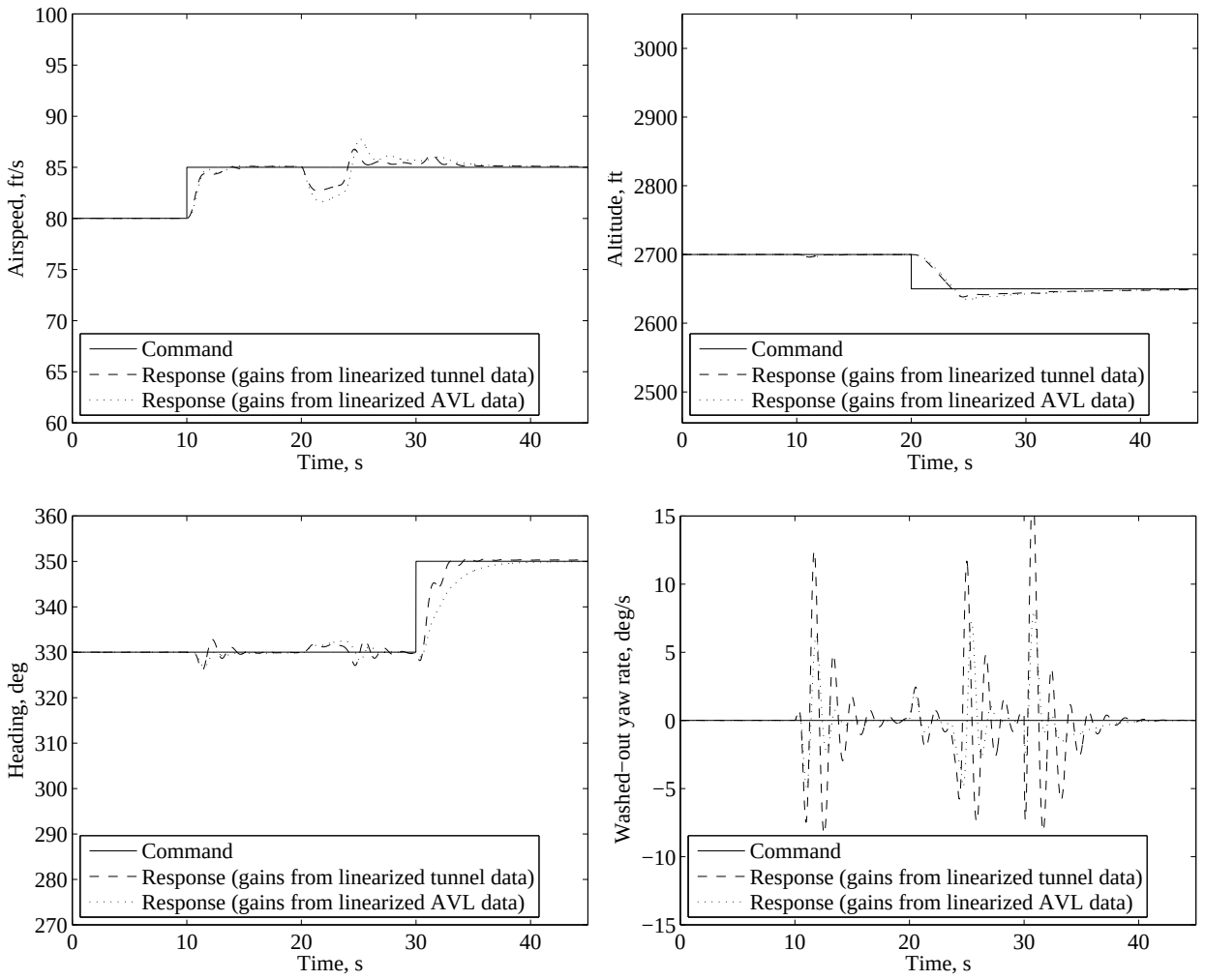


Figure 5.2: Nonlinear simulation of step inputs during cruise using gains from the LQT approach.

When the results of the LQT calculated gains are compared to the results of the gains determined with the root-locus method, the largest difference is in the heading response. The heading response was much faster using the gains from the LQT and this result was to be expected due to the optimality of the method. What was not expected was how well the speed and altitude responses would compare between the two methods. The reader is reminded here that for the root-locus method, the speed was controlled with the throttle and the altitude was controlled with the elevator to account for the SISO nature of the tool. Due to the structure of the state-space matrices in Appendix B, changes in engine speed from

throttle inputs can not provide direct changes to the altitude. The only state effected by changes in engine speed is the forward speed u. The opposite was true for the LQT method, where the speed was controlled with elevator and the altitude was controlled with throttle. This was possible due to the MIMO nature of the tool. For the gains from the root-locus method, the throttle loop (controlling airspeed) had a step response of eight seconds while the gains from the LQT for the throttle loop (controlling altitude) had a response of five seconds. The elevator loop for the gains from the root-locus method (controlling altitude) had a step response of two seconds, which was the same as the response from the LQT gains for the elevator loop (controlling airspeed). This result is interesting to note because similar performance was able to be obtained with the classical method. The only disadvantages with the root-locus approach are the need to change the control structure to accommodate the SISO nature of the method, the time required to produce the result, and the slow response associated with the heading control loop.

5.2 Maneuvering

The maneuvering flight simulation results are presented in the same order the maneuvers were introduced in Sec. 3.2. The results for the basic aerobatic maneuvers are given first, followed by the results for the post-stall maneuvers. The simulations in this section began from level, trimmed flight at a speed of 80 ft/s and a heading of 330 degrees and ended as soon as recovery from the maneuver was initiated. For all of the maneuvering simulations, two sets of gains (from different linear models) were used to compare the results obtained from each method. The first results are for the autonomous loop and are shown in Figs. 5.3 - 5.6. Figure 5.3 illustrates the flight path during the loop with the aircraft starting from the right side and progressing to the left of the figure. Both sets of gains resulted in the characteristic shape of the looping maneuver. Figure 5.4 depicts consecutive screen captures of the visualization environment during the loop simulation and aids in visualizing the aircraft orientation throughout the maneuver. The maneuver began four seconds into the simulation

and ended after 15 seconds, once the aircraft returned to level flight. The aircraft states during the simulation are shown in Figs. 5.5 and 5.6. The gains from both linear models provided great performance in terms of tracking the angular-rate commands and the results from both gain sets were actually quite similar. The airspeed, altitude, and heading during the maneuver behaved in a similar manner as for the pilot-controlled loop, confirming that the simulation performed the maneuver correctly. The angle of attack and the sideslip angle during the loop are plotted in Fig. 5.6 and are relatively similar for both sets of gains. The angle of attack reached a maximum value of 10 degrees while the maximum sideslip angle was only five degrees.

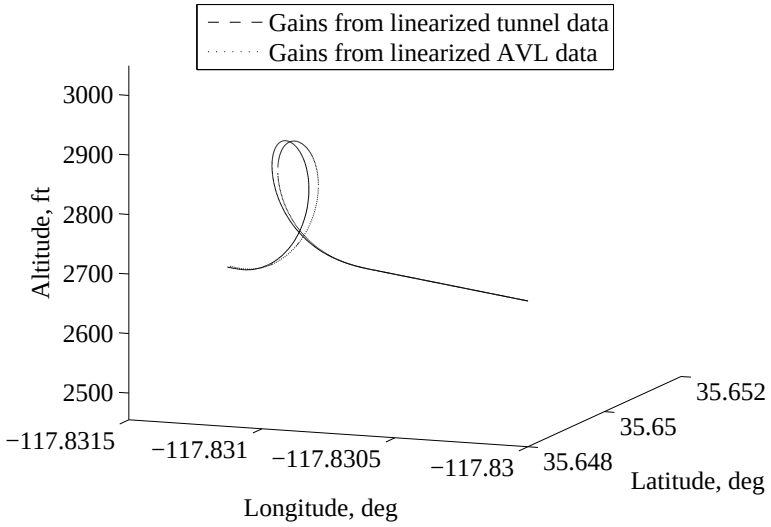
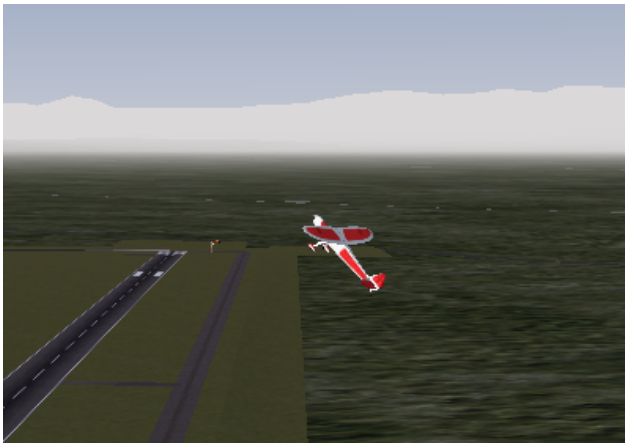


Figure 5.3: Flight path during an autonomous loop.

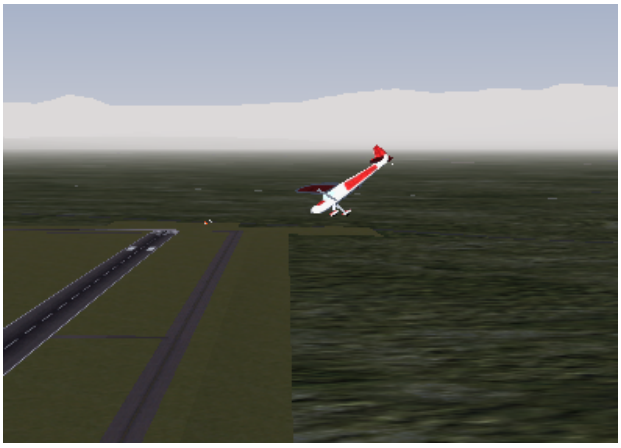
(a) time = 4 s



(b) time = 8 s

(c) time = 10 s

(d) time = 12 s



(e) time = 14 s

(f) time = 15 s

Figure 5.4: Screen capture during an autonomous loop.

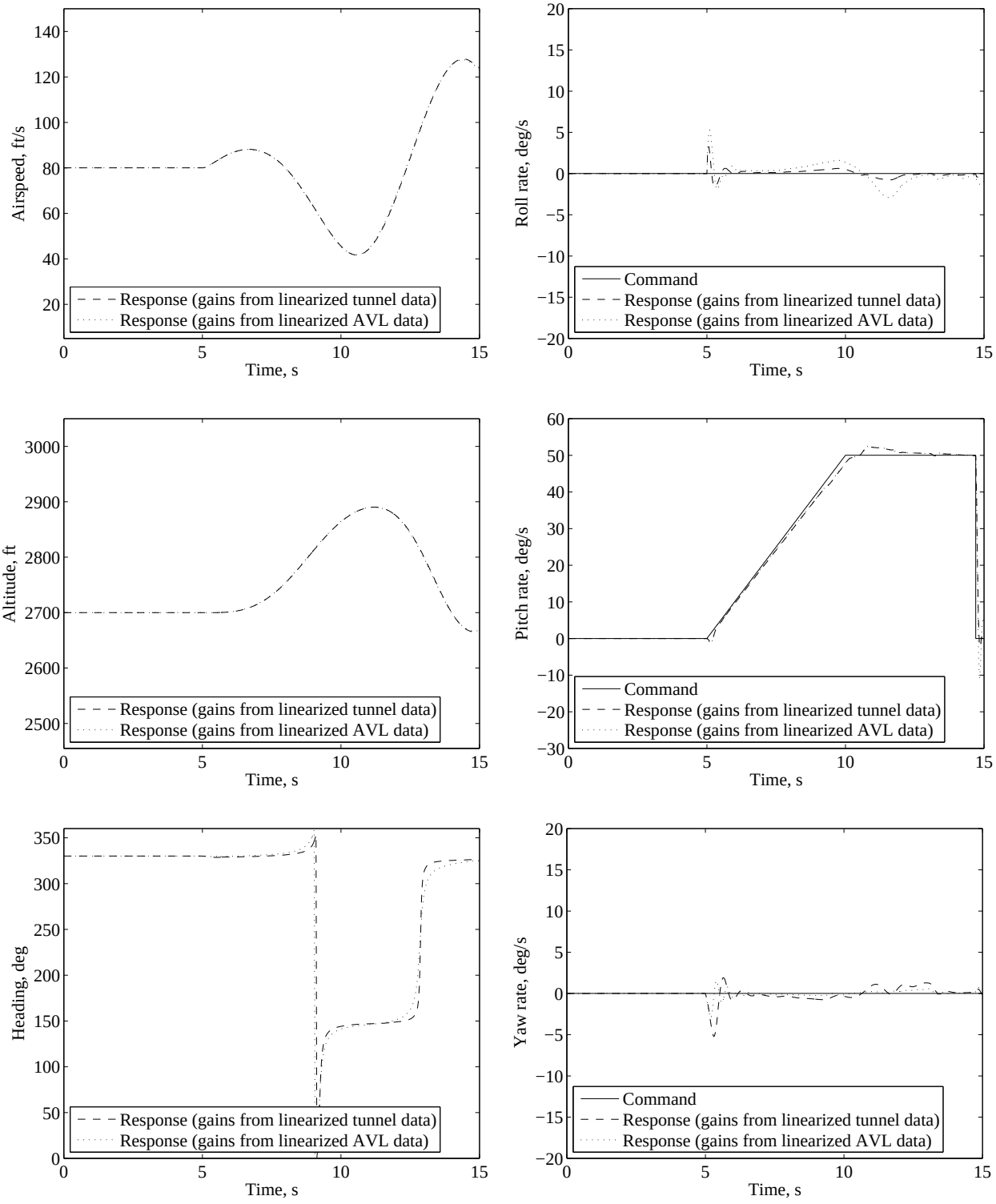


Figure 5.5: Nonlinear simulation comparison during an autonomous loop.

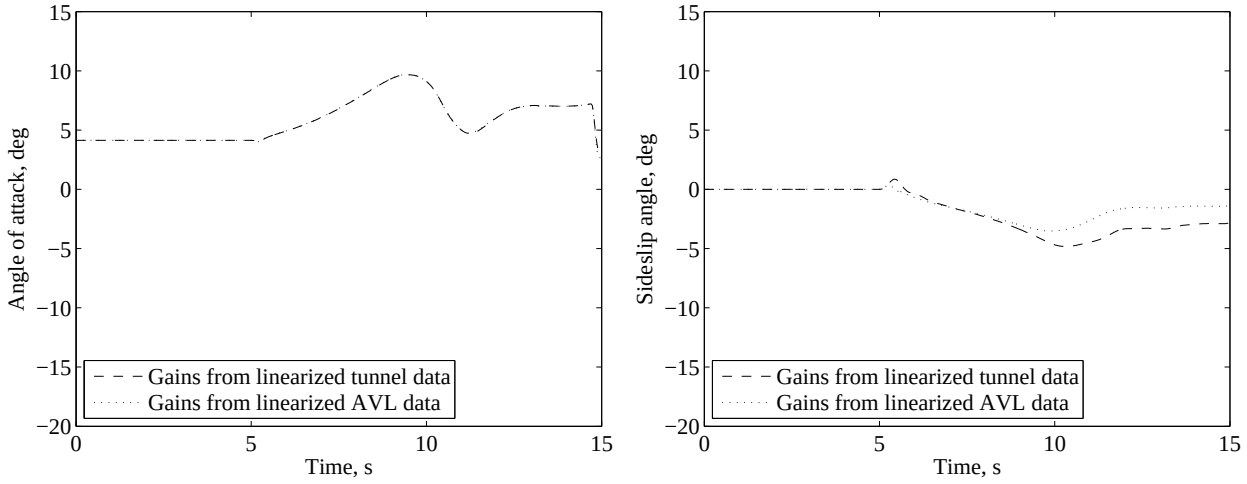


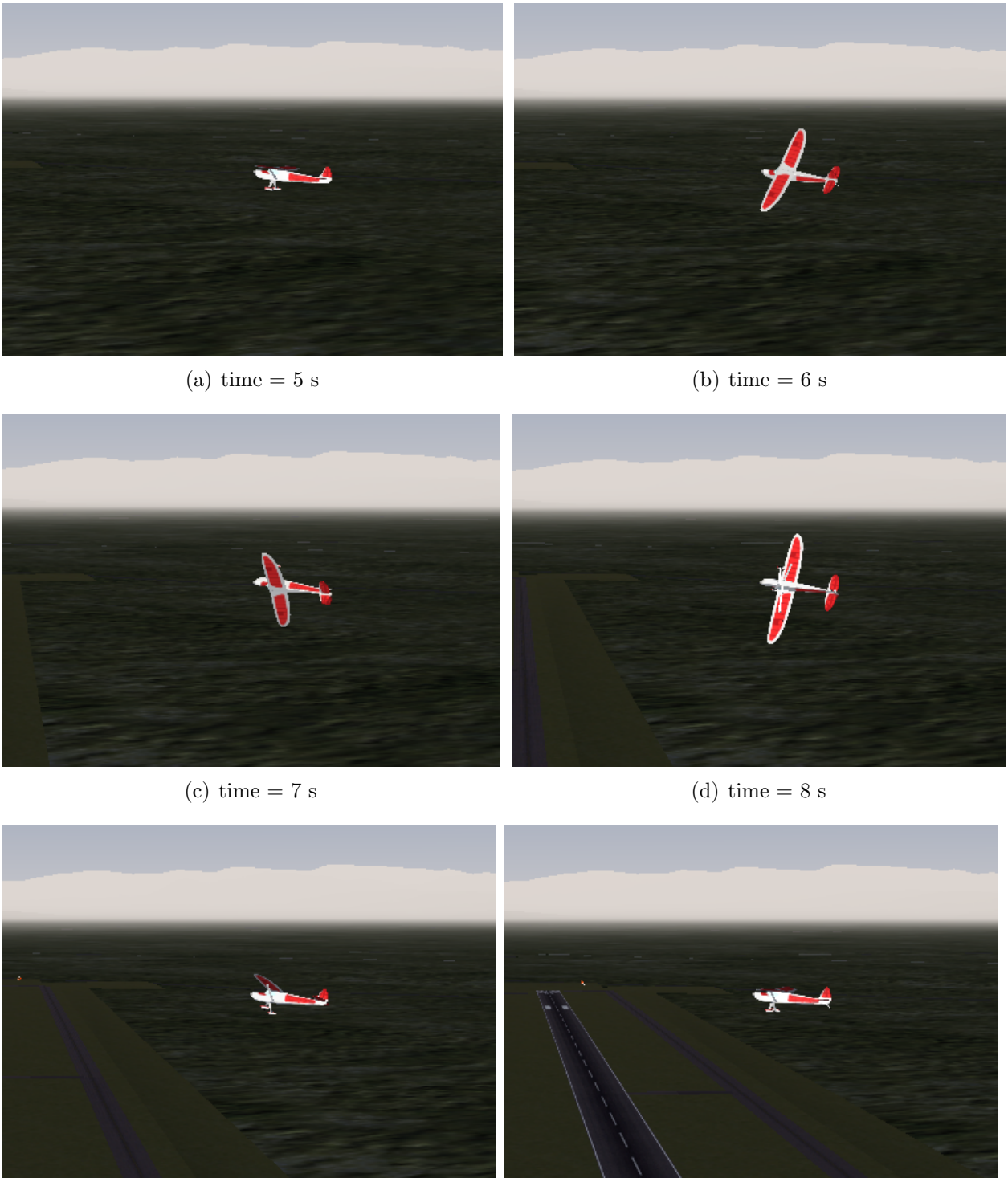
Figure 5.6: Angle of attack and sideslip angle during an autonomous loop.

The simulation results for the roll maneuver are presented next and appear in Figs. 5.7 - 5.11. Both left rolls and right rolls were simulated to ensure that the control strategy would work in both directions despite the moments from the engine. Figure 5.7 depicts consecutive screen captures of the visualization environment during a left roll. The maneuver began a little over four seconds into the simulation with a slight pitch up, to minimize the altitude lost during the roll, and ended after 12 seconds with the aircraft back in straight-and-level flight. Pictures of a right roll are not shown because the motion is nearly identical, just in the opposite direction. The aircraft states during the rolling maneuvers are shown in Figs. 5.8 - 5.11 for both sets of control gains. The altitude and heading remained relatively constant during the maneuver, which was to be expected. The increase in airspeed, as well as the first disturbance seen on the angular rate plots, is due to the throttle increase that took place after two seconds. The throttle was increased from the trim value to full power in an attempt to maximize the available lift while at high bank angles. For the initial simulation of the roll maneuver presented in this section, the recovery back to level flight was excluded. This exclusion explains the deviation in the altitude toward the end of the maneuver. Since only the angular rates were controlled, the aircraft may have had a positive pitch angle when

it stopped rolling, resulting in a gradual climb. This effect is most notable in the altitude plot of Fig. 5.9 for the gains from the EoM model.

Reviewing the angular rate plots, both sets of gains appear to be able to track the roll- and pitch-rate commands with reasonable accuracy. During both rolls, the gains found from the EoM model with AVL data resulted in a slight overshoot after each change in the roll-rate command, but still performed relatively well. The gains from the linearized tunnel data were able to track the changes in roll and pitch rate for both roll directions with great accuracy. The yaw-rate results were not as good, with both sets of gains struggling to track the nonlinear reference command for both roll directions. The yaw-rate command for the roll was a sine function of roll angle so that the yaw rate would be maximum when the aircraft was at 90 degrees of bank. This command, which was determined from the pilot-controlled data, should help to minimize the altitude loss associated with the reduced lift during high bank angle conditions. Despite both sets of gains struggling to track the yaw-rate reference command, the altitude during the maneuver remained relatively constant. From the yaw-rate plots, it appears that the gains from the EoM model with AVL data were able to track the slopes of the reference commands for both roll directions. The gains from the linearized tunnel data were able to exactly track the yaw-rate command for the first half second of the roll. Both sets of gains, however, were able to track the zero yaw-rate command at the end of the maneuver.

Figures 5.10 and 5.11 show the angle of attack and sideslip angle during both rolls. The angle of attack data is pretty similar for both rolls, with the maximum value actually occurring during the initial level-flight portion of the maneuvers. The sideslip data is similar for both rolls initially. The sideslip angles during the rolls have similar magnitudes, with opposite signs for each roll direction. This result is exactly as intuition would lead due to the orientation of the aircraft being similar throughout each roll.



(a) time = 5 s

(b) time = 6 s

(c) time = 7 s

(d) time = 8 s

(e) time = 9 s

(f) time = 11 s

Figure 5.7: Screen capture during an autonomous roll.

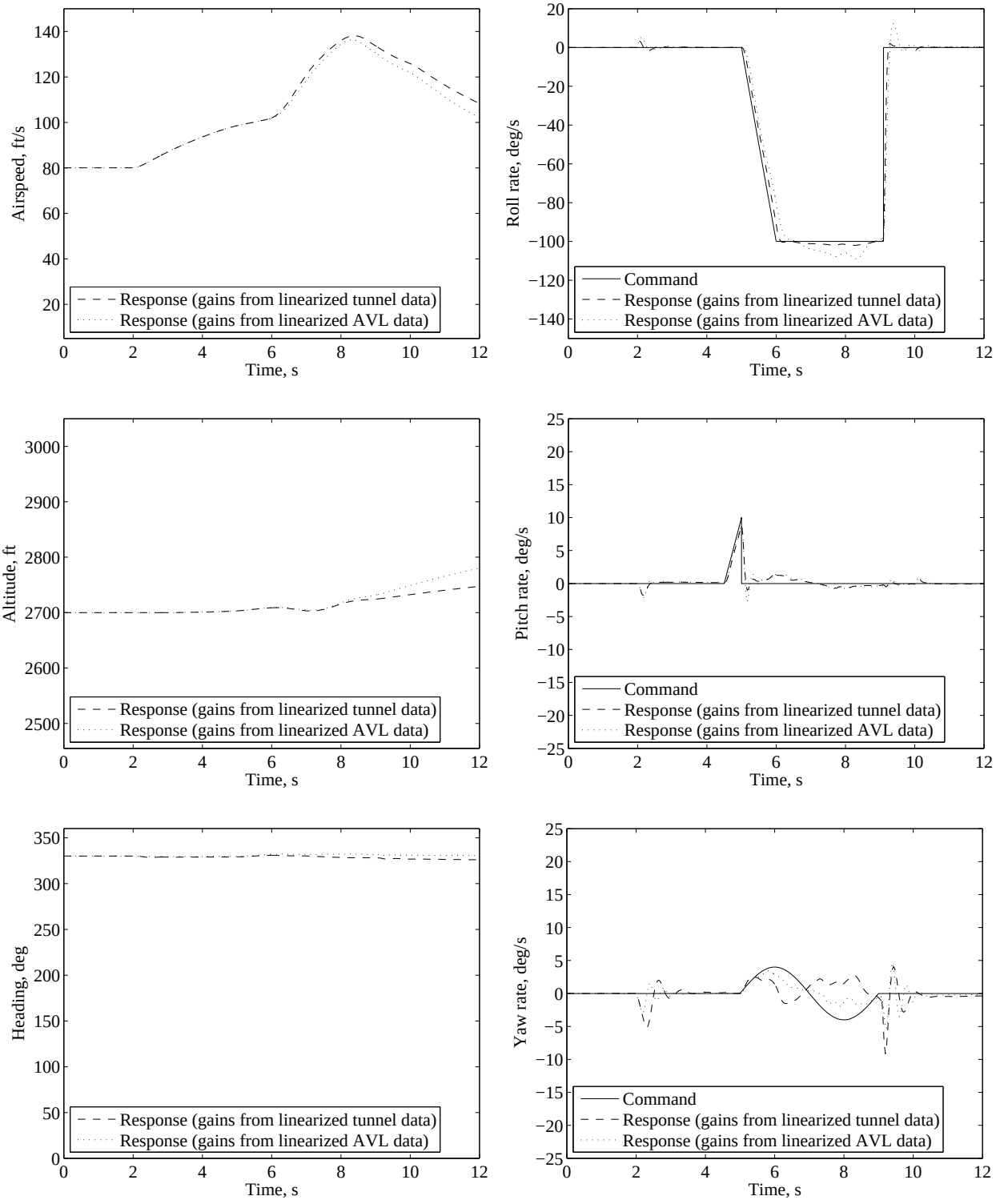


Figure 5.8: Nonlinear simulation comparison during an autonomous left roll.

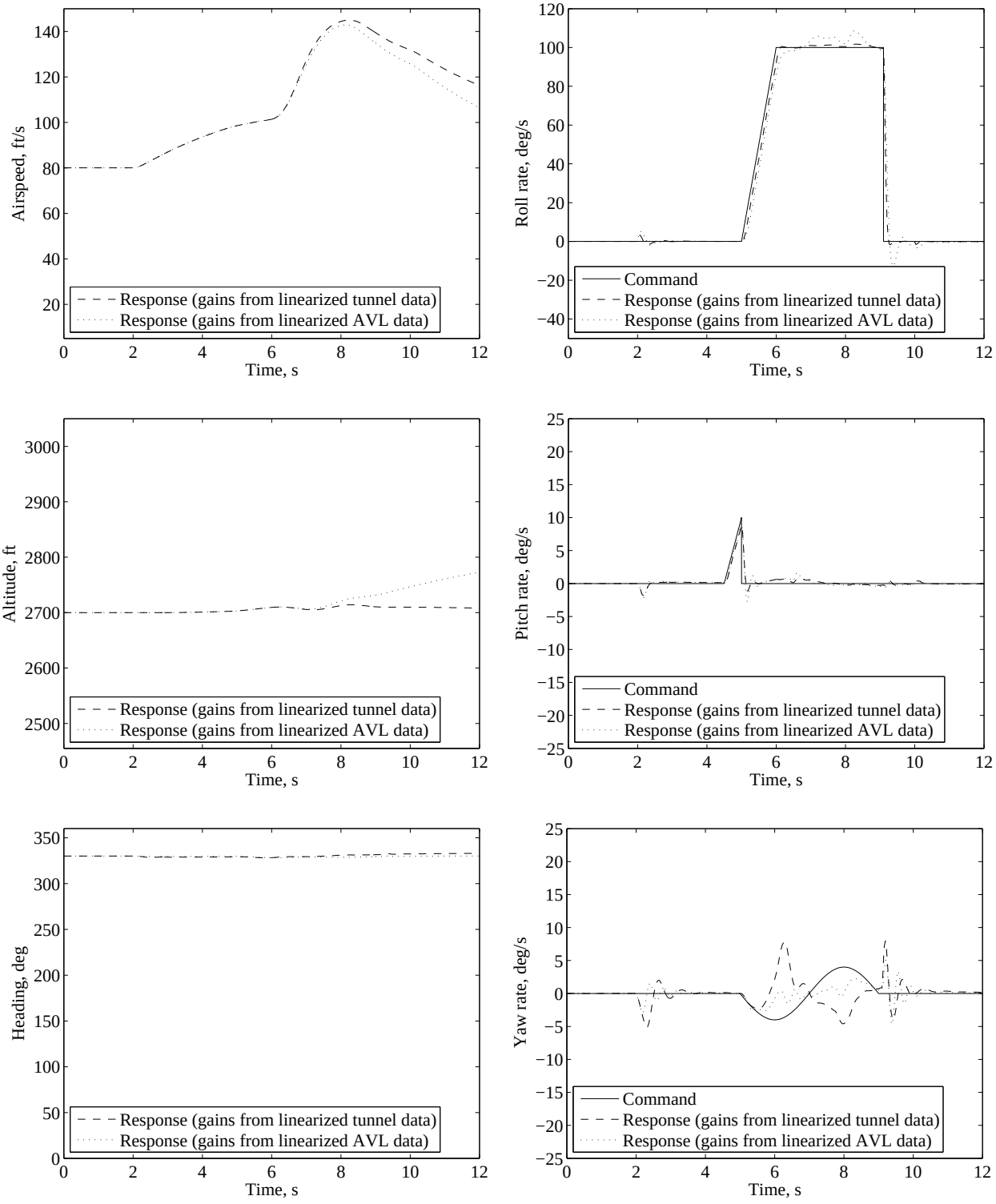
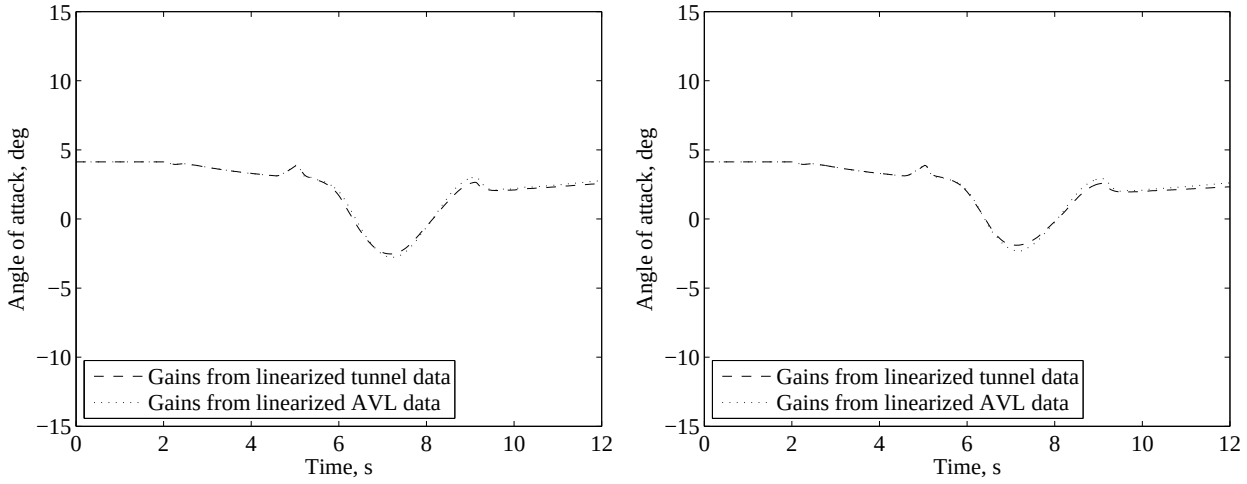
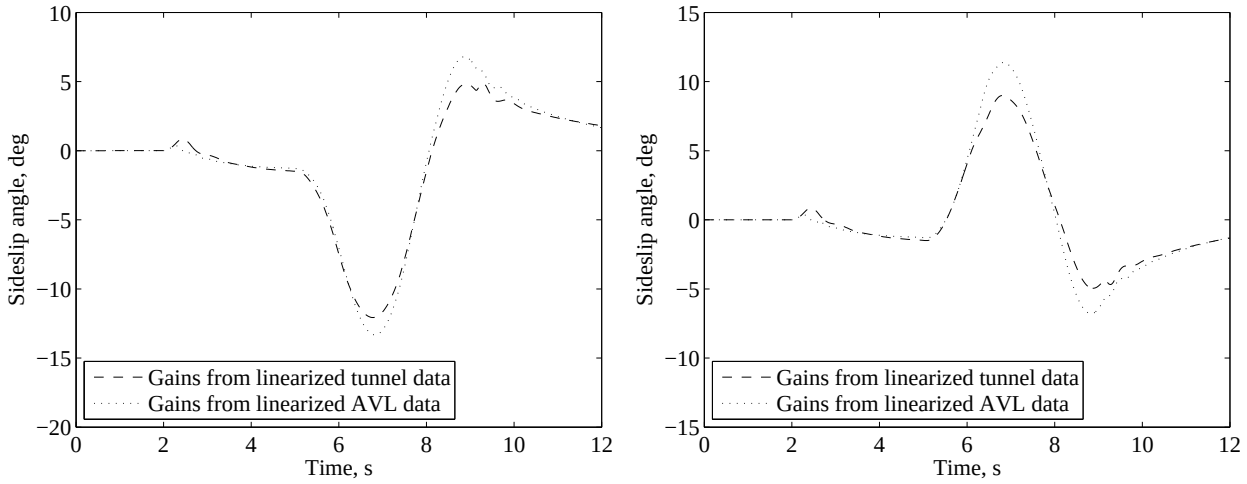


Figure 5.9: Nonlinear simulation comparison during an autonomous right roll.



(a) Left roll (b) Right roll

Figure 5.10: Angle of attack during autonomous rolls.



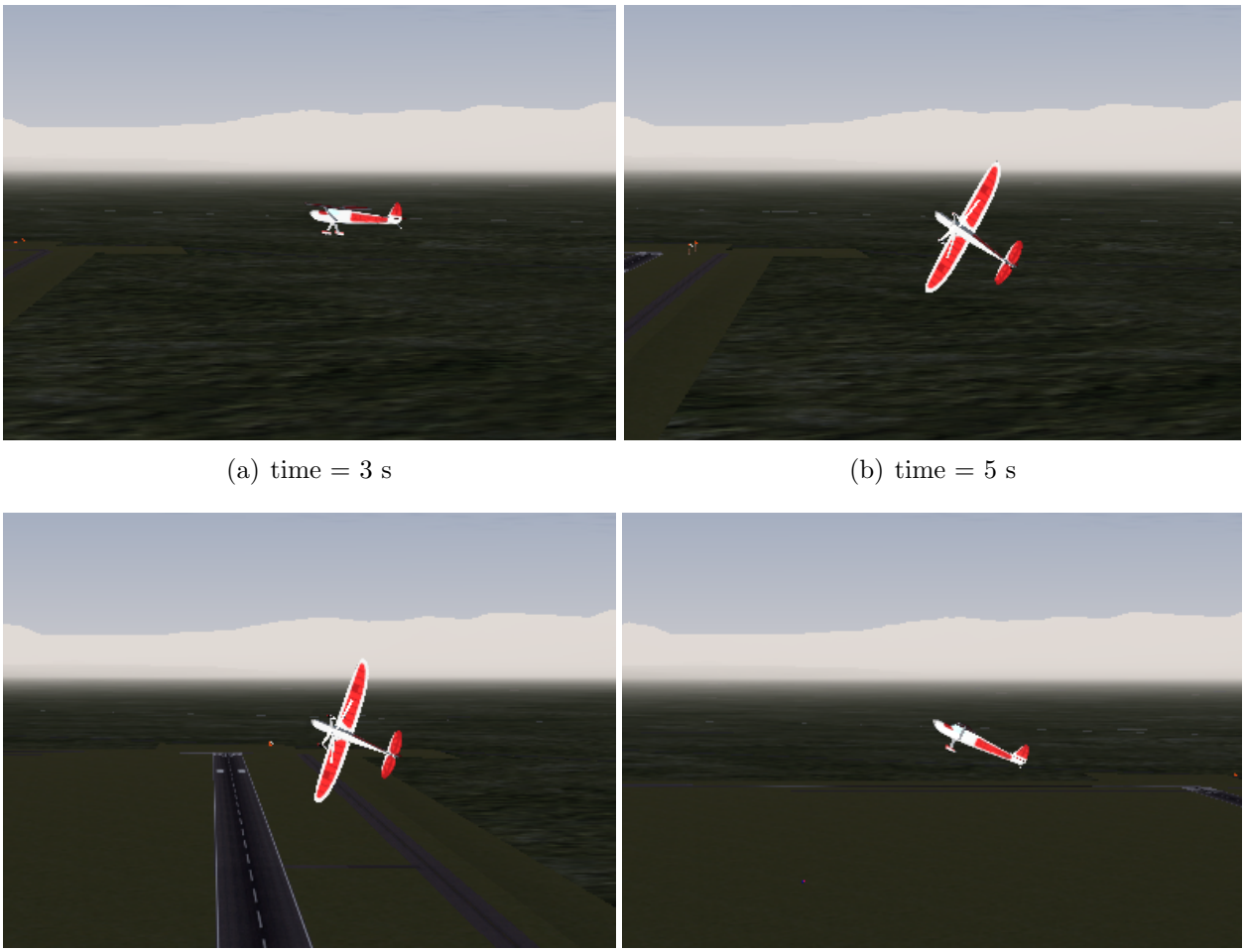
(a) Left roll (b) Right roll

Figure 5.11: Sideslip angle during autonomous rolls.

The results for the simulation of knife-edge flight appear in Figs. 5.12 - 5.17. The knife-edge simulation started from level flight at 80 ft/s. The throttle was increased from the trim value to full power after the first second to increase the airspeed, allowing the side area of the fuselage to generate more lift. Figure 5.12 shows the orientation of the aircraft for right knife-edge flight, but simulations were performed with the aircraft banked to both the right and the left. Knife-edge flight was controlled by the angular-rate controller for the

initial and final portions of the maneuver when the roll angle was less than 45 degrees. Once the roll angle was larger than 45 degrees, the knife-edge controller took over by tracking roll angle, altitude, and heading commands. Figures 5.13 and 5.14 show the initial angular rate commands used to roll the aircraft into the knife-edge orientation. Both sets of control gains were able to track the step command in roll rate quite well for rolls to both the left and right. The pitch and yaw rates were commanded to remain zero for the initial portion of the maneuver and both sets of control gains were able to track the commands. During knife-edge flight, both sets of gains were also able to track the constant heading command. The altitude command is set four feet above the starting altitude to prevent the knife-edge controller from commanding an immediate descent if the actual altitude, when the knife-edge controller takes over, is slightly above the target altitude. For both roll directions, the gains from the linearized tunnel data were able to more closely track the altitude command. When simulating knife-edge flight to the left, the gains from the tunnel data were able to maintain altitude within 25 feet of the reference command. Knife-edge flight to the right proved more difficult with an altitude loss of over 50 feet before returning to the reference command. The gains from the EoM model were also able to maintain control during knife-edge flight, but the slow response to the initial altitude error resulted in a larger deviation from the reference command.

Figure 5.15 shows the roll angle response to the reference command for both roll directions. When the knife-edge controller first took over, the roll angle had to be increased from 45 degrees to 90 degrees and the gains from the linearized tunnel data provided a good response during the transition. For the knife-edge portion of the maneuver, the same gains were able to track the 90 degree reference command with great precision. The transition was not as quick when using the gains from the EoM model. Roughly four seconds passed before the reference command was attained for left knife edge and the settling time was even greater for right knife edge.



(a) time = 3 s

(b) time = 5 s

(c) time = 10 s

(d) time = 22 s

Figure 5.12: Screen capture during autonomous knife-edge flight.

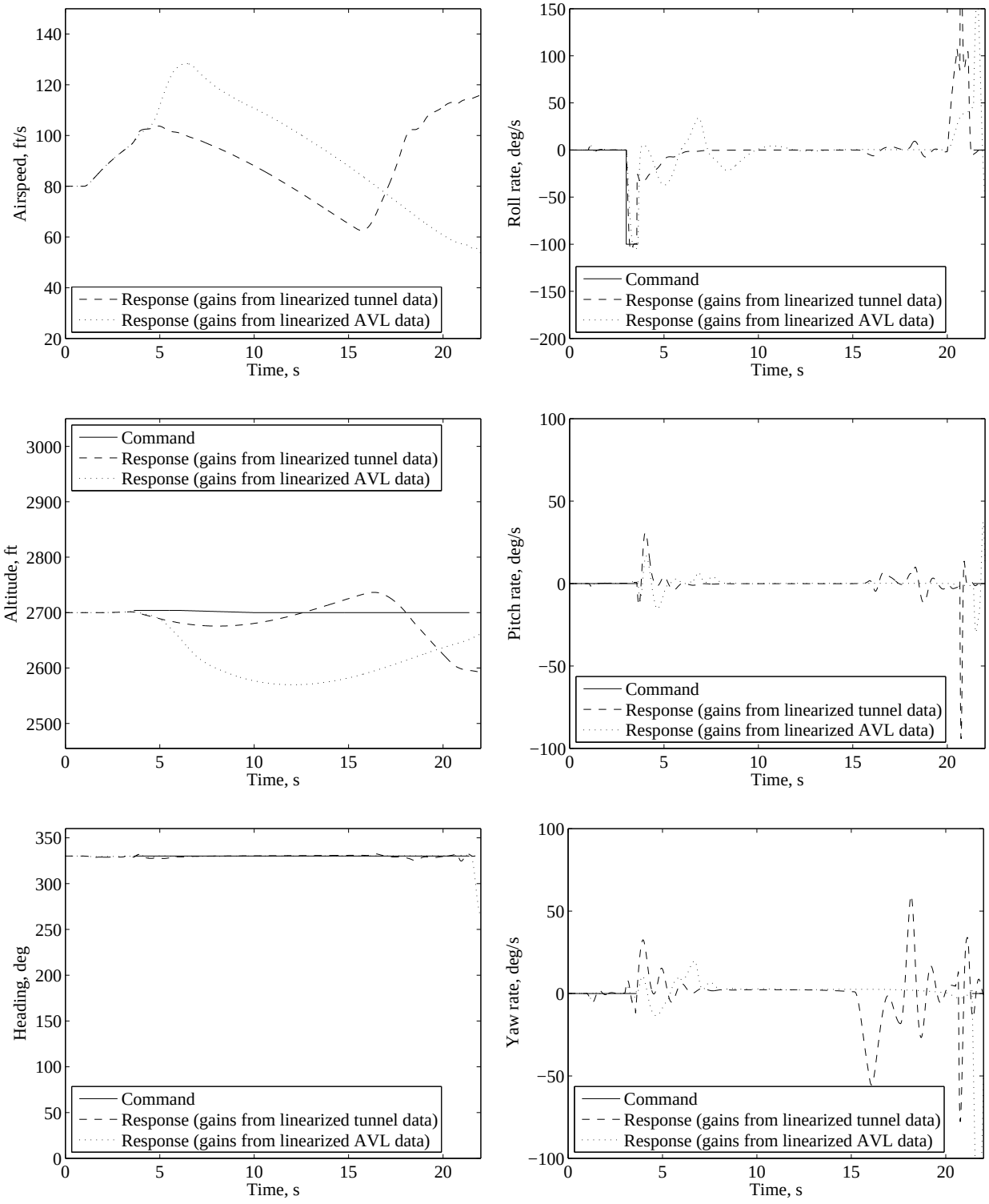


Figure 5.13: Nonlinear simulation comparison during autonomous left knife-edge flight.

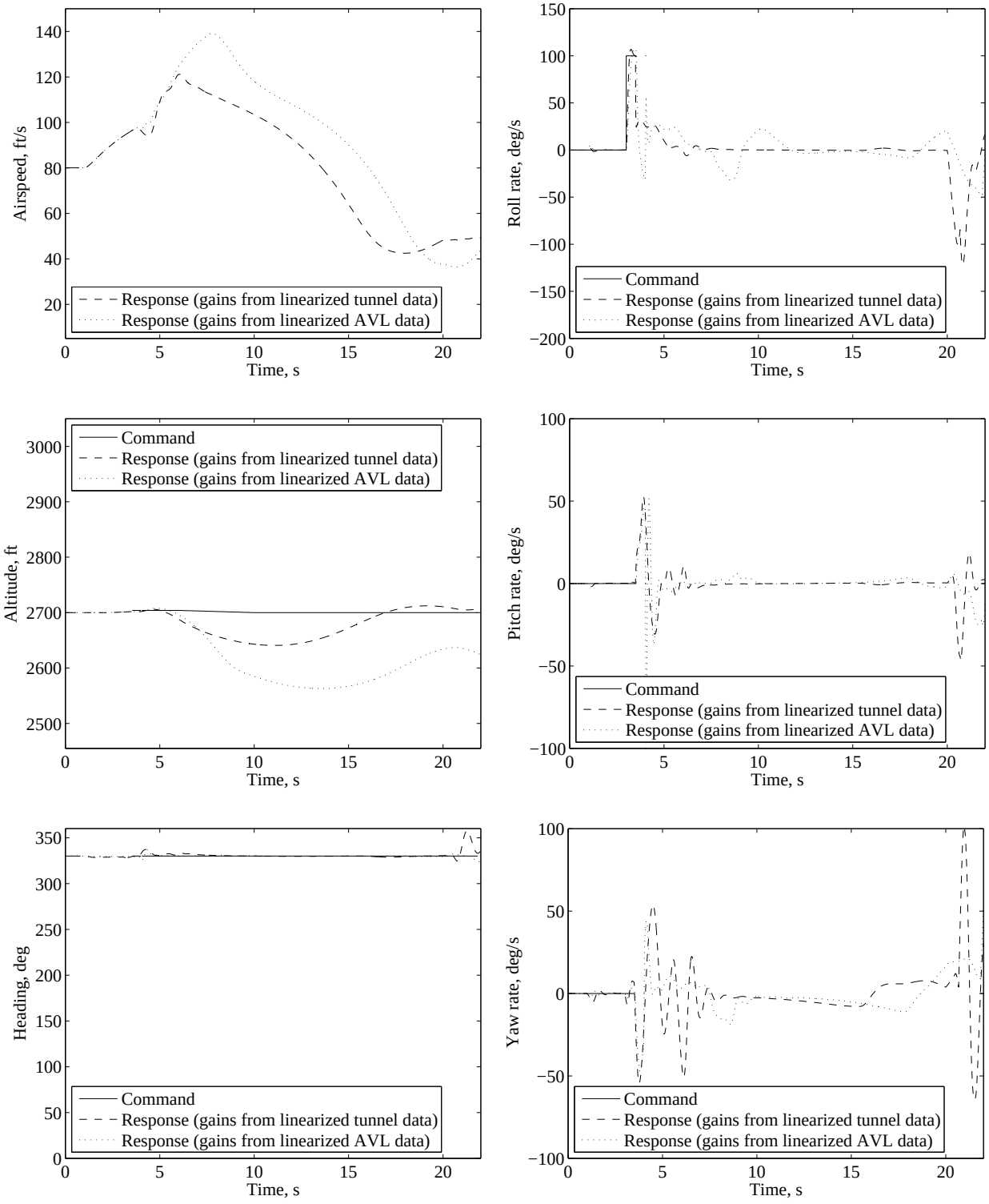
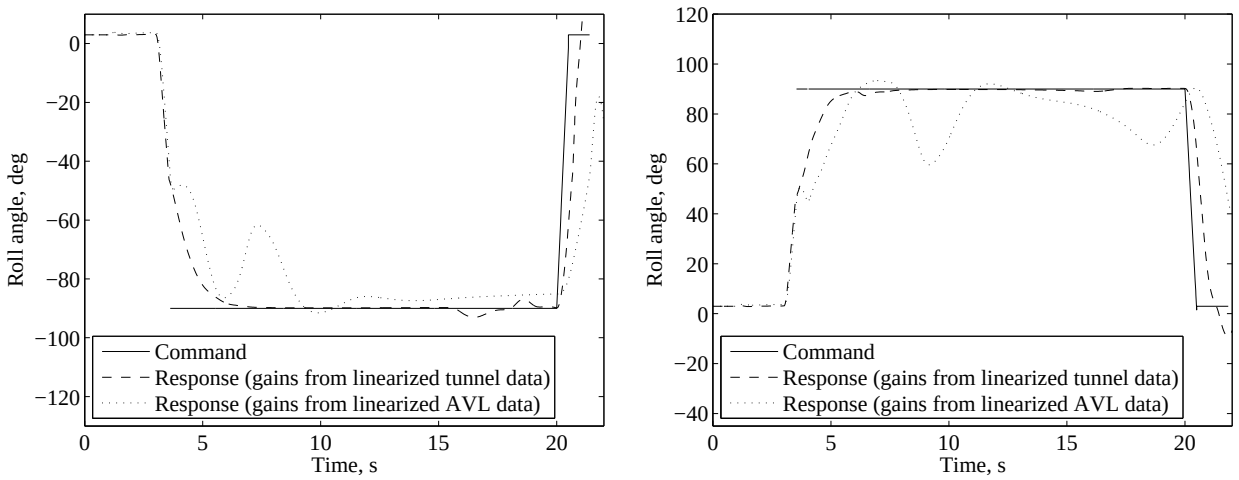


Figure 5.14: Nonlinear simulation comparison during autonomous right knife-edge flight.

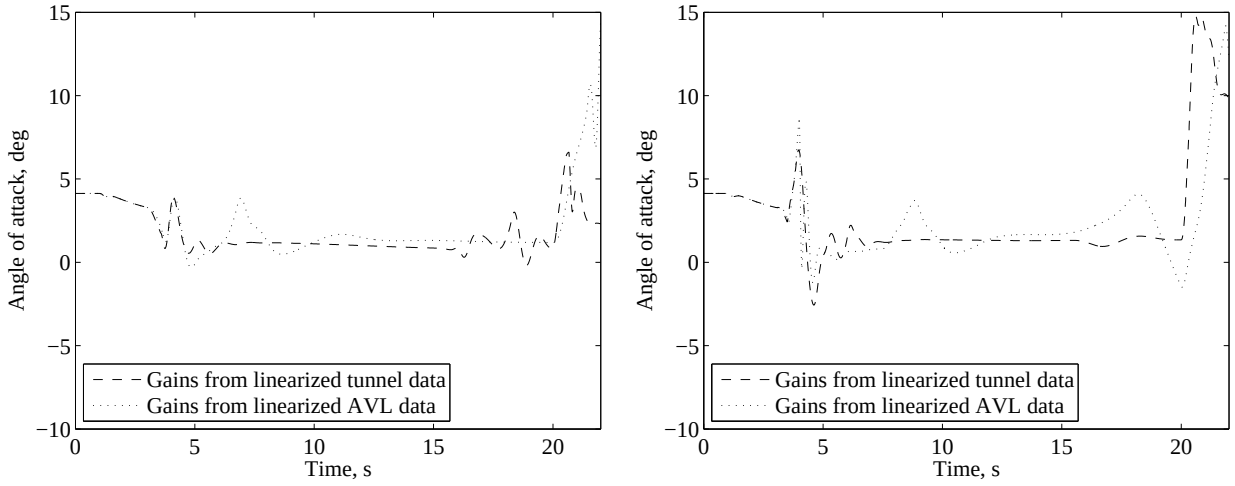


(a) Left knife edge

(b) Right knife edge

Figure 5.15: Roll angle during autonomous knife-edge flight.

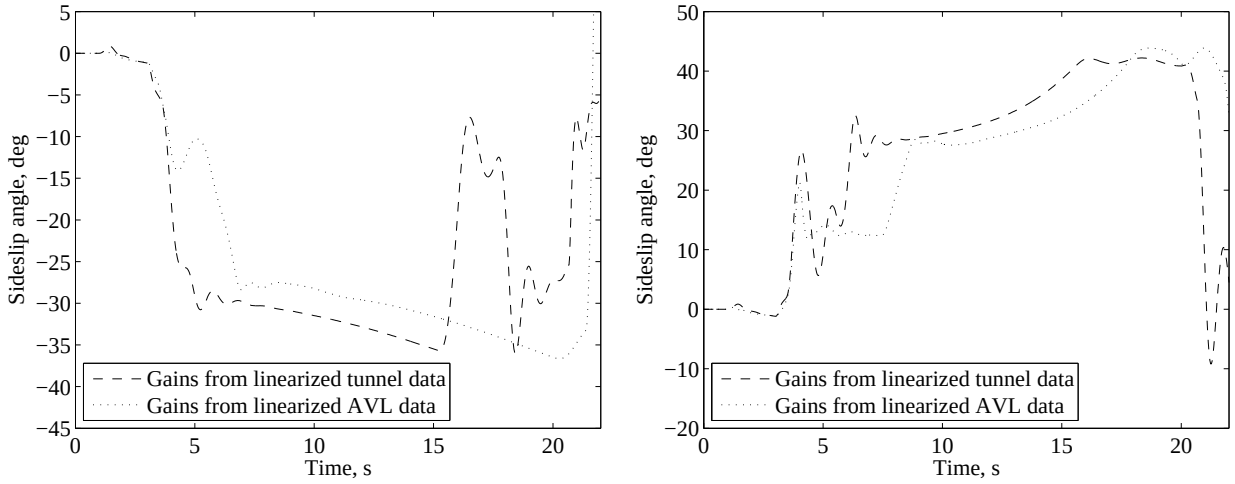
The angle of attack, shown in Fig. 5.16, remained relatively small throughout the maneuver until recovery was initiated. During the roll back to level flight, both roll directions resulted in angle of attack spikes. The spike in angle of attack was caused by the large sideslip angles required during knife edge. When the aircraft was rolled back upright, the roll took place around the X-body axis. This resulted in the sideslip angle being transformed into a combination of angle of attack and pitch angle. Figure 5.17 revels how much sideslip angle was required for the fuselage side to provide the necessary lift to sustain knife-edge flight. When banked to the left, the sideslip angle ranged between 30 and 35 degrees, while for right knife edge, the sideslip angle exceeded 40 degrees.



(a) Left knife edge

(b) Right knife edge

Figure 5.16: Angle of attack during autonomous knife-edge flight.

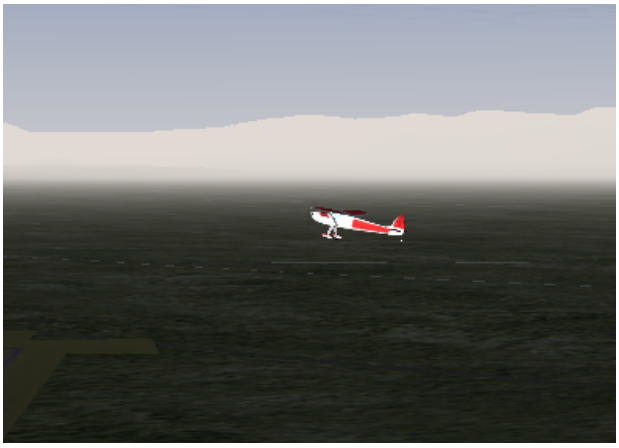


(a) Left knife edge

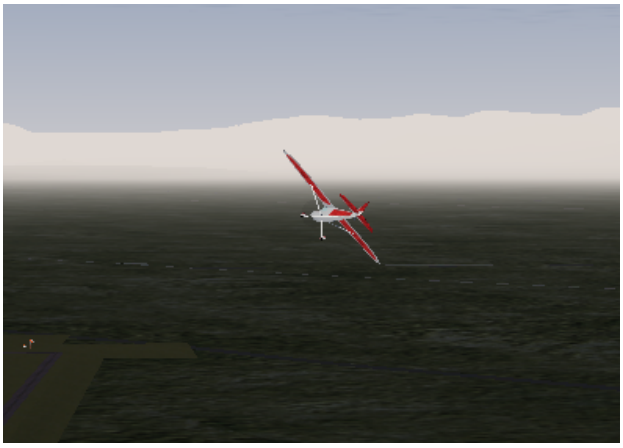
(b) Right knife edge

Figure 5.17: Sideslip angle during autonomous knife-edge flight.

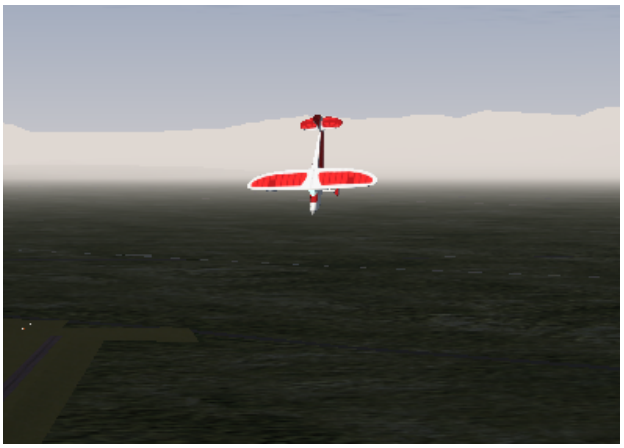
The remainder of the results presented in this section are for the post-stall maneuvers which include the spin, stall turn, and tail slide. Figures 5.18 - 5.23 contain the simulation results for the spin. Both left and right spins were simulated and Fig. 5.18 depicts successive screen captures of the visualization environment during a spin to the right. The spin simulation began at an airspeed of 44 ft/s and an altitude of 3,000 feet. To enter a spin, the aircraft must first be stalled. As the aircraft stalls, pro-spin inputs of aileron, elevator, and rudder induce the spinning motion that characterize the maneuver. The spin simulations in this work never became fully-developed autorotations, so the pro-spin inputs were required throughout the maneuver. For the initial phase of the maneuver, the aircraft was stalled by using the elevator to minimize the airspeed while the ailerons maintained zero roll angle and the rudder maintained constant heading. Once the critical angle of attack was reached, the angular-rate controller was used to induce the spin. The plots of airspeed, heading, and roll angle in Figs. 5.19 - 5.21 show the results for the stall portion of the maneuver. Starting from just above the estimated stall speed, the airspeed was commanded to zero to induce the stall. Both sets of gains tested were able to track the commanded heading and roll angle while minimizing airspeed. As the aircraft began to stall, the angular rate controller induced the spin and the angular rate plots of Figs. 5.19 and 5.20 show the results for this phase of the maneuver. The roll and yaw rate controllers performed the best for both sets of gains. The response from the pitch rate controller exhibited the same shape as the command, but contained oscillations for both sets of gains. Recovery from the spin was initiated by commanding the rates to zero and both sets of gains exhibited fast settling times to the zero reference commands.



(a) time = 3 s

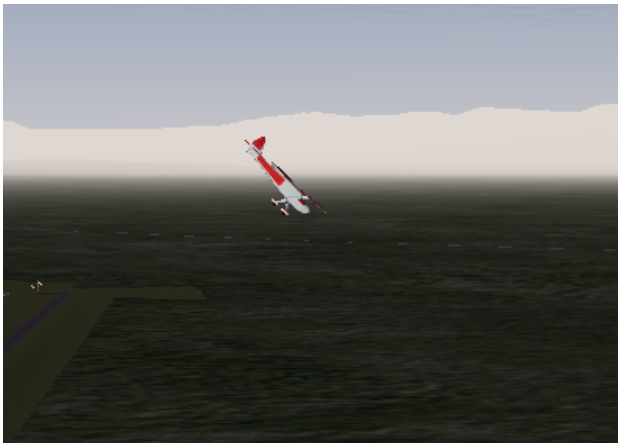


(b) time = 6 s

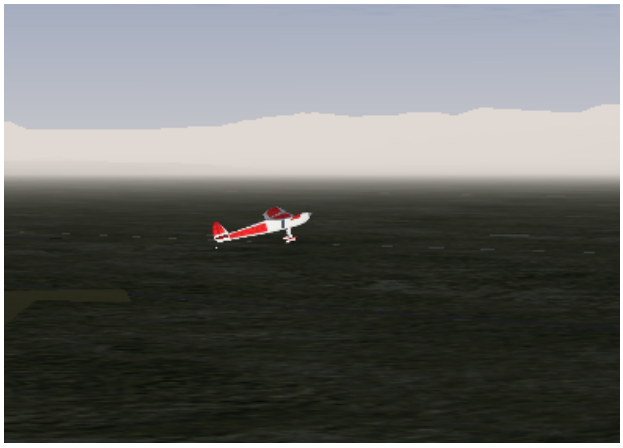


(c) time = 8 s

(d) time = 9 s



(e) time = 10 s



(f) time = 15 s

Figure 5.18: Screen capture during an autonomous spin.

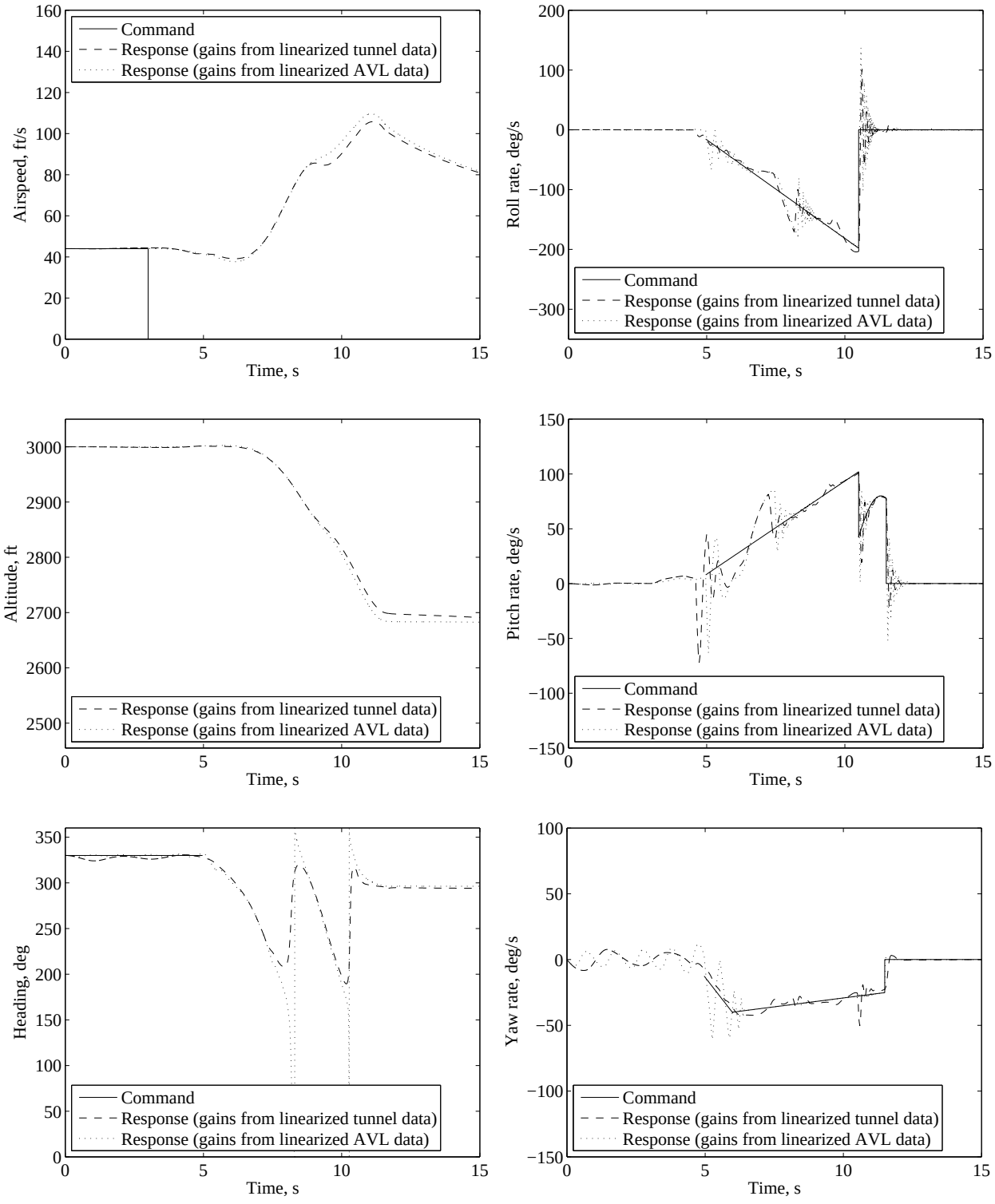


Figure 5.19: Nonlinear simulation comparison during an autonomous left spin.

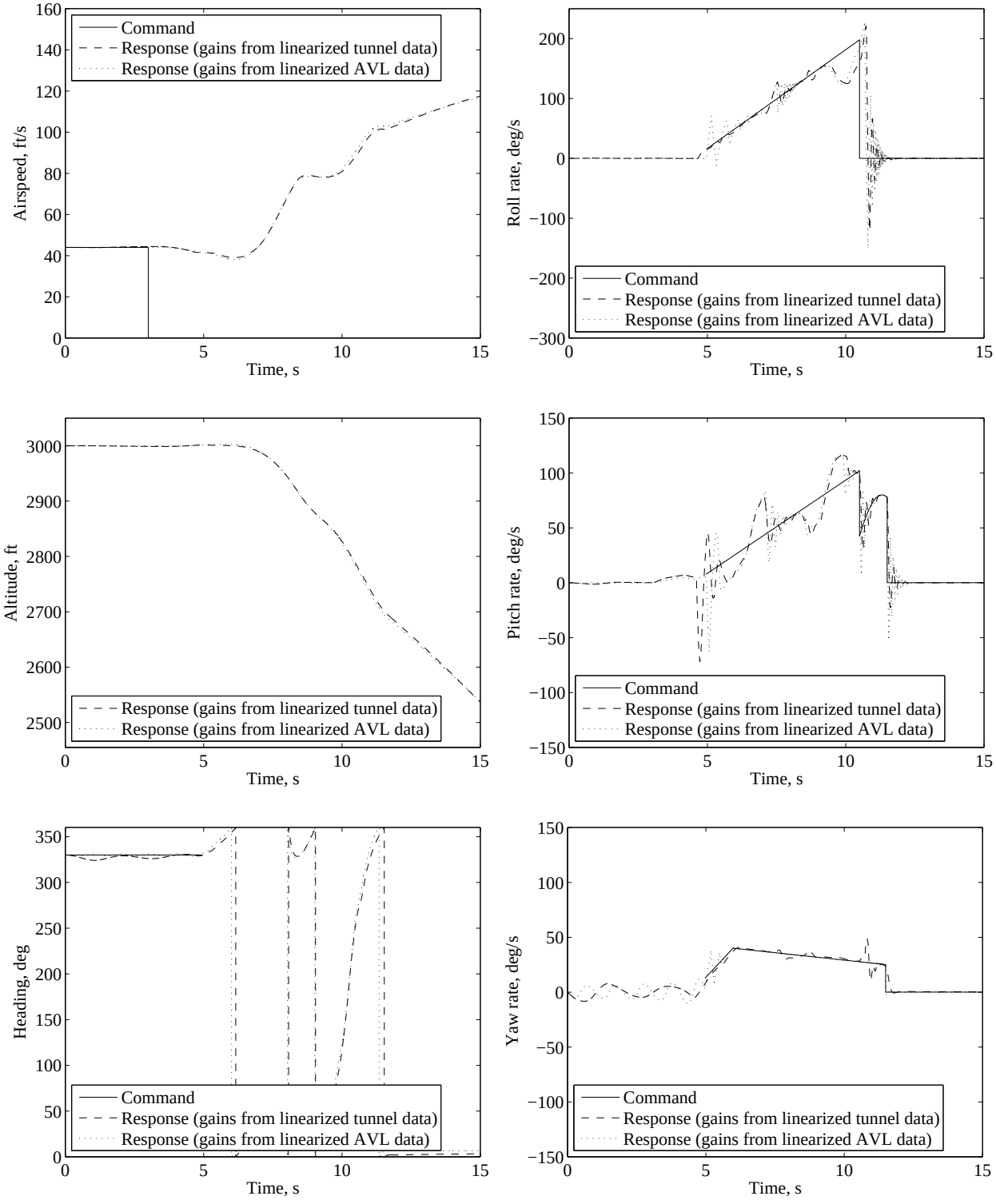
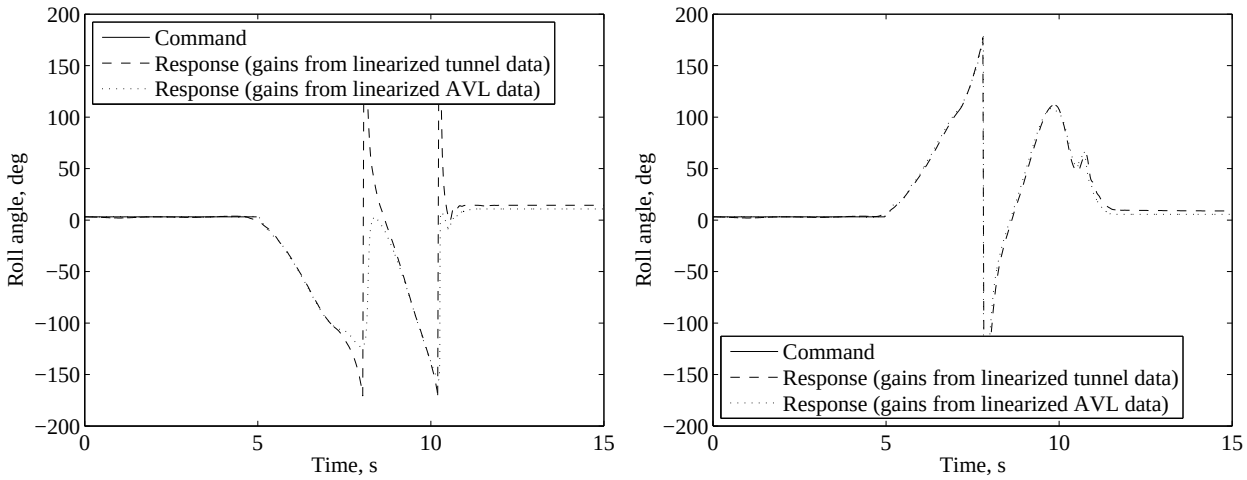


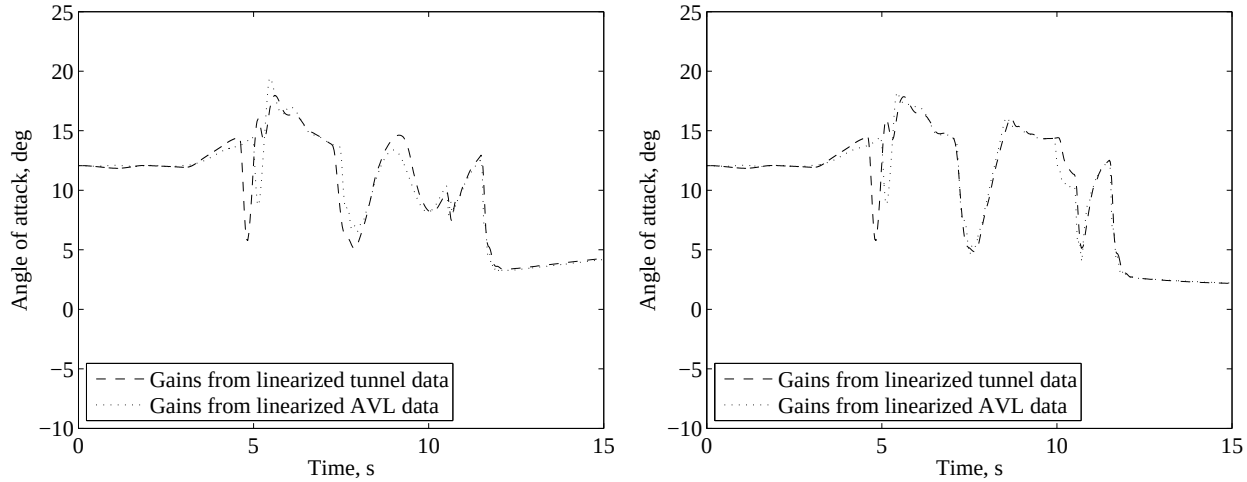
Figure 5.20: Nonlinear simulation comparison during an autonomous right spin.



(a) Left spin (b) Right spin

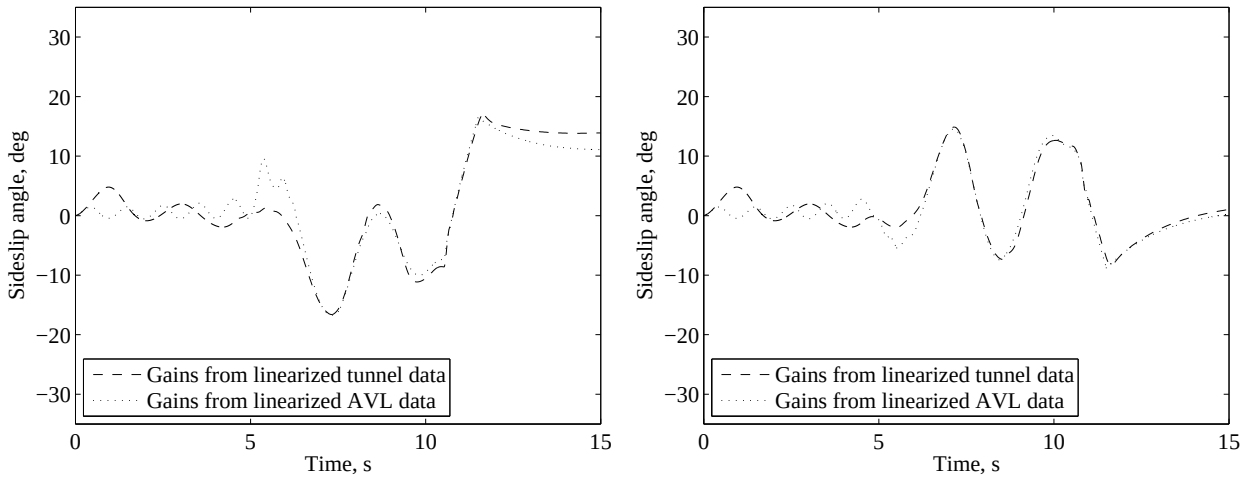
Figure 5.21: Roll angle during an autonomous spin.

The angle of attack and sideslip angle during the spin are shown in Figs. 5.22 and 5.23. For both sets of gains, the angle of attack remained above the critical angle of 14 degrees for the first spin rotation, then decreased during the second rotation and recovery. The sideslip angle exhibited the opposite trend and remained small during the first rotation, then oscillated with an amplitude of nearly 10 degrees during the second rotation.



(a) Left spin (b) Right spin

Figure 5.22: Angle of attack during an autonomous spin.



(a) Left spin

(b) Right spin

Figure 5.23: Sideslip angle during an autonomous spin.

The simulation results for the stall turn are shown in Figs. 5.24 - 5.27 beginning with a plot of the flight path. The simulation began with the aircraft in level flight (at the right side of the figure) and ended once the aircraft was brought back to level flight, heading in the opposite direction. Figure 5.25 contains successive screen captures of the visualization environment during the stall turn simulation and aids in understanding the aircraft orientation throughout the maneuver. The stall turn started with a vertical climb until the combination of engine thrust and momentum could no longer support the climb. The aircraft was then yawed 180 degrees, resulting in the aircraft descending vertically. The final part of the maneuver was a pitch up to return the aircraft to level flight at the same altitude as when the vertical climb began. The aircraft states during the simulation are shown in Figs. 5.26 and 5.27 where the airspeed, altitude, and heading plots are similar to the ones from the pilot-controlled data. At the top of the vertical climb, the speed was well below the estimated stall speed, requiring the use of throttle to provide control authority. The roll-rate command was set to zero for the entirety of the maneuver and the gains from the linearized tunnel data were able to keep the roll rate within ± 10 deg/s. Both sets of gains were able to track pitch rate relatively well, with the exception being the small oscillations that occurred

during the yawing portion of the maneuver. The same trend appears in the yaw-rate plot, where both sets of gains performed relatively well with the exception of the constant yaw rate commanded at the top of the vertical climb. Despite the oscillations in the angular rates, both sets of gains were able to perform the maneuver and return the aircraft back to level flight.

The angle of attack and sideslip plots for the stall turn are shown in Fig. 5.27. The angle of attack remained less than 10 degrees for most of the maneuver. The only spike above 10 degrees occurred just as the aircraft was entering the vertical climb. The sideslip angle remained small during the vertical climb, but rose quickly to an approximate average of 15 degrees during the yawing portion of the maneuver. During the vertical descent, which began around 11 seconds, the sideslip angle gradually decreased and reached approximately five degrees by the end of the simulation.

Figure 5.28 shows the throttle commands used to complete the stall turn. Full power was required during the vertical climb and the yawing portions of the maneuver. The prop wash helped provide additional rudder effectiveness at the top of the climb when the airspeed was minimal. During the vertical descent, the throttle was reduced to prevent excess airspeed from accumulating. Once the recovery back to level flight was initiated, the throttle was returned to the trim value for the remainder of the simulation.

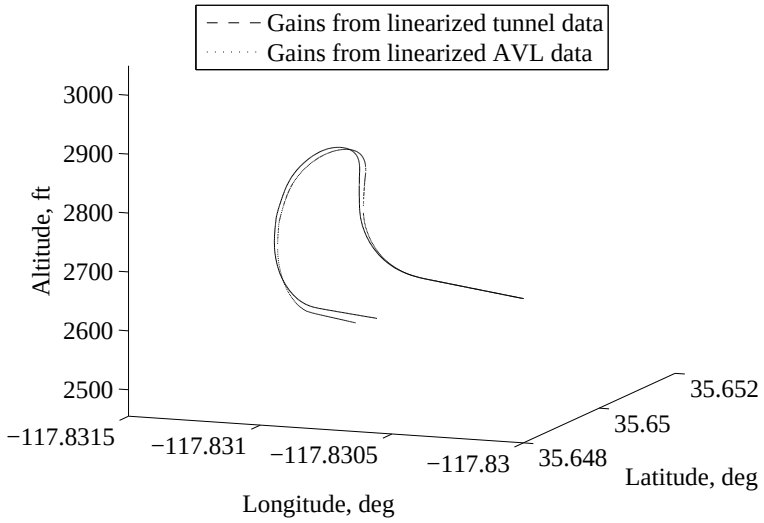
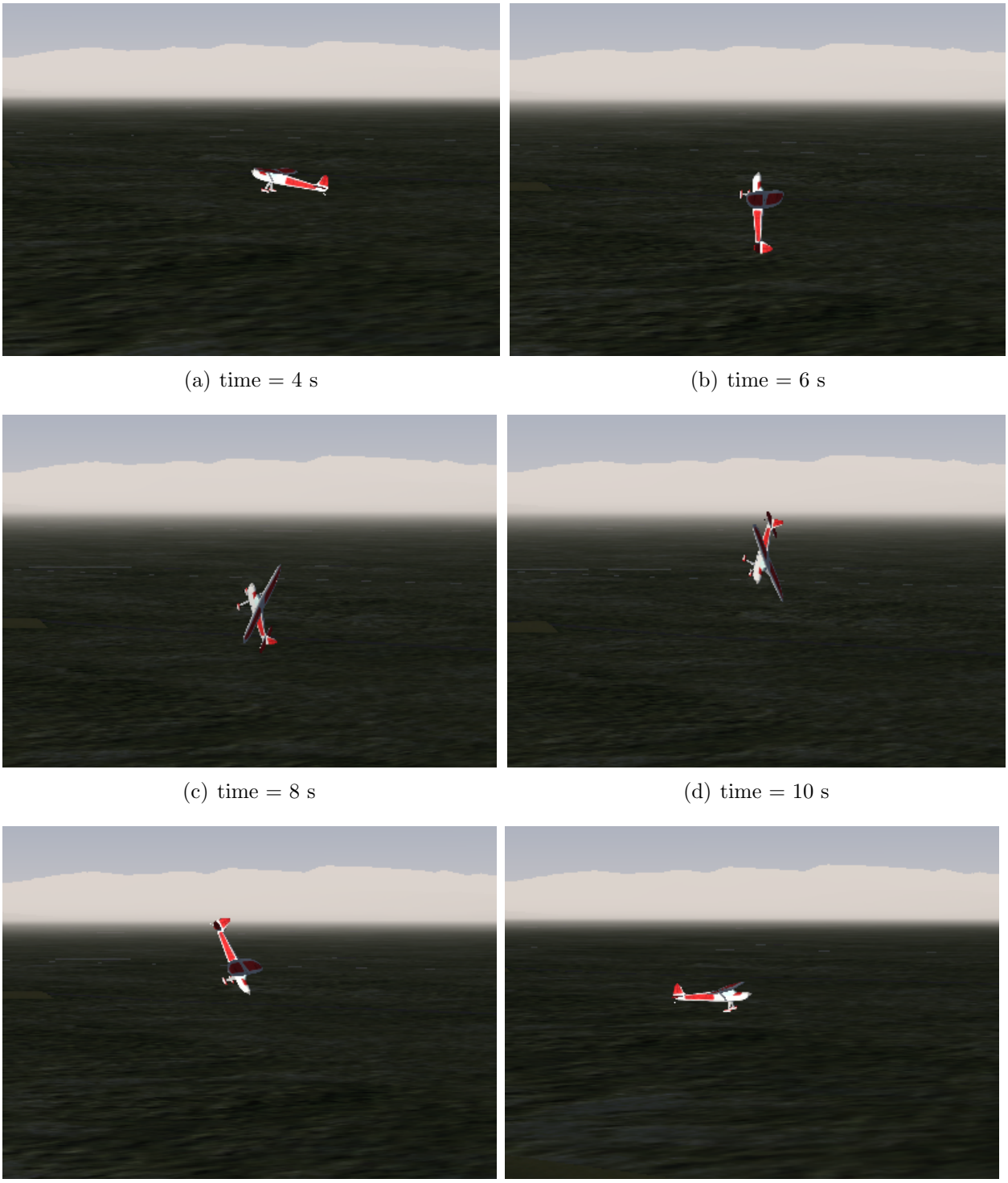


Figure 5.24: Flight path during an autonomous stall turn.



(a) time = 4 s

(b) time = 6 s

(c) time = 8 s

(d) time = 10 s

(e) time = 12 s

(f) time = 14 s

Figure 5.25: Screen capture during an autonomous stall turn.

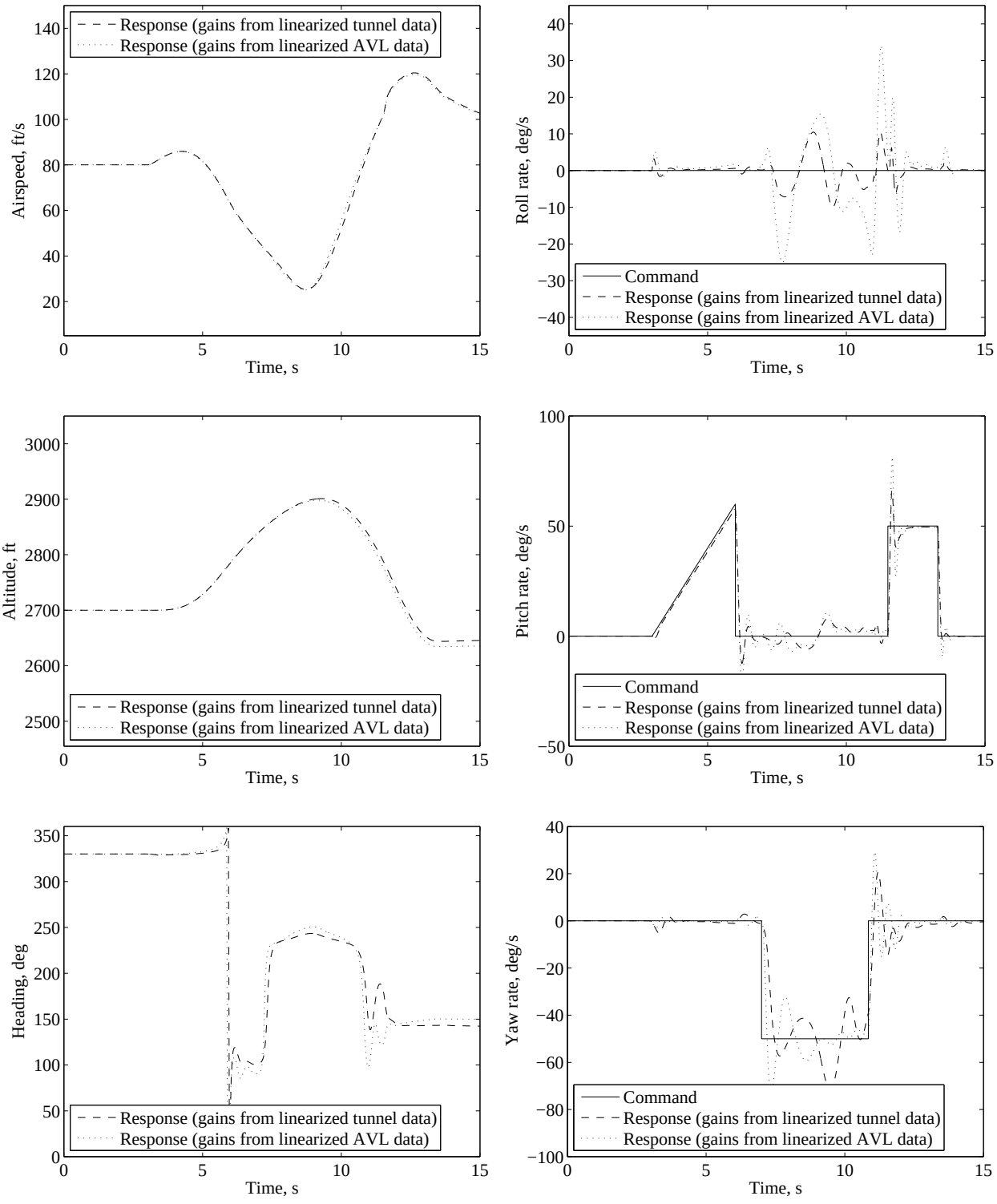


Figure 5.26: Nonlinear simulation comparison during an autonomous stall turn.

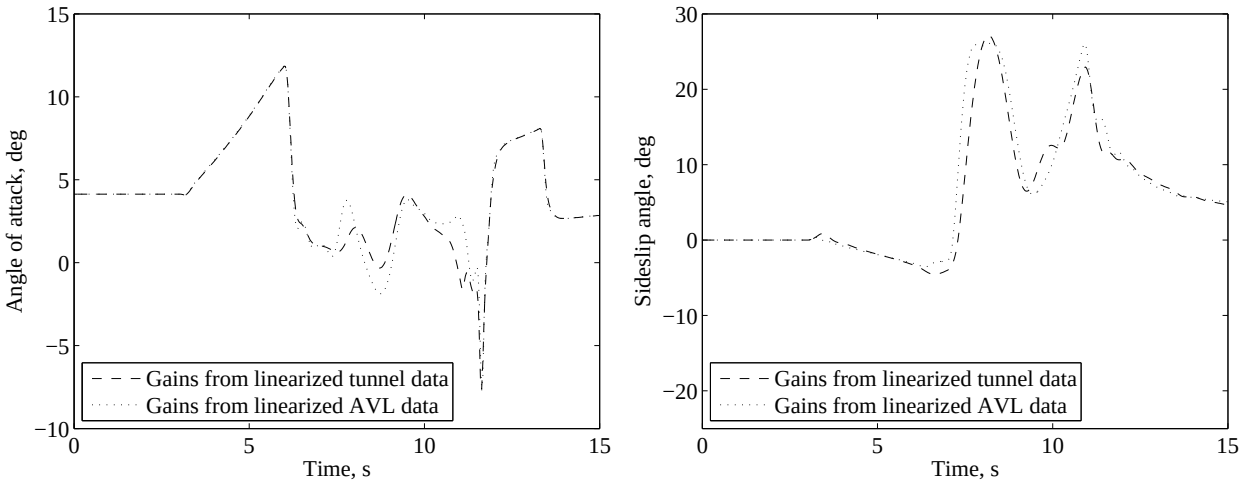


Figure 5.27: Angle of attack and sideslip angle during an autonomous stall turn.

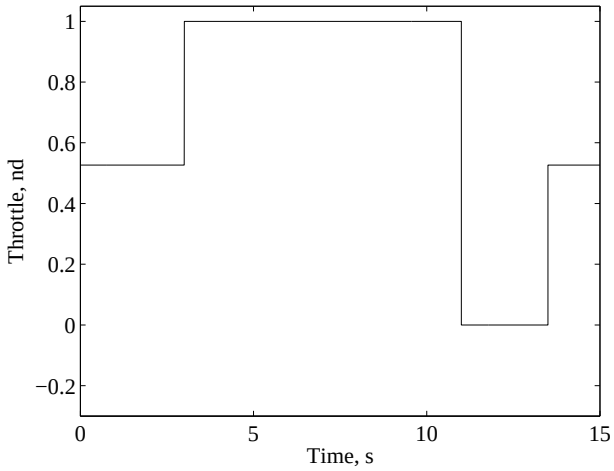


Figure 5.28: Throttle position during an autonomous stall turn.

The results for the simulation of the tail slide are shown in Figs. 5.29 - 5.33 with the first figure showing the flight path during the maneuver. Starting from level flight at the right side of the plot, the tail slide consists of a vertical climb, followed by a vertical descent and a return to level flight. The successive screen captures during the simulation, shown in Fig. 5.30, help to visualize the aircraft orientation throughout the maneuver. At the top of the vertical climb, the pitch rate was held at zero as the airspeed decreased. Once the elevator could no longer track the zero pitch-rate command, the aircraft was commanded to pitch nose down to begin the descent. The tail slide is characterized by the motion at the top of the vertical climb, which for most aircraft results in a tail-first descent for a few feet before nosing over.

The plots of airspeed, altitude, and heading in Fig. 5.31 show that at the top of the climb, the speed decreased to less than 10 ft/s at a maximum altitude of just over 2,900 feet. During the vertical climb, the heading plot depicts a change in heading due to the Euler angle ψ being undefined when the pitch angle is 90 degrees. The plots of the angular rates show the ability of each set of control gains to track the reference commands. The roll and yaw rates were commanded to zero for the entirety of the maneuver and for the initial pitch up and vertical climb, both gain sets were able to track the zero command. Once the airspeed decreased at the top of the vertical climb, however, oscillations of ± 20 deg/s occurred until the airspeed recovered. The gains from the linearized tunnel data performed the best, as expected, and were able to keep the roll rate oscillations to within ± 15 deg/s throughout the maneuver. The pitch rate plot shows the three distinct commands required to perform the tail slide starting with the pitch up to vertical, which both sets of gains were able to track. The pitch down at the top of the climb as well as the pitch up to return to level flight were both accomplished with step inputs. The response from both sets of gains contained overshoot when the airspeed was lowest at the top of the climb. As the airspeed started to rise during the vertical descent, however, both gain sets were able to track the step

inputs with greater accuracy. This difference in performance was expected and is caused by the large deviation from the trim condition experienced during the maneuver.

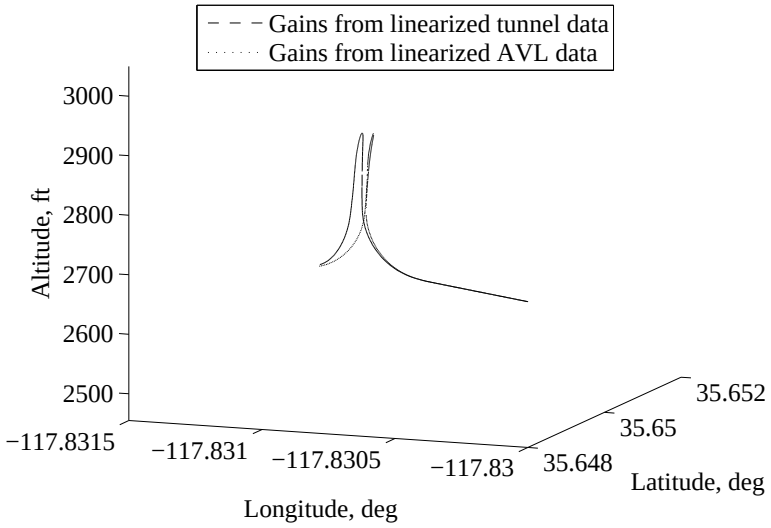
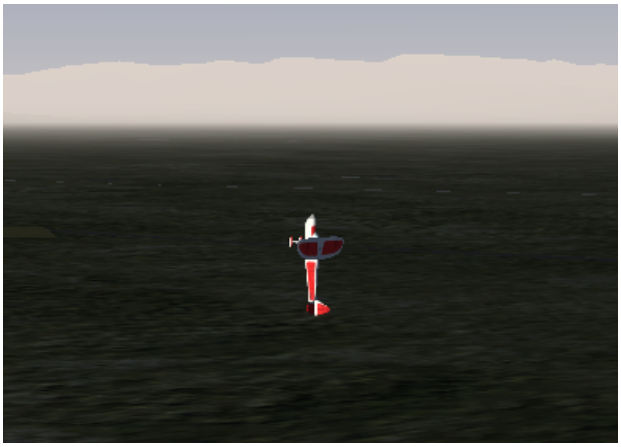


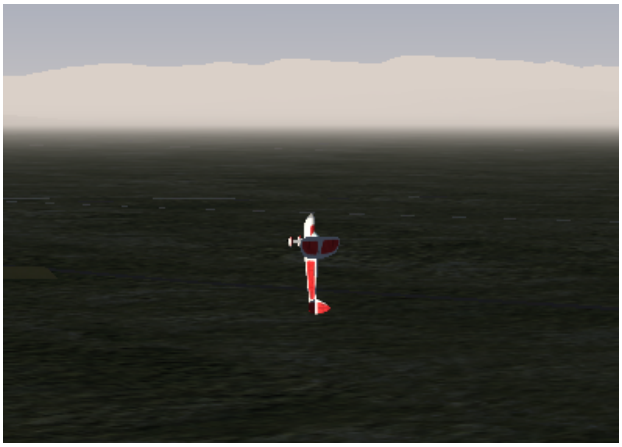
Figure 5.29: Flight path during an autonomous tail slide.

Plots of angle of attack, sideslip angle, and throttle position during the simulation are given in Figs. 5.32 and 5.33. The angle of attack increase associated with the transition to vertical flight is apparent and is similar to the result from the stall turn simulation. The sideslip angle remained relatively small during the pitch up and vertical climb, but oscillated during the vertical descent. Both angles appear to oscillate during the pitch down, at the top of the vertical climb, due to the immediate reversal in the velocity vector. The throttle position plot shows how full power was used for the vertical climb and to provide elevator effectiveness as the airspeed decreased. During the descent, just as for the stall turn, the power was reduced to prevent excessive airspeeds.

(a) time = 4 s

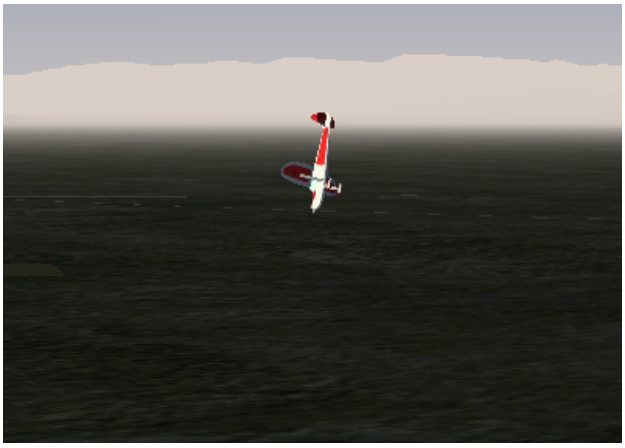


(b) time = 6 s



(c) time = 9 s

(d) time = 10 s



(e) time = 12 s

(f) time = 15 s

Figure 5.30: Screen capture during an autonomous tail slide.

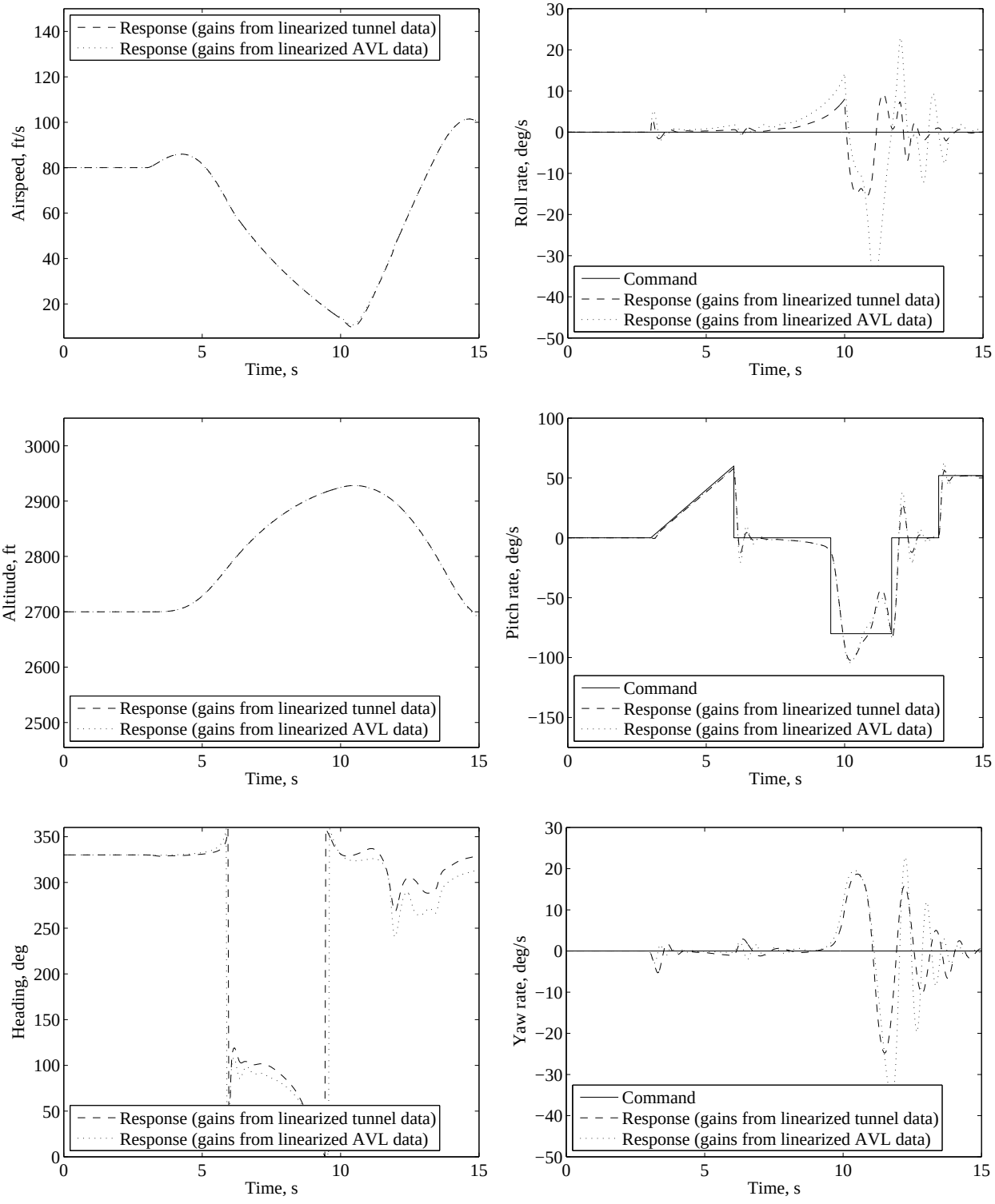


Figure 5.31: Nonlinear simulation comparison during an autonomous tail slide.

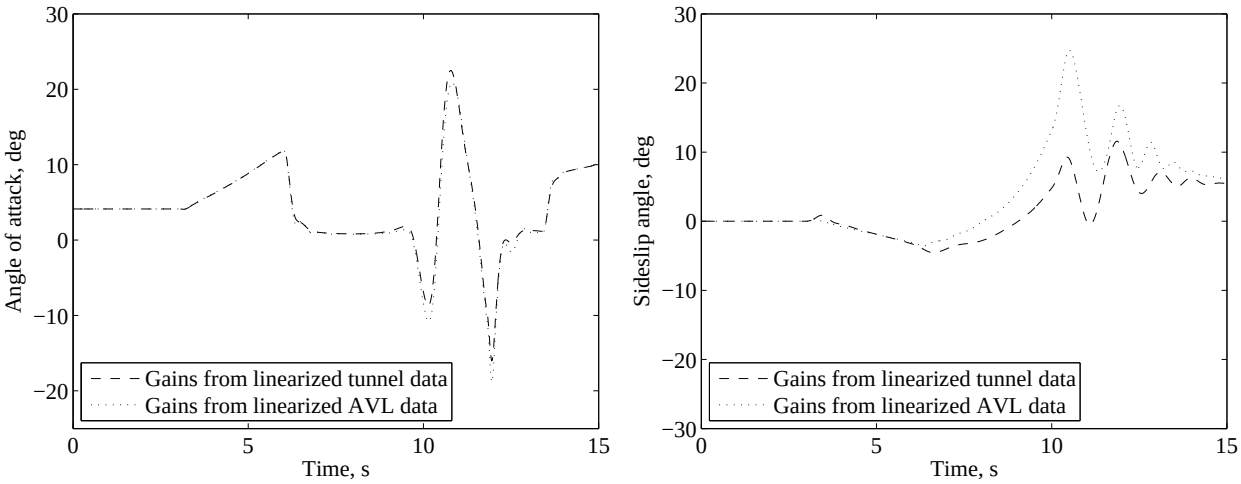


Figure 5.32: Angle of attack and sideslip angle during an autonomous tail slide.

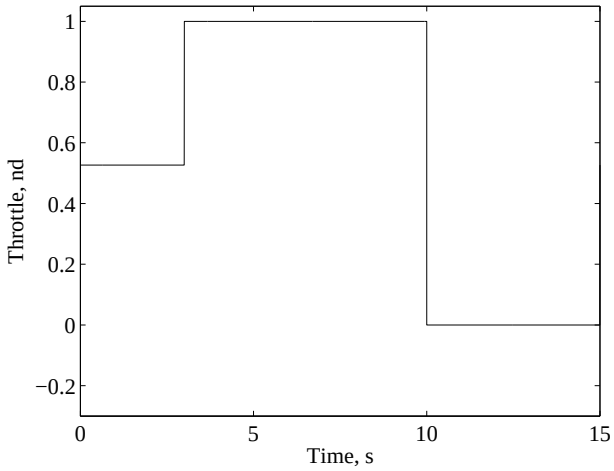


Figure 5.33: Throttle position during an autonomous tail slide.

5.3 Maneuvering and Returning to Cruise

The simulation results in this section combine the methods of the previous two sections to allow transitions from maneuvering flight to cruise flight. The first set of results in this section are for simulations in which the aircraft returned to straight-and-level flight once the maneuver was completed. In addition to the transition, actuator models were also included to test the functionality of the autopilot when the control commands have rate and deflection limits. The control gains for the simulations that include actuator models were recalculated based on the linear aircraft models that contain linear models for the actuators. The second set of results in this section are for simulations in which zero-mean, Gaussian pseudo-random noise was added to the feedback states based on published data for actual sensors. Measurement noise was included to determine the effects of having imperfect state estimates in the feedback path. For the simulations including measurement noise, the same control gains determined using the linear aircraft models with actuator models were used.

An important consideration when reviewing the angular rate plots of this section is that during the transition to level flight, the angular rates are not being controlled directly. What is controlled during the transition is the airspeed, altitude, and heading and the plots of those states show the commands used to return the aircraft back to cruise flight. This methodology results in large angular rate oscillations during recovery from some of the maneuvers. These oscillations, however, are not indicative of the performance of the controller during the transition. The angular rate plots should only be used to determine the performance of the maneuvering controller (during the maneuver) and to get an idea of the aggressiveness used by the transition controller to recover back to level flight.

5.3.1 Including Actuator Models

The first results are for a simulation of the loop with a transition back to cruise flight at the end of the maneuver. Actuator models were included to simulate the time lag, deflection limits, and rate limits associated with electric servo motors. Figure 5.34 shows the plots

of airspeed, altitude, heading, and the angular rates during the loop. With the actuator models included, the gains calculated using the linearized tunnel data resulted in small oscillations around the zero roll- and yaw-rate commands throughout the maneuver. The gains calculated using AVL data seemed to perform slightly better, in terms of tracking roll and yaw rate during the maneuver, by avoiding some of the high frequency oscillations experienced with the first set of gains. In terms of tracking pitch rate, however, the gains from the linearized tunnel data were able to exactly track the reference command, while the gains from AVL data experienced overshoot after each change in command. The plots of airspeed, altitude, and heading appear similar to the plots for the loop from the previous section during the maneuver. For the transition from maneuvering to level flight, both sets of gains provided the ability to acquire and track the reference commands quite well. During the descent portion of the loop the airspeed climbed to 130 ft/s, so the transition back to the level-flight command of 80 ft/s required around four seconds for both sets of gains. Despite the coupled motion of the airspeed and altitude during the transition, both sets of gains were able to return to the commanded altitude in approximately five seconds. The heading response during the transition was the quickest and only required roughly two seconds to reach steady state.

The control deflections during the loop are shown in Fig. 5.35 and were similar for both sets of gains. During the maneuver, the control inputs of elevator, aileron, and rudder were gradual while the throttle was held at full power. During the transition to level flight, the elevator and aileron inputs oscillated for the first few seconds before stabilizing near the trim settings. Due to the altitude difference at the start of the transition, the throttle response rose to full power for a short time before going to zero. As the altitude command was reached, the throttle returned to the trim setting for the final seconds of the simulation.

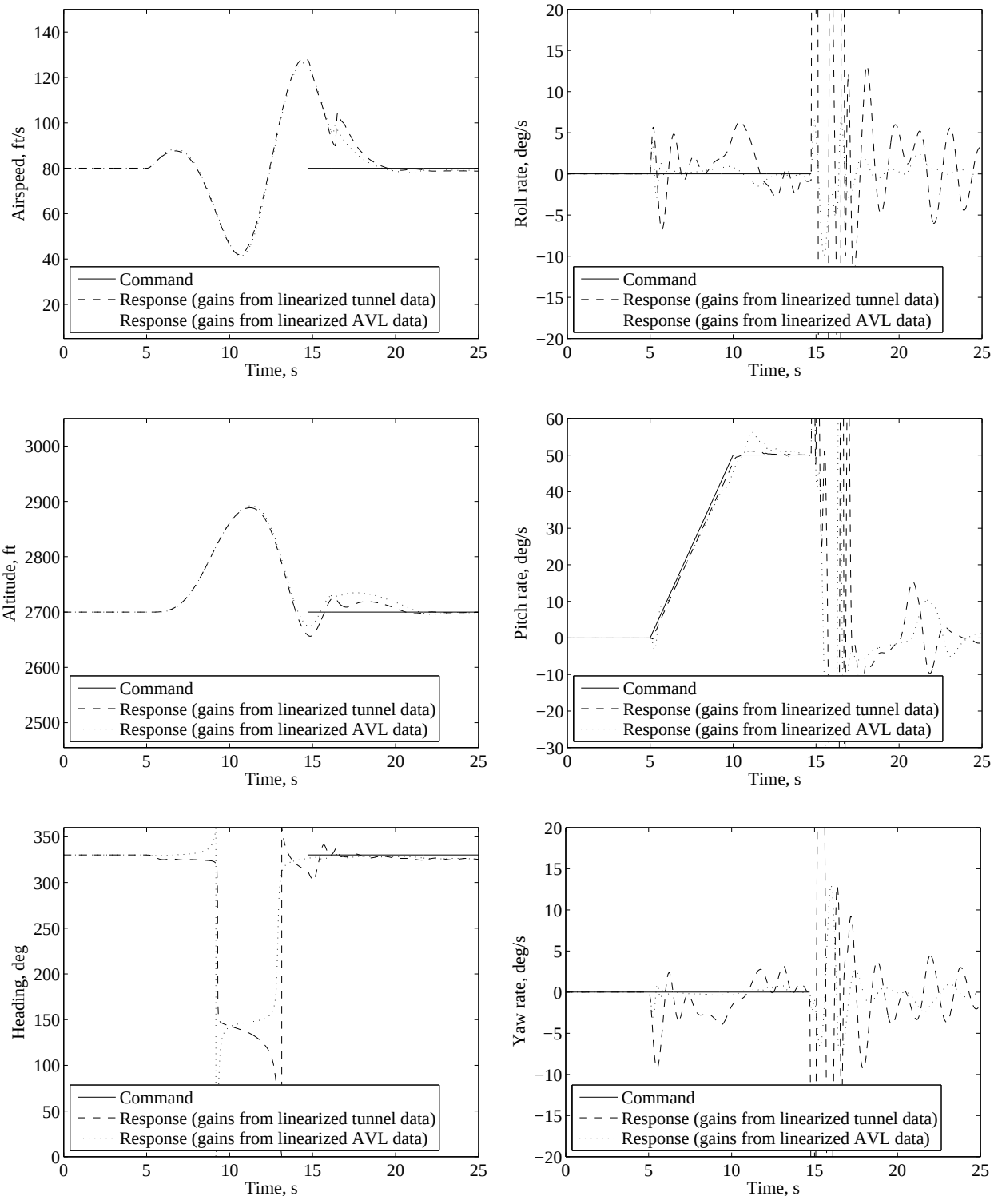


Figure 5.34: Nonlinear simulation comparison during an autonomous loop with actuator models.

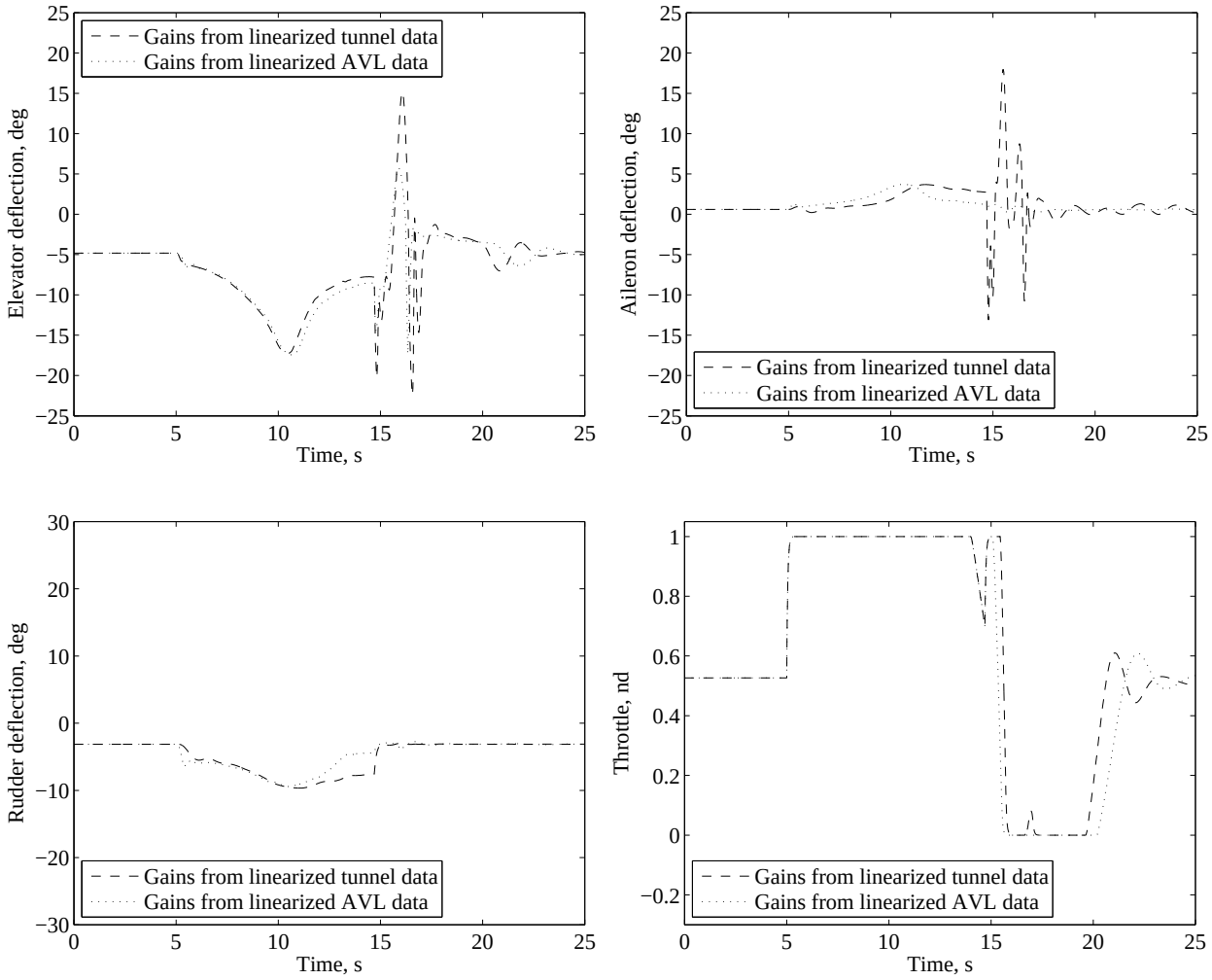


Figure 5.35: Control deflections during an autonomous loop with actuator models.

The results for the roll maneuver are discussed next with Figs. 5.36 and 5.37 showing the aircraft states for both the left and right roll directions respectively. Both sets of gains were able to track the roll-rate command, during the maneuvering portion of the simulation, for both roll directions quite well. The gains from the linearized tunnel data were able to track the pitch-rate command during both rolls, but the gains from the AVL data resulted in oscillations of ± 5 deg/s. The first oscillations occurred when the throttle was increased and they happened again after the initial pitch up to begin the roll. Both sets of gains struggled to track the sinusoidal yaw-rate command, which is the same result discovered in the previous section. The largest difference in yaw rate occurred while using the gains from the linearized tunnel data to perform a right roll. Approximately seven seconds into the simulation, the yaw rate reached 20 ft/s while the command was near zero.

The transition to level flight began after nine seconds and, for the gains from the tunnel data, required about five seconds to return to the commanded cruise speed and nearly 10 seconds to return to the commanded cruise altitude. The time to return to cruising altitude was longer when the gains from AVL data were used, due to the higher altitude deviation that occurred during the transition. To explain the large deviations in altitude during the transition, we first examine the speed when the transition began. As the transition began, the airspeed had reached nearly 150 ft/s. To decrease speed and return to the commanded cruise speed, the control system commanded the aircraft to pitch up. This pitch-up command is evident in the elevator deflection plots of Figs. 5.38 and 5.39 by the spikes to -20 degrees that occurred around the nine second mark. The result of the aircraft pitching up was twofold; the speed decreased due to the rise in induced drag, and the altitude increased due to the rise in lift. To reduce the altitude, the control system commanded the throttle to zero which is shown in the throttle position plots of Figs. 5.38 and 5.39. The drag on the aircraft then determined the time to return to the commanded altitude. Once the airspeed and altitude both stabilized near their commanded values, the elevator and throttle inputs settled near their trim values. The heading remained relatively constant throughout

the maneuver, so the aileron and rudder inputs settled to their respective trim values shortly after the transition began.

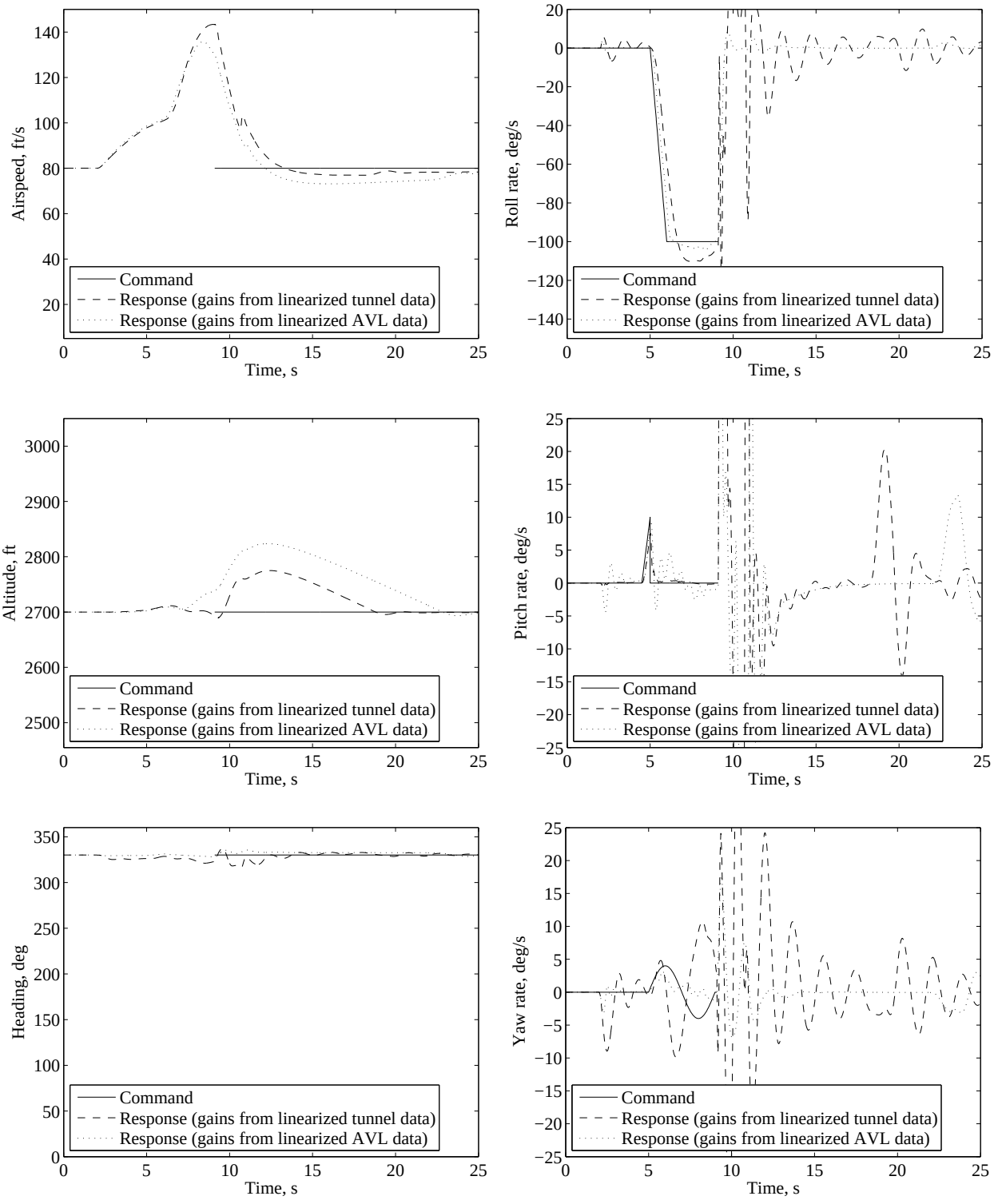


Figure 5.36: Nonlinear simulation comparison during an autonomous left roll with actuator models.

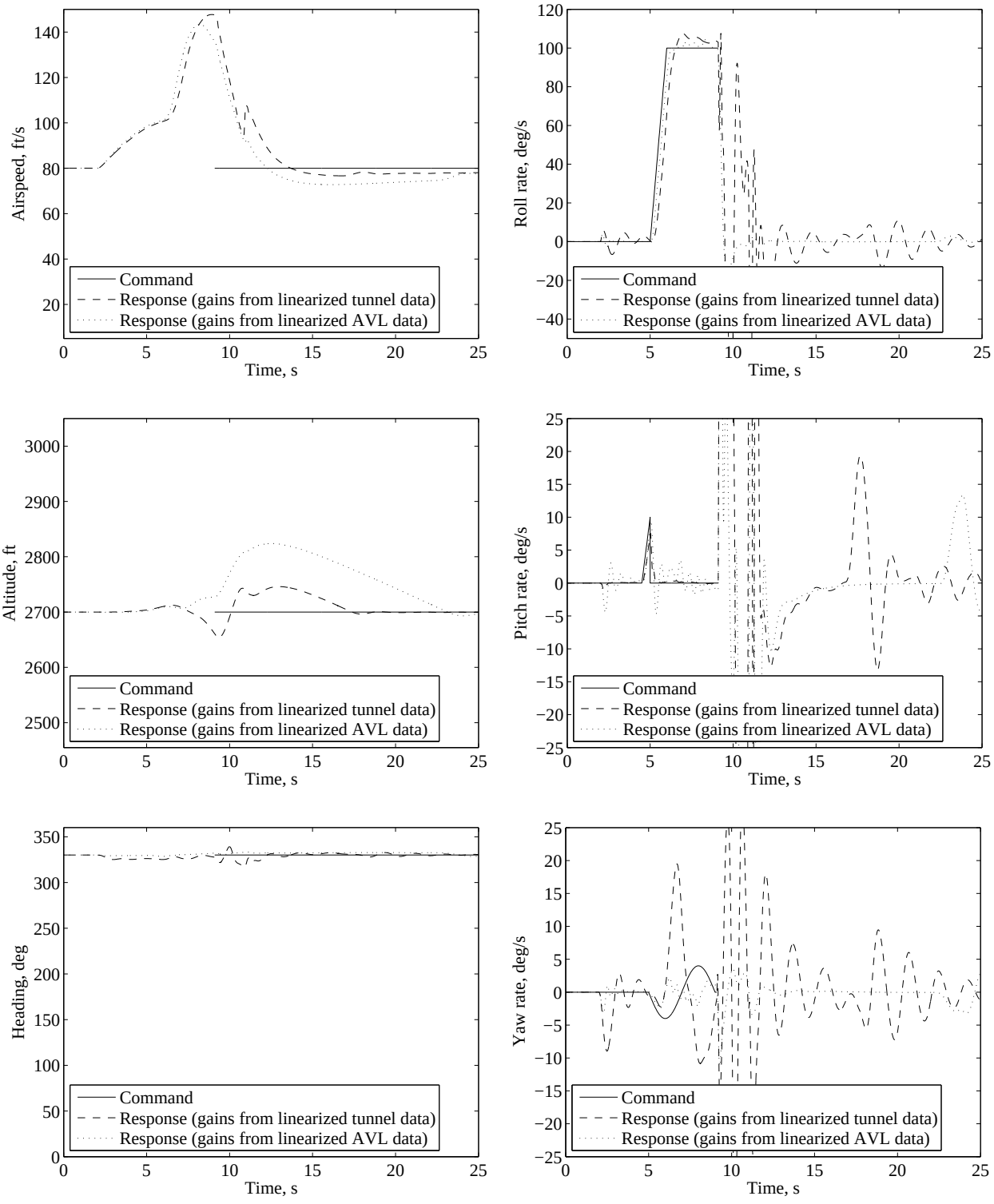


Figure 5.37: Nonlinear simulation comparison during an autonomous right roll with actuator models.

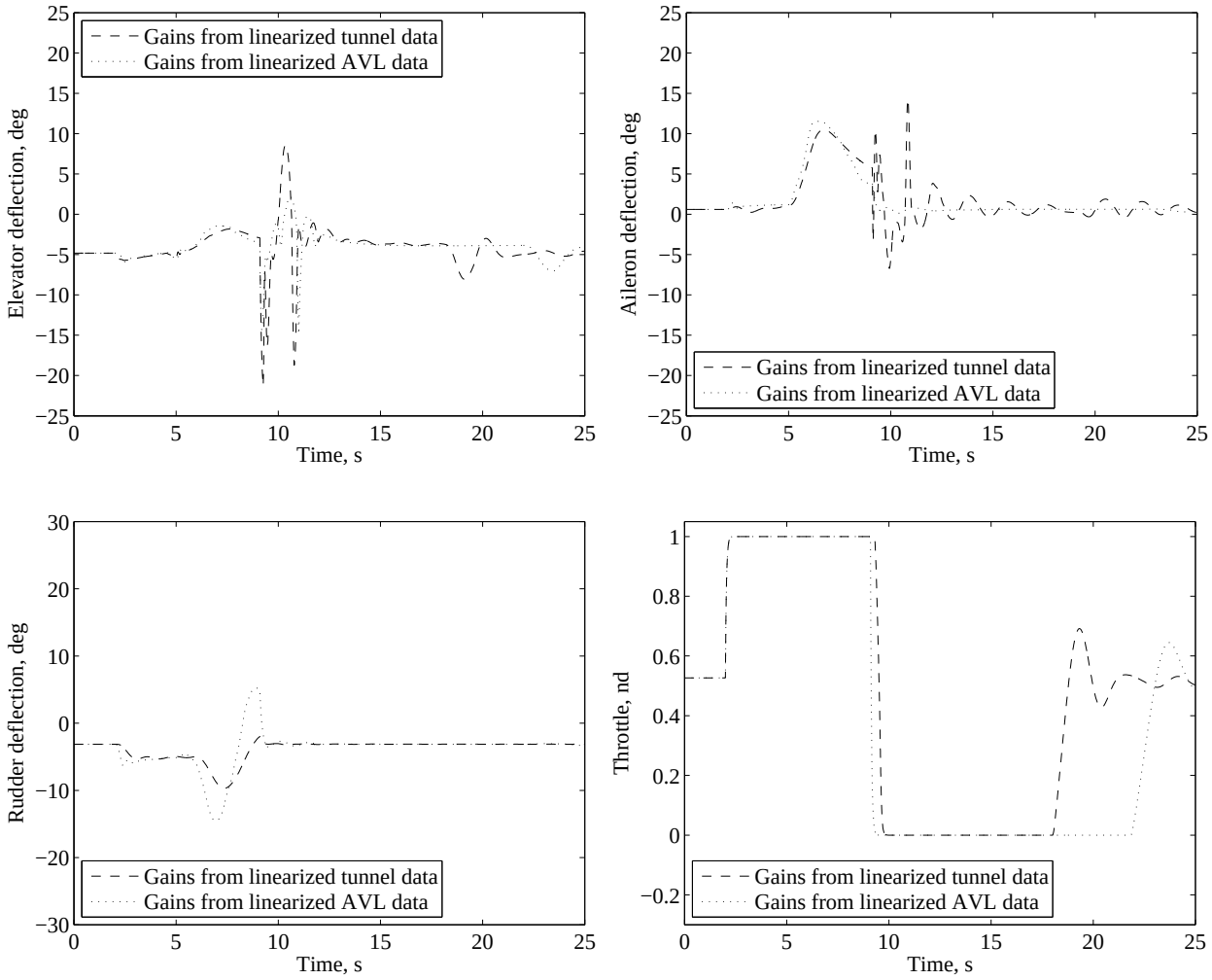


Figure 5.38: Control deflections during an autonomous left roll with actuator models.

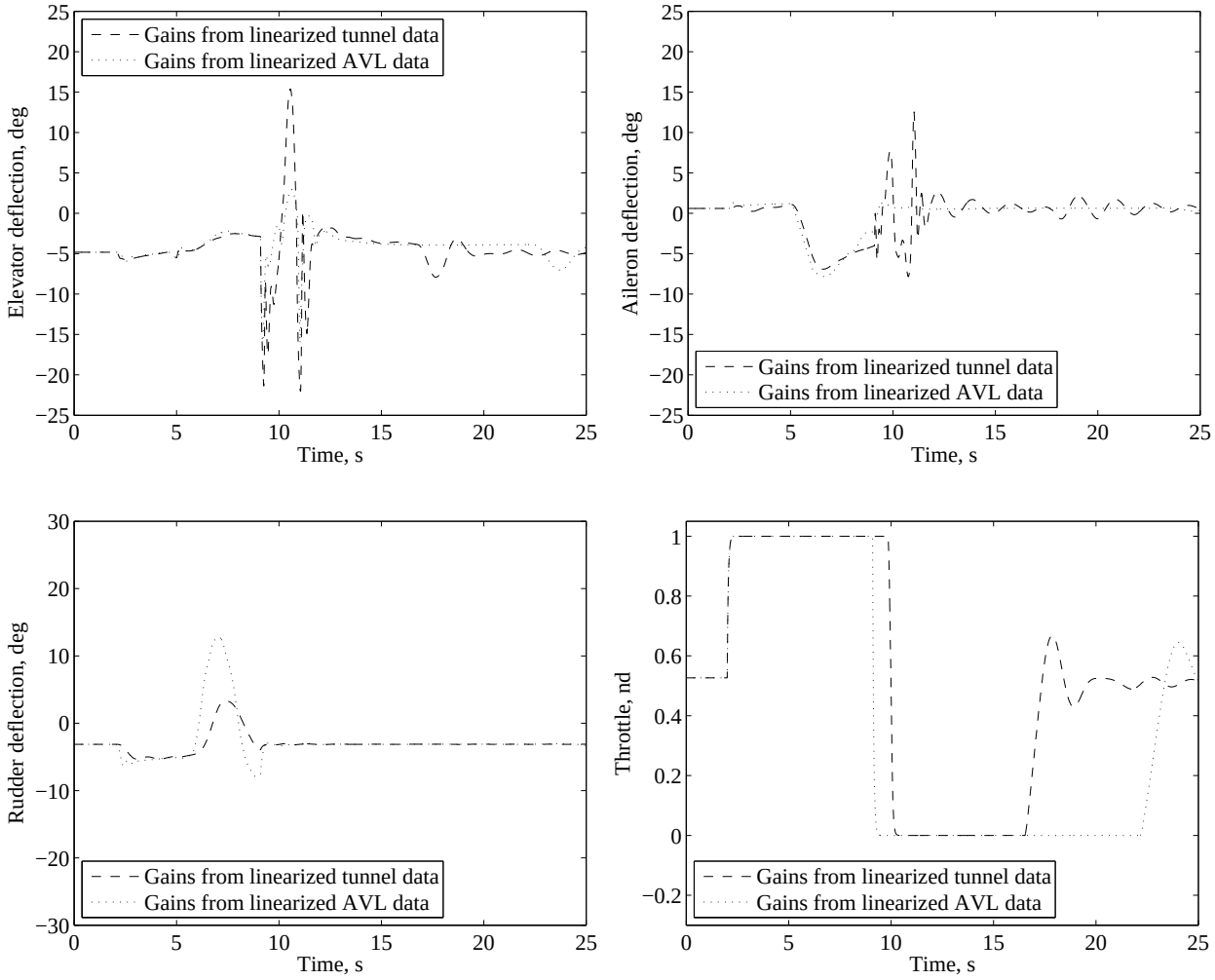
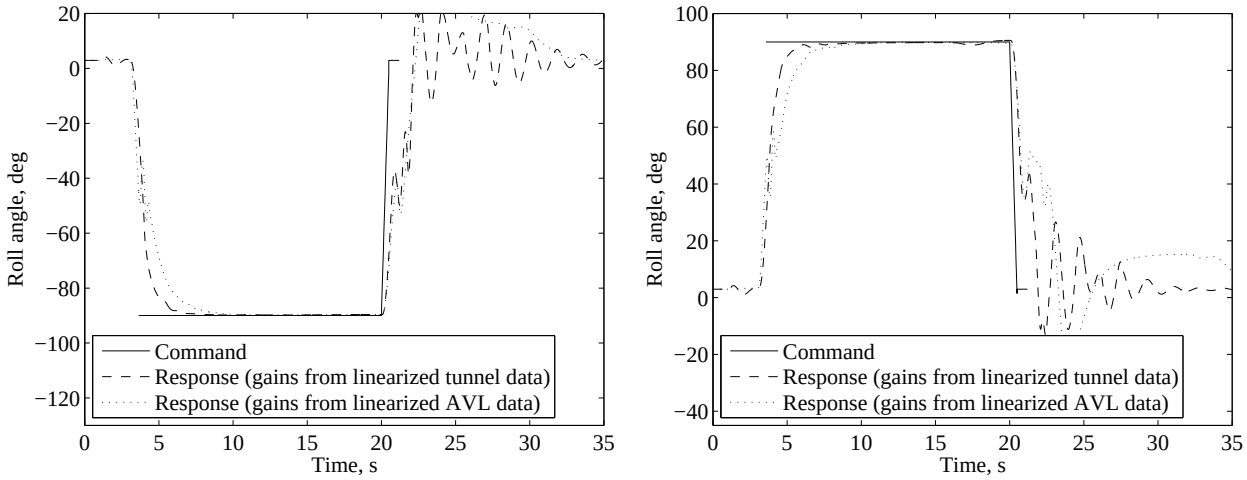


Figure 5.39: Control deflections during an autonomous right roll with actuator models.

Figures 5.40 - 5.44 show the results for knife-edge flight. To enter the knife-edge attitude, the angular rate controller was used and both sets of gains were able to track the rate commands even with the actuator models included. As the bank angle increased and the roll-angle controller took over, the gains from AVL data actually performed better with the actuator models included than they did in the previous section without the models. Both sets of gains were able to track the 90 degree roll-angle command as well as the heading command with great precision throughout the maneuver. The altitude deviation during knife-edge flight increased for both sets of gains when the actuator models were included in the simulation. The gains from AVL data suffered the largest altitude loss during right

knife-edge flight, while the gains from the tunnel data suffered their largest deviation during left knife-edge flight.

The transition from knife-edge flight to level flight began after 21 seconds and took approximately five seconds using the gains from the tunnel data. The gains from the AVL data required slightly longer due to the larger deviation in altitude during the maneuver. Figures 5.43 and 5.44 show that for the knife-edge portion of the maneuver, the rudder actuator became saturated while attempting to maintain altitude. As the transition to level flight began, the rudder deflection quickly returned to its trim value and full power was used to return to the commanded altitude. The elevator and aileron actuators were controlled more vigorously during the transition by the gains from the tunnel data, despite having smaller airspeed and heading errors when compared to the simulation using gains from AVL data.



(a) Left knife edge

(b) Right knife edge

Figure 5.40: Roll angle during autonomous knife-edge flight with actuator models.

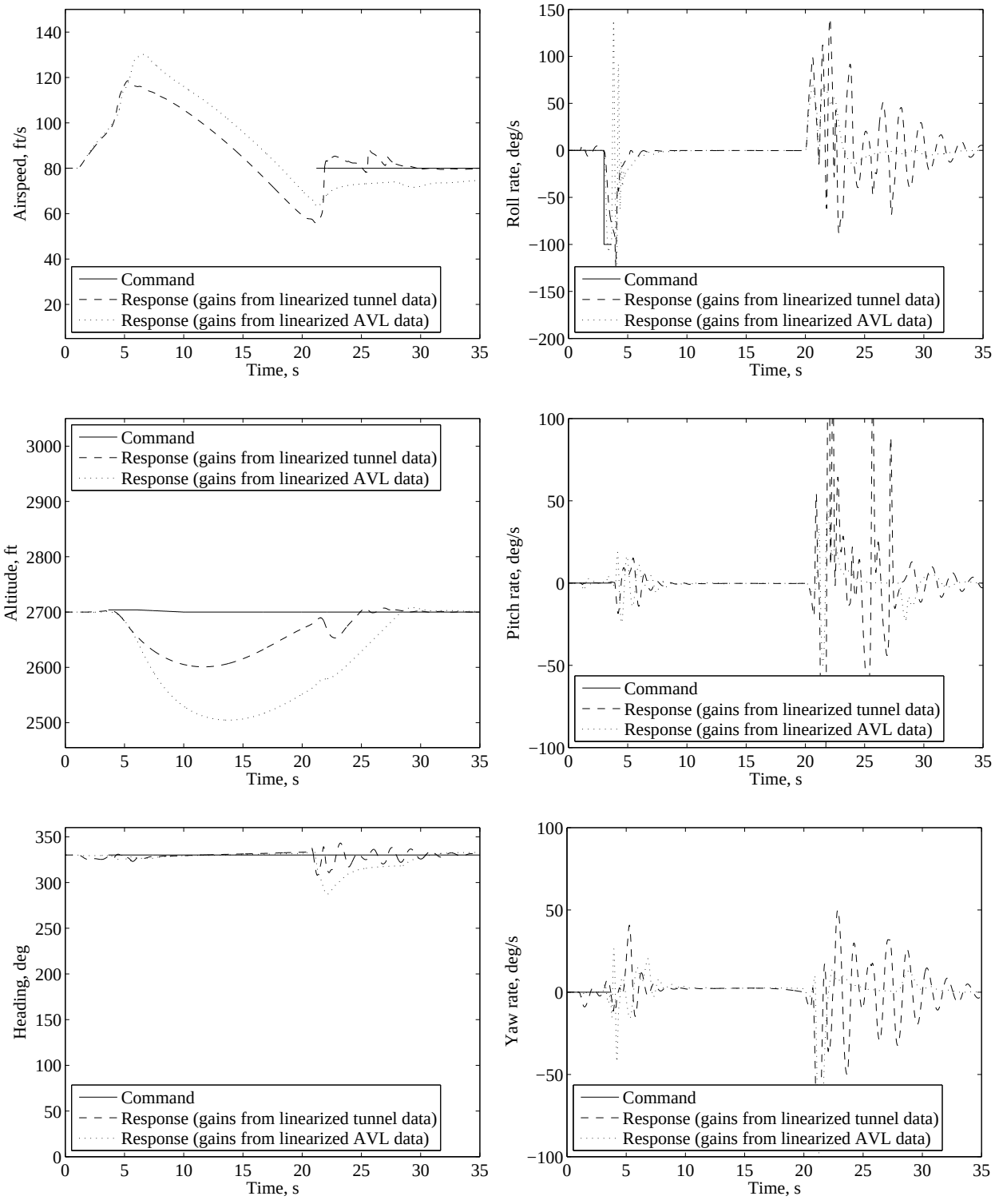


Figure 5.41: Nonlinear simulation comparison during autonomous left knife-edge flight with actuator models.

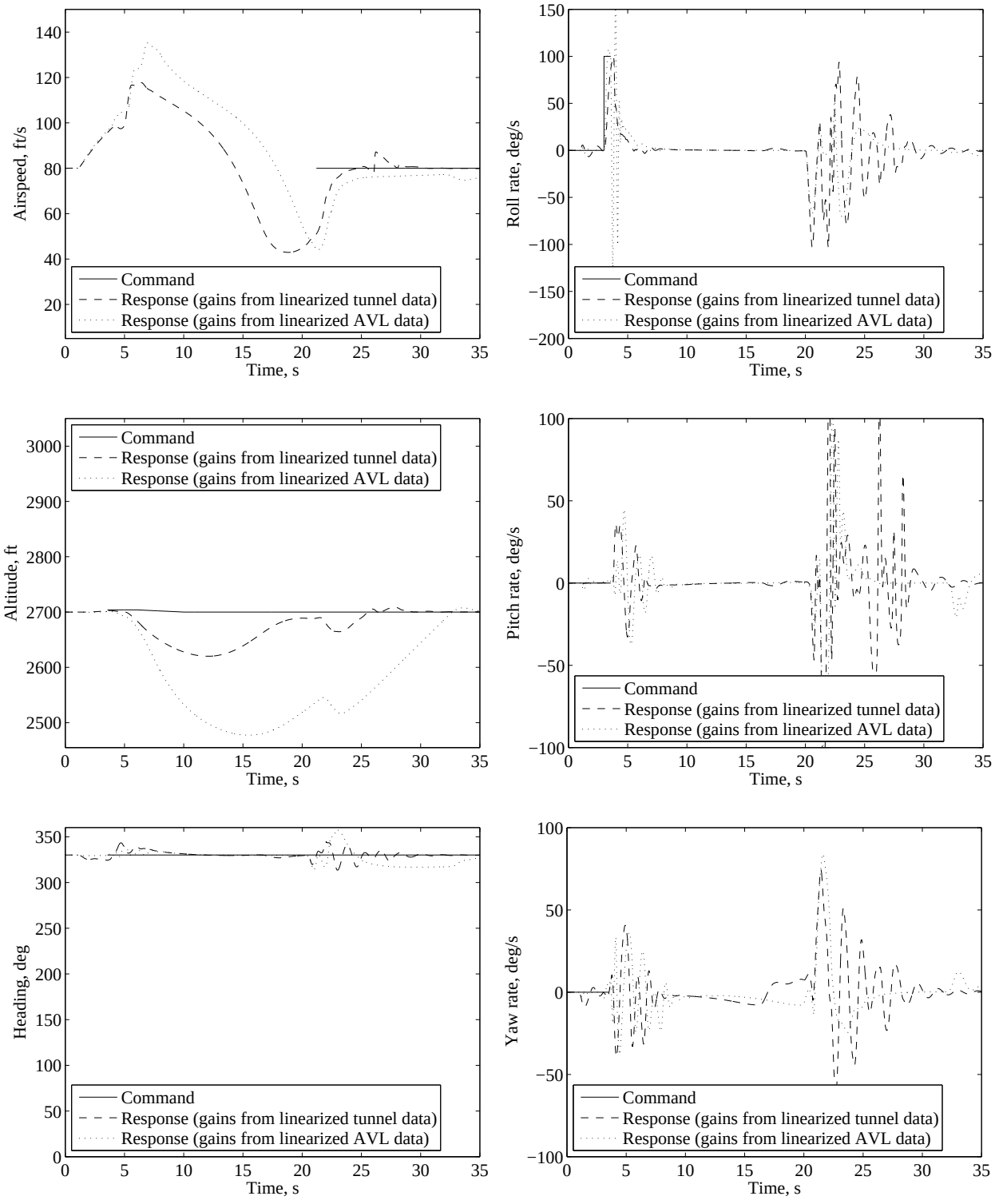


Figure 5.42: Nonlinear simulation comparison during autonomous right knife-edge flight with actuator models.

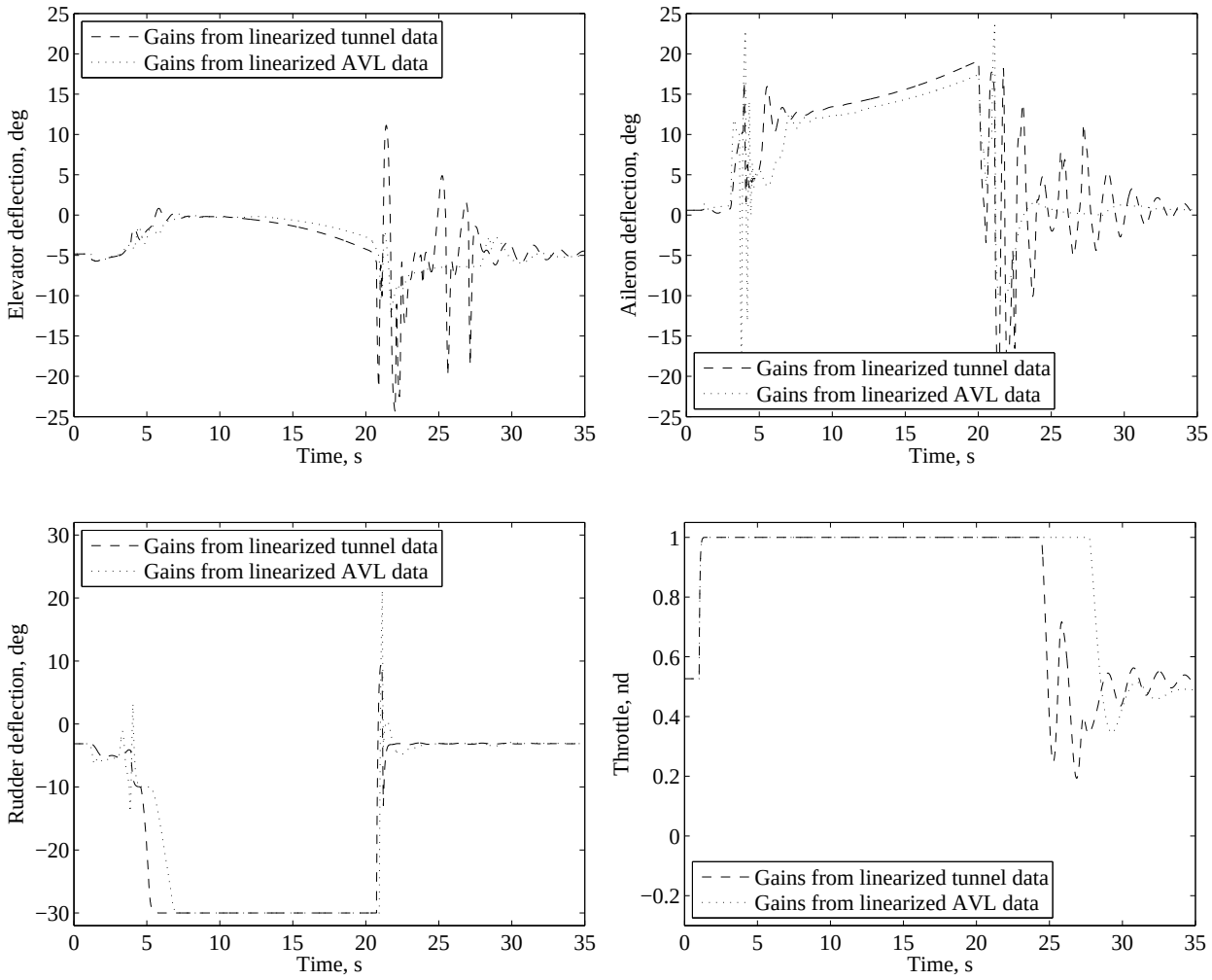


Figure 5.43: Control deflections during autonomous left knife-edge flight with actuator models.

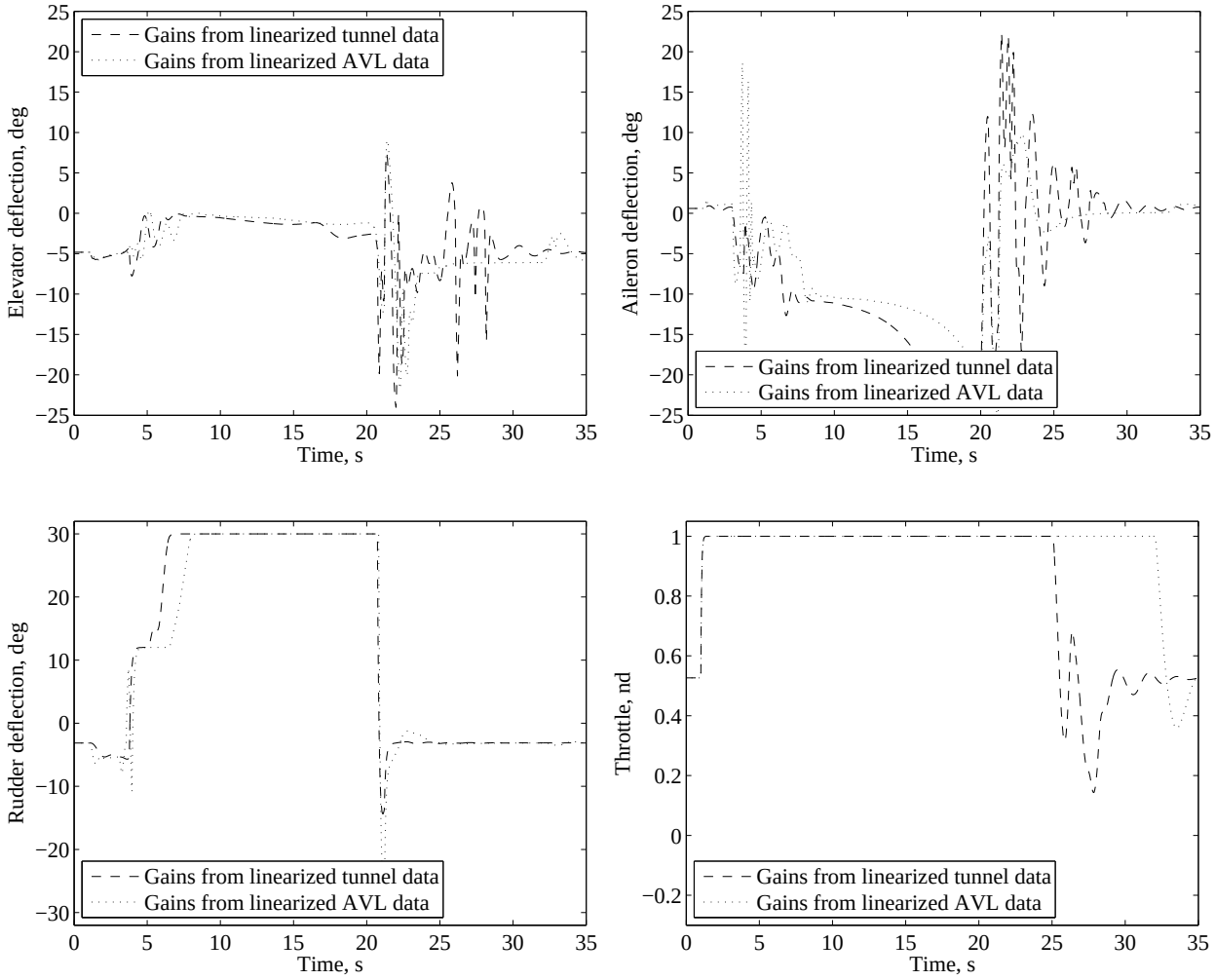
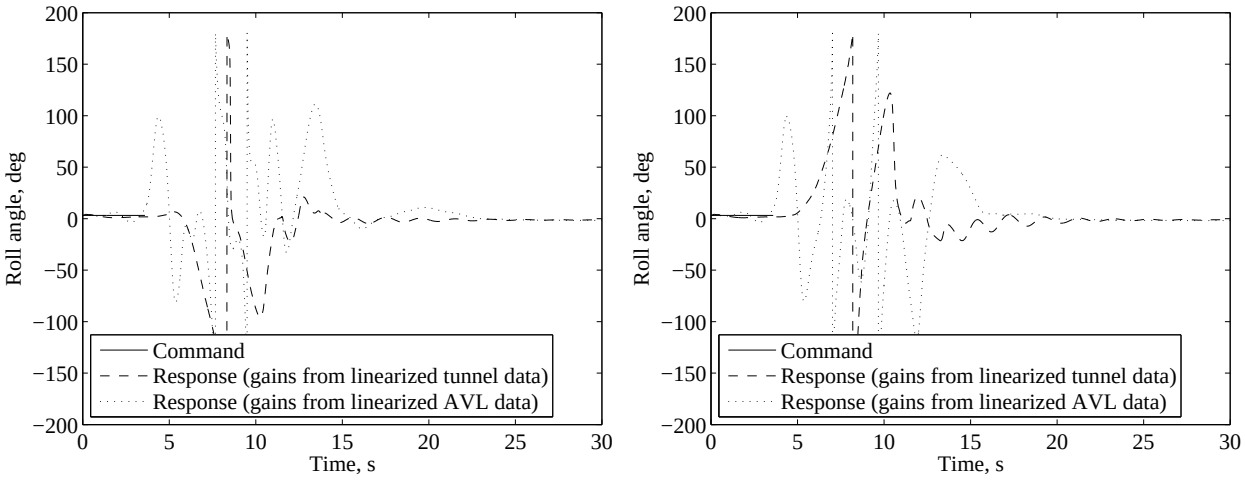


Figure 5.44: Control deflections during autonomous right knife-edge flight with actuator models.

The remainder of the simulation results in this section, with actuator models and transitions from maneuvering to level flight included, are for the post-stall maneuvers and Figs. 5.45 - 5.49 show the results for the spin. The results for the roll-angle and airspeed control loops during the initial stalling portion of the maneuver are similar to the spin results from the previous section. The results for the heading loop indicate that when the actuator models were included, the gains from the tunnel data were not able to track the reference command and the gains from the AVL data were not able to stabilize the closed-loop system. When using the gains from AVL data, the heading oscillations appeared to grow in amplitude during

the stalling portion of the maneuver and resulted in yaw rate oscillations up to 100 deg/s. This heading loop result appeared to be the same for both left and right spins, which makes intuitive sense based on the stalling portion of the maneuver being the same for both spin directions. The gains from the tunnel data were able to track the roll- and pitch-rate commands during the spinning portion of both maneuvers, but were unable to track the yaw-rate commands beyond the initial ramp up. The gains from the AVL data were not able to control the spinning motion as evident by the large oscillations in the angular rate plots. The oscillations in the aileron and rudder deflection plots of Figs. 5.48 and 5.49 correspond to the oscillations in the angular rates and confirm that the gains from AVL data were not able to control the spin. Despite the results for the spin, both sets of gains were able to control the transition from maneuvering to level flight and recovery took roughly five seconds for both gain sets.



(a) Left spin

(b) Right spin

Figure 5.45: Roll angle during an autonomous spin with actuator models.

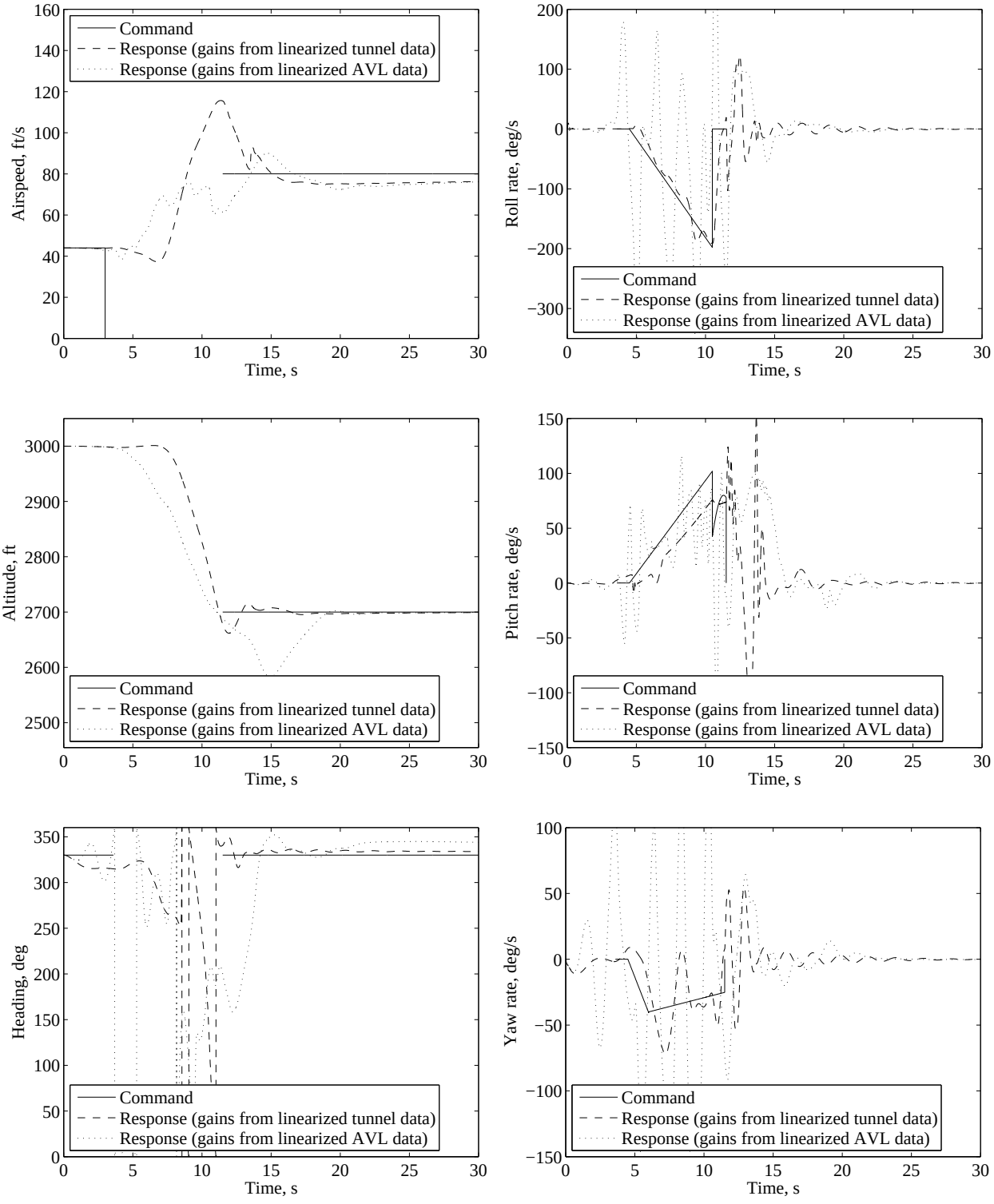


Figure 5.46: Nonlinear simulation comparison during an autonomous left spin with actuator models.

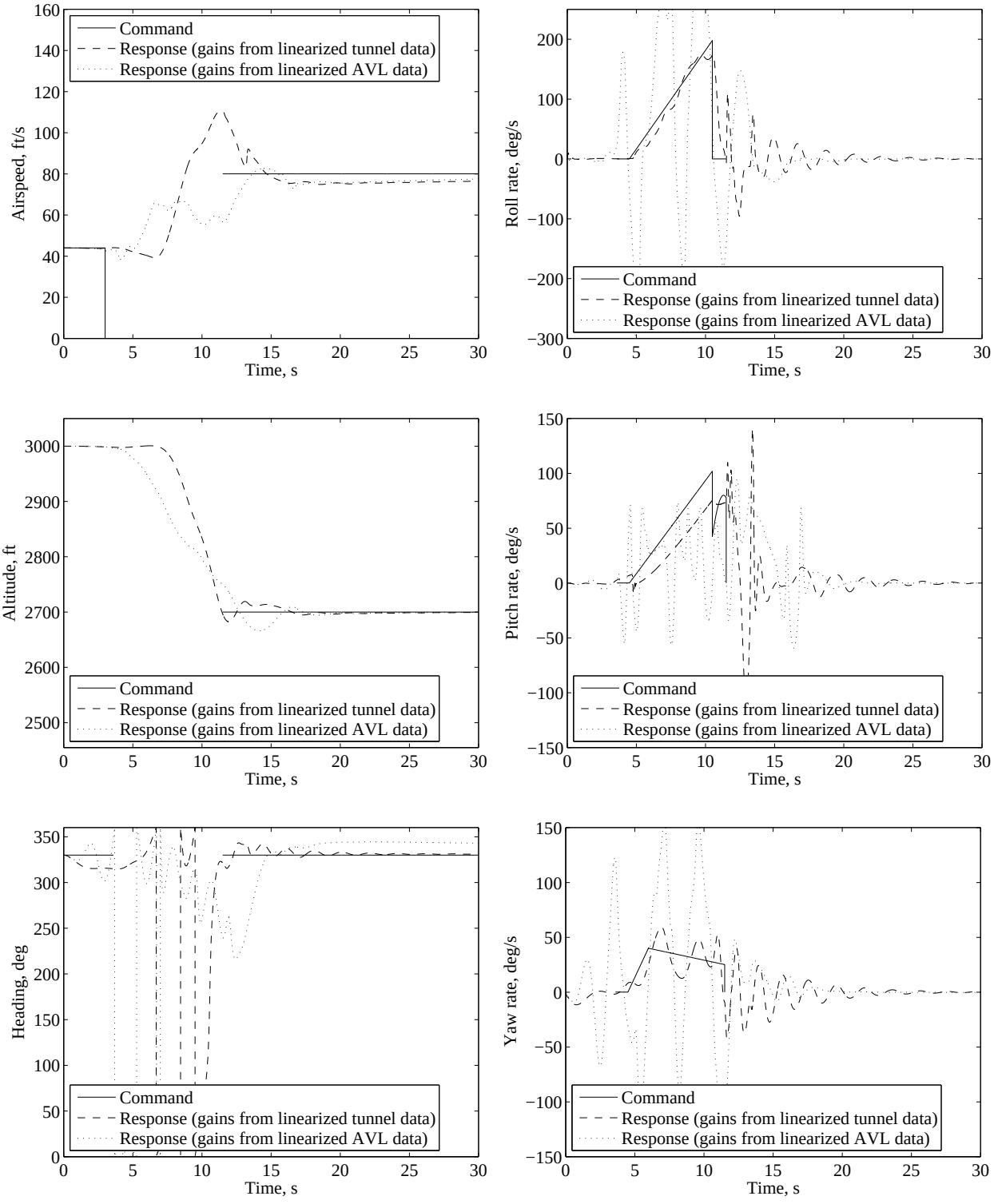


Figure 5.47: Nonlinear simulation comparison during an autonomous right spin with actuator models.

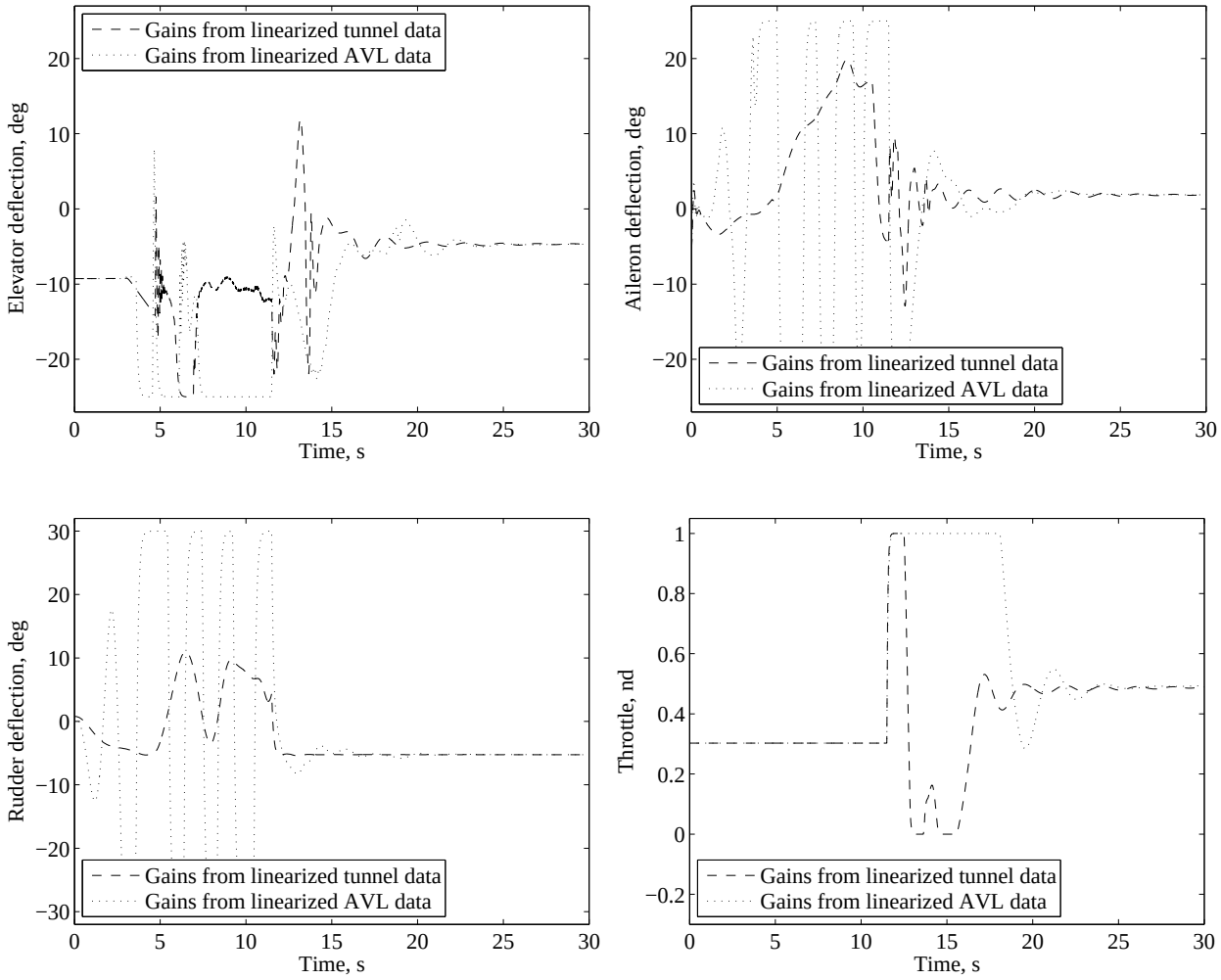


Figure 5.48: Control deflections during an autonomous left spin with actuator models.

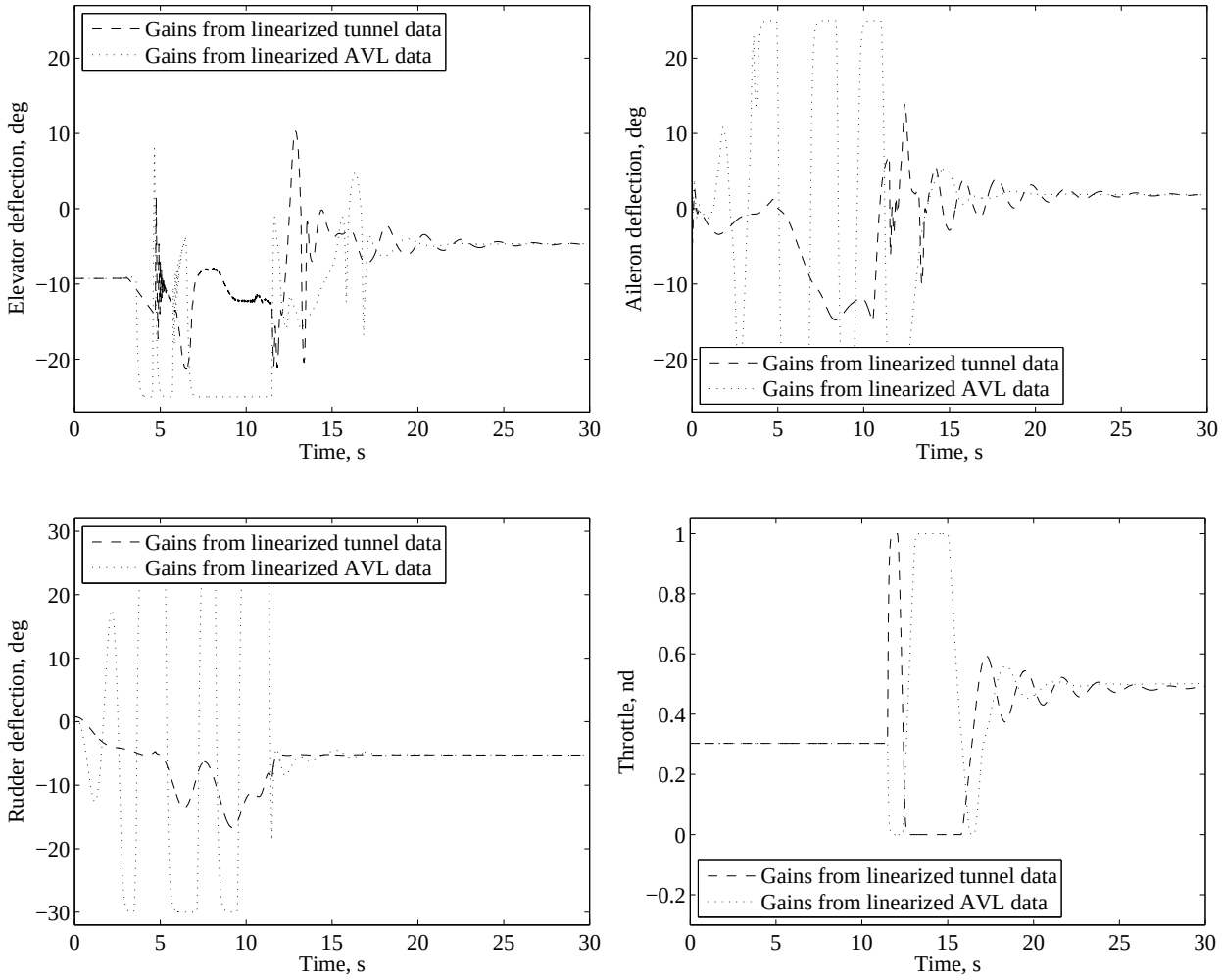


Figure 5.49: Control deflections during an autonomous right spin with actuator models.

The results from the stall turn are presented next with Fig. 5.50 showing the control deflections during the maneuver and Fig. 5.51 showing the aircraft states. Both sets of gains were able to maintain control during the vertical climbing portion of the maneuver. As the airspeed declined, the elevator effectiveness was reduced resulting in an increase in the commanded deflection until the deflection limit was reached. As the elevator actuator saturated, the pitch rate oscillated slightly when the gains from the AVL data were used. The gains from the linearized tunnel data, however, resulted in good pitch-rate tracking throughout the maneuver. Both gain sets struggled to track the roll- and yaw-rate commands exactly, but the overall execution of the maneuver was not affected. Recovery from the stall

turn occurred in five seconds for both sets of gains with the gains from the AVL data resulting in a small amount of steady state heading error.

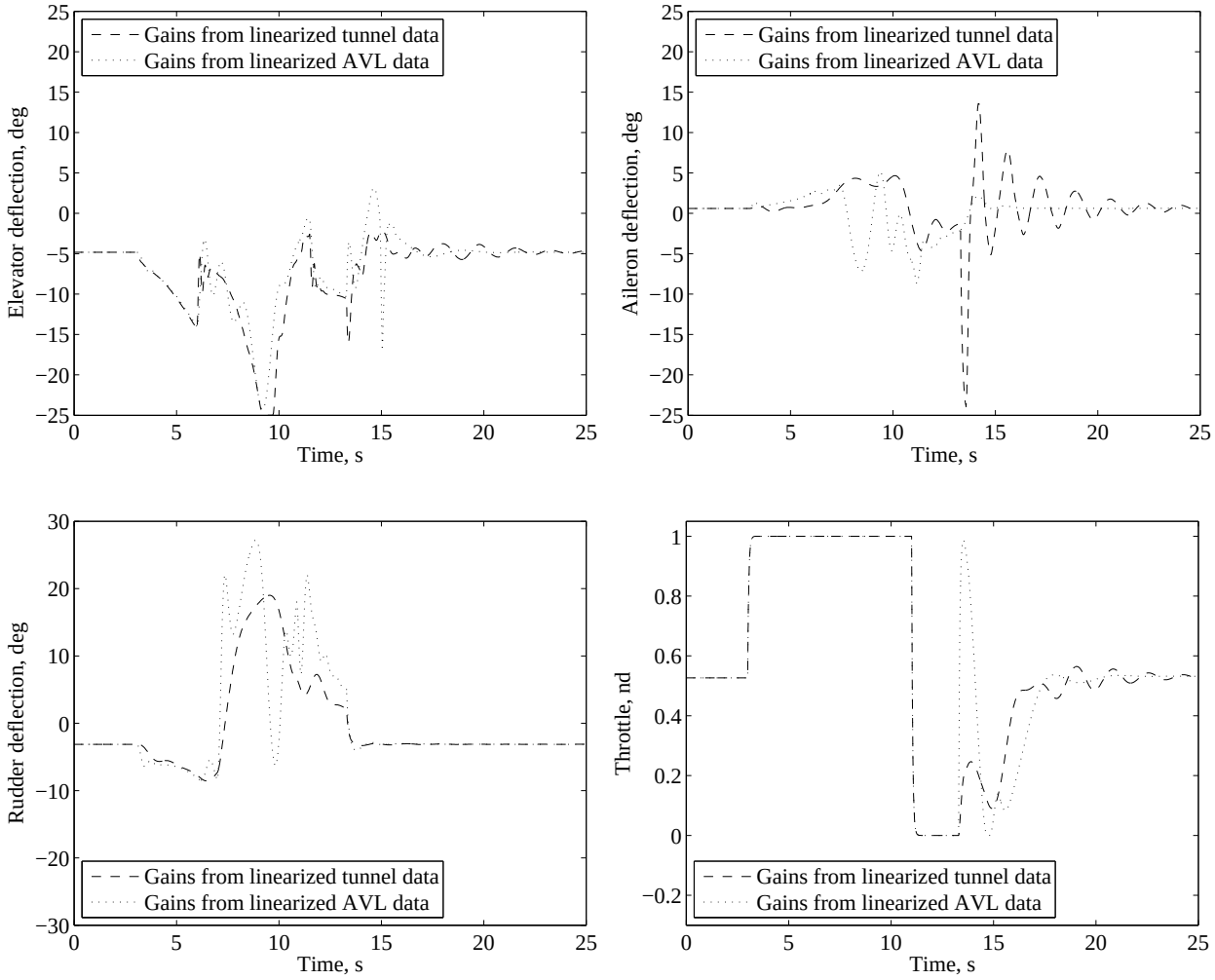


Figure 5.50: Control deflections during an autonomous stall turn with actuator models.

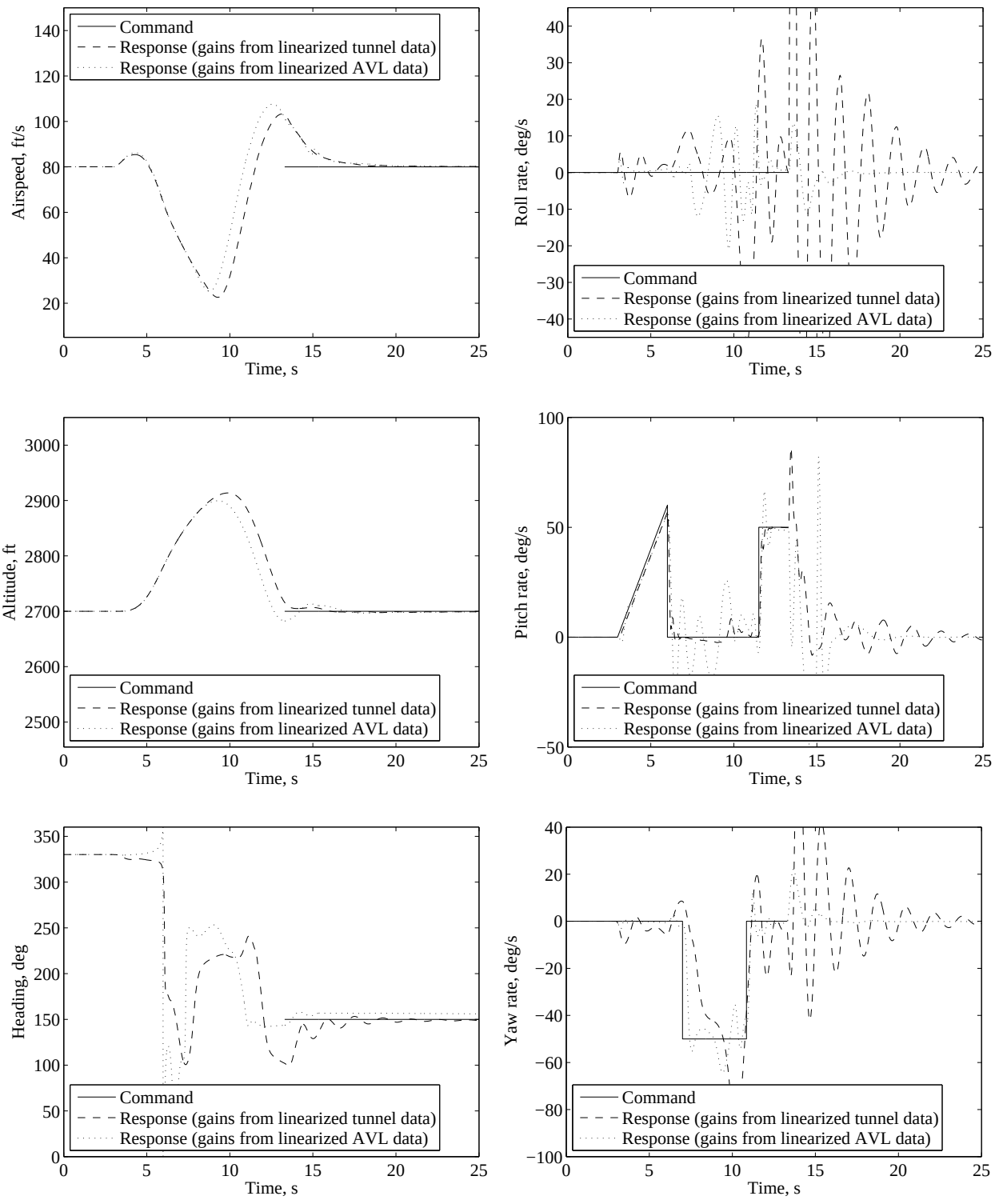


Figure 5.51: Nonlinear simulation comparison during an autonomous stall turn with actuator models.

The final results of this section are for the tail slide and Fig. 5.52 shows the control deflections during the maneuver with the aircraft states plotted in Fig. 5.53. During the beginning of the maneuver, the results are the same as for the stall turn. At the top of the vertical climb the elevator reached the deflection limit just before the minimum airspeed. As the elevator was no longer able to counter the nose-down pitching moment, the pitch rate was commanded to -80 deg/s to execute the pitch over into the vertical descent. Both sets of gains were able to control the pitch over despite having the actuator models included in the simulation. During the vertical descent, the throttle was reduced to zero to prevent excess airspeed from accumulating. As the starting altitude was reached, a positive pitch-rate command returned the aircraft to level flight before the cruise controller took over to return the aircraft to the original airspeed, altitude, and heading. Recovery from the maneuver took roughly seven seconds for both sets of gains with the gains from the tunnel data resulting in slight heading oscillations before settling on the steady-state value.

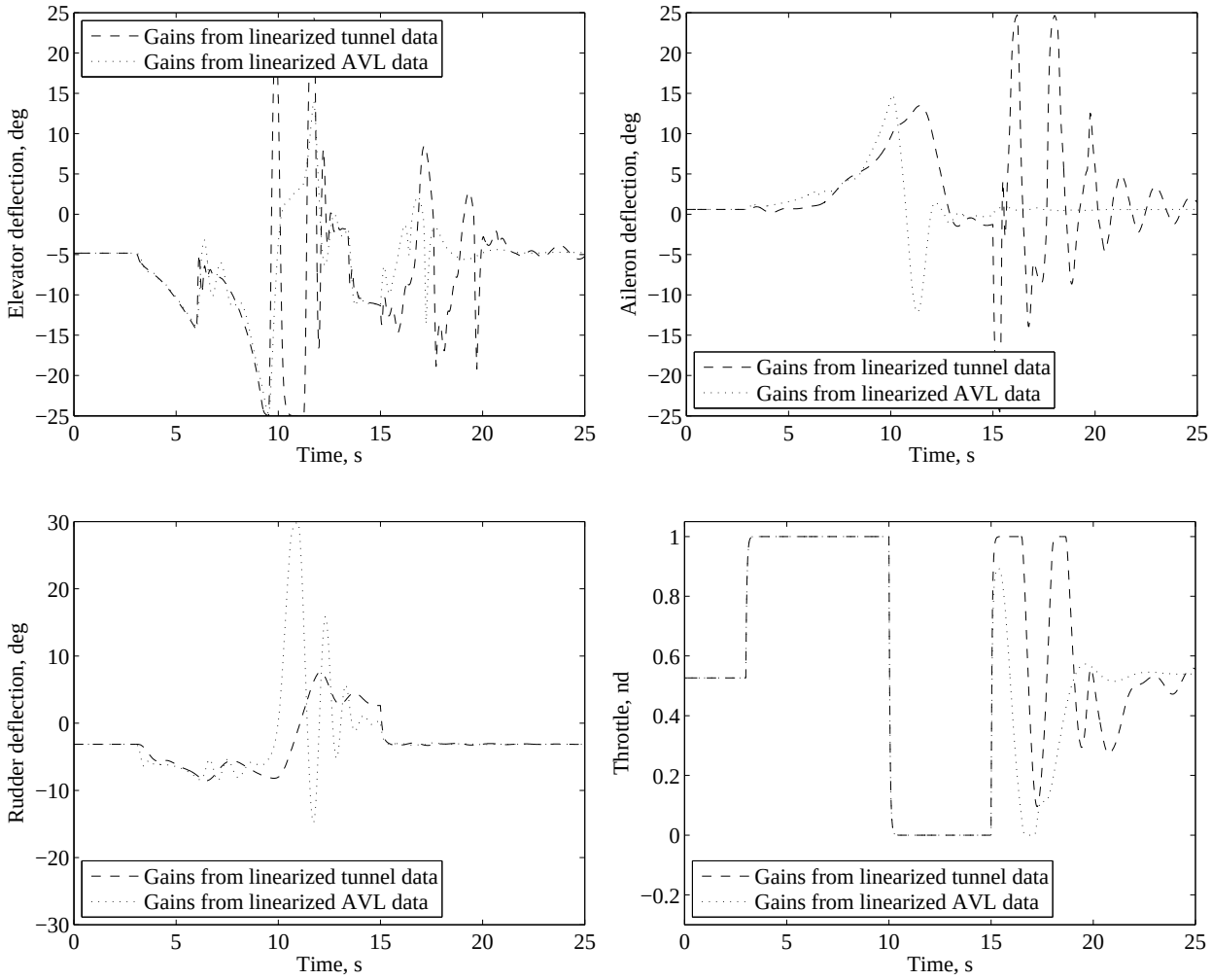


Figure 5.52: Control deflections during an autonomous tail slide with actuator models.

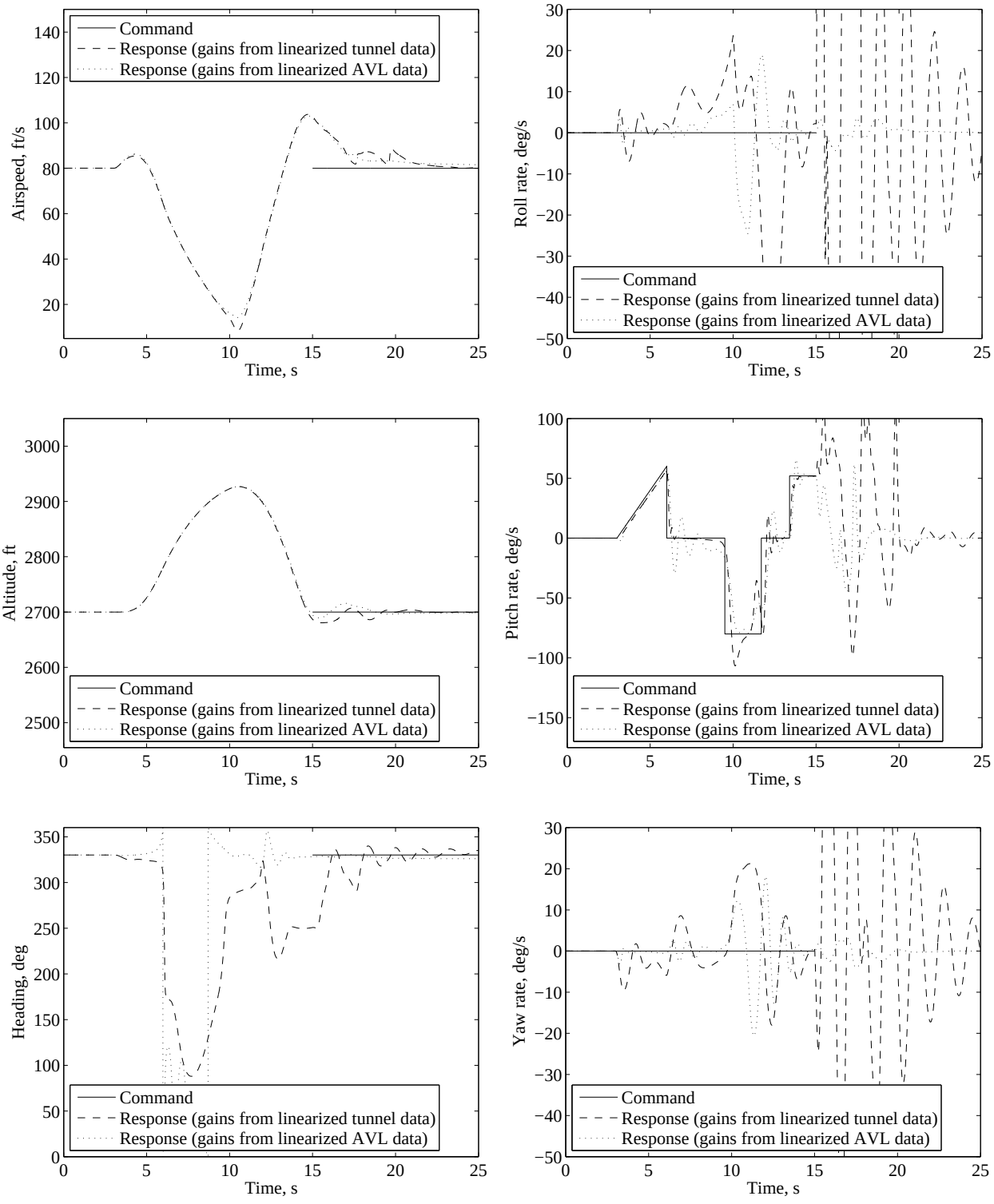


Figure 5.53: Nonlinear simulation comparison during an autonomous tail slide with actuator models.

5.3.2 Including Actuators and Measurement Noise

The simulations in this subsection use the same control gains and actuator models as the previous subsection with the addition of measurement noise to determine the effects of having imperfect measurements in the feedback path. The only aircraft states shown in these results are the ones used to control the maneuvers and the transitions to level flight. Due to the scattered nature of the plots for each set of gains, the results are presented in separate side-by-side plots for clarity. The reason for the exclusion of the control deflections was to avoid repeating the results from the previous subsection for the actuator models because the results are similar for both sets of simulations. The main difference in the control deflections with the inclusion of measurement noise is slight oscillations appear in the actuator commands. The oscillations are limited based on the rate limits in the actuator models and were most pronounced during the steady-state portions of the simulations. As an example, the elevator and aileron control deflections during a loop using the gains from the tunnel data, with measurement noise included, are shown in Fig. 5.54 along with the data from Fig. 5.35 in which measurement noise was not included. The plots reveal that the only difference in the actuator commands occurred during the recovery to cruise.

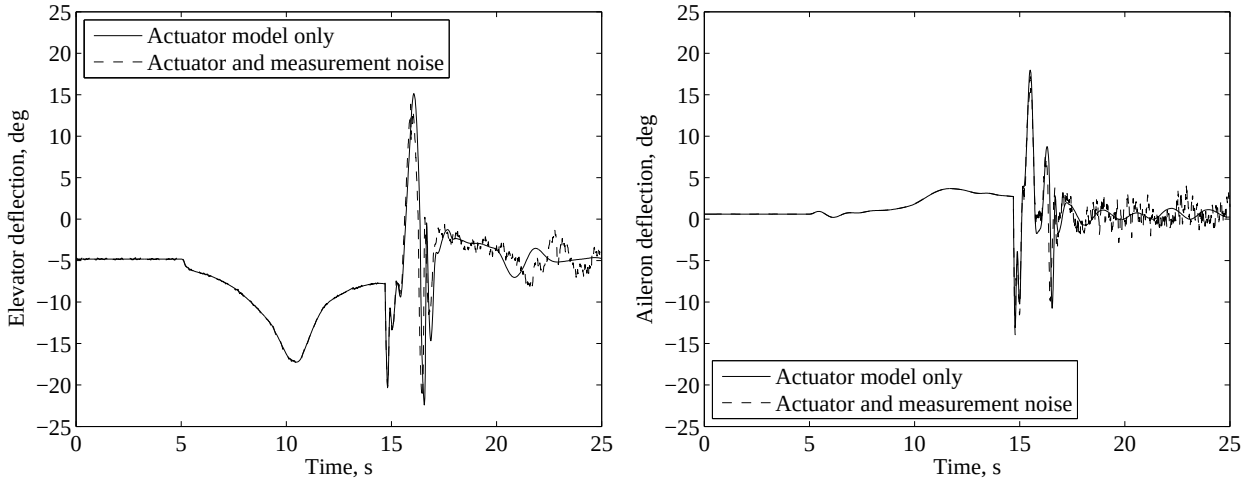


Figure 5.54: Comparison of control deflections during an autonomous loop with and without measurement noise.

Figs. 5.55 and 5.56 show the remainder of the results for the loop simulation with measurement noise included. Both sets of gains produced similar results for airspeed, altitude, and heading throughout the simulation and the results with imperfect sensors are similar to the results with perfect sensors. The angular rate plots are also similar, suggesting that the loop control system was capable of performing the maneuver despite having imperfect state measurements. The gains from the linearized tunnel data were able to track the pitch-rate command better than the gains from AVL data, but the reverse was true for the roll- and yaw-rate commands. The ± 5 deg/s oscillations that appear in the roll- and yaw-rate response for the gains from the tunnel data, however, did not affect the overall ability to perform the maneuver. Airspeed and heading returned to the cruise commands within five seconds using both sets of gains with the altitude response using the AVL gains taking slightly longer.

The results for the roll maneuvers are discussed next with Figs. 5.57 and 5.58 showing the airspeed, altitude, and heading plots while Figs. 5.59 and 5.60 depict the angular rates. Both sets of gains were able to perform rolls in both directions and the speed, altitude, and heading plots look similar for both gain sets. Similar to previous results, when measurement noise was included the gains from the tunnel data were able to more closely track the pitch-rate commands while the gains from the AVL data were able to more closely track the roll-rate commands for both roll directions. Both gain sets struggled to track the yaw-rate commands during the maneuvers with the gains from AVL data performing the best overall. The gains from the tunnel data were able to track the yaw-rate commands during the initial portion of the maneuver, but once the aircraft rolled past 90 degrees, the response diverged from the commands until the maneuver ended. Despite the similarity in the airspeed at the start of the transition from maneuvering to level flight for both gain sets, the gains from AVL data resulted in a higher altitude and hence a longer recovery time.

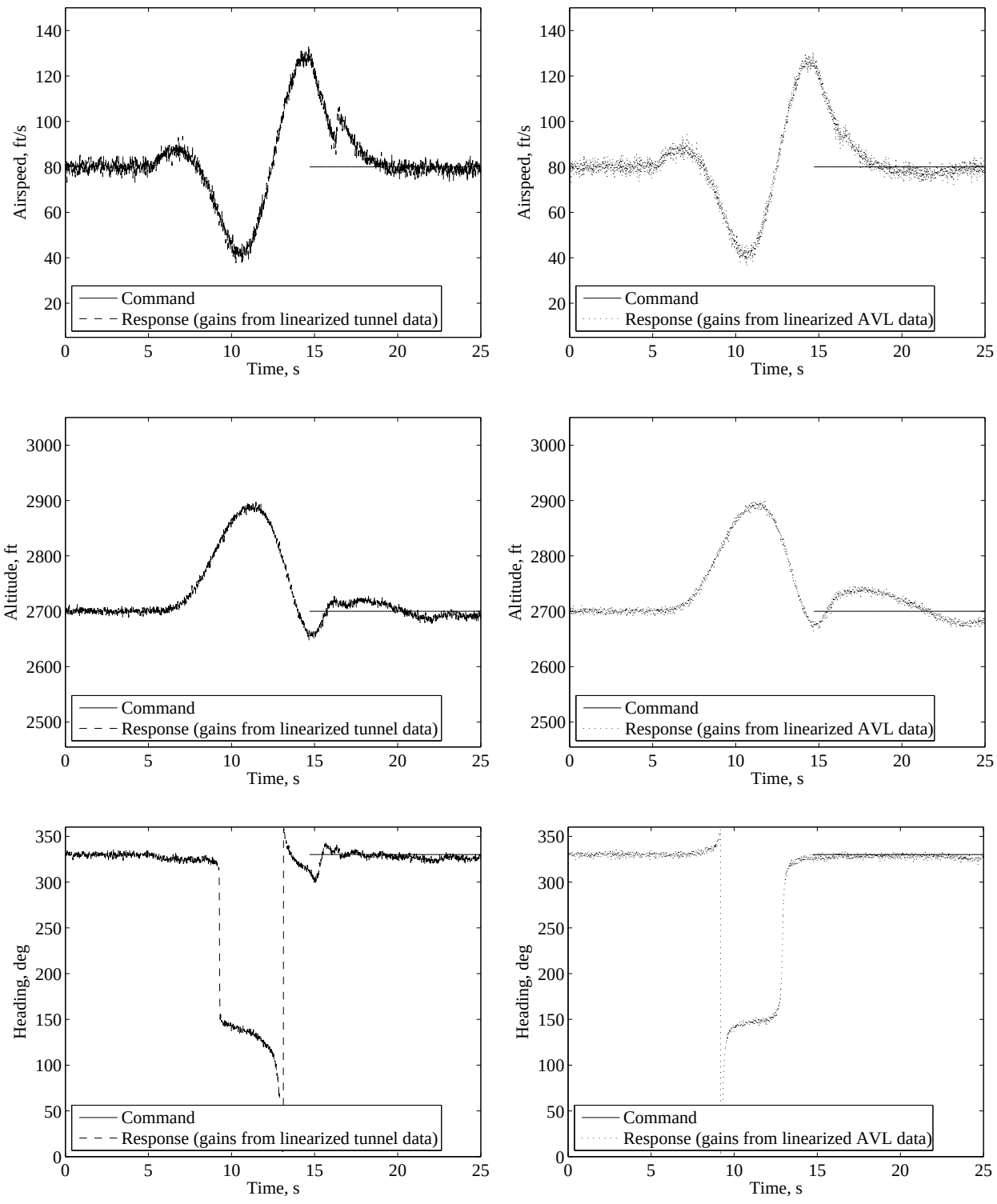


Figure 5.55: Simulation comparison during an autonomous loop with actuator models and measurement noise.

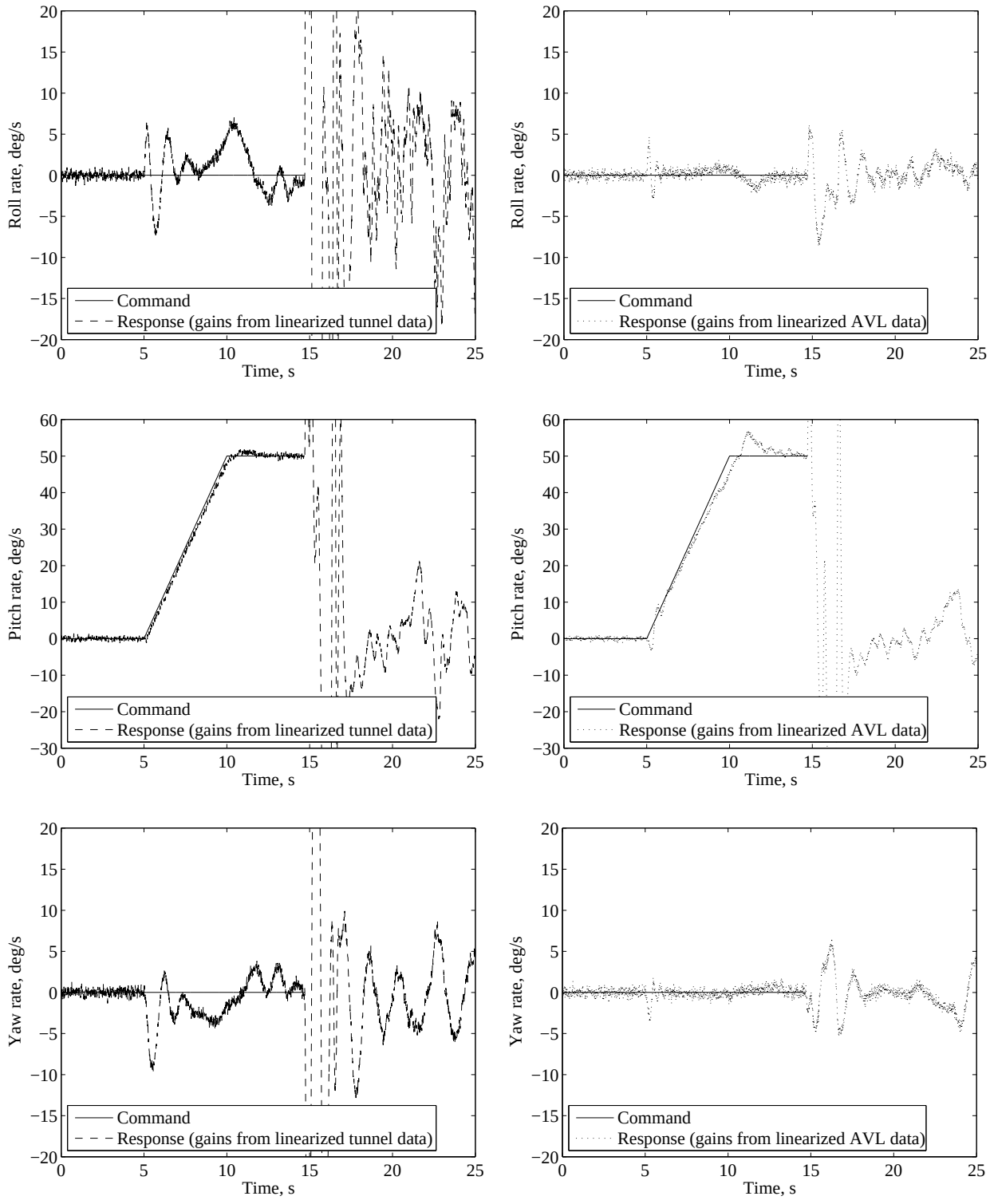


Figure 5.56: Angular rates during an autonomous loop with actuator models and measurement noise.

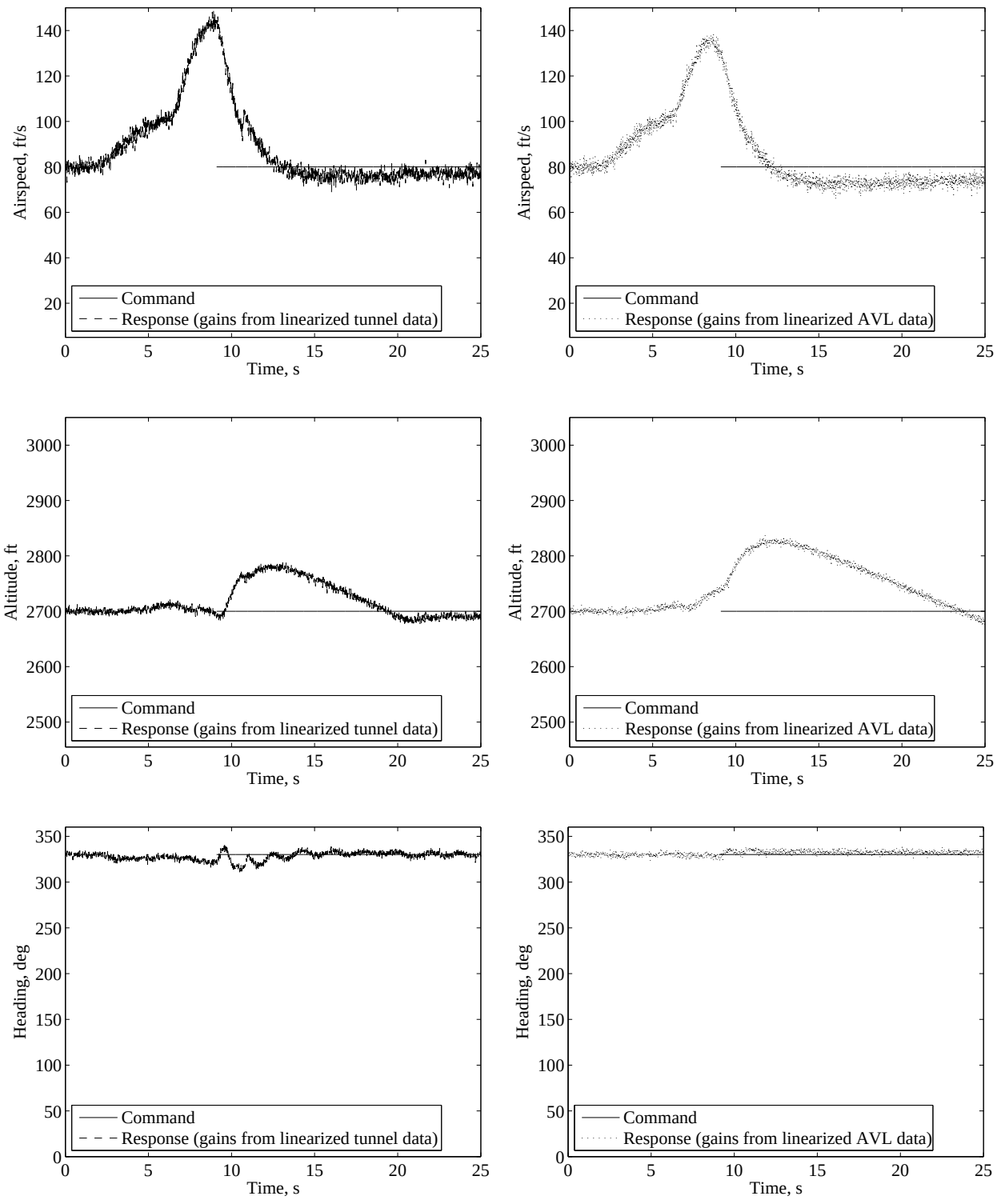


Figure 5.57: Simulation comparison during an autonomous left roll with actuator models and measurement noise.

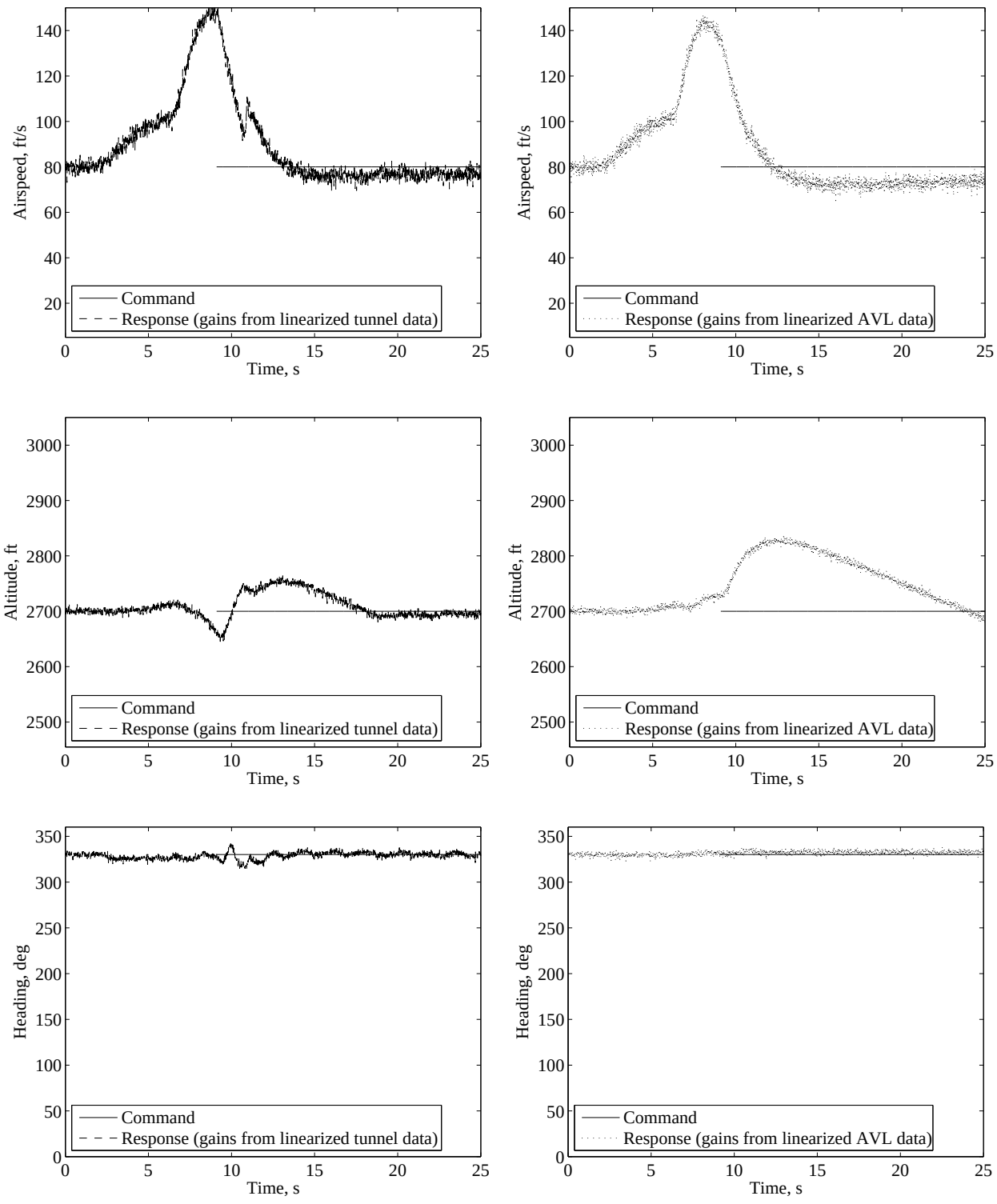


Figure 5.58: Simulation comparison during an autonomous right roll with actuator models and measurement noise.

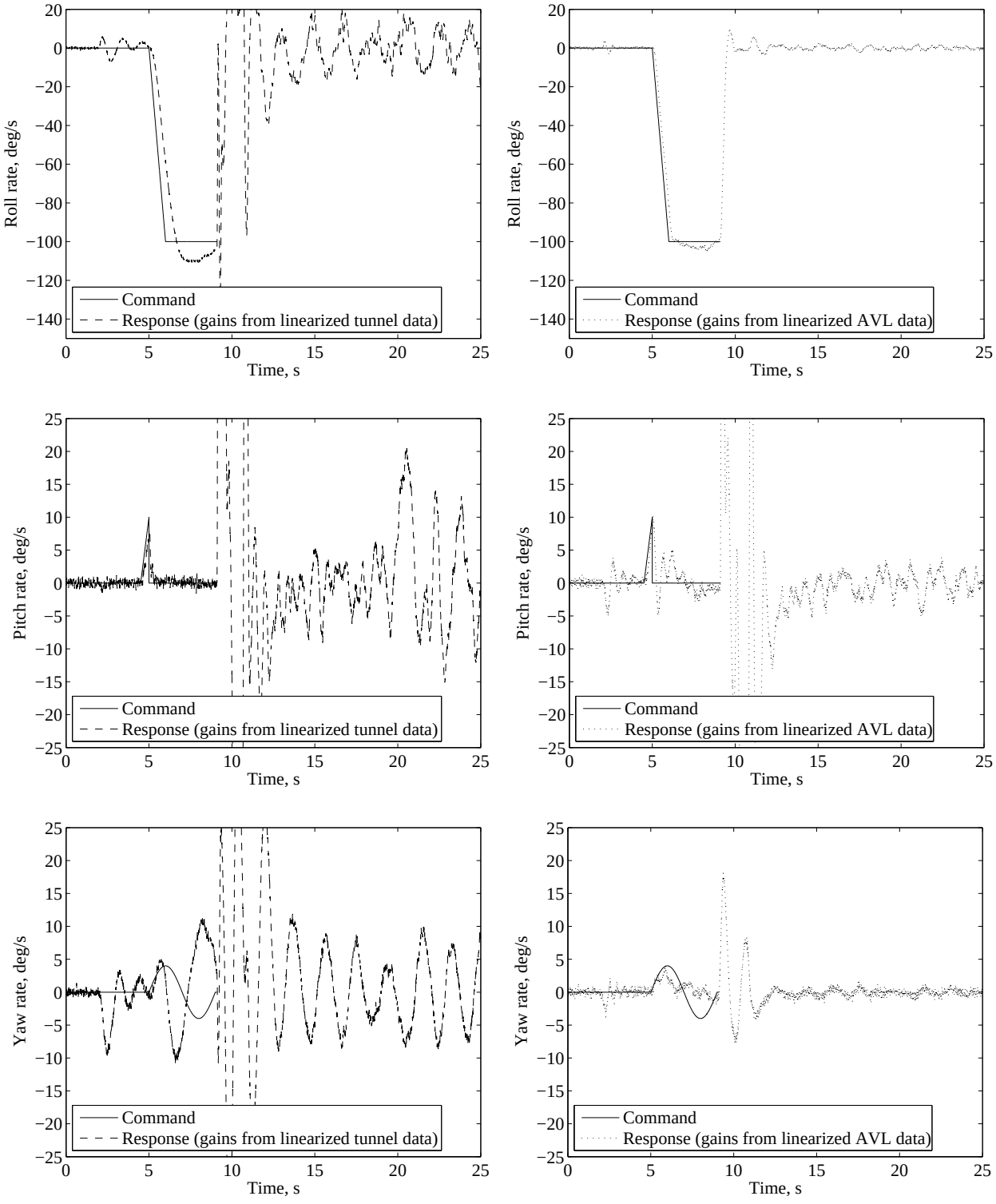


Figure 5.59: Angular rates during an autonomous left roll with actuator models and measurement noise.

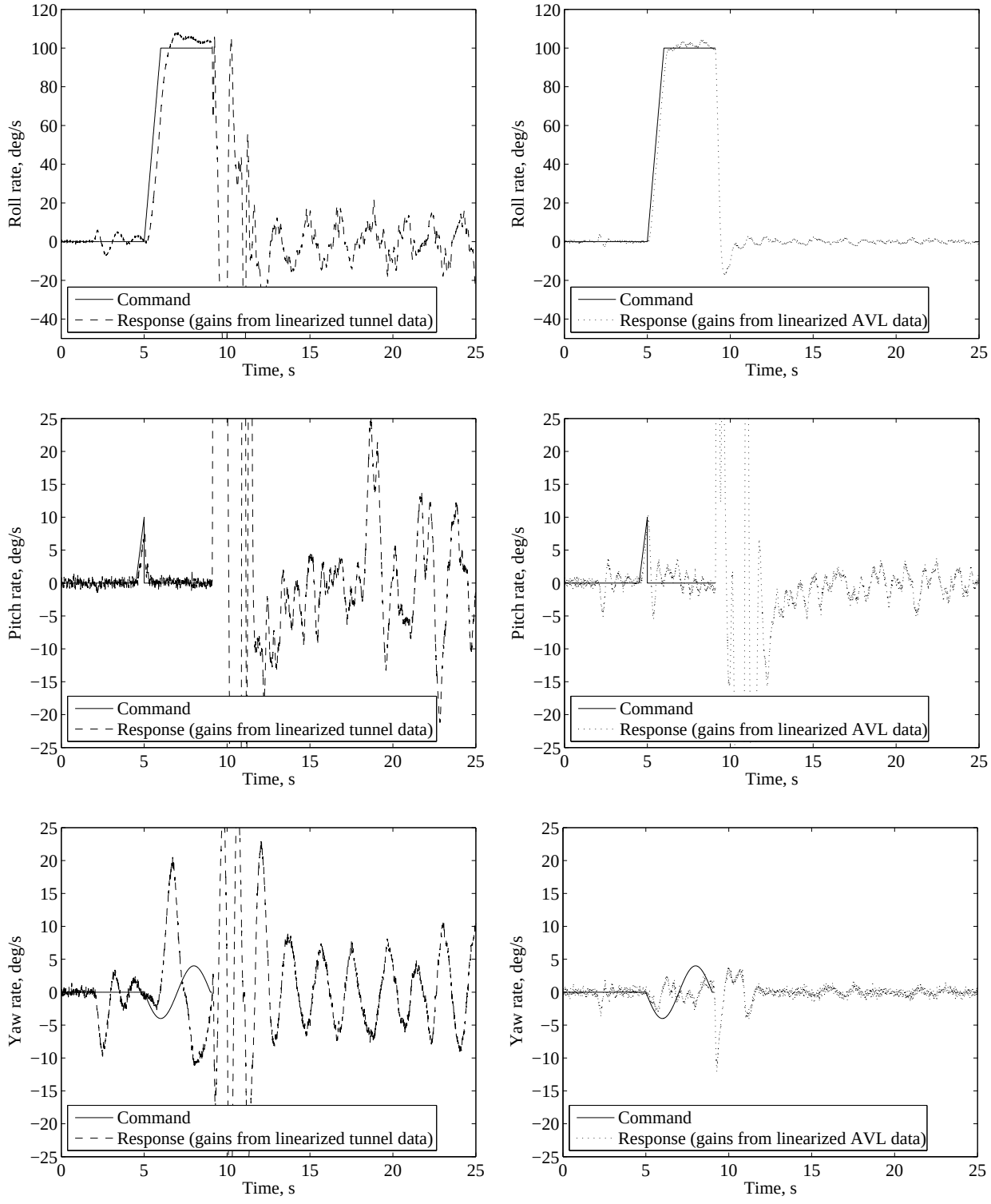


Figure 5.60: Angular rates during an autonomous right roll with actuator models and measurement noise.

The knife-edge simulation results are presented in Figs. 5.61 - 5.66. The results for the roll-angle controller are shown first and both sets of gains were able to track the roll-angle commands throughout the maneuver. Looking at the angular rate plots of Figs. 5.63 and 5.64, the transition into the knife-edge orientation was easily controlled by both gain sets. As the roll angle increased beyond 45 degrees and the roll-angle controller took over, the roll angle was commanded to 90 degrees immediately and the step response from both sets of gains contained no overshoot. During the maneuver, the heading was tracked perfectly by both gain sets with the altitude and airspeed results appearing similar to the results without measurement noise. The largest loss in altitude occurred while using the gains from AVL data to perform right knife-edge flight. With measurement noise included, the transition from knife edge to level flight contained a small amount of steady-state error in speed and altitude when using the gains from AVL data. The heading plot of Fig. 5.65 shows that the steady-state heading error was corrected at the 30 second mark, but that correction did not occur for the airspeed error.

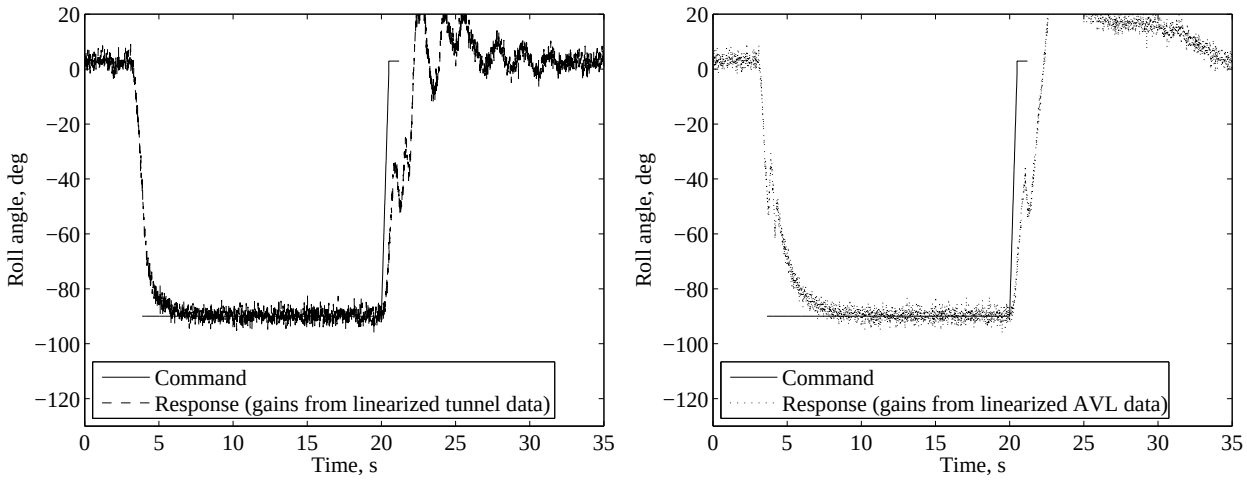


Figure 5.61: Roll angle during autonomous left knife-edge flight with actuator models and measurement noise.

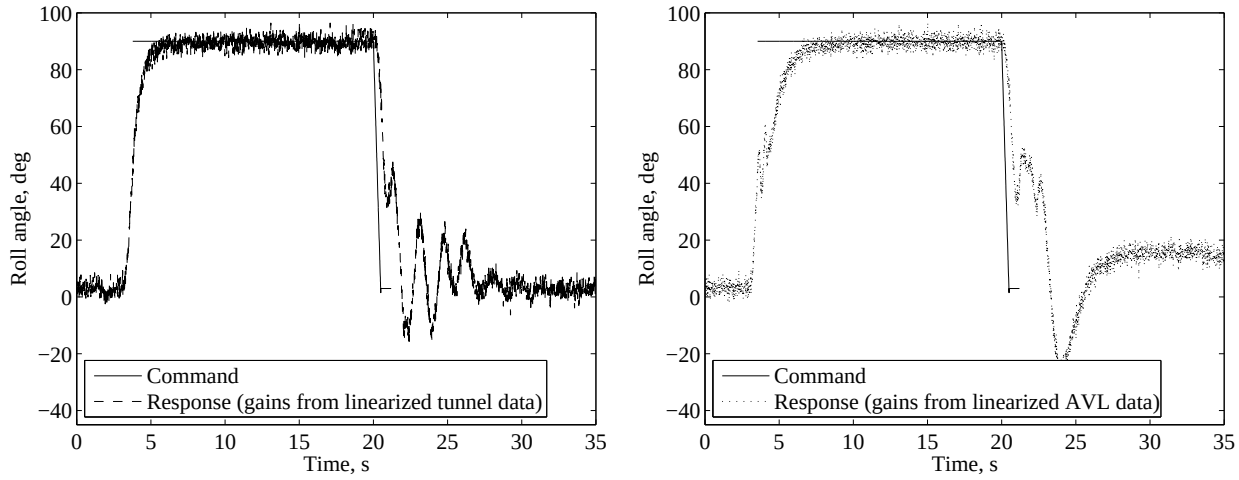


Figure 5.62: Roll angle during autonomous right knife-edge flight with actuator models and measurement noise.

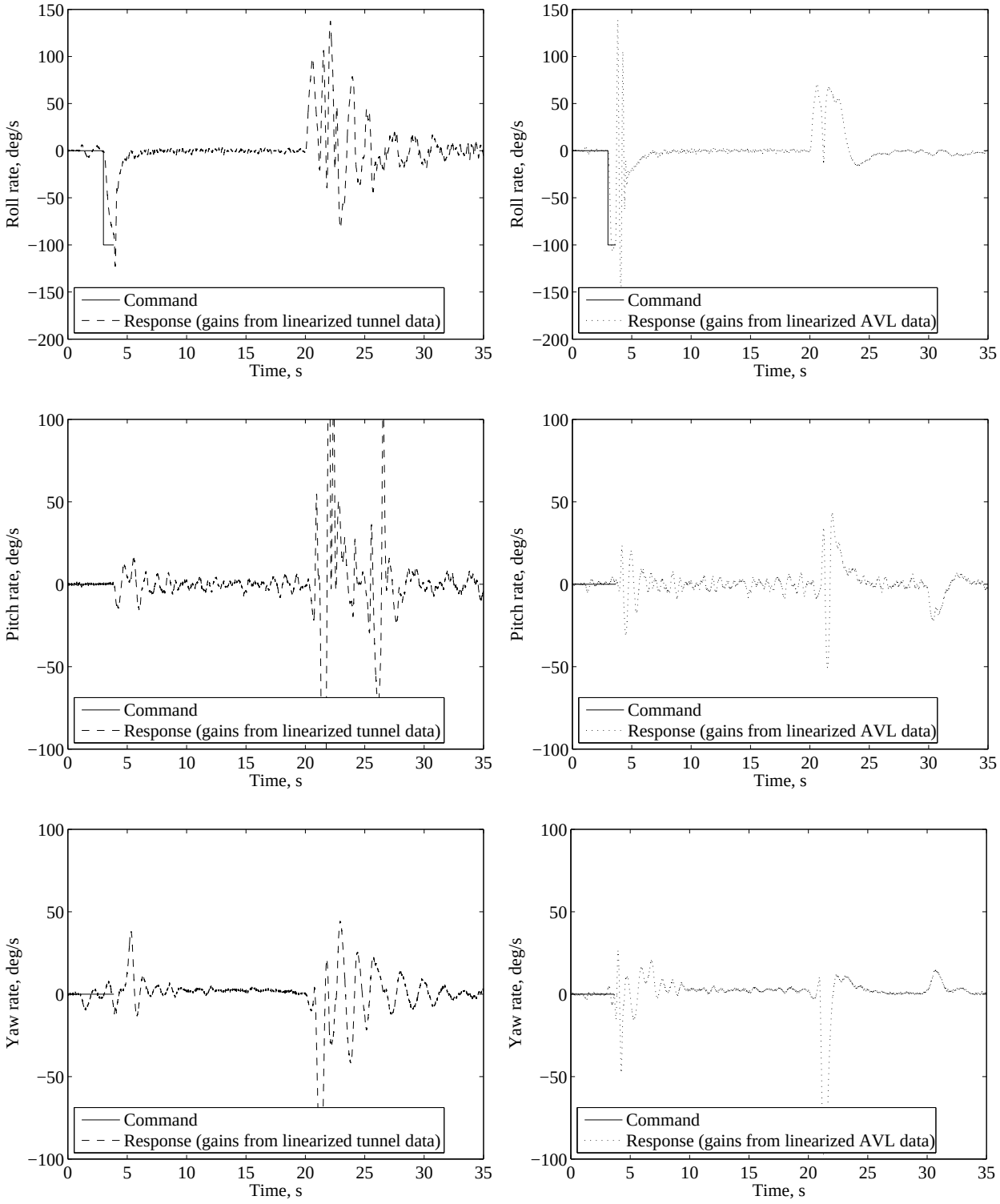


Figure 5.63: Angular rates during autonomous left knife-edge flight with actuator models and measurement noise.

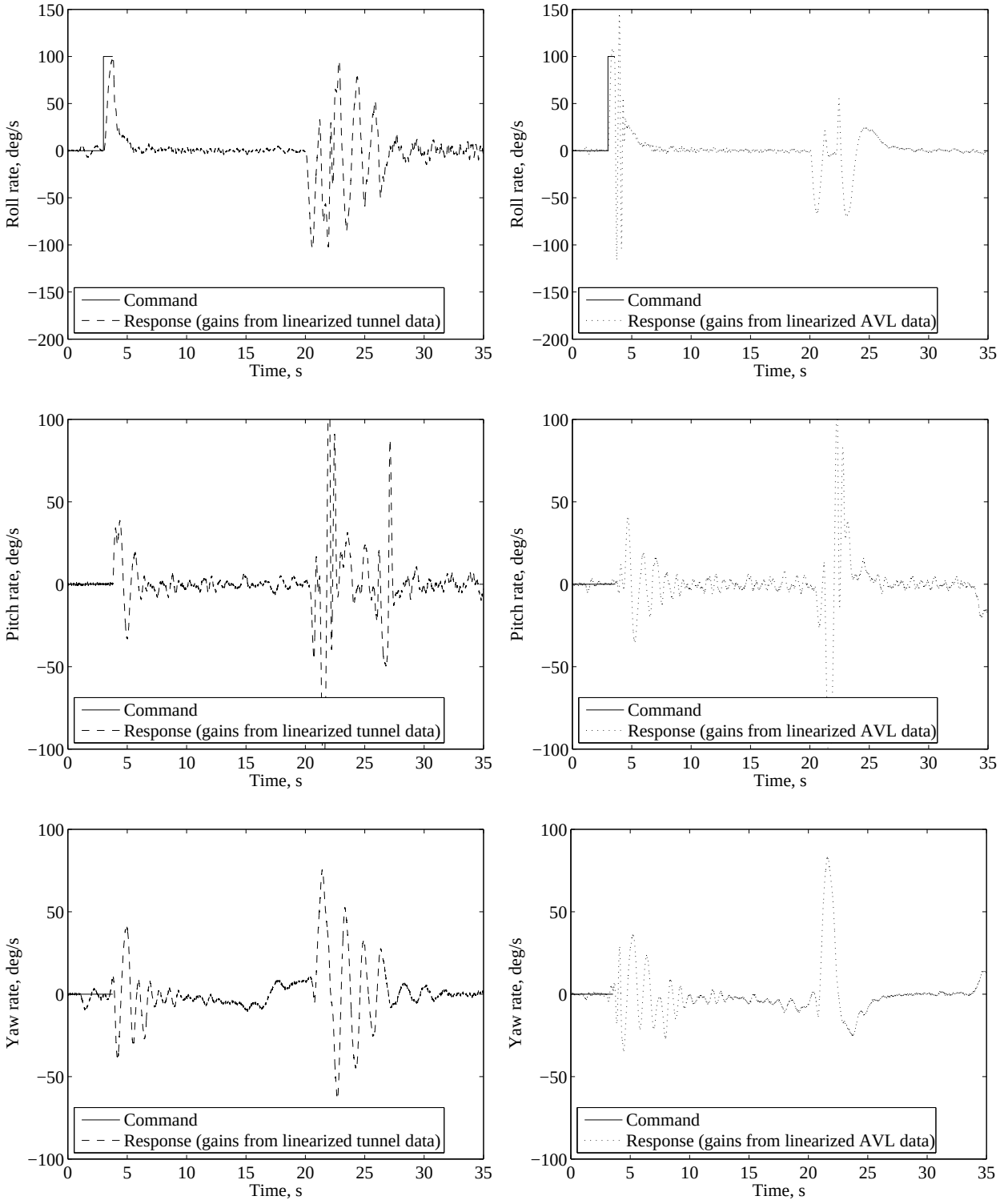


Figure 5.64: Angular rates during autonomous right knife-edge flight with actuator models and measurement noise.

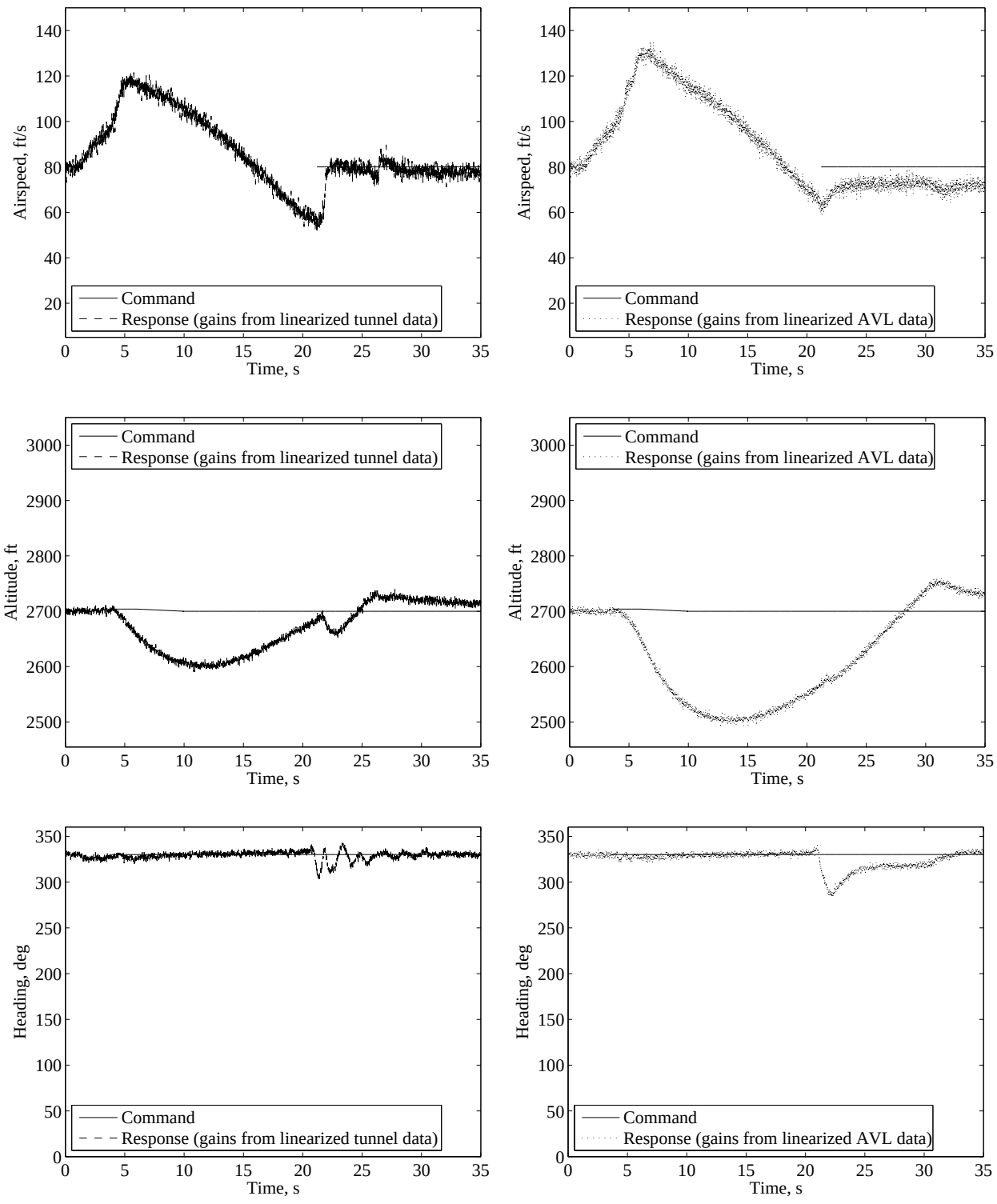


Figure 5.65: Simulation comparison during autonomous left knife-edge flight with actuator models and measurement noise.

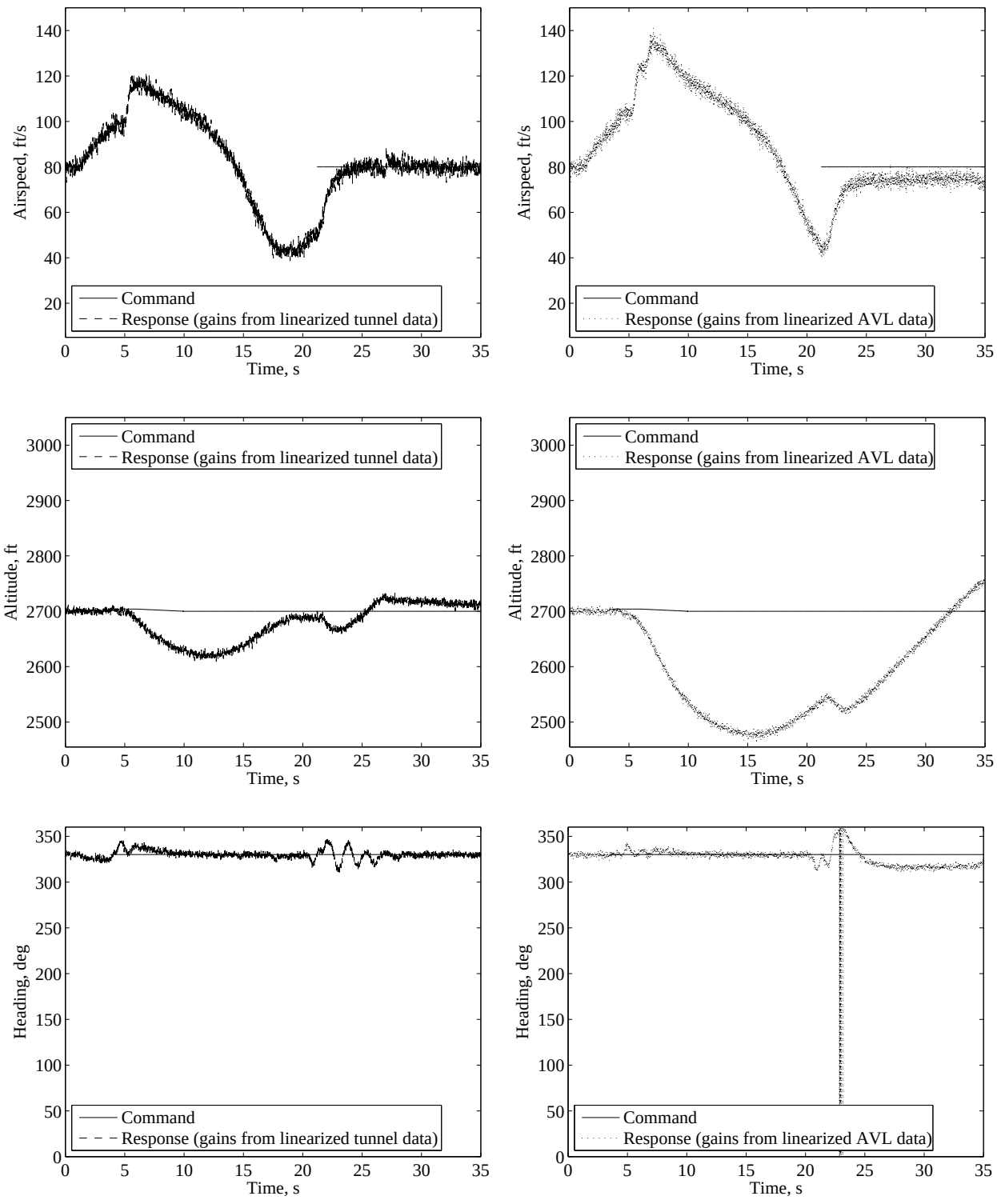
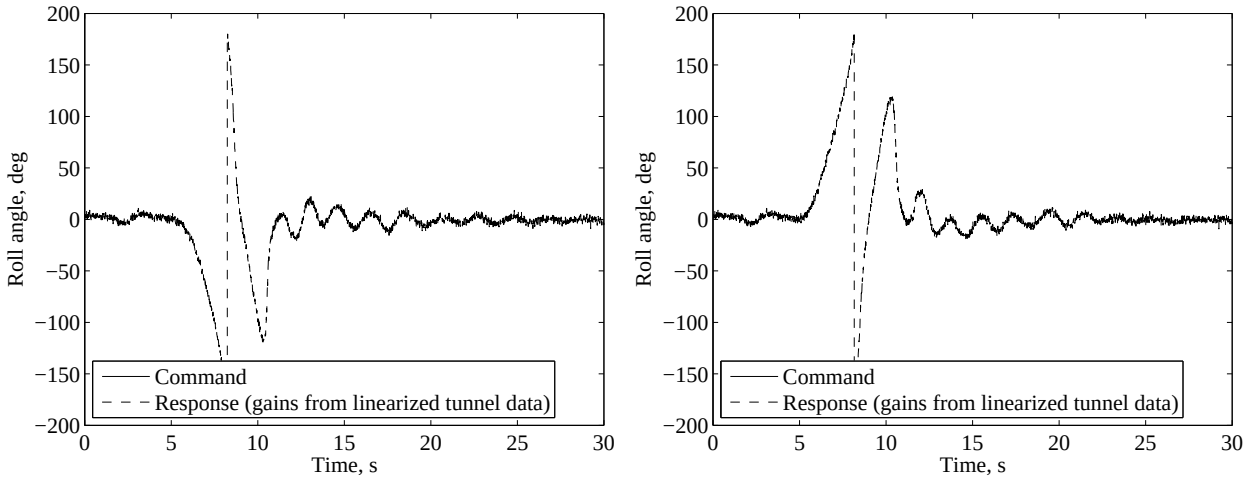


Figure 5.66: Simulation comparison during autonomous right knife-edge flight with actuator models and measurement noise.

For the spin results, shown in Figs. 5.67 - 5.69, the plots using the gains from AVL data are excluded. The vortex lattice aerodynamic modeling method used in AVL is inviscid, so the stability derivatives from AVL contain error for flight conditions near stall. In the spin results from the previous subsection, the gains from the AVL data (determined using a pre-stall aerodynamic model) were unable to maintain control of the aircraft. Since the gains from the AVL data failed in that simulation, there was no reason to retest the gains with measurement noise.

The roll angle plots appear first for both the left and right spin. During the initial stalling portion of the spin, the ailerons were used to keep the wings level and the roll angle plots indicate that the gains from the tunnel data were able to track the zero roll-angle command. During the stalling portion of the maneuver, the airspeed was commanded to zero until the angle of attack exceeded the critical angle. Examining the plots of airspeed reveals that the speed command to slow down the airplane did not occur. Due to the noisy angle of attack measurements, the spin controller took over command before the zero speed command was given.

As the spin controller took over and the spinning motion began, the roll- and pitch-rate gains were able to track the reference commands fairly well for both spin directions. The yaw-rate controller did not perform as well in terms of tracking the commands, but was able to track the initial ramp up to start the spin itself. With the inclusion of measurement noise, more time was required to recover from the spin than with perfect state estimates with the altitude and heading returning to their level-flight commands after 15 seconds. The airspeed, however, recovered in five seconds, but contained a small amount of steady-state error for both spin directions.



(a) Left spin

(b) Right spin

Figure 5.67: Roll angle during an autonomous spin with actuator models and measurement noise.

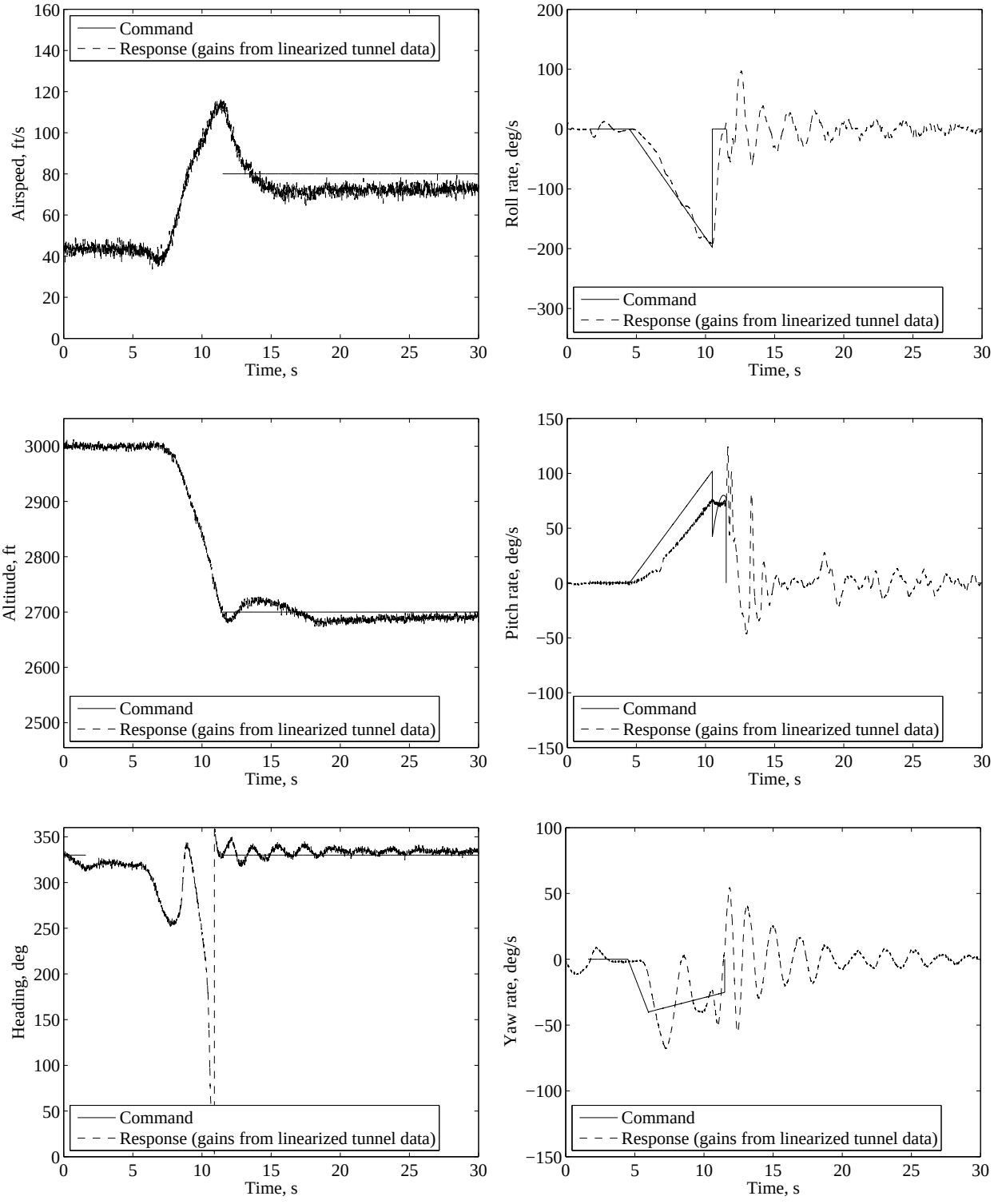


Figure 5.68: Simulation of an autonomous left spin with actuator models and measurement noise.

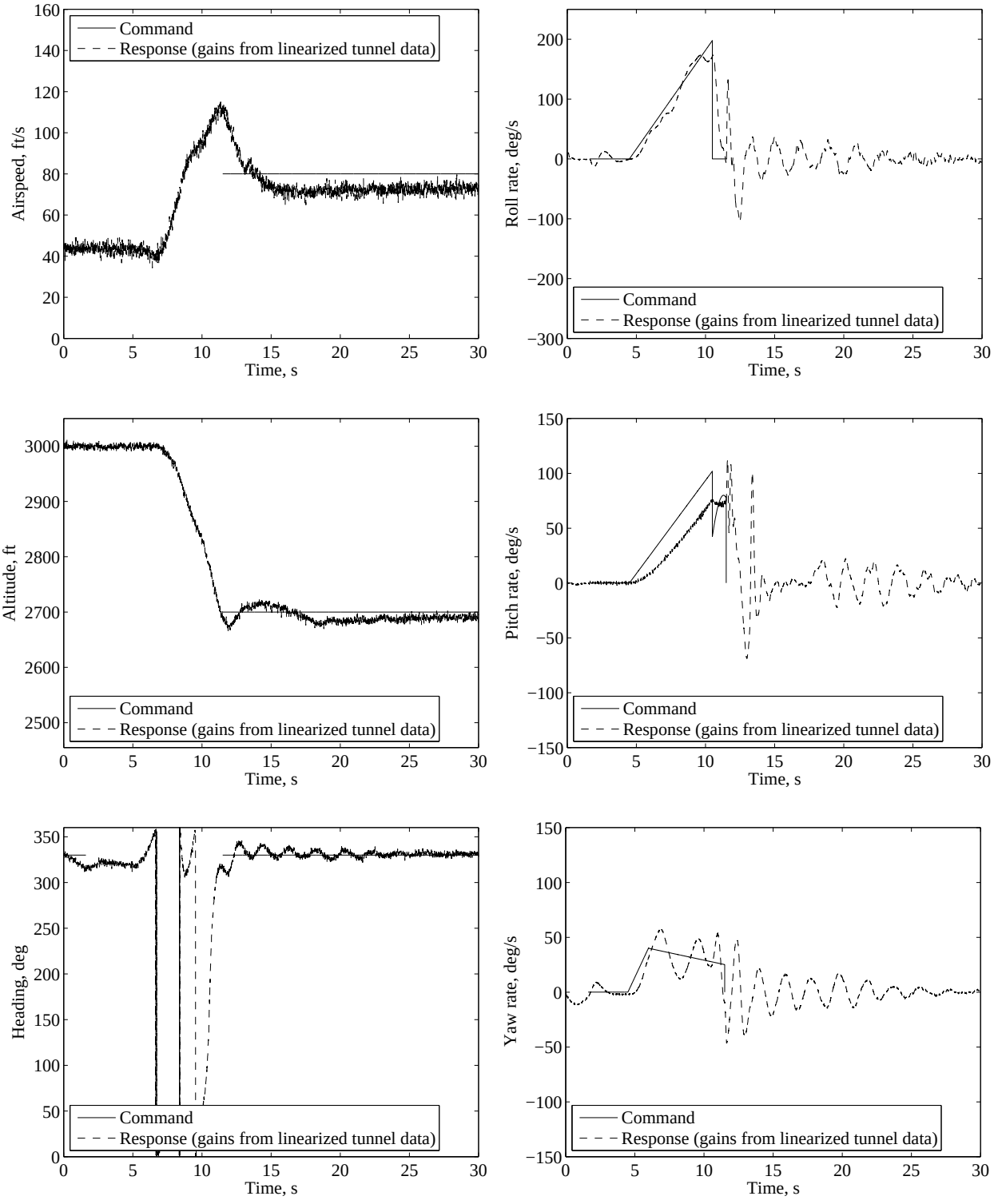


Figure 5.69: Simulation of an autonomous right spin with actuator models and measurement noise.

Figures 5.70 and 5.71 contain the results for the stall turn simulation with measurement noise. Both sets of gains were able to complete the maneuver and the speed, altitude, and heading plots appear similar for both. During the pitch up into the vertical climb, the gains from AVL data were able to keep the oscillations in roll and yaw rate smaller than the gains from the tunnel data. The pitch-rate commands during the vertical climb and turn, however, were tracked better by the gains from the tunnel data. The yaw rate plot suggests that the gains from the tunnel data were more sensitive to the off-trim condition experienced as the airspeed decreased at the top of the climb. During the yawing portion of the maneuver, the yaw-rate response was slower with more overshoot than the response using the gains from AVL. For both gain sets the recovery to the level-flight commands of airspeed and altitude occurred within five seconds. The response to the heading command during the transition was slower for the gains from the tunnel data than for the gains from AVL data, but was still able to reach steady state in only seven seconds.

The final results in this subsection appear in Figs. 5.72 and 5.73 and correspond to the tail slide. Similar to the stall turn results, the pitch-rate commands were tracked better by the gains from the tunnel data while the roll- and yaw-rate commands were tracked better by the gains from AVL data. The airspeed during the tail slide decreased more than during the stall turn and both sets of gains were still able to track the required pitch-rate command to begin the vertical descent. During the recovery to level flight, the gains from AVL data were again able to stabilize the heading quicker than the gains from the tunnel data. Despite the performance differences between the two sets of gains, both were able to complete the maneuver and return the aircraft to level flight.

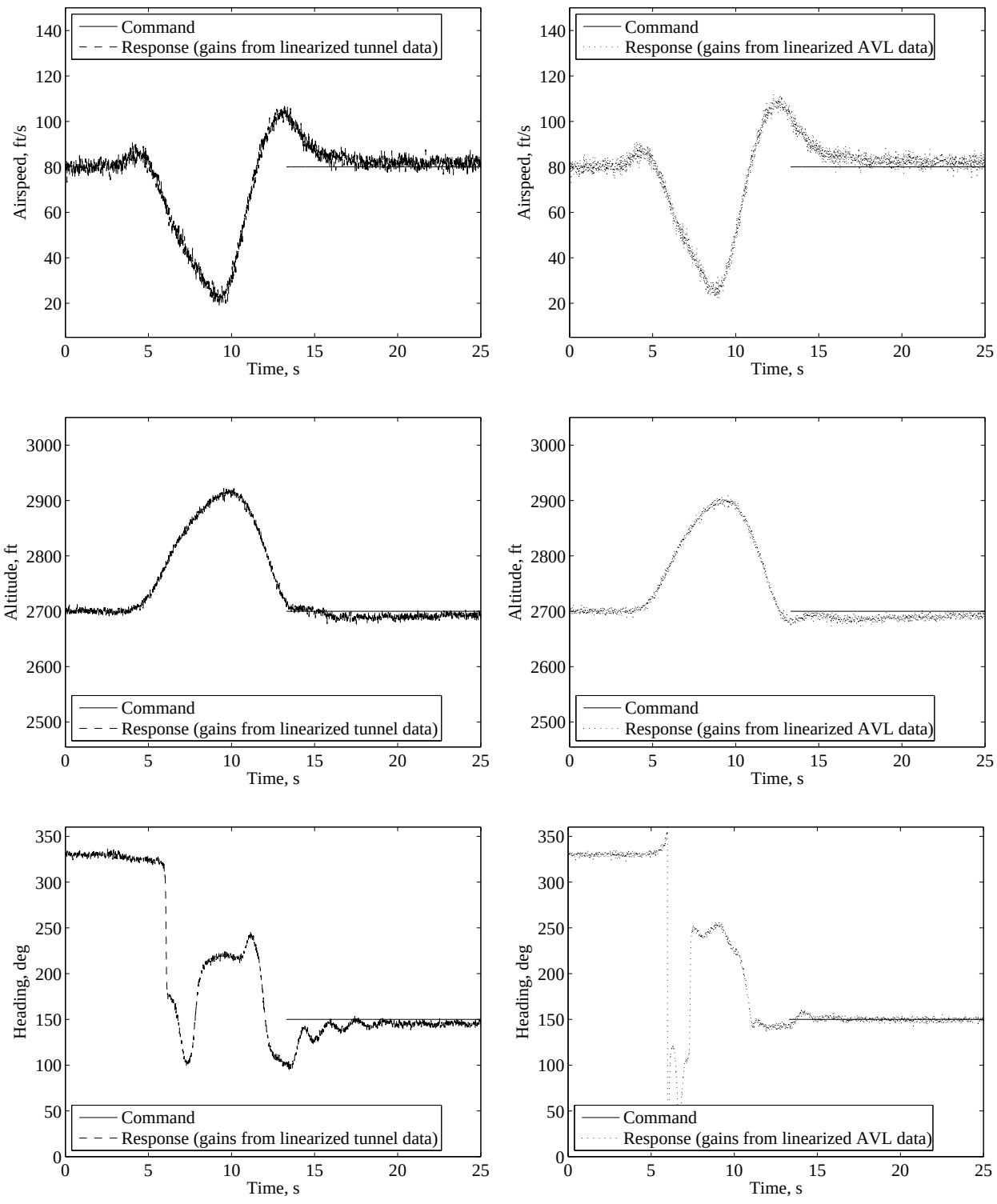


Figure 5.70: Simulation comparison during an autonomous stall turn with actuator models and measurement noise.

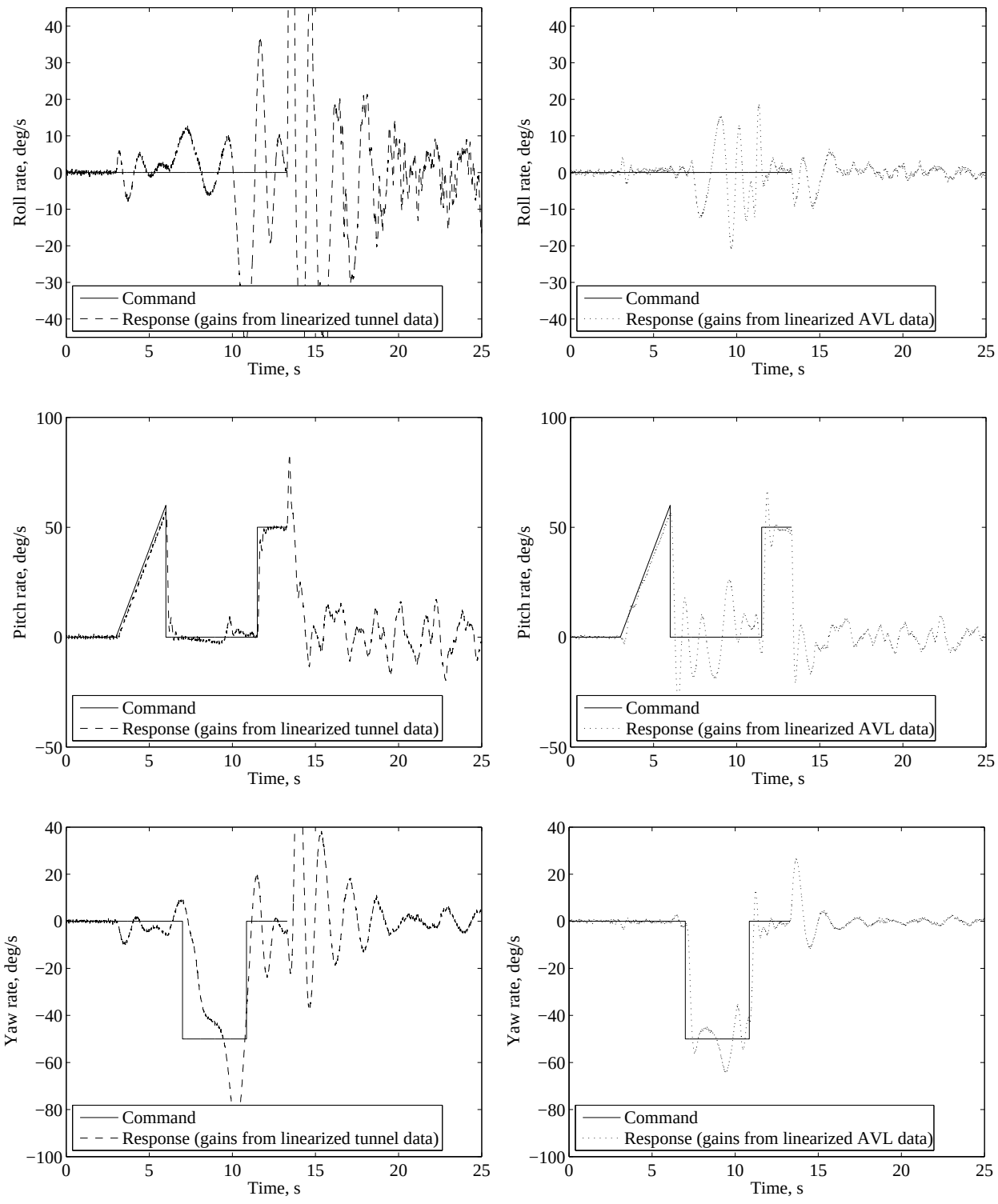


Figure 5.71: Angular rates during an autonomous stall turn with actuator models and measurement noise.

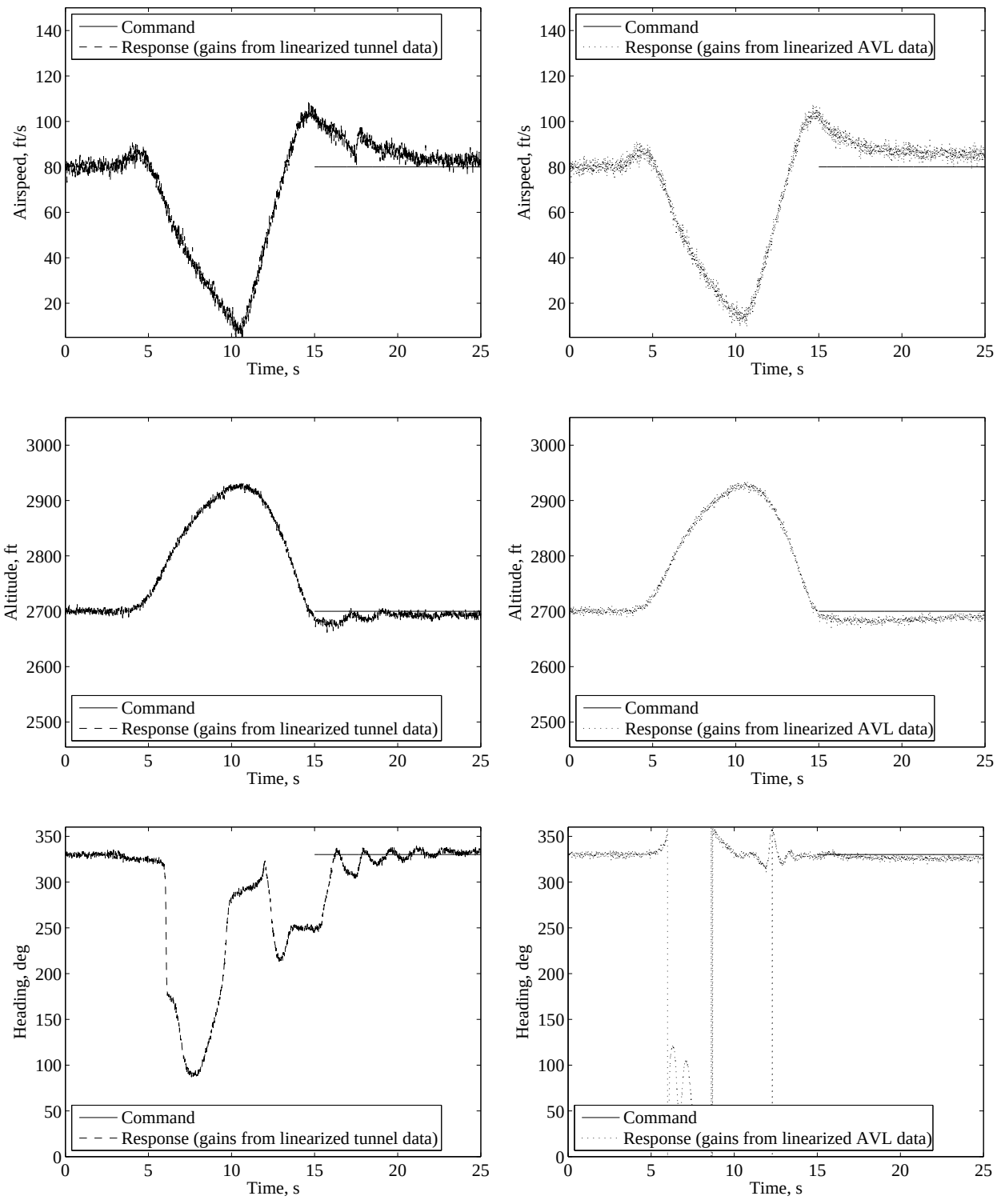


Figure 5.72: Simulation comparison during an autonomous tail slide with actuator models and measurement noise.

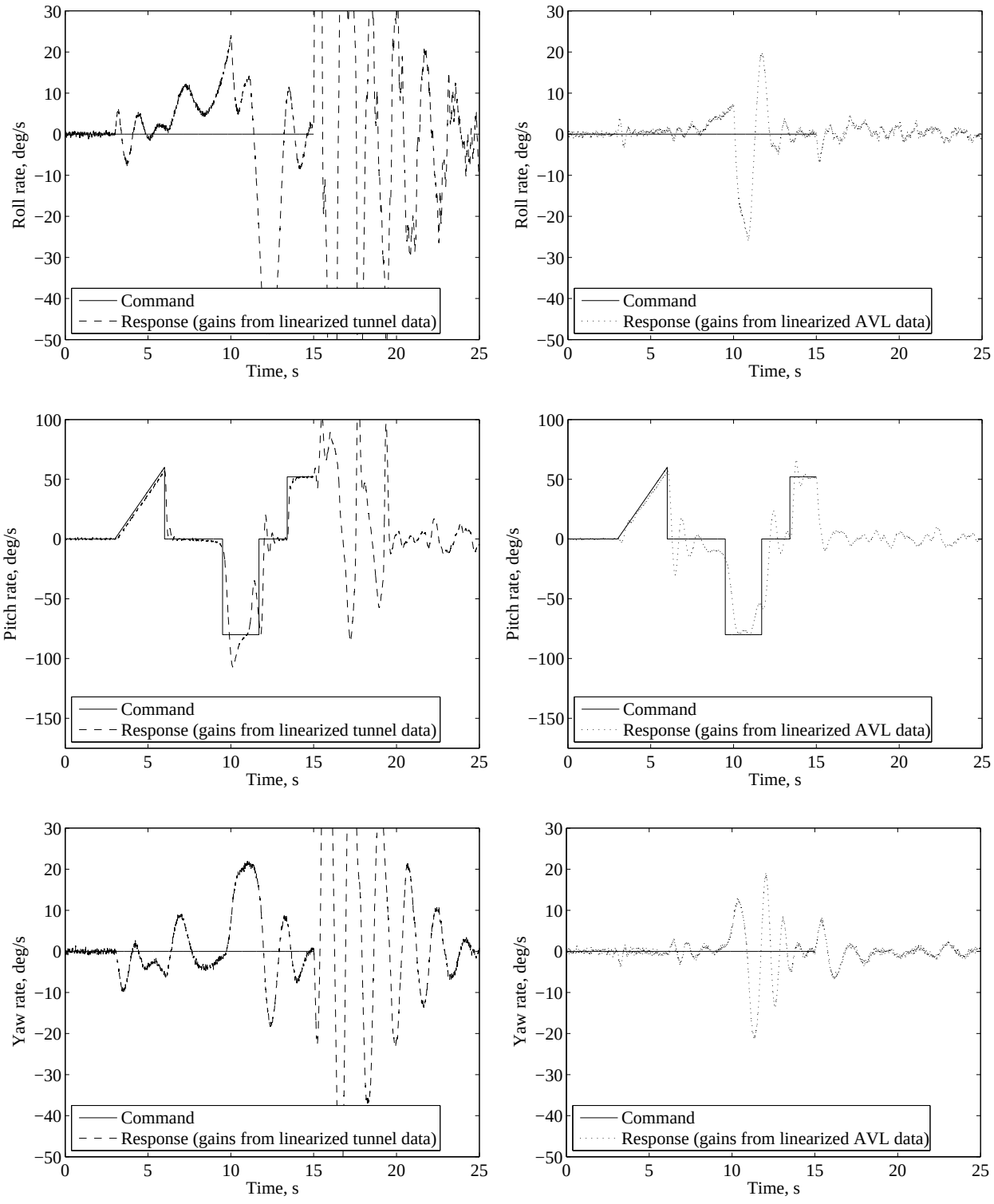


Figure 5.73: Angular rates during an autonomous tail slide with actuator models and measurement noise.

5.4 Gain Scheduled Maneuvers

For the results in this section, the gains for each controller were scheduled based on airspeed. Similar to the results in Sec. 5.3.2, actuator models were included in the simulation as well as measurement noise. The first results with gain scheduling are for the loop and are shown in Figs. 5.74 and 5.75. Comparing the angular rate plots to those of Fig. 5.56 reveals that for the gains from the tunnel data, the zero roll- and yaw-rate commands were tracked more closely with scheduled gains. This occurred between the five and eight second marks while the airspeed was greater than the trim speed. For both sets of gains, the initial oscillations in pitch rate suggest that the scheduled gains were more sensitive to the measurement noise. When the airspeed was below 64 ft/s (between the eight and 14 second marks), the angular-rate response tended to oscillate around the reference command for both sets of scheduled gains. The oscillations were not as bad for the gains from AVL data, however, and only lasted about three seconds. Despite the oscillations during the final seconds of the maneuver, the scheduled gains from AVL data resulted in a quicker recovery to the cruise altitude than the constant gains.

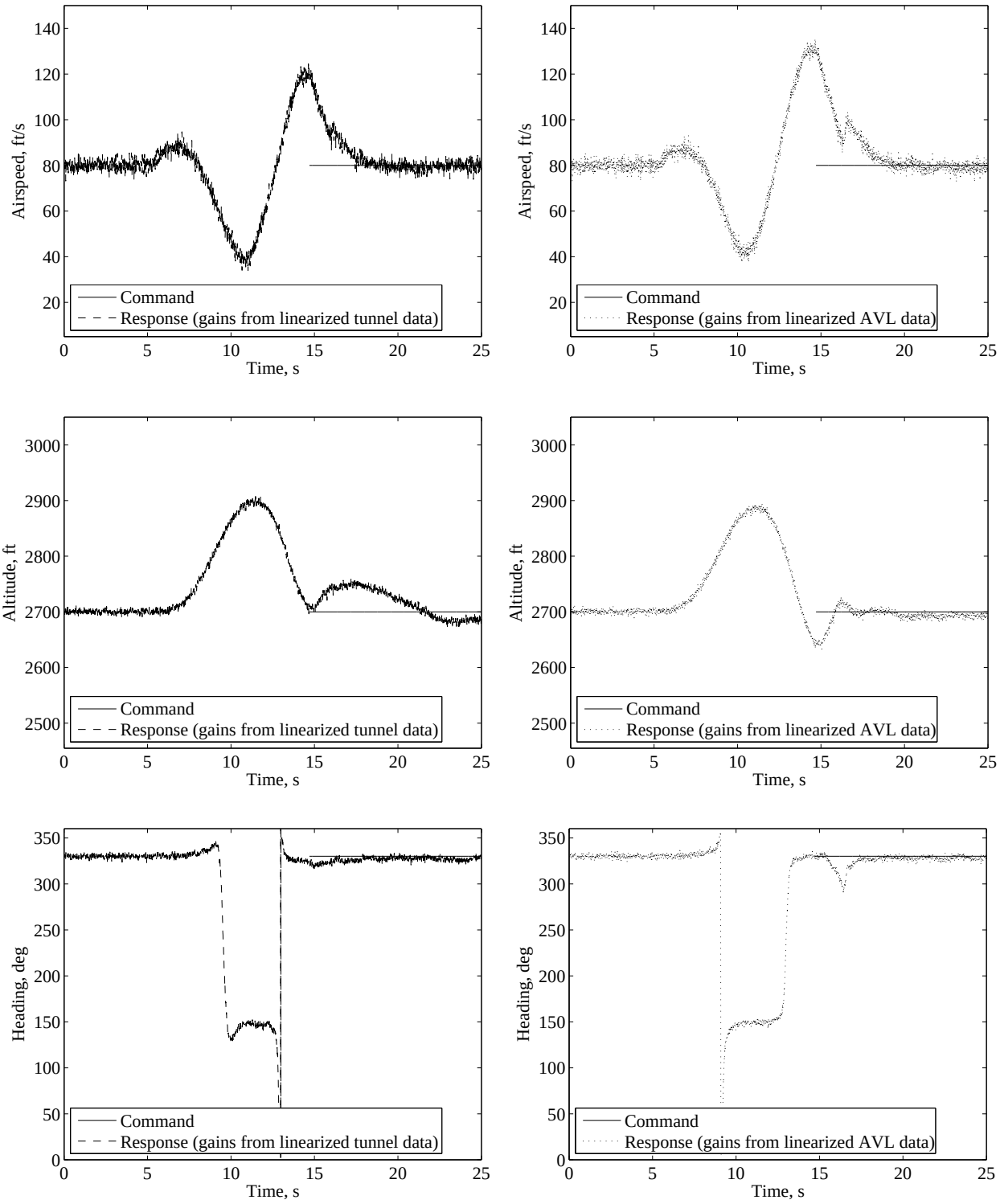


Figure 5.74: Simulation comparison during an autonomous loop with scheduled gains.

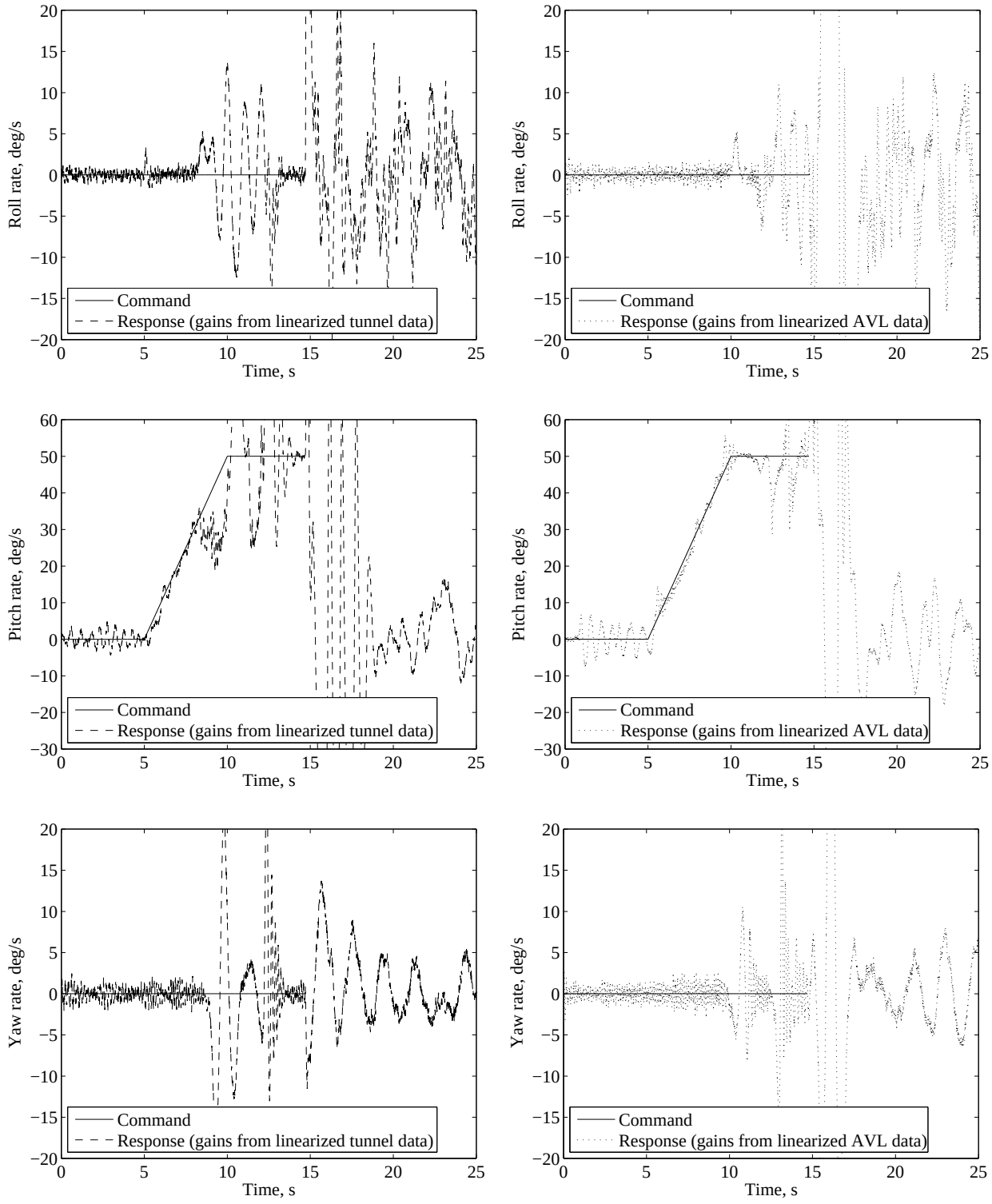


Figure 5.75: Angular rates during an autonomous loop with scheduled gains.

The simulation results for the roll maneuvers using scheduled gains are shown in Figs. 5.76 - 5.79. Comparing the angular rate plots to those of Figs. 5.59 and 5.60 reveals that the scheduled gains from the tunnel data were able to more closely track the roll- and yaw-rate commands. For the first time in any of the results presented, the sinusoidal yaw-rate command for both roll directions was closely tracked by the scheduled gains. The oscillations in roll and yaw rate visible when constant gains from the tunnel data were used are no longer apparent. The ramp in roll rate was also tracked with greater precision and there was no overshoot associated with the constant roll-rate command for both roll directions. The pitch-rate response, however, contained more oscillations while using the scheduled gains from the tunnel data than when constant gains were used. Examination of the airspeed plots shows that the better performance with the scheduled gains occurred after the airspeed had increased beyond the trim condition used to determine the constant gains. This result confirms intuition because as the airspeed rises, the constant gains are operating further and further from their design condition.

The same performance increase from the gain schedule was not experienced for the gains from AVL data, however. Comparing their angular rate plots to those of Figs. 5.59 and 5.60 shows that the tracking ability decreased when the gains were scheduled. While the scheduled gains were still able to complete the maneuver, the angular-rate responses contain more oscillations than the responses from the constant gains. This was especially true for the constant 100 deg/s roll-rate command where the constant gains were able to track the command perfectly while the scheduled gains resulted in oscillations of ± 10 deg/s. Similar to the constant gain result, the scheduled gains from AVL data were also not able to track the sinusoidal yaw-rate command.

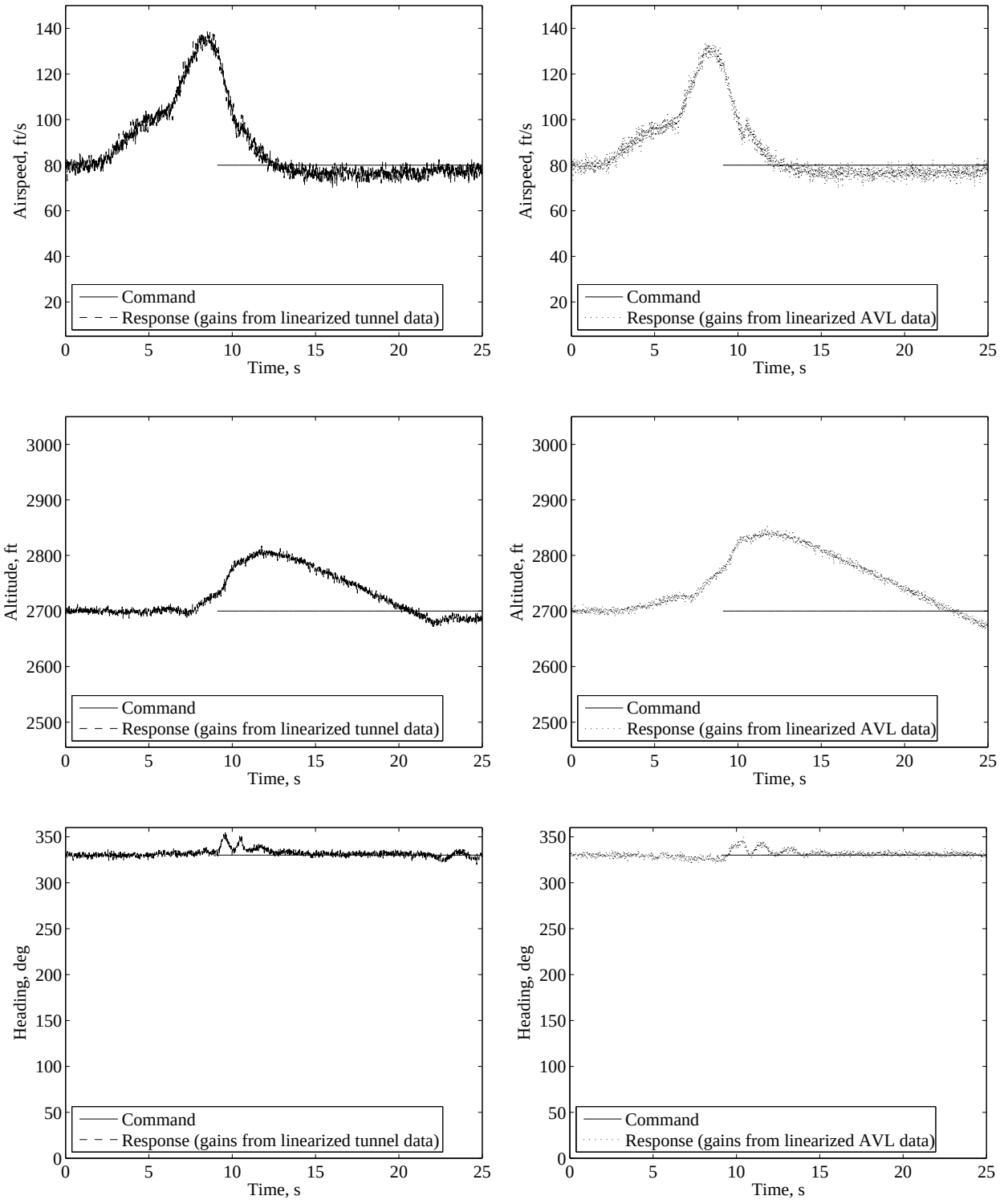


Figure 5.76: Simulation comparison during an autonomous left roll with scheduled gains.

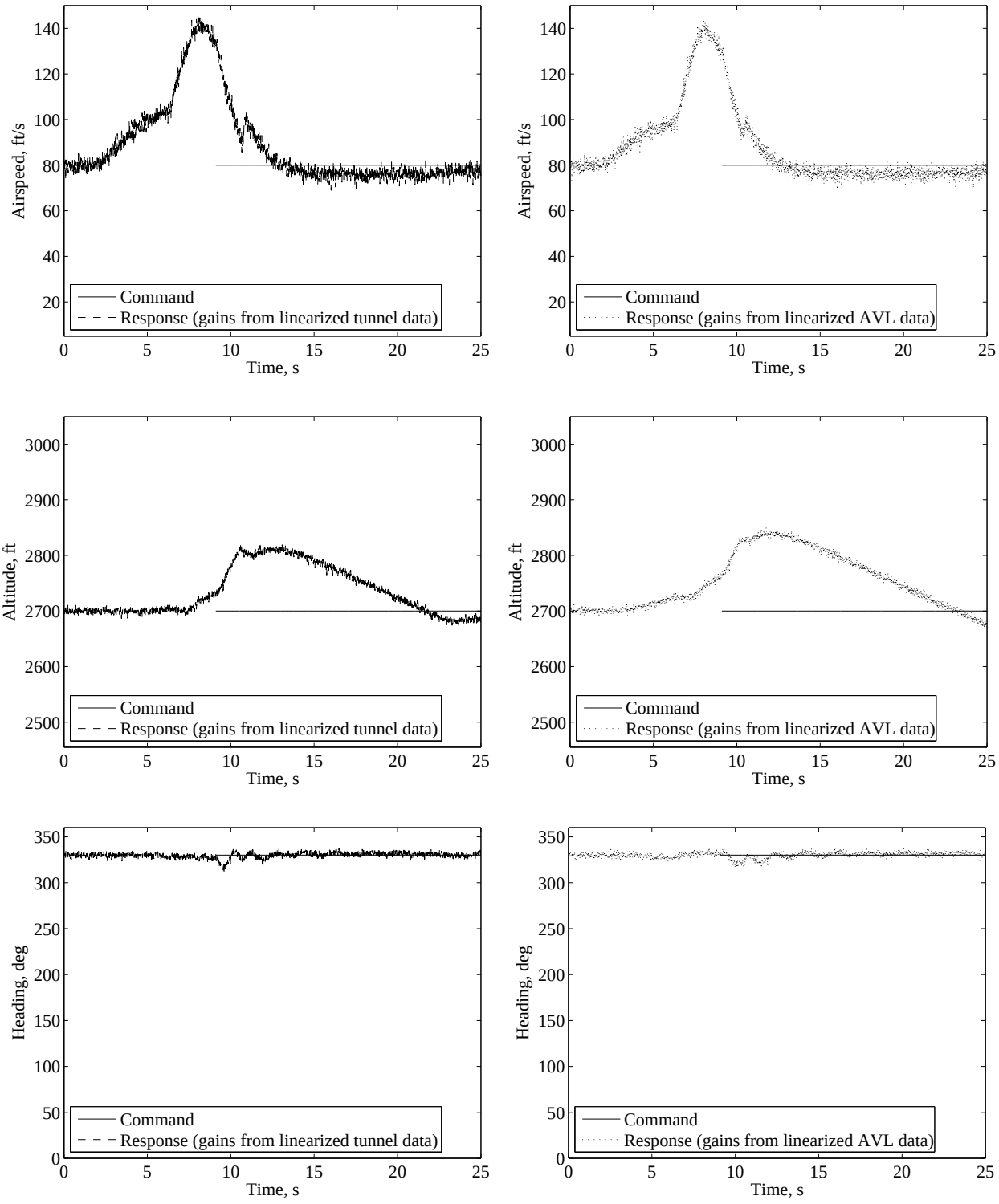


Figure 5.77: Simulation comparison during an autonomous right roll with scheduled gains.

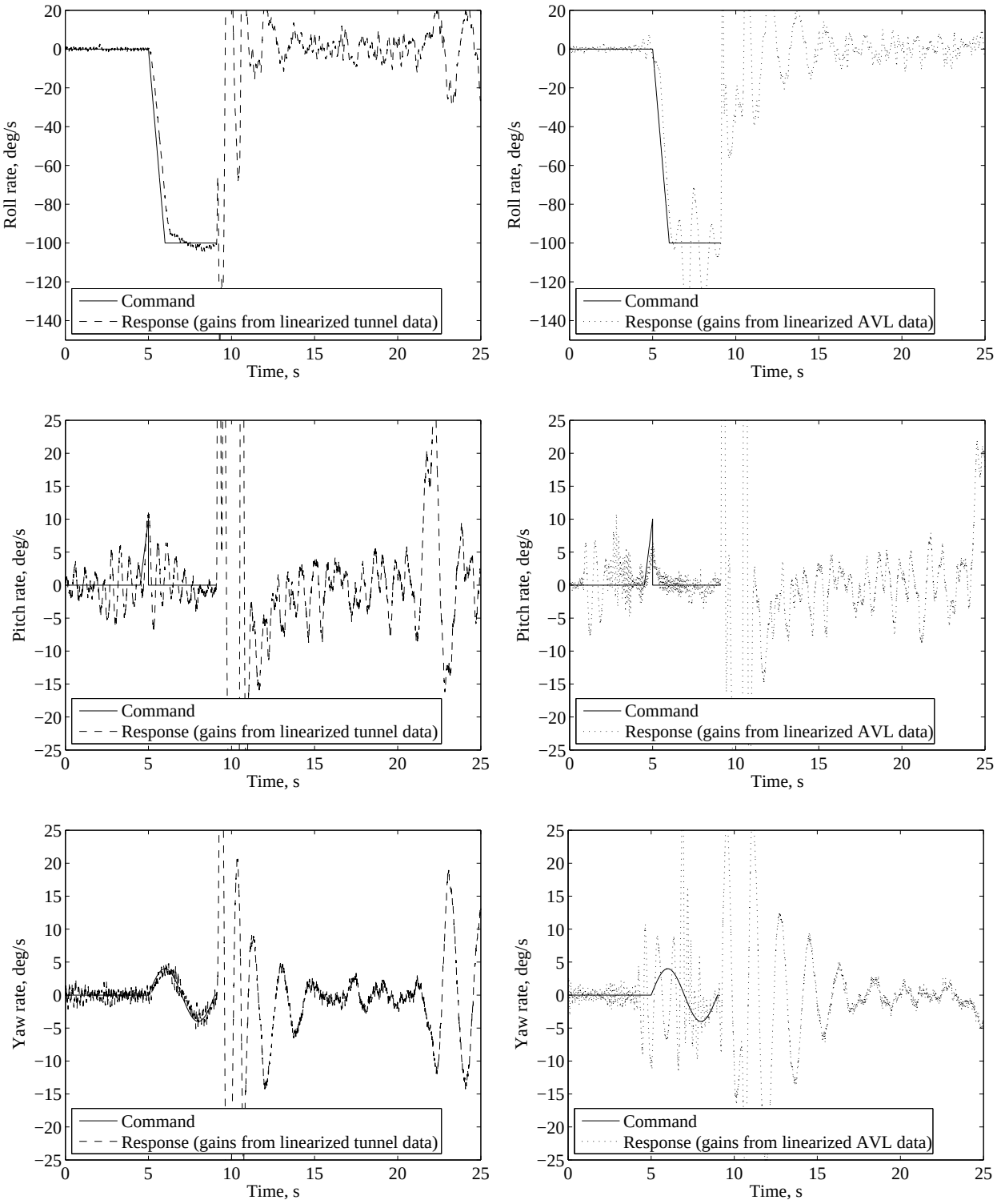


Figure 5.78: Angular rates during an autonomous left roll with scheduled gains.

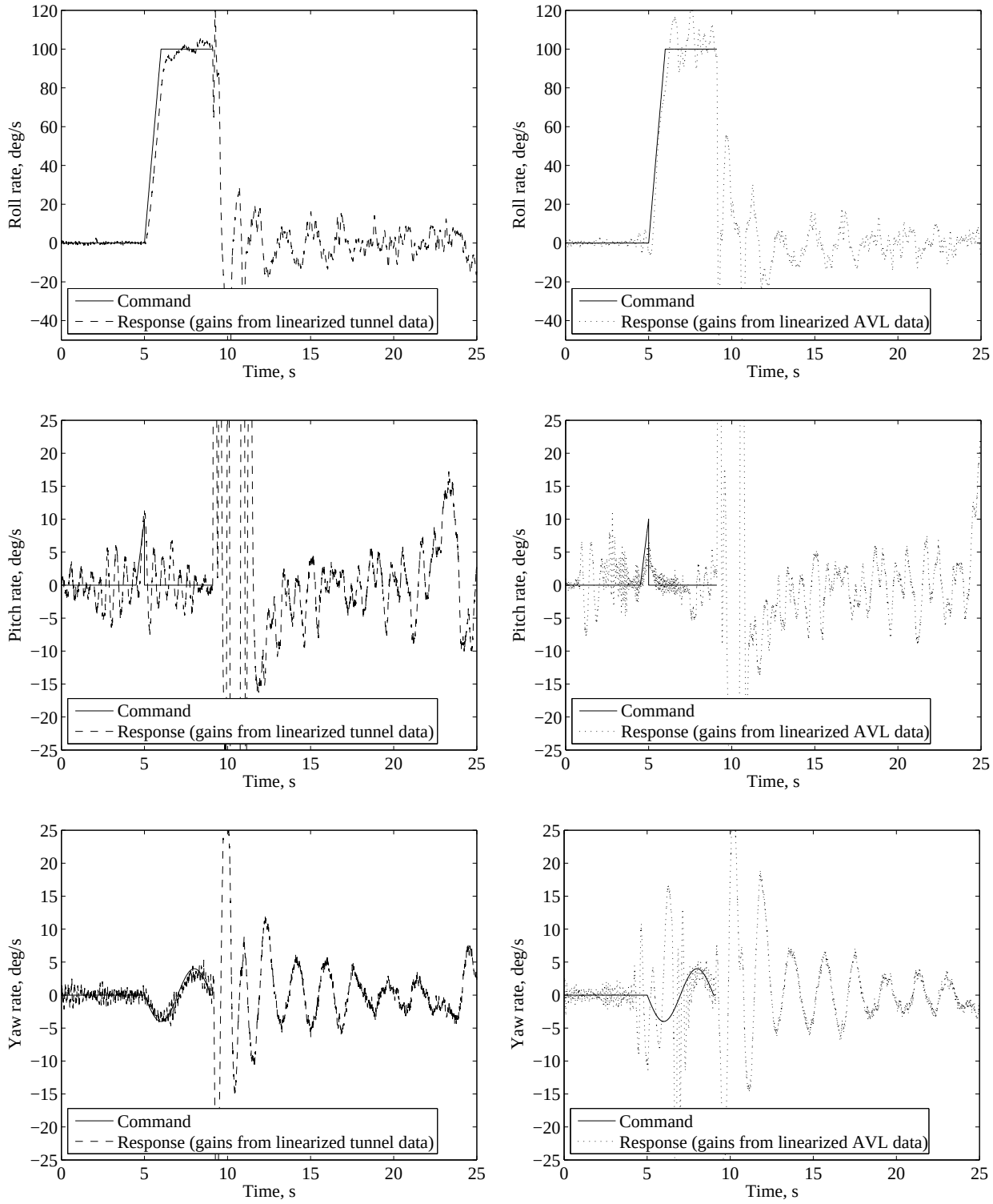


Figure 5.79: Angular rates during an autonomous right roll with scheduled gains.

The knife-edge simulation results using scheduled gains are presented next in Figs. 5.80 - 5.85. The roll angle plots are similar to Figs. 5.61 and 5.62 using both sets of constant gains. Comparing the angular rate plots, using the gains from the tunnel data, to Figs. 5.63 and 5.64 reveals that the scheduled gains were able to more closely track the roll- and yaw-rate commands during the initial portion of the maneuver. The pitch-rate response using the gains from the tunnel data did not change much with the gain schedule and neither did the angular rate responses using the gains from AVL data. During the knife-edge portion of the maneuver, both sets of constant gains were able to track the heading command just as well as the scheduled gains. The altitude tracking, however, improved slightly using the scheduled gains from AVL data for right knife-edge flight. Using constant gains the altitude decreased roughly 225 feet during a right knife edge, but the scheduled gains were able to keep the altitude loss to only 175 feet. The scheduled gains from AVL data were also able to reduce the steady-state error in airspeed and eliminate the error in altitude during the transition to level flight from both knife-edge directions.

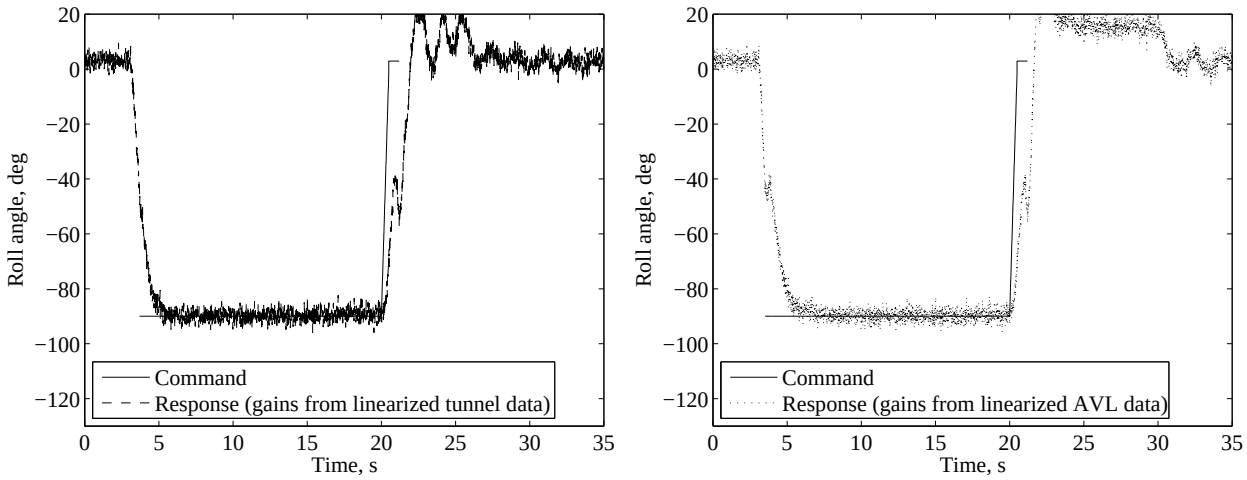


Figure 5.80: Roll angle during autonomous left knife-edge flight with scheduled gains.

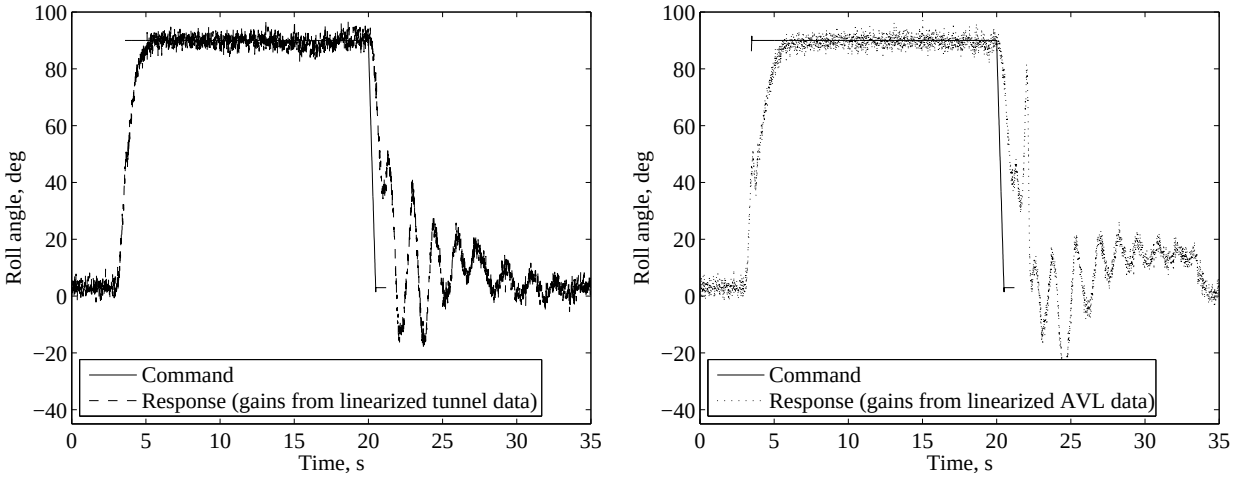


Figure 5.81: Roll angle during autonomous right knife-edge flight with scheduled gains.

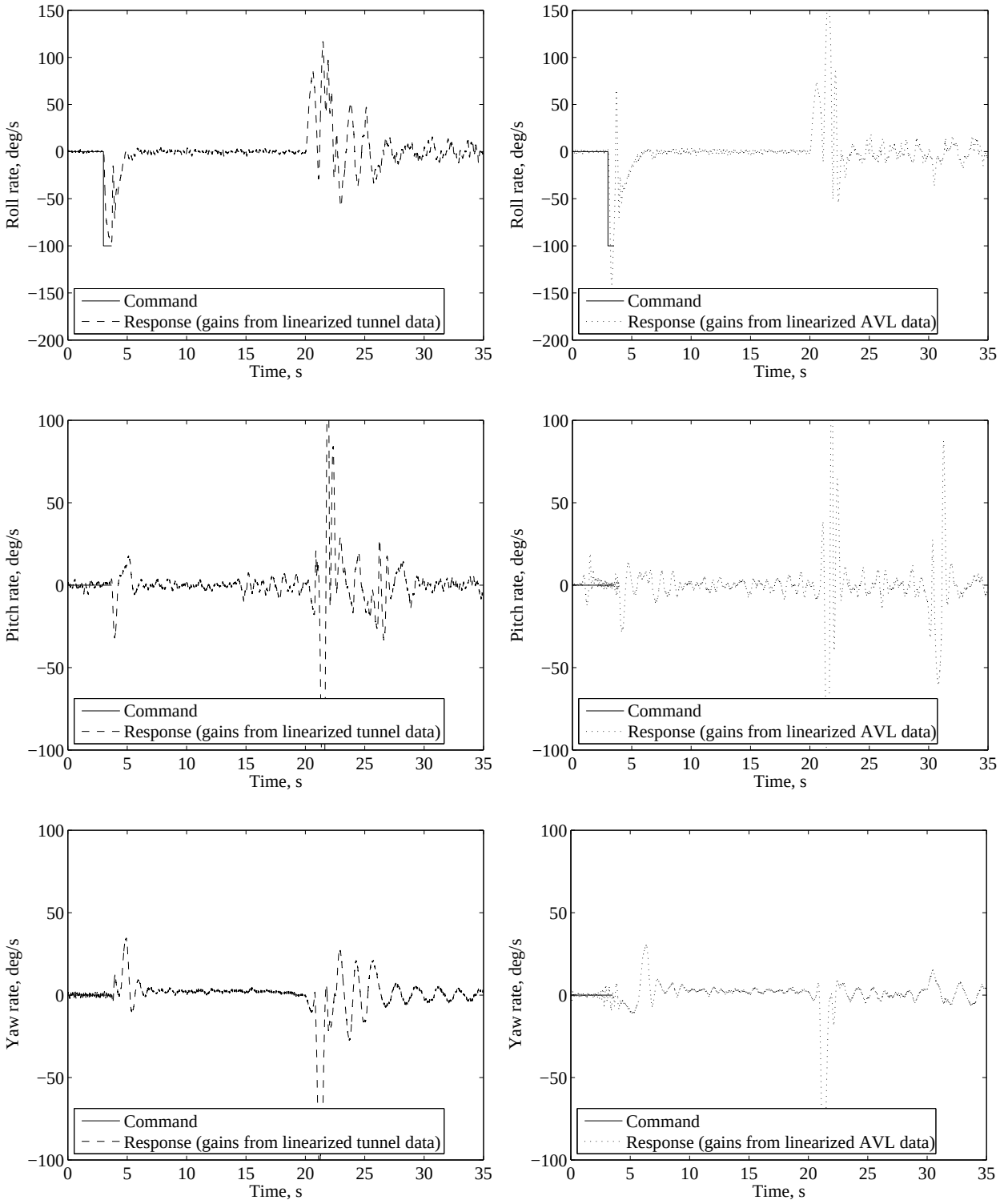


Figure 5.82: Angular rates during autonomous left knife-edge flight with scheduled gains.

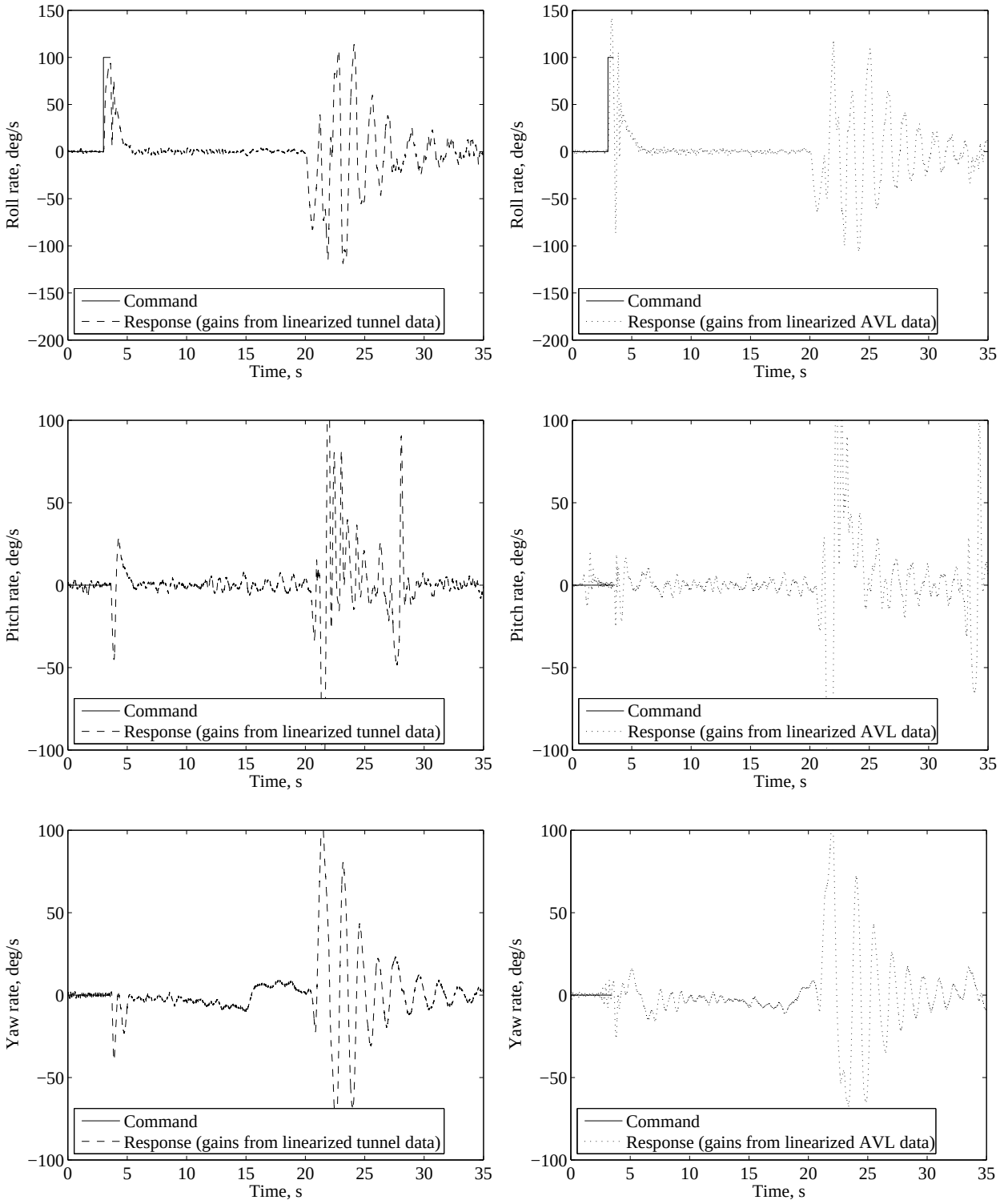


Figure 5.83: Angular rates during autonomous right knife-edge flight with scheduled gains.

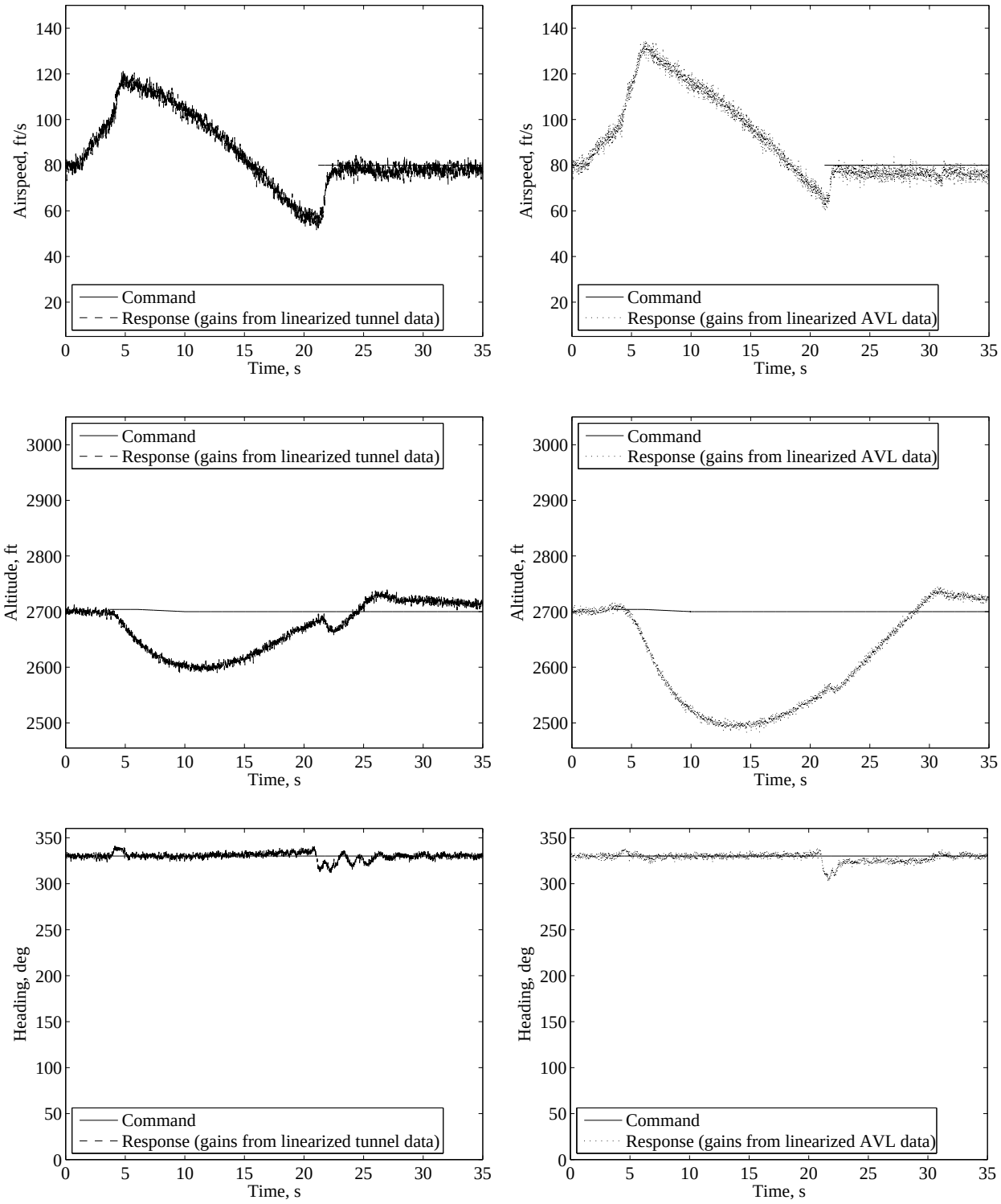


Figure 5.84: Simulation comparison during autonomous left knife-edge flight with scheduled gains.

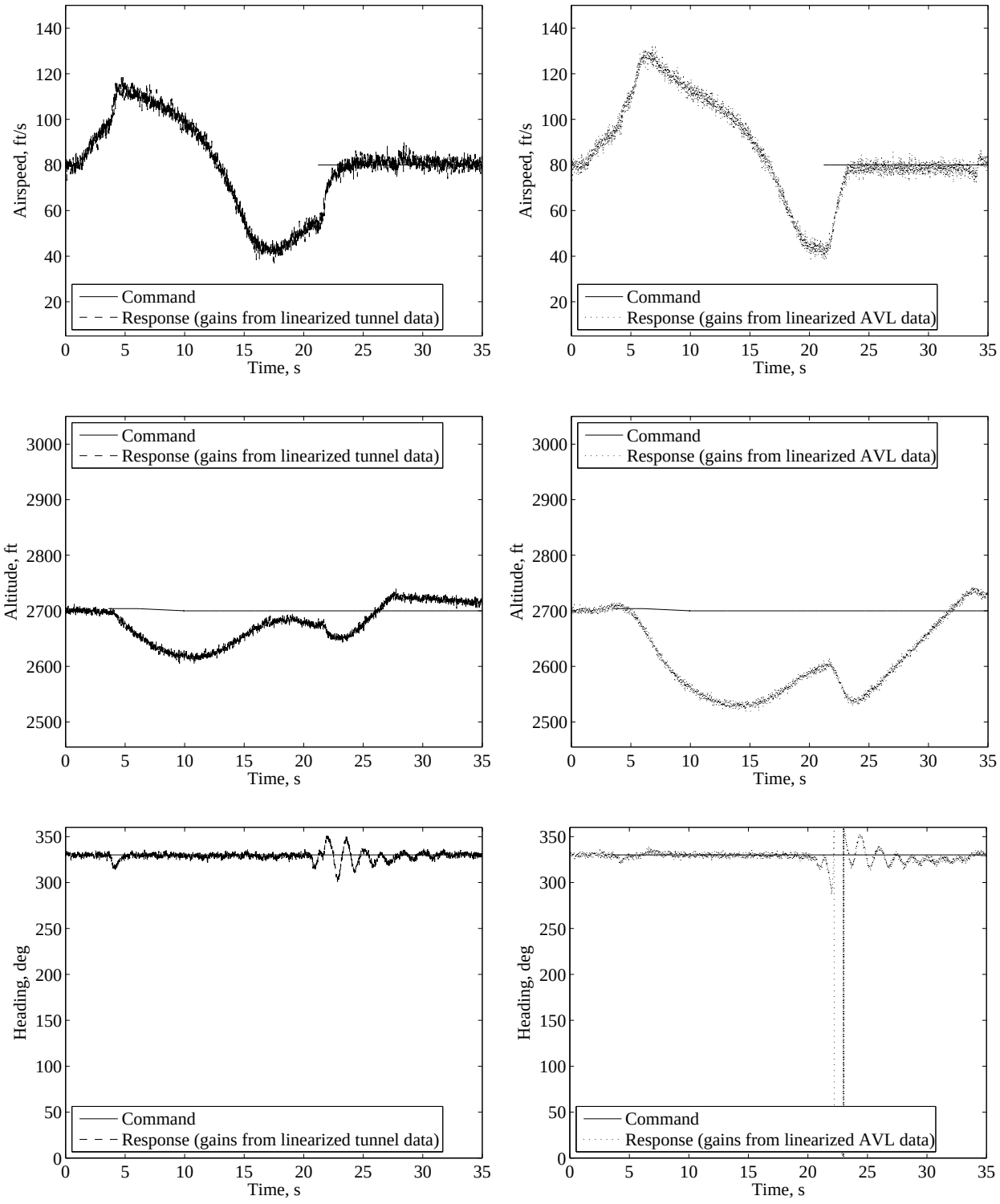
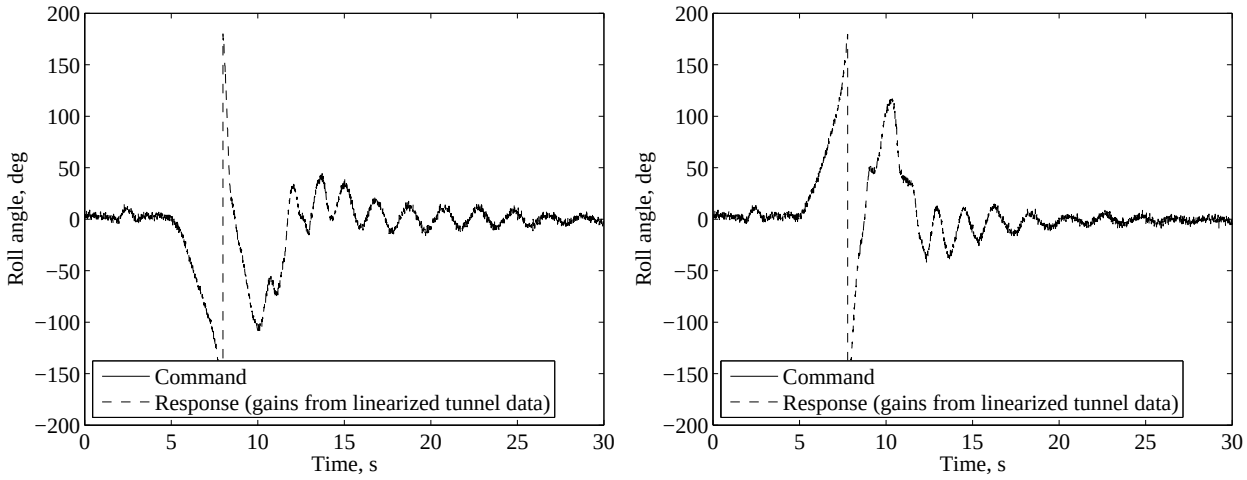


Figure 5.85: Simulation comparison during autonomous right knife-edge flight with scheduled gains.

Plots of the spin results are shown in Figs. 5.86 - 5.88 where the plots using the gains from AVL data are again excluded. Since the gains for the stall controller were not scheduled, the roll angle plots as well as the plots of airspeed and heading are similar to Figs. 5.67, 5.68, and 5.69 for both spin directions. During the spin, the roll-rate response using scheduled gains did not track the reference command as well as the constant gains. The pitch-rate response from the scheduled gains contained more oscillation than for the constant gains, but the average value remained closer to the command throughout the maneuver for both spin directions. When the gains were held constant, the response was steady but did not attain the same slope as the command. The yaw-rate response improved when the gains were scheduled and for left spins, the response was able to track the commands quite well. During right spins the yaw-rate response contained a few more high frequency oscillations than while spinning to the left, but the overall response was better than when the gains were held constant.



(a) Left spin

(b) Right spin

Figure 5.86: Roll angle during an autonomous spin with scheduled gains.

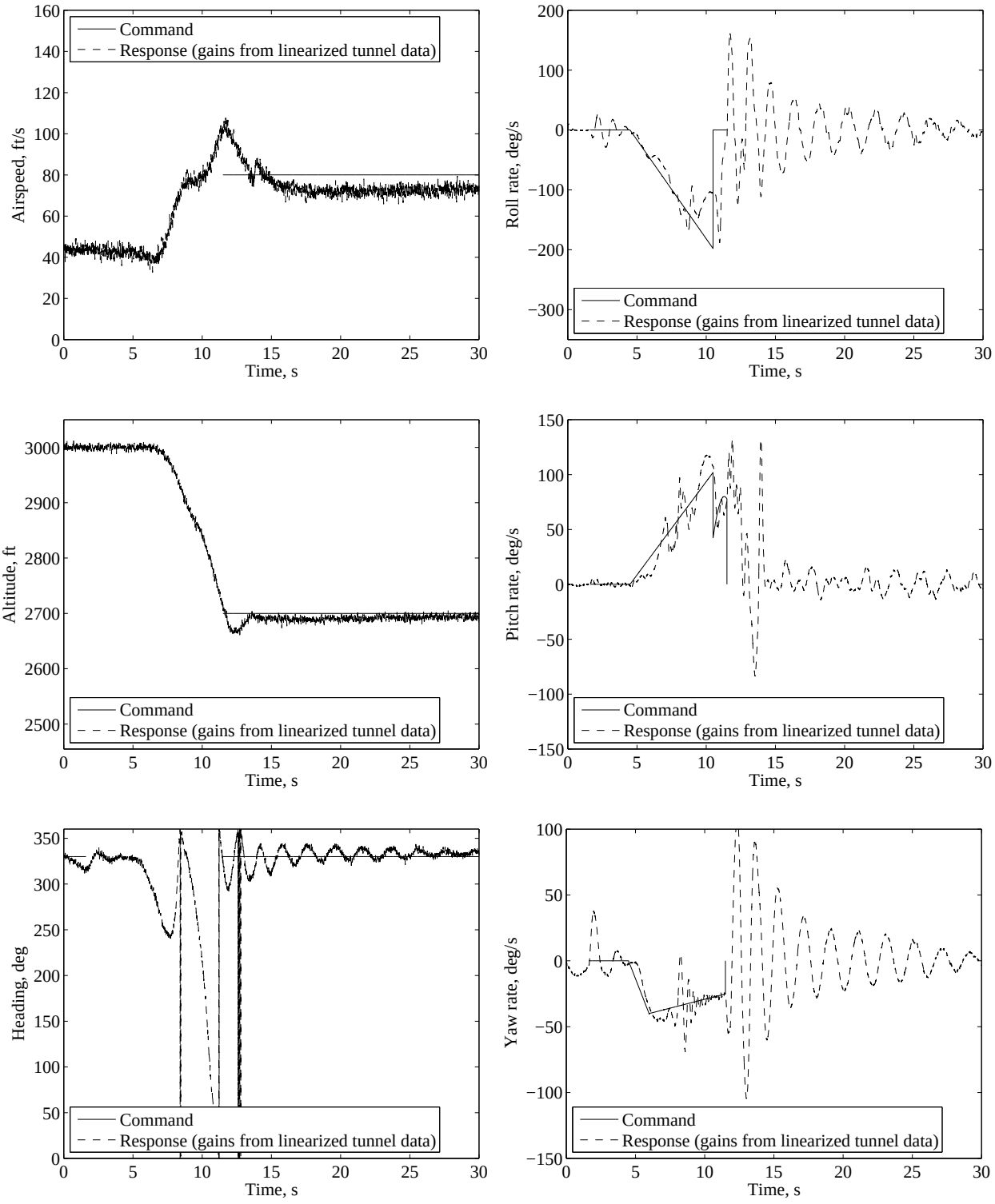


Figure 5.87: Simulation of an autonomous left spin with scheduled gains.

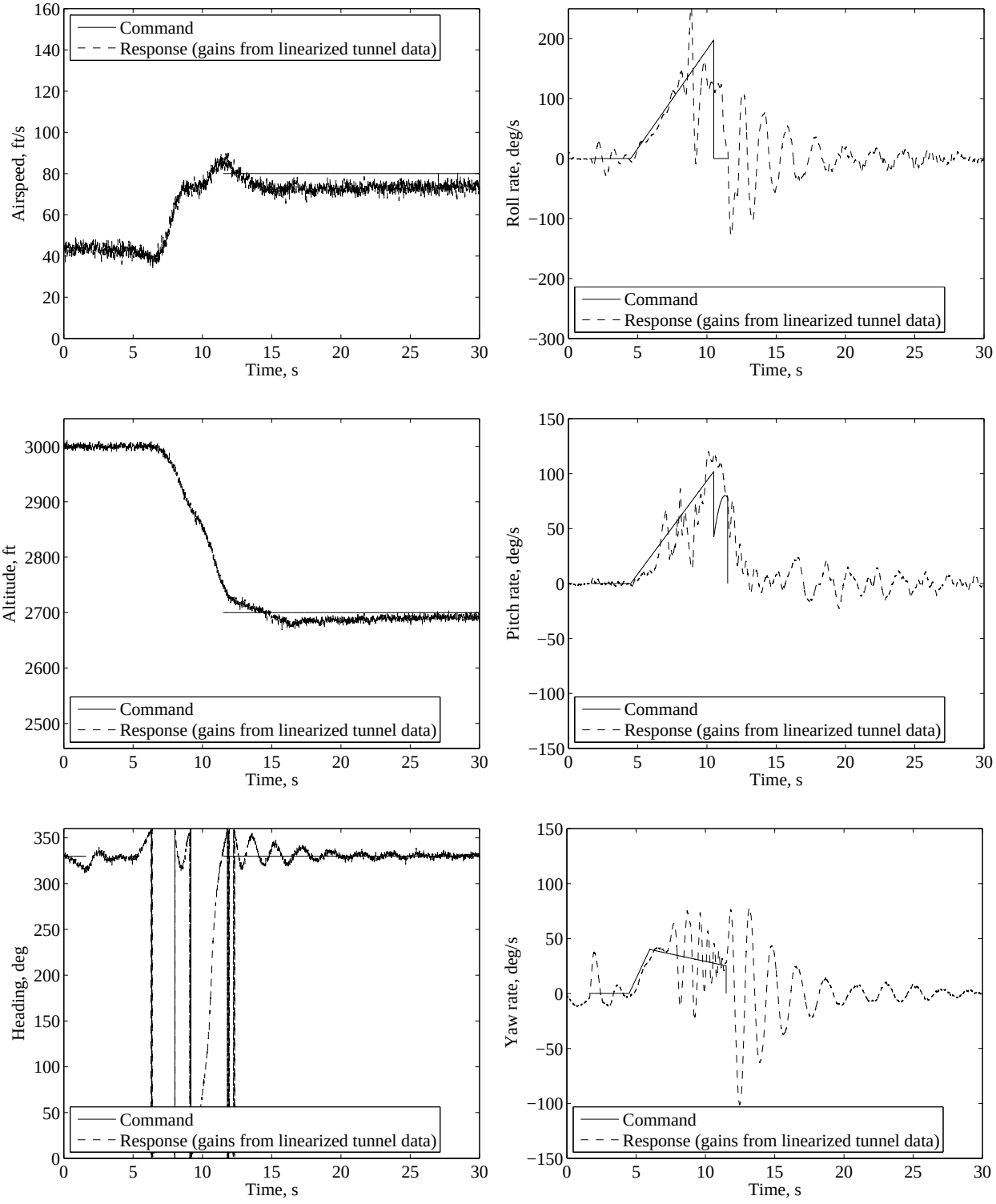


Figure 5.88: Simulation of an autonomous right spin with scheduled gains.

Figures 5.89 and 5.90 contain the results for the stall turn simulation with scheduled gains. Comparing the angular rate plots using the gains from the tunnel data to Fig. 5.71 shows that the roll-rate response was similar to the response from constant gains. The pitch-rate response with scheduled gains was slightly worse than with constant gains because the peak command was never reached and the response during the yawing portion of the maneuver was more oscillatory. The yaw-rate response, however, was better with scheduled gains because it contained less oscillation during the vertical portions of the maneuver and the initial step response to the yaw-rate command was quicker. In the case of the gains from AVL data, the angular rate responses were worse using the scheduled gains than when using constant gains. The scheduled gains produced more oscillation in the response for both the roll rate and the yaw rate during the second half of the maneuver. The pitch-rate response was not as bad, but did oscillate more than when using constant gains. The most notable difference in the pitch-rate response occurred at the top of the vertical climb while the airspeed was at its lowest value. Despite the difference in the angular-rate responses, the plots of airspeed, altitude, and heading for both sets of scheduled gains are similar to Fig. 5.70 and the control system was still able to complete the maneuver and return the aircraft to level flight.

The final results appear in Figs. 5.91 and 5.92 and correspond to the tail slide. Similar to the results from the stall turn, the plots of airspeed, altitude, and heading resemble the plots in Fig. 5.72 for the constant gains. Comparing the angular rate plots to those of Fig. 5.73 reveals that the scheduled gains from the tunnel data resulted in better roll- and yaw-rate responses during the vertical climb portion of the maneuver. The pitch-rate response was also better with the scheduled gains because the overshoot in the nose-down command was reduced and the settling time for the zero command during the vertical descent was shorter. In the case of the gains from AVL data, all of the angular rate responses were worse with the scheduled gains than with the constant gains. The roll-rate response using the scheduled

gains was similar for the first half of the maneuver, but quickly starting oscillating with higher frequency and amplitude than the response using constant gains.

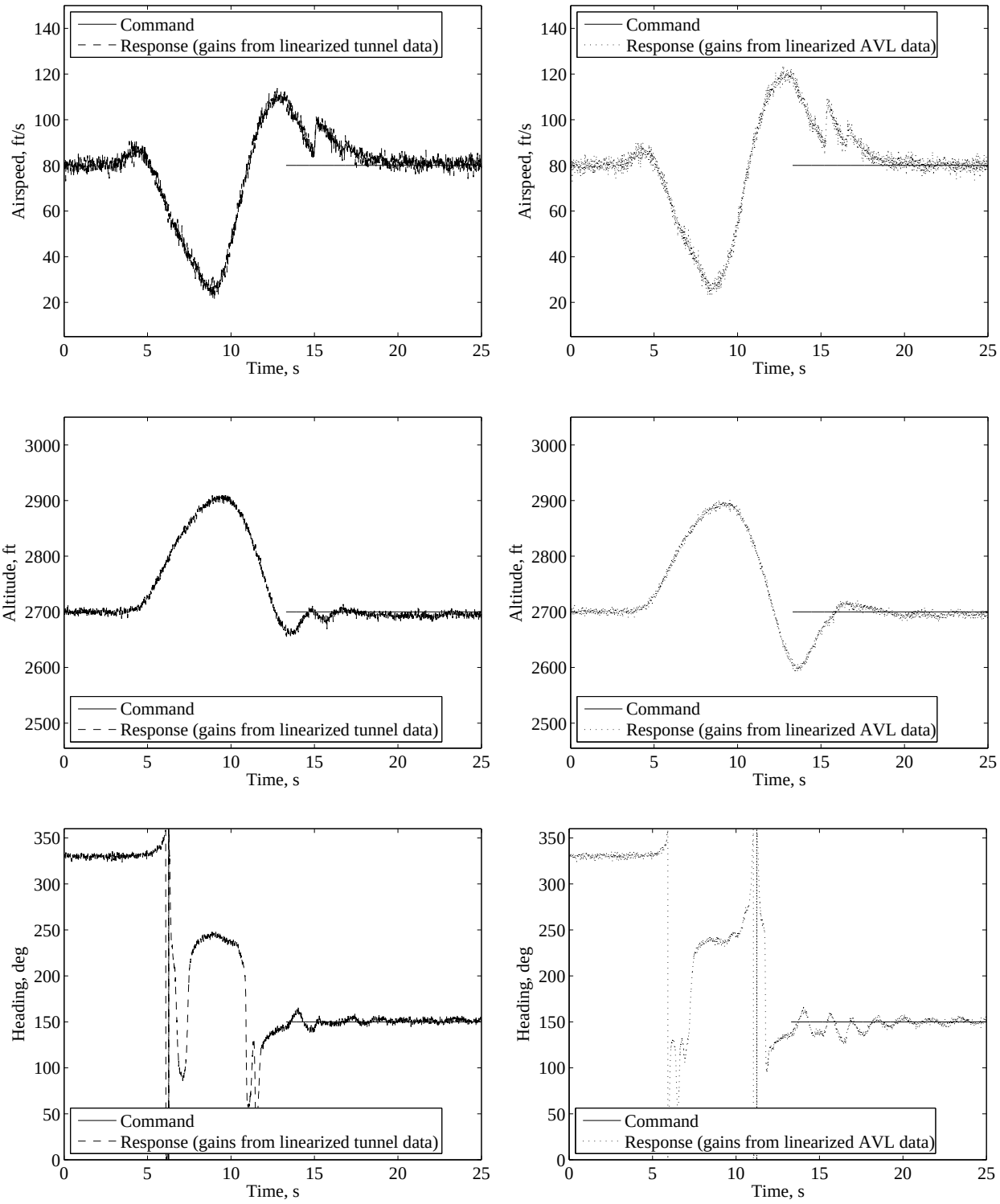


Figure 5.89: Simulation comparison during an autonomous stall turn with scheduled gains.

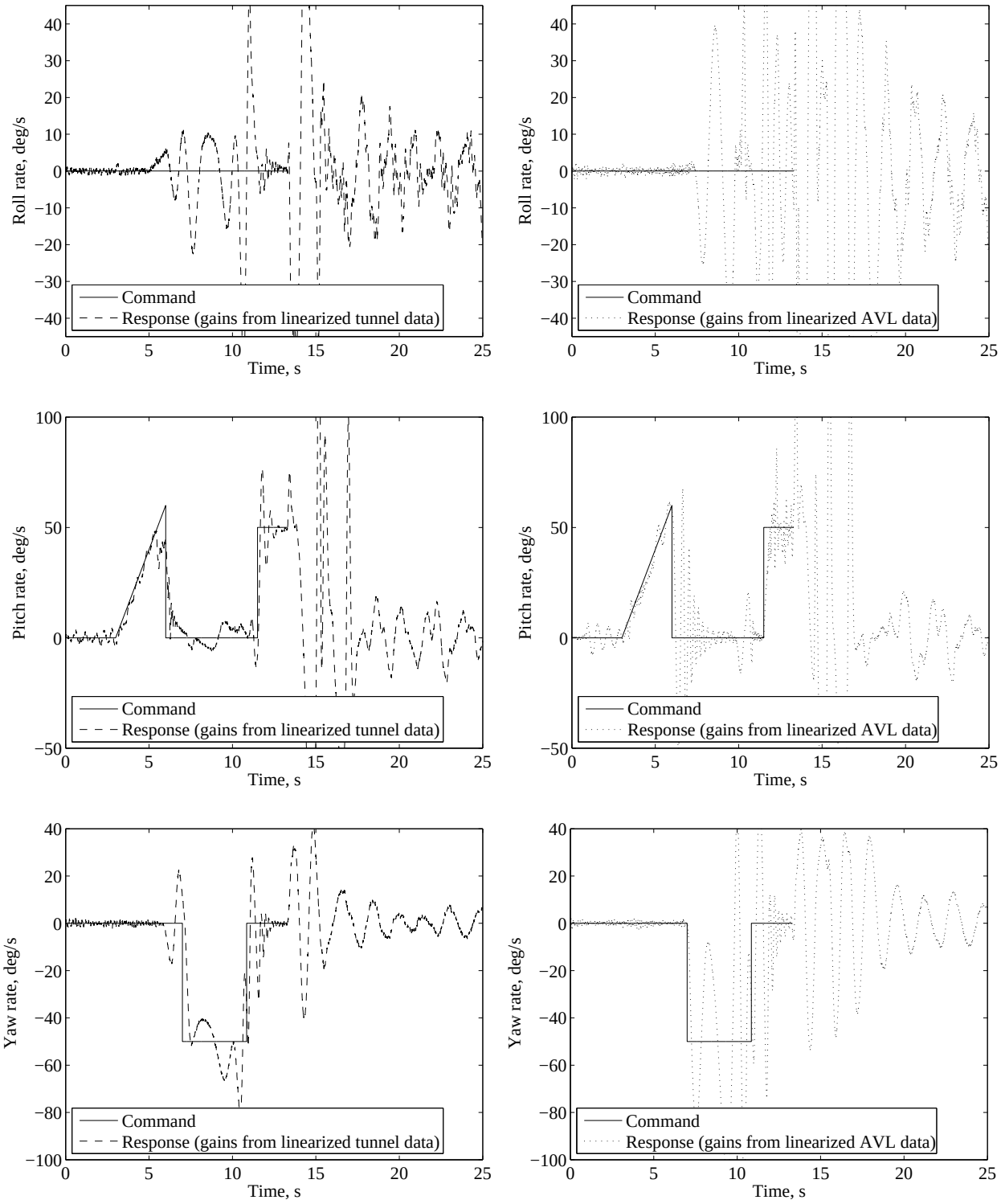


Figure 5.90: Angular rates during an autonomous stall turn with scheduled gains.

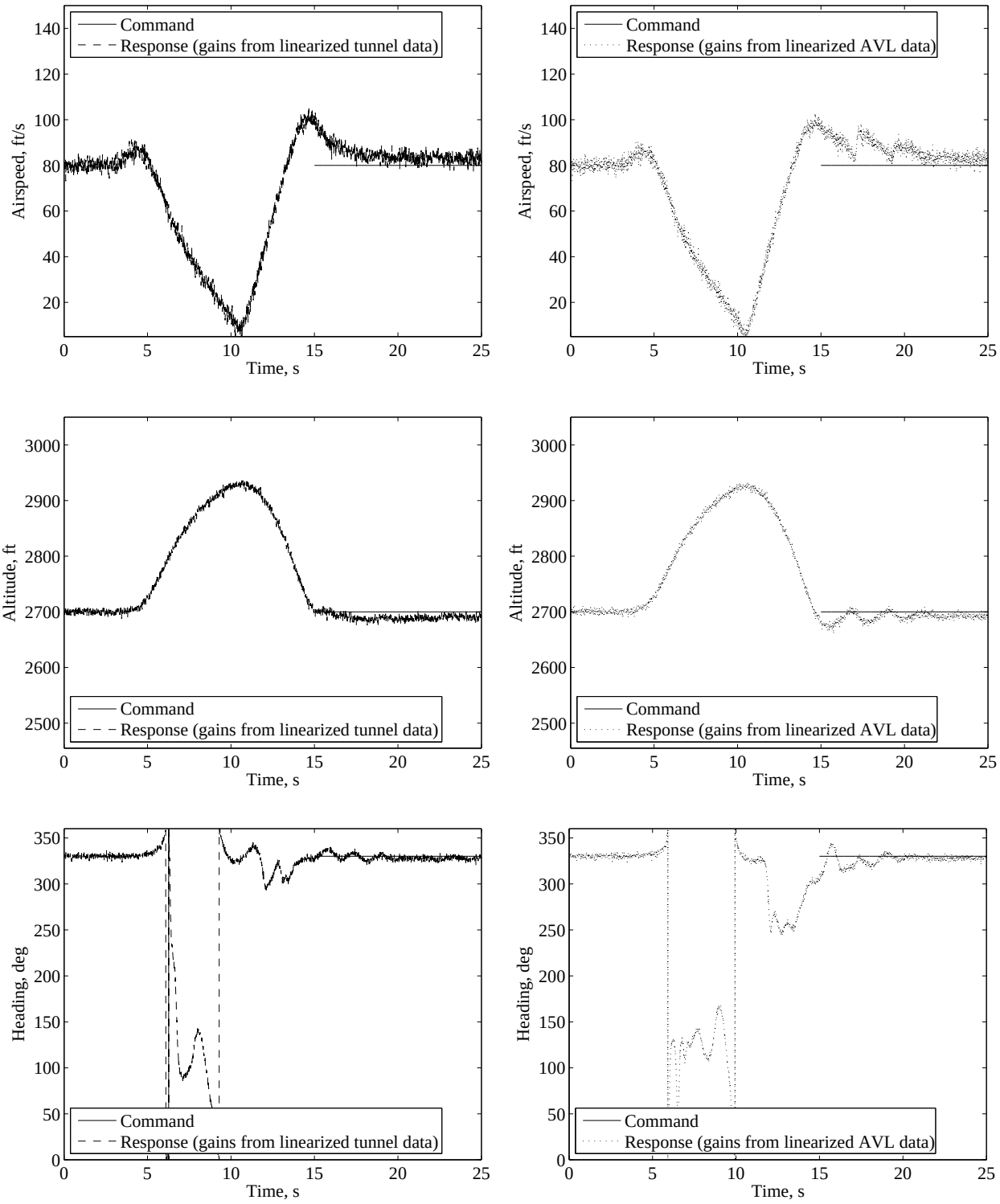


Figure 5.91: Simulation comparison during an autonomous tail slide with scheduled gains.

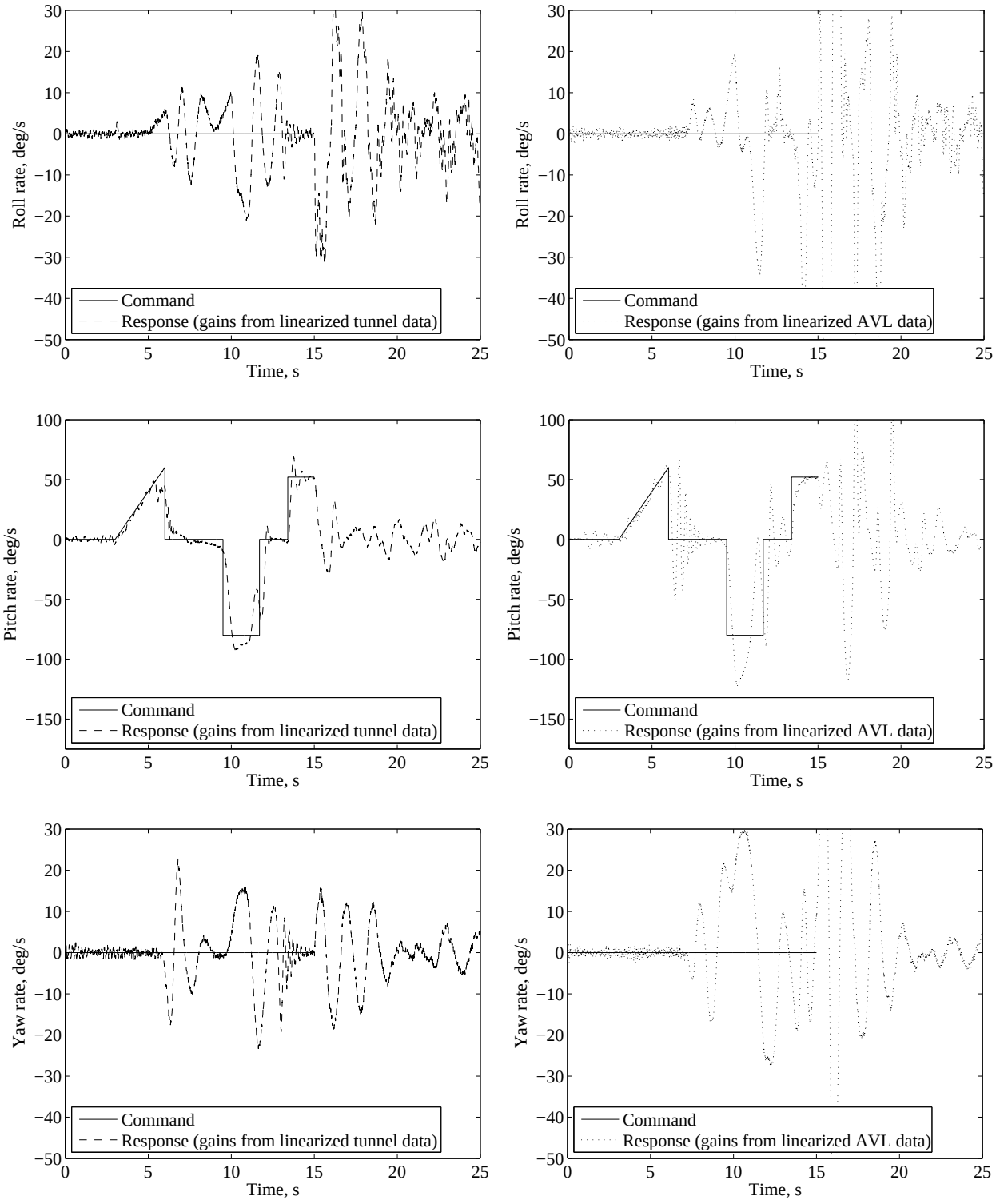


Figure 5.92: Angular rates during an autonomous tail slide with scheduled gains.

Chapter 6

Accomplishments and Future Development

6.1 Summary

Throughout this dissertation, a control system to autonomously perform aerobatics with a UA was developed. The conventional approach of path-following guidance for maneuver control of a fixed-wing aircraft was replaced with a pilot-inspired approach in which recorded state time histories of pilot-controlled aerobatics were used to develop the necessary control commands. Focus was placed on developing a control system to perform autonomous aerobatics that did not require the generation of a flight path and relaxed the constraint on the exact position of the aircraft. For level flight control as well as transitions between maneuvers, a conventional autopilot based on tracking airspeed, altitude, and heading was employed.

To test the developed control system, a nonlinear simulation was created that incorporated both static and forced oscillation wind tunnel data which included thrust and control surface deflection effects. An electric engine model was created based on the model of a similar engine presented in [73] to represent the propulsive forces and moments on the aircraft. Moments from the engine torque, gyroscopic effects associated with a rotating propeller and engine, and the position of the thrust line relative to the CG were included. A quaternion attitude description was used in the nonlinear equations of motion for the simulation to avoid the singularity associated with Euler angles that occurs when the pitch angle is 90 degrees. The quaternion was initialized at the start of the simulation from Euler angles since the quaternion rotation axis is not obvious for compound rotations consisting of yaw, pitch, and roll combined. Euler angles were also used in the feedback path of the control system because of their intuitive nature. Control actuators were modeled in the simulation as simple lags

to test the effects of a time delay between when a control surface deflection command was sent and when the output was achieved. Sensor models were included in the simulation to determine the effects of having imperfect state measurements in the feedback path of the control system. Zero-mean, Gaussian pseudo-random noise was added to the aircraft states used as feedback in the control structure based on manufacturer's published data for a commercially available gyroscope, accelerometer, magnetometer, and multi-port pressure probe. An open-source flight simulation environment was used to visualize the aircraft motion and to verify each simulation was performing as expected.

For cruise flight and transitions between maneuvers, proportional-integral-derivative (PID) control was used to maintain a desired airspeed, altitude, and heading. Autonomous aerobatic maneuvers consisting of loops, rolls, spins, stall turns, and tail slides were controlled by an angular-rate controller whose commands were based on pilot-controlled data. Knife-edge flight was performed with a mode-switching mechanism between an angular-rate controller and an one that maintained a desired roll angle, altitude, and heading. Proportional-integral (PI) control was used in the angular-rate controller in an attempt to minimize the steady-state error possible with proportional only control. An open source flight simulator was used to record state time histories of pilot-controlled aerobatics to determine the command inputs necessary to complete the maneuvers.

Two sets of linear aircraft models were used during the gain tuning process. The first set of models was found by numerically linearizing the nonlinear simulation about a specified trim condition. The second set of models was found from the fully-coupled, linearized perturbation equations of motion which were developed in conjunction with this research. Along with the linearized perturbation equations, linear aerodynamic data from a vortex lattice method was used to estimate the stability derivatives of the aircraft. These two approaches (state-space models found by linearizing the simulation and models found by substituting the stability derivatives from a vortex lattice method into the perturbation equations) were used to demonstrate the effects of inaccurate aerodynamic data on the overall performance

of the control system. This was considered an important comparison as wind tunnel data for new test platforms is costly to obtain and higher fidelity computational aerodynamic tools can require long computational times. The lack of propeller and fuselage modeling abilities with vortex lattice methods, however, led to an investigation based on previous work by the author and a colleague [75] as to the best way to include the aerodynamic effects of both items in the model. The level of accuracy of the aerodynamic data when using the fuselage model and when including the destabilizing effect of the propeller was examined and compared to the wind tunnel data of the test aircraft.

The control gains for the cruise flight controller as well as the maneuvering flight controllers were calculated using an infinite-horizon optimal control technique known as a linear quadratic tracker (LQT). The simulation results for the cruise flight controller using the optimal gains were compared to results using gains determined from the classical root-locus method to show the time and performance advantages afforded by employing a well-studied optimal control technique. The LQT technique allowed multi-input/multi-output (MIMO) control systems to be designed quickly with guaranteed closed-loop stability and good time responses. The difference between the LQT and the popular linear quadratic regulator (LQR) was discussed with the LQT regulating the tracking error (the difference between the command and the performance output) to zero as opposed to regulating the perturbations in the states to zero. Multiple performance criterions were discussed and a modification to the standard cost function in the form of time weighting was shown to reduce the number of design variables while correcting any observability issues with the weighting matrices. The system matrices to apply the LQT technique to cruise flight with a compensator of desired structure were derived and a description of how to apply the method to maneuvering flight control design was given. A gain scheduling technique based on airspeed was provided and used to show how the use of multiple linear models throughout the maneuver affected the overall performance of the autopilot.

6.2 Conclusions

The following conclusions are drawn from the derivations and simulations performed in this work.

- The importance of a detailed parasite drag estimate became apparent while performing a comparison of vortex lattice modeling techniques for propeller driven airplanes. Modeling the fuselage in AVL as a combination of vertical and horizontal lifting surfaces was observed to more closely predict the stability derivatives C_{Y_β}, C_{L_α}, C_{m_α}, and C_{n_β}. Using a similar approach to model the propeller was shown to improve the estimate of the pitch stiffness (C_{m_α}). The improvement in these quantities is important as the lift-curve slope and pitch stiffness are the two most important longitudinal stability derivatives and the yaw stiffness is the second most important lateral stability derivative [21]. The lift-curve slope is important because it affects the damping of the pitching motion due to disturbances as well as the maneuverability of the aircraft. The pitch stiffness is important because it determines static stability and affects the dynamic pitching motion. The yaw stiffness is associated with the weathercock stability and plays a large role in the aircraft dutch roll mode. Error in these stability derivatives can result in incorrectly optimized control gains due to a falsely stable open-loop aircraft model.

- Pilot-controlled simulation data revealed that a pilot's commands during most aerobatic maneuvers are similar to those of an angular-rate controller. Also, it was determined that the commanded rates to perform most maneuvers could be represented by piecewise constant or piecewise linear functions of time. The fact that a pilot's commands are similar to an angular-rate controller is a nice convenience since the feedback in the controller is in the body axes. Since the measured outputs are in body axes and the aircraft control surfaces act in the body axes, the controls do not change with aircraft attitude. This means that the same control loops can be used

despite the orientation of the aircraft. This same idea could also be used to control body axis accelerations without regard to aircraft orientation and is in fact the basis of most path-following guidance schemes [12]. It was discovered that, due to the mostly feedforward nature in which a pilot controls aerobatic maneuvers, the aircraft states of airspeed, altitude, and heading are largely repeatable for maneuvers that begin and end at the same trim condition. This repetition allows those states to be used at the end of a simulation to confirm that the aircraft completed the specific maneuver that was commanded.

- An optimal gain tuning approach, based on linear quadratic methods, was applied to both normal and maneuvering flight control design and was shown to provide better and more repeatable results than the classical root-locus method. The technique of combining the coupled aircraft dynamics with actuator models and compensator dynamics of a desired form was derived and used during the gain optimization. A modification to the standard performance criterion in the form of time-weighting was shown to yield good results with only one design parameter. The closed-loop system, using the gains from a time-weighted performance criterion, was determined to be robust to imperfect state estimates by including measurement noise in the feedback path during the simulations. The most important discovery in regard to the application of the LQT technique was in selecting the initial stabilizing gain matrix. It was found that not only must the initial gains stabilize the closed-loop aircraft, but they must provide that stability with an adequate gain margin. If one or more of the poles are only marginally stable, a slight change to the control gains during the optimization process could cause one of the nested Lyapunov equations to not have a unique solution. This would occur if the eigenvalues α_1, α_2, ..., α_n of the system state matrix $\mathbf{A}$ and β_1, β_2, ..., β_n of the system input matrix $\mathbf{B}$ do not satisfy $\alpha_i + \beta_j \neq 0$ for all pairs (i, j) [91]. If this happens, the gain optimization fails and the initial gain matrix should be suspected as the cause. The second discovery made while implementing the LQT

technique is that the approach is inherently sensitive to the inclusion of states that are not controlled by the feedback structure. This means, for example, that if altitude and heading are not being controlled, those states along with the corresponding rows and columns of the state-space matrices should be removed before minimizing the cost function. Another revelation was that any time control gains for throttle control were optimized, a good linear engine model was required in order to determine gains that would stabilize the closed-loop aircraft. In terms of the actual minimization of the performance criterion (remember this work employed MATLAB's `fminsearch` function) another lesson learned was that the value of the cost would stop decreasing after the first few iterations of the optimization. The workaround used to overcome this issue was to successively restart the minimization using the previous gain values and then stop the function calls once progress was no longer made by restarting. This issue is one that could be explored in more detail in the future.

- A performance comparison was conducted to determine the effects of using vortex lattice data instead of wind tunnel data to design maneuvering control systems with an infinite-horizon optimal control approach. It was determined that vortex lattice data can be used to obtain aircraft models for control development as long as models at low airspeeds are not needed. Due to the nature of the vortex lattice method, the stability derivative estimates for flight regimes that have the potential for flow separation will contain considerable error. Despite the errors introduced at low speeds, it was discovered that vortex lattice data can be used to calculate control gains for both cruise and maneuvering flight as long as the gains are held constant. The limit to using vortex lattice data occurs at the extremes of the flight envelope. Good results were not obtained with the aerodynamic modeling of flight conditions involving large angles of attack or large sideslip angles and extremely slow speeds were also found to be error prone. The use of vortex lattice data for maneuvering flight control development is, however, a huge discovery for small businesses and independent research firms as well

as research institutions because the acquisition of wind tunnel data can be expensive and requires a considerable amount of time. Using the methods derived in this work, it is the author's belief that the same technique could be applied to multiple test aircraft platforms in a reasonable amount of time without much cost. This should allow for an increase in the study of maneuvering flight control which could be a benefit to both military and civilian aviation whether the vehicles are manned or unmanned.

- In terms of the nonlinear simulation, the first thing learned while developing the simulation of the spin was that the dynamics of the maneuver are difficult to model. This is an area that has been studied for many years and it was discovered that more wind tunnel data would be needed to fully model the autorotations associated with a spinning aircraft. As a result of the nonlinear simulations of the remaining maneuvers investigated in this work, it was determined that feedforward throttle control may not be the best approach. It did not have an effect on the performance of the maneuver, but it is believed that if throttle were actively controlled throughout the maneuver the transition time between maneuver recovery and level flight could be reduced. Another solution to the same problem could be to gradually change the speed and altitude commands based on their respective values at the end of the maneuver. The transition time was not an objective for this work, but is a topic that has room for improvement.

- The simulation results which compared the differences between using constant gains and scheduled gains for maneuvering flight control revealed that gain scheduling can improve control system performance for maneuvers that involve large variations in airspeed such as stall turns and tail slides. While it was not needed to complete the maneuvers, the tracking ability of the controller was improved during off-trim flight conditions when scheduled gains were used. For the systems tested, the inclusion of scheduled gains caused the controller to be more sensitive to measurement noise as more oscillations were observed in the response. When using vortex lattice data to

determine the aircraft models, scheduled gains were found to result in worse responses to the reference commands than constant gains due to the errors introduced by the method at low airspeeds.

6.3 Recommendations

The following recommendations provide ideas for continued research to expand the developed method into an implementable control system capable of increasing the safety of general aviation flight and handling the problems associated with modern warfare.

- Acquire and instrument the test aircraft used in the simulations with sensors similar to the ones assumed in this work. Flight testing could then be conducted to verify that the developed approach works as well for an actual aircraft as it did for the simulated aircraft.

- Improve the simulation and further the investigation of autonomous spin recovery by more accurately modeling the out-of-control motions experienced during the spin. The rotary derivatives obtained with steady motion rotary balance tunnel tests could be blended with the forced oscillation data using the Direct Resolution method as in [92] or a Kalviste method as described in [93] to more closely simulate real flight motions. This improvement would allow for more accurate tests of the spin controller developed in this work.

- Investigate the autopilot's robustness to environment effects by including steady wind, wind gusts, and turbulence in the simulation. These effects may act to destabilize the aircraft or reduce the maneuvering performance and a modification to the gain tuning method may be required.

- Develop a selection algorithm to trigger a specific maneuver based on some condition experienced in flight. This functionality could be used to command maneuvers to track a target as well as to avoid structures, people, other aircraft, or enemy fire.

Bibliography

[1] Reitz, B. C., *Collaborative UAS Control to Increase Deconfliction Ability in the NAS*, Master's thesis, Auburn University, 2009, http://etd.auburn.edu/etd/handle/10415/1783 [retrieved 17 July 2009].

[2] "2013 General Aviation Statistical Databook & Industry Outlook," General Aviation Manufacturers Association, 2014, http://www.gama.aero/files/GAMA%202013%20Databook-Updated-LowRes.pdf [retrieved 16 July 2014].

[3] "Stall/spin: Entry point for crash and burn?" Aircraft Owners and Pilots Association, www.aopa.org/asf/ntsb/stall_spin.html [retrieved 14 July 2014].

[4] "Brief on Autonomy Initiatives in the US DoD," Department of Defense, November 2012, http://www.defenseinnovationmarketplace.mil/resources/Autonomy-PSC_Briefing_DistroA_RE.pdf [retrieved 16 July 2014].

[5] "Defense Budget Priorities," Department of Defense, January 2012, www.defenseinnovationmarketplace.mil/resources/012612_Defense_Budget_Priorities.pdf [retrieved 6 July 2012].

[6] "Quadrennial Defense Review 2014," Department of Defense, March 2014, http://www.defense.gov/pubs/2014_Quadrennial_Defense_Review.pdf [retrieved 16 July 2014].

[7] Friedman, N., *Unmanned Combat Air Systems: A New Kind of Carrier Aviation*, Naval Institute Press, 2010.

[8] Mattei, G. and Monaco, S., "Nonlinear Autopilot Design for an Asymmetric Missile Using Robust Backstepping Control," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 5, September 2014, pp. 1462–1476.

[9] Jang, J.-W., Yang, L., Fritz, M., Nguyen, L. H., Johnson, W. R., and Hart, J. J., "Design and Analysis of Morpheus Lander Flight Control System," AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA, August 2014.

[10] Nichols, J. W., Sun, L., Beard, R. W., and McLain, T., "Aerial Rendezvous of Small Unmanned Aircraft Using a Passive Towed Cable System," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, July 2014, pp. 1131–1142.

[11] Park, S., Deyst, J., and How, J. P., "Performance and Lyapunov Stability of a Nonlinear Path-Following Guidance Method," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, November 2007, pp. 1718–1728.

[12] Park, S., "Autonomous Aerobatics on Commanded Path," *Aerospace Science and Technology*, 2011, http://dx.doi.org/10.1016/j.ast.2011.06.007 [retrieved 20 January 2012].

[13] Park, S., *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles*, Ph.D. thesis, Massachusetts Institute of Technology, 2004, http://hdl.handle.net/1721.1/16662 [retrieved 17 July 2013].

[14] Peddle, I. K. and Jones, T., "Acceleration-based 3D Flight Control for UAVs: Strategy and Longitudinal Design," *Advances in Flight Control Systems*, edited by A. Balint, InTech, 2011, http://dx.doi.org/10.5772/15237 [retrieved 20 February 2012].

[15] How, J. P., McGrew, J. S., Frank, A. A., and Hines, G. H., "Autonomous Aircraft Flight Control for Constrained Environments," IEEE International Conference on Robotics and Automation, Pasadena, CA, May 2008.

[16] Frank, A., McGrew, J. S., Valenti, M., Levine, D., and How, J. P., "Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane," AIAA 2007-6318, AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, August 2007.

[17] Matsumoto, T., Konno, A., Suzuki, R., Oosedo, A., Go, K., and Uchiyama, M., "Agile Turnaround Using Post-Stall Maneuvers for Tail-Sitter VTOL UAVs," International Conference on Intelligent Robots and Systems, Taipei, Taiwan, October 2010, pp. 1612–1617.

[18] Sobolic, F. M., *Agile Flight Control Techniques for a Fixed-Wing Aircraft*, Master's thesis, Massachusetts Institute of Technology, 2009, http://acl.mit.edu/papers/SobolicSM.pdf [retrieved 23 January 2012].

[19] Knoebel, N. B. and McLain, T. W., "Adaptive Quaternion Control of a Miniature Tailsitter UAV," American Control Conference, Seattle, WA, June 2008, pp. 2340–2345.

[20] Jun-fang, F. and Zhong, S., "Missile Longitudinal Autopilot Design Using Backstepping Approach," IEEE Aerospace Conference, March 2010, http://dx.doi.org/10.1109/AERO.2010.5446743 [retrieved 6 February 2012].

[21] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, 2nd ed., John Wiley & Sons, Inc., 2003.

[22] Sobel, K. and Shapiro, E., "Eigenstructure Assignment for Design of Multimode Flight Control Systems," *IEEE Control Systems Magazine*, Vol. 5, No. 2, May 1985, pp. 9–15.

[23] Srinathkumar, S., "Eigenvalue/eigenvector Assignment using Output Feedback," *IEEE Transactions on Automatic Control*, Vol. 23, No. 1, February 1978, pp. 79–81.

[24] Klein, G. and Moore, B. C., "Eigenvalue-Generalized Eigenvector Assignment with State Feedback," *IEEE Transactions on Automatic Control*, Vol. 22, No. 1, February 1977, pp. 140–141.

[25] Moorhouse, D. J. and Woodcock, R. J., "Background Information and User Guide for MIL-F-8785C," AFWAL-TR-81-3109, July 1982.

[26] Alag, G. S. and Duke, E. L., "Development of Control Laws for a Flight Test Maneuver Autopilot," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 4, July 1986, pp. 441–445.

[27] Kopecki, G., "Aircraft Trajectory Model Following Control with the Use of Linear Quadratic Regulator Control Laws," AIAA 2012-4828, AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, August 2012.

[28] Cramer, E. J. and Lee, T. P., "Test Flight of LQR Missile Guidance," AIAA 92-4532, AIAA Astrodynamics Conference, Hilton Head, SC, August 1992.

[29] Starin, S. R., Yedavalli, R. K., and Sparks, A. G., "Design of a LQR Controller of Reduced Inputs for Multiple Spacecraft Formation Flying," *Proceedings of the 2001 American Control Conference*, Vol. 2, 2001, pp. 1327–1332.

[30] Stevens, B. L., Lewis, F. L., and Al-Sunni, F., "Aircraft Flight Controls Design Using Output Feedback," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, January 1992, pp. 238–246.

[31] Johnson, E. N., Turbe, M. A., Wu, A. D., Kannan, S. K., and Neidhoefer, J. C., "Flight Test Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hover," AIAA 2006-6775, AIAA Guidance, Navigation, and Control Conference, Keystone, CO, August 2006.

[32] Bruner, H. S., *A Model Following Inverse Controller with Adaptive Compensation for General Aviation Aircraft*, Ph.D. thesis, Wichita State University, 2007, `http://soar.wichita.edu/dspace/bitstream/handle/10057/1073/d07001.pdf` [retrieved 1 May 2012].

[33] Hough, W. J., *Autonomous Aerobatic Flight of a Fixed Wing Unmanned Aerial Vehicle*, Master's thesis, University of Stellenbosch, 2007, `http://hdl.handle.net/10019.1/2730` [retrieved 20 February 2012].

[34] Jourdan, D. B., Piedmonte, M. D., Gavrilets, V., Vos, D. W., and McCormick, J., "Enhancing UAV Survivability Through Damage Tolerant Control," AIAA 2010-7548, AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario Canada, August 2010.

[35] AlSwailem, S. I., *Application of Robust Control in Unmanned Vehicle Flight Control System Design*, Ph.D. thesis, Cranfield University, 2004, `https://dspace.lib.cranfield.ac.uk/bitstream/1826/136/2/ThesisMasterV2.pdf` [retrieved 1 May 2012].

[36] Doyle, J. and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/-Modern Synthesis," *IEEE Transactions on Automatic Control*, Vol. 26, No. 1, February 1981, pp. 4–16.

[37] Kalman, R. L., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Fluids Engineering*, Vol. 82, No. 1, March 1960, pp. 35–45.

[38] Athans, M., "A Tutorial on the LQG/LTR Method," *Proceedings of the 1986 American Control Conference*, Seattle, WA, June 1986, pp. 1289–1296.

[39] Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, 2nd ed., John Wiley & Sons, Inc., 1991.

[40] Leith, D. J. and Leithead, W. E., "Survey of Gain-scheduling Analysis and Design," *International Journal of Control*, Vol. 73, No. 11, 2000, pp. 1001–1025, `http://dx.doi.org/10.1080/002071700411304` [retrieved 5 June 2012].

[41] Smith, R. L., "An Autopilot Design Methodology for Bank-to-Turn Missiles," AFATL TR-89-49, August 1989, `http://www.dtic.mil/get-tr-doc/pdf?AD=ADA213379` [retrieved 5 February 2014].

[42] Myrand-Lapierre, V., Desbiens, A., Gagnon, E., Wong, F., and Poulin, E., "Transitions Between Level Flight and Hovering for a Fixed-Wing Mini Aerial Vehicle," American Control Conference, Baltimore, MD, July 2010, pp. 530–535.

[43] Krogh, K. J., *Developing a Framework for Control of Agile Aircraft Platforms in Autonomous Hover*, Master's thesis, University of Washington, 2009, `www.aa.washington.edu/research/afsl/publications/kroghthesis.pdf` [retrieved 23 January 2012].

[44] Gavrilets, V., *Autonomous Aerobatic Maneuvering of Miniature Helicopters*, Ph.D. thesis, Massachusetts Institute of Technology, 2003, `http://hdl.handle.net/1721.1/17609` [retrieved 28 January 2014].

[45] Mason, W. H., "Applied Computational Aerodynamics," March 1998, `http://www.dept.aoe.vt.edu/~mason/Mason_f/CAtxtTop.html` [retrieved 16 October 2014].

[46] Margason, R. J. and Lamar, J. E., "Vortex-Lattice FORTRAN Program for Estimating Subsonic Aerodynamic Characteristics of Complex Planforms," NASA TN D-6142, February 1971.

[47] Lamar, J. E. and Gloss, B. B., "Subsonic Aerodynamic Characteristics of Interacting Lifting Surfaces with Separated Flow Around Sharp Edges Predicted by a Vortex-Lattice Method," NASA TN D-7921, September 1975.

[48] Herbert, H. E. and Lamar, J. E., "Production Version of the Extended NASA-Langley Vortex Lattice FORTRAN Computer Program," NASA TM 83304, April 1982.

[49] Lan, C. E., "A Quasi-Vortex-Lattice Method in Thin Wing Theory," *Journal of Aircraft*, Vol. 11, No. 9, September 1974, pp. 518–527.

[50] Miranda, L. R., Elliot, R. D., and Baker, W. M., "A Generalized Vortex Lattice Method for Subsonic and Supersonic Flow Applications," NASA CR-2865, December 1977.

[51] Drela, M. and Youngren, H., *AVL 3.30 User Primer*, 2010, `http://web.mit.edu/drela/Public/web/avl/avl_doc.txt` [retrieved January 2014].

[52] Kramer, B. R., "Experimental Evaluation of Superposition Techniques Applied to Dynamic Aerodynamics (Invited)," AIAA 2002-0700, AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2002.

[53] Kay, J., "Acquiring and Modeling Unsteady Aerodynamic Characteristics," AIAA 2000-3907, AIAA Atmospheric Flight Mechanics Conference, Denver, CO, August 2000.

[54] Dickes, E. G., Ralston, J. N., and Lawson, K. P., "Application of Large-Angle Data for Flight Simulation," AIAA 2000-4584, AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, CO, August 2000.

[55] O'Conner, C. J., Ralston, J. N., and Fitzgerald, T., "Evaluation of the NAWC/AD F/A-18 C/D Simulation Including Database Coverage and Dynamic Data Implementation Techniques," AIAA 96-3365, AIAA Atmospheric Flight Mechanics Conference, San Diego, CA, July 1996.

[56] Ogburn, M. E., Nguyen, L. T., and Hoffler, K. D., "Modeling of Large-Amplitude High-Angle-of-Attack Maneuvers," AIAA 88-4357, AIAA Atmospheric Flight Mechanics Conference, Minneapolis, MN, August 1988.

[57] Chambers, J. R. and Stough, III, H. P., "Summary of NASA Stall/Spin Research for General Aviation Configurations," AIAA 86-2597, AIAA General Aviation Technology Conference, Anaheim, CA, 1986.

[58] Cunningham, K., Foster, J. V., Morelli, E. A., and Murch, A. M., "Practical Application of a Subscale Transport Aircraft for Flight Research in Control Upset and Failure Conditions," AIAA 2008-6200, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Honolulu, HI, August 2008.

[59] Murch, A. M., *Aerodynamic Modeling of Post-Stall and Spin Dynamics of Large Transport Airplanes*, Master's thesis, Georgia Institute of Technology, 2007, `http://hdl.handle.net/1853/18855` [retrieved 16 January 2014].

[60] Murch, A. M. and Foster, J. V., "Recent NASA Research on Aerodynamic Modeling of Post-Stall and Spin Dynamics of Large Transport Airplanes," AIAA 2007-0463, AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2007.

[61] "General Operating and Flight Rules," *Federal Aviation Regulations*, 14 CFR Part 91, U.S. Department of Transportation, 2014.

[62] Owens, D. B., Cox, D. E., and Morelli, E. A., "Development of a Low-Cost Sub-Scale Aircraft for Flight Research: The FASER Project," AIAA 2006-3306, AIAA Aerodynamic Measurement Technology and Ground Testing Conference, San Francisco, CA, June 2006.

[63] Napolitano, M. R., *Aircraft Dynamics: From Modeling to Simulation*, John Wiley & Sons, Inc., 2012.

[64] Etkin, B. and Reid, L. D., *Dynamics of Flight: Stability and Control*, 3rd ed., John Wiley & Sons, Inc., 1996.

[65] Schmidt, D. K., *Modern Flight Dynamics*, McGraw-Hill, 2012.

[66] Meriam, J. L. and Kraige, L. G., *Engineering Mechanics: Dynamics*, 7th ed., John Wiley & Sons, Inc., 2012.

[67] Owens, D. B. and Aubuchon, V., "FASER Static Test 153 NASA LaRC," July 2006, `http://hdl.handle.net/11299/119574` [retrieved 19 August 2013].

[68] Owens, D. B. and Hoe, G., "FASER Forced Oscillation Test 158 NASA LaRC," August 2011, `http://hdl.handle.net/11299/119573` [retrieved 19 August 2013].

[69] Hoe, G., Owens, D. B., and Denham, C., "Forced Oscillation Wind Tunnel Testing for FASER Flight Research Aircraft," AIAA 2012-4645, AIAA Atmospheric Flight Mechanics Conference, Minneapolis, MN, August 2012.

[70] Monzon, B. R., *Nonlinear Simulation Development for a Sub-Scale Research Airplane*, Master's thesis, George Washington University, 2001.

[71] Garza, F. R. and Morelli, E. A., "A Collection of Nonlinear Aircraft Simulations in MATLAB," NASA TM-2003-212145, January 2003.

[72] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, 3rd ed., American Institute of Aeronautics and Astronautics, Inc., 1999.

[73] Paw, Y. C., *Synthesis and Validation of Flight Control for UAV*, Ph.D. thesis, The University of Minnesota, 2009, `http://hdl.handle.net/11299/58727` [retrieved 9 August 2013].

[74] Anderson, Jr., J. D., *Fundamentals of Aerodynamics*, 3rd ed., McGraw-Hill, 2001.

[75] Welstead, J. R., Reitz, B. C., and Crouse, Jr., G. L., "Modeling Fuselage Aerodynamic Effects in Aircraft Design Optimization," AIAA 2012-0394, AIAA Aerospace Sciences Meeting and Exhibit, Nashville, TN, January 2012.

[76] Perkins, C. D. and Hage, R. E., *Airplane Performance Stability and Control*, John Wiley & Sons, Inc., 1949.

[77] Torenbeek, E., *Synthesis of Subsonic Airplane Design*, Delft University Press, 1982.

[78] Lowry, J. G. and Polhamus, E. C., "A Method for Predicting Lift Increments due to Flap Deflection at Low Angles of Attack in Incompressible Flow," Tech. rep., National Advisory Committee on Aeronautics, NACA TN-3911, January 1957.

[79] Zipfel, P. H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, American Institute of Aeronautics and Astronautics, Inc., 2000.

[80] Pamadi, B. N., *Performance, Stability, Dynamics, and Control of Airplanes*, 2nd ed., American Institute of Aeronautics and Astronautics, Inc., 2004.

[81] Rysdyk, R., "Unmanned Aerial Vehicle Path Following for Target Observation in Wind," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, September 2006, pp. 1092–1100.

[82] Hall, J. K., Beard, R. W., and McLain, T. W., "Quaternion Control for Autonomous Path Following Maneuvers," AIAA 2012-2483, AIAA Infotech@Aerospace Conference, Garden Grove, CA, June 2012.

[83] Knospe, C., "PID Control," *IEEE Control Systems*, Vol. 26, No. 1, February 2006, pp. 30–31, `http://dx.doi.org/10.1109/MCS.2006.1580151` [retrieved 8 July 2012].

[84] O'Brien, Jr., R. T. and Howe, III, J. M., "Optimal PID Controller Design using Standard Optimal Control Techniques," American Control Conference, Seattle, WA, June 2008, pp. 4733–4738.

[85] Nelson, R. C., *Flight Stability and Automatic Control*, 2nd ed., McGraw-Hill, 1998.

[86] Ang, K. H., Chong, G., and Li, Y., "PID Control System Analysis, Design, and Technology," *IEEE Transactions on Control Systems Technology*, Vol. 13, No. 4, July 2005, pp. 559–576.

[87] Anderson, Jr., J. D., *Introduction to Flight*, 5th ed., McGraw-Hill, 2005.

[88] Bryson, Jr., A. E. and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor & Francis Group, 1975.

[89] Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal of Optimization*, Vol. 9, No. 1, 1998, pp. 112–147.

[90] Andry, Jr., A. N., Shapiro, E. Y., and Chung, J. C., "Eigenstructure Assignment for Linear Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-19, No. 5, September 1983, pp. 711–729.

[91] Penzl, T., "Numerical Solution of Generalized Lyapunov Equations," *Advances in Computational Mathematics*, Vol. 8, 1998, pp. 33–48.

[92] Bihrle, Jr., W. and Barnhart, B., "Spin Prediction Techniques," *Journal of Aircraft*, Vol. 20, No. 2, February 1983, pp. 97–101.

[93] Kalviste, J., "Use of Rotary Balance and Forced Oscillation Test Data in a Six Degrees of Freedom Simulation," AIAA 82-1364, AIAA Atmospheric Flight Mechanics Conference, San Diego, CA, August 1982.

Appendix A

AVL Input and Mass Files

A.1 Input File

```
FASER
0.072                       Mach
0    0    0            iYsym    iZsym    Zsym
8.28    1.4225    6.29   Sref    cref    bref
1.0335    0    0.2362            Xref    Yref    Zref
0.03413                      CDp (optional profile drag)
#============================================
# Main Wing
SURFACE
Main Wing
#Nchord Cspace    |   Nspan Sspace
6         1            20    -2
YDUPLICATE
    0
ANGLE
    0
INDEX
    1
TRANSLATE
# dX     dY   dZ
1.05    0    0.236
SECTION
#      Xle   Yle   Zle   chord   Ainc
    -0.3322    0    0    1.19    0
CLAF
1.1
NACA
0012
SECTION
    -0.3322    0.1675    0    1.19    0
CLAF
1.1
NACA
0012
```

```
SECTION
    -0.3322    0.37    0    1.45    0
CLAF
1.1
NACA
0012
SECTION
    -0.3322    1.68    0    1.45    0
CLAF
1.1
NACA
0012
CONTROL
#name    gain    Xhinge    XYZhvec     SgnDup
Aileron    .75    0.82    0    0    0     -1
SECTION
    -0.3322    3.02    0    1.38    0
CLAF
1.1
NACA
0012
CONTROL
Aileron    .75    0.86    0    0    0     -1
#============================================
# Horizontal Tail
SURFACE
H Stab
6    1    20    -2
YDUPLICATE
    0
ANGLE
    0
INDEX
    1
TRANSLATE
4.182    0    0
SECTION
    -0.27   0.1675    0    0.71    0
CLAF
1.1
SECTION
    -0.27   0.37    0    1.05    0
CLAF
1.1
CONTROL
    Elevator    .8    0.67    0    0    0     1
```

```
SECTION
    -0.27    1.22    0    1.05    0
CLAF
1.1
CONTROL
    Elevator    .8    0.67    0    0    0     1
#============================================
# Vertical Tail
SURFACE
V Stab
6    1    20    -2
ANGLE
    0
INDEX
    1
TRANSLATE
3.5678    0   0
SECTION
    0    0    0    1.57    0
CLAF
1.1
CONTROL
    Rudder    -1    0.64    0    0    0     1
SECTION
    0.83    0    0.5    0.425    0
CLAF
1.1
CONTROL
    Rudder    -1    0.40    0    0    0     1
#============================================
# Horizontal Fuselage
SURFACE
H Fuse
12    1    5    -2.5
YDUPLICATE
    0
INDEX
    1
TRANSLATE
-.2461    0    0
SECTION
   0    0    0    4.8681    0
NACA
    0009
SECTION
   0    0.1675    0    4.8681    0
```

```
NACA
    0009
#============================================
# Vertical Fuselage Upper
SURFACE
V Fuse Upper
12    1    5    0.0
INDEX
    1
TRANSLATE
-.2461    0    0
SECTION
    0    0    0    3.8139    0
NACA
    0009
SECTION
    0    0    0.1675   2.636    0
NACA
    0009
SECTION
    0.9639    0    0.236    1.19    0
NACA
    0009
#============================================
# Vertical Fuselage Lower
SURFACE
V Fuse Lower
12    1    5    0.0
INDEX
    1
TRANSLATE
-.2461    0    0
SECTION
    0    0    0    3.8139    0
NACA
    0009
SECTION
    0    0   -0.1675    2.636    0
NACA
    0009
SECTION
    0.9639    0    -0.236    1.19    0

NACA
    0009
```

```
#============================================
# Horizontal Propeller
SURFACE
H Prop
2    1    10    -2
YDUPLICATE
    0
INDEX
    1
TRANSLATE
-.2461    0    0
SECTION
    0    .1675    0    .026    0
SECTION
    0    .5833    0    .026    0
#============================================
# Vertical Propeller Upper
SURFACE
V Prop Upper
2    1    10    -2
INDEX
    1
ANGLE
    -12
TRANSLATE
-.2461    0    0
SECTION
    0    0    0.1675    .026    0
SECTION
    0    0    0.5833    .026    0
#============================================
# Vertical Propeller Lower
SURFACE
V Prop Lower
2    1    10    -2
INDEX
    1
ANGLE
    -12
TRANSLATE
-.2461    0    0
SECTION
    0    0    -0.1675    .026    0
SECTION
    0    0    -0.5833    .026    0
```

A.2 Mass File

```
# FASER
# The Lunit and Munit values scale the mass, xyz, and inertia table data below.
# Lunit value will also scale all lengths and areas in the AVL input file.
Lunit = 1 ft
Munit = 1 slug
Tunit = 1 s
#=========================
g   = 32.2
# for 3000 ft
#rho = 0.002175
# for 2700 ft
rho = 0.002195
#=========================
# Mass & Inertia breakdown.
# x y z  is location of item's own CG.
# Ixx... are item's inertias about item's own CG.
#
# x,y,z system here must be exactly the same one used in the .avl input file
#     (same orientation, same origin location, same length units)
#
# Mass    Xcg    Ycg   Zcg      Ixx      Iyy      Izz    Ixy    Ixz   Iyz
0.5078   1.0335   0   0.2362   0.6319   0.7446   1.2542   0.   -0.14   0.
```

Appendix B

State-Space Matrices

Using the EoM developed in Sec. 2.4.1 as well as the stability derivatives from AVL, the state and control matrices assuming unaccelerated flight during cruise with an airspeed of 80 ft/s at 2,700 ft were estimated to be:

$$\mathbf{A}_{EoM}(:, 1:6) = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{cccccc} \phi & \theta & \psi & p & q & r \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0.003587 & 0.06844 \\ 0 & 0 & 0 & 0 & 0.9986 & -0.05234 \\ 0 & 0 & 0 & 0 & 0.05246 & 1.001 \\ 0 & 0 & 0 & -7.908 & 0 & 1.541 \\ 0 & 0 & 0 & 0 & -8.985 & 0 \\ 0 & 0 & 0 & 0.2533 & 0 & -0.9174 \\ 0 & -32.12 & 0 & 0 & -4.858 & 0 \\ 32.08 & -0.1152 & 0 & 5.922 & 0 & -79.19 \\ -1.681 & -2.199 & 0 & 0 & 70.88 & 0 \\ -0.2856 & -80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 57.3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

$$\mathbf{A}_{EoM}(:, 7:12) = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{ccccc} u & v & w & p_D & \omega_{\text{engine}} \end{array} \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.0002 & -0.3885 & 0.002982 & 0 & 0 \\ 0.03733 & 0 & -0.5447 & 0 & 0 \\ 0.00126 & 0.1609 & -0.01838 & 0 & 0 \\ -0.3541 & 0 & 0.8253 & 0 & 0.02597 \\ -0.04193 & -0.3209 & -0.002874 & 0 & 0 \\ -0.3811 & 0 & -6.169 & 0 & 0 \\ -0.06837 & 0.05221 & 0.9963 & 0 & 0 \\ 25.99 & 0 & 1.852 & -0.03284 & -5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

$$\mathbf{A}_{EoM}(:, 13:16) = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{ccccc} x_w & \delta_e & \delta_a & \delta_r & \delta_T \end{array} \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.454 & 0.03554 & 2.955 \\ 0 & -1.275 & 0 & 0 & 0 \\ 0 & 0 & 0.1024 & -0.1869 & 0-0.374 \\ 0 & 0.04074 & 0 & 0 & 0 \\ 0 & 0 & -0.0056 & 0.1298 & 0 \\ 0 & -0.5944 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3384 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -11.43 & 0 & 0 & 0 \\ 0 & 0 & -11.43 & 0 & 0 \\ 0 & 0 & 0 & -9.52 & 0 \\ 0 & 0 & 0 & 0 & -20 \end{array}\right] \end{array}$$

$$
\mathbf{B}_{EoM} = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{cccc} u_e & u_a & u_r & u_T \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 11.43 & 0 & 0 & 0 \\ 0 & 11.43 & 0 & 0 \\ 0 & 0 & 9.52 & 0 \\ 0 & 0 & 0 & 20 \end{array}\right] \end{array}
$$

Linearizing the simulation during cruise with an airspeed of 80 ft/s at 2,700 ft gives state and control matrices of:

$$\mathbf{A}_{sim}(:, 1:6) = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{cccccc} \phi & \theta & \psi & p & q & r \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0.003704 & 0.07204 \\ 0 & 0 & 0 & 0 & 0.9987 & -0.05134 \\ 0 & 0 & 0 & 0 & 0.05148 & 1.001 \\ 0 & 0 & 0 & -9.272 & 0.0607 & 2.03 \\ 0 & 0 & 0 & 0 & -7.758 & 0.4501 \\ 0 & 0 & 0 & 0.668 & -0.274 & -1.632 \\ 0 & -32.05 & 0 & 0 & -5.763 & 0 \\ 32.01 & -0.1187 & 0 & 5.646 & 0 & -78.45 \\ -1.646 & -2.309 & 0 & 0 & 79.79 & 0 \\ -0.2951 & -80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 57.3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

$$\mathbf{A}_{sim}(:, 7:12) = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{ccccc} u & v & w & p_D & \omega_{\text{engine}} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.01334 & -0.3641 & 0.0317 & -3.2\text{x}10^{-5} & -0.002156 \\ 0.1059 & 0.01936 & -0.3906 & 0 & -0.005723 \\ 0.008806 & 0.2089 & -0.004145 & 3.6\text{x}10^{-6} & -0.0002949 \\ -0.3424 & -0.03163 & 0.6009 & 6.9\text{x}10^{-5} & 0.02629 \\ -0.002552 & -0.6075 & -0.1695 & -4.9\text{x}10^{-5} & -0.001948 \\ -0.2835 & 0.1696 & -6.358 & -9.5\text{x}10^{-4} & -0.004386 \\ -0.07195 & 0.05121 & 0.9961 & 0 & 0 \\ 27.02 & 0 & 1.952 & -0.03187 & -4.959 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{A}_{sim}(:, 13:16) = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{ccccc} x_w & \delta_e & \delta_a & \delta_r & \delta_T \end{array} \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.843 & 0.06888 & 3.365 \\ 0 & -1.529 & 0 & 0 & 0 \\ 0 & 0 & 0.2058 & 0.3089 & -0.3756 \\ 0 & 0.205 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4765 & 0 \\ 0 & -0.1628 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3353 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -11.43 & 0 & 0 & 0 \\ 0 & 0 & -11.43 & 0 & 0 \\ 0 & 0 & 0 & -9.52 & 0 \\ 0 & 0 & 0 & 0 & -20 \end{array}\right] \end{array}$$

$$
\mathbf{B}_{sim} = \begin{array}{c} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ p_D \\ \omega_{\text{engine}} \\ x_w \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_T \end{array} \begin{array}{c} \begin{array}{cccc} u_e & u_a & u_r & u_T \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 11.43 & 0 & 0 & 0 \\ 0 & 11.43 & 0 & 0 \\ 0 & 0 & 9.52 & 0 \\ 0 & 0 & 0 & 20 \end{array}\right] \end{array}
$$

Appendix C

Time-Weighted Cost

This is the MATLAB function file created to compute the time-weighted cost derived in Sec. 4.2.2.

```
function [J] = CalcCost_LQT(Ap,Bp,Ch,Cp,Cphi,Cpsi,Crw,CVw,Gp,H,kexp,Kvec,...
                            nint,R,r0,sizeK)

[A,B,G,C,D,F,I] = matrices(Ap,Bp,Ch,Cp,Cphi,Cpsi,Crw,CVw,Gp,Kvec(5),...
                           Kvec(6),Kvec(7),Kvec(8),nint,sizeK);

K = [Kvec(1) Kvec(2) Kvec(3)  0       0       0       0        0;...
       0       0       0    Kvec(4)   0       0       0        0;...
       0       0       0      0    Kvec(9)    0       0        0;...
       0       0       0      0       0    Kvec(10) Kvec(11) Kvec(12)];

[Ac,Bc,Cc,Dc] = CLmatrices(A,B,G,C,D,F,I,K);

if (max(real(eig(Ac))) >= 0)
    J = 1e15;
    return
end

P = H'*H;
for i = 1:kexp
    P = lyap(Ac', P);
end

try
    Pk = lyap(Ac', factorial(kexp)*P + C'*K'*R*K*C);
catch
    J = 1e10;
    return
end

xbar = -Ac\Bc*r0;
X = xbar*xbar';

J = 0.5*trace(Pk*X);
end
```