2014

FILTERED-DYNAMIC-INVERSION CONTROL FOR FIXED-WING UNMANNED AERIAL SYSTEMS

Jon Mullen
University of Kentucky, jon-mullen@hotmail.com

FILTERED-DYNAMIC-INVERSION CONTROL FOR FIXED-WING UNMANNED AERIAL SYSTEMS

THESIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering in the College of Engineering at the University of Kentucky

by

Jon Mullen

Lexington, Kentucky

Directors: Dr. Sean Bailey & Dr. Jesse B. Hoagg,
Professors of Mechanical Engineering

Lexington, Kentucky

2014

ABSTRACT OF THESIS

FILTERED-DYNAMIC-INVERSION CONTROL FOR FIXED-WING UNMANNED AERIAL SYSTEMS

Instrumented umanned aerial vehicles represent a new way of measuring turbulence in the atmospheric boundary layer. However, autonomous measurements require control methods with disturbance-rejection and altitude command-following capabilities. Filtered dynamic inversion is a control method with desirable disturbance-rejection and command-following properties, and this controller requires limited model information. We implement filtered dynamic inversion as the pitch controller in an altitude-hold autopilot. We design and numerically simulate the continuous-time and discrete-time filtered-dynamic-inversion controllers with anti-windup on a non-linear aircraft model. Finally, we present results from a flight experiment comparing the filtered-dynamic-inversion controller to a classical proportional-integral controller. The experimental results show that the filtered-dynamic-inversion controller performs better than a proportional-integral controller at certain values of the parameter.

KEYWORDS: Filtered dynamic inversion, Unmanned Aerial Vehicle, Altitude hold, Disturbance rejection, Command following

Jon Mullen
September 29, 2014

FILTERED-DYNAMIC-INVERSION CONTROL FOR FIXED-WING UNMANNED AERIAL SYSTEMS

by

Jon Mullen

Dr. Sean Bailey & Dr. Jesse B. Hoagg

Directors of Thesis

Dr. James M. McDonough,

Director of Graduate Studies

September 29, 2014

Acknowledgments

I would like to thank my mom, dad, and sister. They have always stood behind me, and although moving halfway across the country for school wasn't easy on anyone, they've put what they know is best for me above the pain of being apart.

Doing this research and writing this thesis in a year never could have happened without the guidance and support of my advisors, Dr. Bailey and Dr. Hoagg. I always left their offices feeling like I had a ton of work to do, but like it was well within my reach. I would also like to thank Dr. Seigler for serving on my committee.

For supporting flight testing and making good decisions on the fly, I'd like to thank Luke Briggs, Jonathan Boustani, Nick Willard, Bob Cooper and Joe.

In no particular order, the people who supported me in one way or another and to whom I'm immensely grateful: Niko Lemaç, Nick Millevoi, Josh Feinberg, Dan Walinksky, Alex Alverson, Heidi, Mike Haley, Rita Griffith, Amy Craiglow, Tom Suggs, Tony D'Amato and the Ford Motor Company, Steely Dan and Barack Obama.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation for Unmanned Aerial Vehicle Altitude Tracking

Understanding the structure of turbulence within the Earth's atmospheric boundary layer (ABL) is important for both for the sake of atmospheric science and for furthering fundamental turbulence research. Phenomena like pollutant transport and localized weather happen in the ABL, and the ABL plays a role in global climate change dynamics [1, 2]. Meanwhile, turbulence researchers can benefit from studying parts of the ABL as canonical flows. Obtaining a spatial description of the structure and organization of turbulence is thus of theoretical interest. Towers are not suitable for this work because they are stationary, and thus reliant on Taylor's frozen flow hypothesis, and are typically instrumented with anemometers having poor temporal response [3, 4]. Manned aircraft have been used to measure atmospheric turbulence; characterize wind, temperature and humidity profiles; and track pollutant dispersion [5–9]. However, manned aircraft are costly, and conducting measurements close to the ground would be dangerous.

Unmanned aerial vehicles (UAVs) offer an innovative way to provide safe, low-cost, autonomous measurement of the ABL [10–12]. However, the requirements of a UAV meant for ABL research are unique. Most properties of the ABL are functions of height, thus the UAV must track altitude well. Additionally, air velocity sensors measure only the relative velocity of air with respect to the airframe, thus it is desirable to minimize unnecessary movement of the airframe.

The turbulence we wish to study constitutes a disturbance from the point of view of the UAV's control system. The anemometer's signal is in a system apart from the autopilot, and thus is not available for feedback. Therefore, the wind must be treated as an unknown-and-unmeasured disturbance. The dynamics of a fixed-wing, subsonic aircraft are generally well-understood, but are also nonlinear and sensitive to environmental conditions. Furthermore, affordable hardware solutions for a UAV's autopilot are limited in processing power. Thus, a linear controller that excels at disturbance rejection and requires minimal model information is ideal.

1.2 Overview of Filtered Dynamic Inversion

Filtered dynamic inversion is a high-parameter-stabilizing controller for multiple-input-multiple-output minimum-phase systems. [13, 14]. For a sufficiently large parameter, the average power of the performance is arbitrarily small. The controller is designed using minimal model information, specifically, the plant's relative degree and high-frequency-gain matrix. In the case of a single-input-single-output system, these assumptions can be weakened to require knowledge of the plant's relative degree, the sign of the high-frequency-gain, and an upper-bound on the magnitude of the high-frequency-gain.

The model information required to design the filtered-dynamic-inversion controller for pitch angle error to elevator deflection can be found using knowledge of fixed-wing aircraft dynamics and computational aerodynamics software. First, the relative degree from elevator deflection to pitch angle is usually taken as two; three for commanded elevator deflection to pitch angle if the servo dyanmics are first-order. Second, for an elevator located on the plane's tail, the sign of the high-frequency-gain, i.e. leading nonzero Markov parameter, is negative. Finally, an upper-bound on the magnitude of the high-frequency-gain can be estimated using computational aerodynamics software. Thus, we expect the filtered-dynamic-inversion controller to be appropriate for the UAV altitude-tracking problem.

In this thesis, we implement the filtered dynamic inversion controller as the pitch controller in an altitude-hold autopilot. In Chapter 5, we design and numerically simulate the filtered dynamic inversion controller with anti-windup with a nonlinear aircraft model. In Chapter 6, we discretize the controller designed in Chapter 5 and numerically simulate the discrete-time controller with the same continuous-time nonlinear aircraft model. Finally, in Chapter 8, we present results from a flight experiment comparing the filtered dynamic inversion controller to a baseline proportional-integral controller.

1.3 Summary of Chapters

Chapter 2

We present a nonlinear model for an aircraft. In particular, we develop the six-degree-of-freedom kinematics and dynamics for an aircraft, and a nonlinear model of the aerodynamic forces and moments that act on the aircraft. In addition, we linearize the longitudinal equations of motion.

Chapter 3

We review the filtered-dynamic-inversion controller. We provide the closed-loop stability and performance properties. In addition, we augment the filtered-dynamic-inversion controller with an anti-windup strategy.

Chapter 4

We describe the AeroWorks EDGE 540T fixed-wing aircraft, which is used as the experimental testbed for the filtered dynamic inversion controller. We compute the model parameters of the test platform using computational fluid dynamics software.

Chapter 5

We construct the filtered dynamic inversion controller in continuous-time for use as a pitch control on the EDGE UAV. We construct a speed control loop and an altitude-to-pitch-command outer loop. We simulate the nonlinear aircraft dynamics developed in Chapter 2 with the test platform model developed in Chapter 4.

Chapter 6

We discretize the filtered-dynamic-inversion control, which was developed in Chapter 3 and designed in continuous time in Chapter 5. We augment the discrete-time filtered-dynamic-inversion and proportional-integral pitch controllers with an anti-windup strategy. Finally, we present results from simulations of the continuous-time aircraft dynamics with the discrete-time altitude autopilot in feedback.

Chapter 7

We describe the experiment used to compare the altitude-tracking capabilities of the autopilot with the filtered dynamic inversion controller to the autopilot with the proportional-integral controller.

Chapter 8

We present results from the experiment described in Chapter 7 with the EDGE test platform described in Chapter 4. We compare the results to simulation results from Chapters 5 and 6.

Chapter 2 Background: Nonlinear Aircraft Model

In this chapter, we present a nonlinear model for an aircraft. We develop the six-degree-of-freedom kinematics and dynamics for an aircraft, and a nonlinear model of the aerodynamic forces and moments that act on the aircraft. In addition, we linearize the longitudinal equations of motion about a constant-velocity, constant-altitude, wings-level flight condition.

2.1 Aircraft Kinematics

The inertial frame $F_{\rm I}$ is a frame in which Newton's second law is valid. The inertial frame $F_{\rm I}$ is centered at $O_{\rm I}$ with orthogonal unit vectors $\hat{\imath}_{\rm I}$, $\hat{\jmath}_{\rm I}$, and $\hat{k}_{\rm I}$. The body frame $F_{\rm B}$ is fixed to the aircraft at the aircraft's center of mass $O_{\rm B}$ with orthogonal unit vectors $\hat{\imath}_{\rm B}$, $\hat{\jmath}_{\rm B}$, and $\hat{k}_{\rm B}$ as shown in Figure 2.1.

The position of $O_{\rm B}$ relative to $O_{\rm I}$ is

$$\vec{r} = X\hat{\imath}_{\rm I} + Y\hat{\jmath}_{\rm I} + Z\hat{k}_{\rm I},$$

and the velocity of $O_{\rm B}$ relative to $O_{\rm I}$ with respect to $F_{\rm I}$ is

$$\vec{v} \triangleq \overset{\rm I\cdot}{\vec{r}} = \dot{X}\hat{\imath}_{\rm I} + \dot{Y}\hat{\jmath}_{\rm I} + \dot{Z}\hat{k}_{\rm I}.$$

The angular velocity of $F_{\rm B}$ relative to $F_{\rm I}$ is $\vec{\omega}$.

Let $[\,\cdot\,]_{\rm I}$ denote a physical vector resolved in the inertial frame, and let $[\,\cdot\,]_{\rm B}$ denote a physical vector resolved in the body frame. Next, $\vec{v}$ and $\vec{\omega}$ are resolved in the body

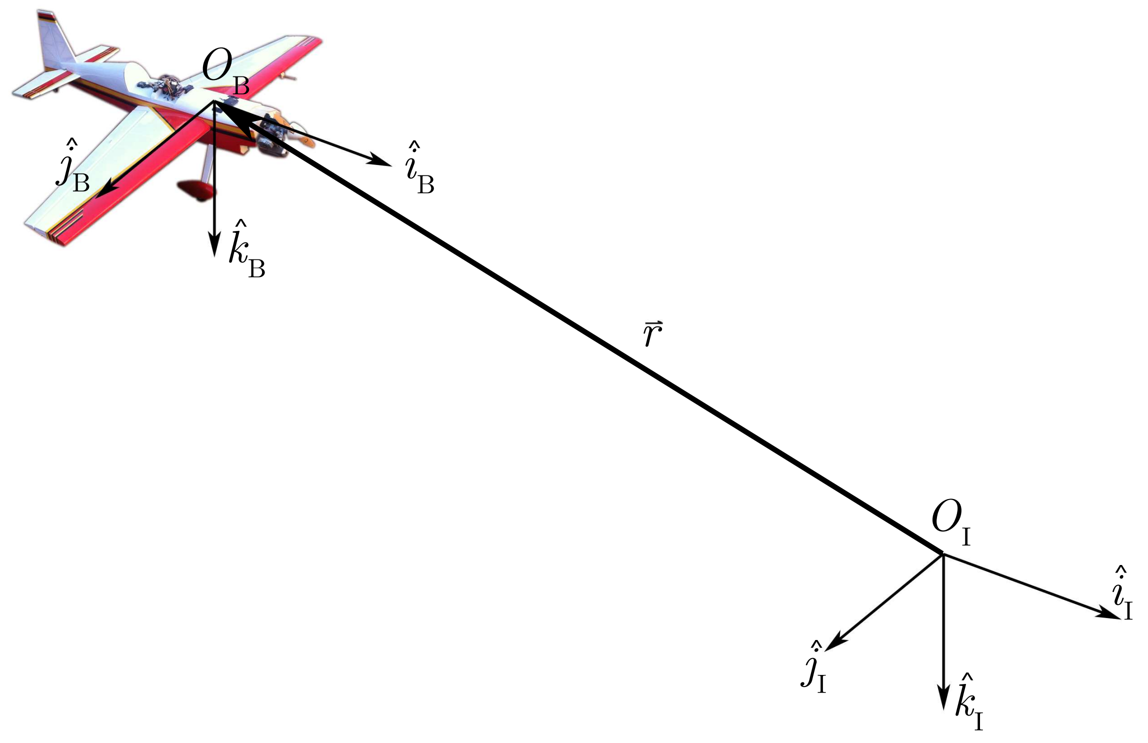


Figure 2.1: *Inertial and Body Frames.* The inertial frame F_I is centered at O_I with orthogonal unit vectors $\hat{\imath}_\mathrm{I}$, $\hat{\jmath}_\mathrm{I}$, and $\hat{k}_\mathrm{I}$. The body frame F_B is fixed to the aircraft at the aircraft's center of mass O_B with orthogonal unit vectors $\hat{\imath}_\mathrm{B}$, $\hat{\jmath}_\mathrm{B}$, and $\hat{k}_\mathrm{B}$.

frame and written as

$$v_\mathrm{B} \triangleq [\vec{v}]_\mathrm{B} = \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \qquad \omega_\mathrm{B} \triangleq [\vec{\omega}]_\mathrm{B} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \tag{2.1}$$

Moreover, the skew-symmetric matrix associated with ω_B is given by

$$\Omega \triangleq \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}. \tag{2.2}$$

Let ϕ, θ, and ψ be Euler angles defined by a 3-2-1 rotation sequence, which is standard in flight dynamics [15–17]. The orientation matrix of F_{B} relative to F_{I} is

$$C_{\mathrm{BI}} \triangleq \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix},$$

where

$$\begin{aligned}
C_{11} &\triangleq (\cos\theta)(\cos\psi), \\
C_{12} &\triangleq (\cos\theta)(\sin\psi), \\
C_{13} &\triangleq -\sin\theta, \\
C_{21} &\triangleq -(\cos\phi)(\sin\psi) + (\sin\phi)(\sin\theta)(\cos\psi), \\
C_{22} &\triangleq (\cos\phi)(\cos\psi) + (\sin\phi)(\sin\theta)(\sin\psi), \\
C_{23} &\triangleq (\sin\phi)(\cos\theta) \\
C_{31} &\triangleq (\sin\phi)(\sin\psi) + (\cos\phi)(\sin\theta)(\cos\psi), \\
C_{32} &\triangleq -(\sin\phi)(\cos\psi) + (\cos\phi)(\sin\theta)(\sin\psi), \\
C_{33} &\triangleq (\cos\phi)(\cos\theta),
\end{aligned}$$

and define $C_{\mathrm{IB}} \triangleq C_{\mathrm{BI}}^{-1} = C_{\mathrm{BI}}^{\mathrm{T}}$, which is the orientation matrix of F_{I} relative to F_{B}. Thus, the velocity of the aircraft's center of mass resolved in F_{I} is

$$[\vec{v}]_{\mathrm{I}} = C_{\mathrm{IB}} v_{\mathrm{B}}. \tag{2.3}$$

From Euler's kinematic equations [16], we obtain

$$\begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} = \begin{bmatrix} 1 & (\tan\theta)(\sin\phi) & (\tan\theta)(\cos\phi) \\ 0 & \cos\phi & -\sin\phi \\ 0 & (\sin\phi)/(\cos\theta) & (\cos\phi)/(\cos\theta) \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \tag{2.4}$$

2.2 Aircraft Dynamics

Let m be the aircraft's mass and $I_{\rm c}$ be the aircraft's mass moment of inertia about the center of mass. Resolving $I_{\rm c}$ in $F_{\rm B}$ yields

$$[I_{\rm c}]_{\rm B} = \begin{bmatrix} I_{\rm xx} & -I_{\rm xy} & -I_{\rm xz} \\ -I_{\rm xy} & I_{\rm yy} & -I_{\rm yz} \\ -I_{\rm xz} & -I_{\rm yz} & I_{\rm zz} \end{bmatrix}. \tag{2.5}$$

We make the following assumptions:

(2.A1) Flight is at low speed and low altitude.

(2.A2) The aircraft is rigid.

(2.A3) The aircraft mass m is constant.

(2.A4) The aircraft is symmetric about the $\hat{\imath}_{\rm B}$-$\hat{k}_{\rm B}$ plane.

(2.A5) The thrust force acts through the center of mass.

(2.A6) The thrust force is in the $\hat{\imath}_{\rm B}$-$\hat{k}_{\rm B}$ plane.

Assumption (2.A1) implies that a point fixed on the Earth's surface can be the origin of the inertial frame and the Earth can be treated as flat. Assumption (2.A2) implies that no part of the aircraft moves in the body frame relative to the center of mass. Assumption (2.A3) implies that ${\rm d}m/{\rm d}t = 0$. Furthermore, (2.A2) and (2.A3) imply that the location of the center of mass in $F_{\rm B}$ is time invariant, and that ${\rm d}([I_{\rm c}]_{\rm I})/{\rm d}t = 0$. Assumption (2.A4) implies that $I_{\rm xy} = I_{\rm yz} = 0$. Assumption (2.A5) implies that there are no moments about the center of mass due to the thrust force. Finally, (2.A6) implies that the thrust force in the $\hat{\jmath}_{\rm B}$ direction is zero.

Let $\vec{F}_{\rm a}$ be the aerodynamic force, and let $\vec{F}_{\rm T}$ be the thrust force. The aerodynamic force resolved in $F_{\rm B}$ is written as $[\vec{F}_{\rm a}]_{\rm B} = [X_{\rm a}\ Y_{\rm a}\ Z_{\rm a}]^{\rm T}$, and the thrust force resolved in F_{B} is written as $[\vec{F}_{\mathrm{T}}]_{\mathrm{B}} = [X_{\mathrm{T}}\ 0\ Z_{\mathrm{T}}]^{\mathrm{T}}$, where the entry in the direction of $\hat{\jmath}_{\mathrm{B}}$ is zero because of (2.A6).

Let $\vec{M}_{\mathrm{c}}$ be the moment about the aircraft's center of mass due to the aerodynamic forces. The aerodynamic moment $\vec{M}_{\mathrm{c}}$ resolved in F_{B} is written as $[\vec{M}_{\mathrm{c}}]_{\mathrm{B}} = [L\ M\ N]^{\mathrm{T}}$.

Thus, Newton's second law in the body frame yields

$$m[\vec{g}]_{\mathrm{B}} + [\vec{F}_{\mathrm{a}}]_{\mathrm{B}} + [\vec{F}_{\mathrm{T}}]_{\mathrm{B}} = m\dot{v}_{\mathrm{B}} + m\Omega v_{\mathrm{B}}, \tag{2.6}$$

where $\vec{g} = g\hat{k}_{\mathrm{I}}$ is the acceleration due to gravity. Moreover, Euler's equation yields

$$[\vec{M}_{\mathrm{c}}]_{\mathrm{B}} = [I_{\mathrm{c}}]_{\mathrm{B}}\dot{\omega}_{\mathrm{B}} + \Omega[I_{\mathrm{c}}]_{\mathrm{B}}\omega_{\mathrm{B}}. \tag{2.7}$$

Finally, combining (2.1), (2.2), and (2.5)–(2.7) gives

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = -\begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}\begin{bmatrix} U \\ V \\ W \end{bmatrix} + \frac{1}{m}\begin{bmatrix} -mg\sin\theta + X_{\mathrm{a}} + X_{\mathrm{T}} \\ mg(\sin\phi)(\cos\theta) + Y_{\mathrm{a}} \\ mg(\cos\phi)(\cos\theta) + Z_{\mathrm{a}} + Z_{\mathrm{T}} \end{bmatrix}, \tag{2.8}$$

$$\begin{aligned}\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = {} & \begin{bmatrix} I_{\mathrm{xx}} & 0 & -I_{\mathrm{xz}} \\ 0 & I_{\mathrm{yy}} & 0 \\ -I_{\mathrm{xz}} & 0 & I_{\mathrm{zz}} \end{bmatrix}^{-1}\left(\begin{bmatrix} L \\ M \\ N \end{bmatrix}\right. \\ & \left. - \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}\begin{bmatrix} I_{\mathrm{xx}} & 0 & -I_{\mathrm{xz}} \\ 0 & I_{\mathrm{yy}} & 0 \\ -I_{\mathrm{xz}} & 0 & I_{\mathrm{zz}} \end{bmatrix}\begin{bmatrix} P \\ Q \\ R \end{bmatrix}\right).\end{aligned} \tag{2.9}$$

2.3 Aerodynamic Forces and Moments

In this section, we develop a nonlinear model of the aerodynamic force $[\vec{F}_{\rm a}]_{\rm B}$ and aerodynamic moment $[\vec{M}_{\rm c}]_{\rm B}$. Let $\vec{W}$ denote the velocity of the wind, and define

$$\vec{v}_{\rm r} \triangleq \vec{v} - \vec{W}, \tag{2.10}$$

which we resolve in the body frame as $[\vec{v}_{\rm r}]_{\rm B} = [U_{\rm r}\ V_{\rm r}\ W_{\rm r}]^{\rm T}$. Next, define the total velocity, angle of attack, and sideslip angle by

$$\begin{aligned} V_{\rm T} &\triangleq \sqrt{U_{\rm r}^2 + V_{\rm r}^2 + W_{\rm r}^2}, \\ \alpha &\triangleq \tan^{-1}\frac{W_{\rm r}}{U_{\rm r}}, \\ \beta &\triangleq \sin^{-1}\frac{V_{\rm r}}{V_{\rm T}}, \end{aligned}$$

respectively. We assume the aerodynamic forces $X_{\rm a}$, $Y_{\rm a}$, and $Z_{\rm a}$ can be expressed as the functions

$$X_{\rm a} = X_{\rm a}(V_{\rm T}, \alpha, \beta, Q, \delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}), \tag{2.11}$$

$$Y_{\rm a} = Y_{\rm a}(V_{\rm T}, \alpha, \beta, P, R, \delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}), \tag{2.12}$$

$$Z_{\rm a} = Z_{\rm a}(V_{\rm T}, \alpha, \beta, Q, \delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}), \tag{2.13}$$

where $\delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}$ are the elevator, rudder, and aileron deflections, respectively. Moreover, we assume the aerodynamic moments L, M, and N can be expressed as the functions

$$L = L(V_{\rm T}, \alpha, \beta, P, R, \delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}), \tag{2.14}$$

$$M = M(V_{\rm T}, \alpha, \beta, Q, \delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}), \tag{2.15}$$

$$N = N(V_{\rm T}, \alpha, \beta, P, R, \delta_{\rm e}, \delta_{\rm r}, \delta_{\rm a}). \tag{2.16}$$

There is no accepted closed-form expression for the aerodynamic forces and moments acting on an airframe with arbitrary attitude, velocity, angular velocity, and control deflections. However, wind tunnel experiments and computational fluid dynamics software are able to estimate the first-order Taylor series expansions of (2.11)–(2.16). Thus, we use a Taylor series expansion to approximate the aerodynamic forces and moments (2.11)–(2.16) in a neighborhood of $\omega_{\rm B} = 0$ and $\delta_{\rm e} = \delta_{\rm r} = \delta_{\rm a} = 0$.

To model the aerodynamic forces and moments, we expand (2.11)–(2.16) as a Taylor series to the first order,

$$\begin{aligned} X_{\rm a} &= X_{\rm a}(V_{\rm T}, \alpha, \beta, 0, 0, 0, 0) + \frac{\partial X_{\rm a}}{\partial Q}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} Q \\ &\quad + \frac{\partial X_{\rm a}}{\partial \delta_{\rm e}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm e} + \frac{\partial X_{\rm a}}{\partial \delta_{\rm r}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm r} + \frac{\partial X_{\rm a}}{\partial \delta_{\rm a}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm a}, \qquad (2.17) \\ Y_{\rm a} &= Y_{\rm a}(V_{\rm T}, \alpha, \beta, 0, 0, 0, 0, 0) + \frac{\partial Y_{\rm a}}{\partial P}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} P + \frac{\partial Y_{\rm a}}{\partial R}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} R \\ &\quad + \frac{\partial Y_{\rm a}}{\partial \delta_{\rm e}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm e} + \frac{\partial Y_{\rm a}}{\partial \delta_{\rm r}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm r} + \frac{\partial Y_{\rm a}}{\partial \delta_{\rm a}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm a}, \qquad (2.18) \\ Z_{\rm a} &= Z_{\rm a}(V_{\rm T}, \alpha, \beta, 0, 0, 0, 0) + \frac{\partial Z_{\rm a}}{\partial Q}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} Q \\ &\quad + \frac{\partial Z_{\rm a}}{\partial \delta_{\rm e}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm e} + \frac{\partial Z_{\rm a}}{\partial \delta_{\rm r}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm r} + \frac{\partial Z_{\rm a}}{\partial \delta_{\rm a}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm a}, \qquad (2.19) \\ L &= L(V_{\rm T}, \alpha, \beta, 0, 0, 0, 0, 0) + \frac{\partial L}{\partial P}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} P + \frac{\partial L}{\partial R}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} R \\ &\quad + \frac{\partial L}{\partial \delta_{\rm e}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm e} + \frac{\partial L}{\partial \delta_{\rm r}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm r} + \frac{\partial L}{\partial \delta_{\rm a}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm a}, \qquad (2.20) \\ M &= M(V_{\rm T}, \alpha, \beta, 0, 0, 0, 0) + \frac{\partial M}{\partial Q}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} Q \\ &\quad + \frac{\partial M}{\partial \delta_{\rm e}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm e} + \frac{\partial M}{\partial \delta_{\rm r}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm r} + \frac{\partial M}{\partial \delta_{\rm a}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0)} \delta_{\rm a}, \qquad (2.21) \\ N &= N(V_{\rm T}, \alpha, \beta, 0, 0, 0, 0, 0) + \frac{\partial N}{\partial P}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} P + \frac{\partial N}{\partial R}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} R \\ &\quad + \frac{\partial N}{\partial \delta_{\rm e}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm e} + \frac{\partial N}{\partial \delta_{\rm r}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm r} + \frac{\partial N}{\partial \delta_{\rm a}}\Big|_{(V_{\rm T},\alpha,\beta,0,0,0,0,0)} \delta_{\rm a}. \qquad (2.22) \end{aligned}$$

Thus, the nonlinear aircraft model is given by (2.3), (2.4), (2.8), (2.9), and (2.17)–(2.22).

2.4 Linearization of Longitudinal Equations

In this section, we linearize the equations of motion (2.3), (2.4), (2.8), (2.9), and (2.17)–(2.22) about a constant-velocity, constant-altitude, wings-level flight condition.

Since we assume that the plane is in longitudinal flight, it follows that $\phi = V = P = R = Y_{\mathrm{a}} = L = N = 0$. We also assume that there is no wind, that is, $\vec{W} = 0$, which, using (2.10), implies that $U_{\mathrm{r}} \equiv U$ and $W_{\mathrm{r}} \equiv W$. Thus, (2.3), (2.4), (2.8), and (2.9) become

$$\begin{bmatrix} m\dot{U} \\ m\dot{W} \\ I_{\mathrm{yy}}\dot{Q} \\ \dot{\theta} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} X_{\mathrm{a}} + X_{\mathrm{T}} - mg\sin\theta - mWQ \\ Z_{\mathrm{a}} + mg\cos\theta + mUQ \\ M \\ Q \\ -U\sin\theta + W\cos\theta \end{bmatrix}. \tag{2.23}$$

Next, we assume small motion in U, W, θ, X_{a}, X_{T}, Z_{a}, Z_{T}, M, and δ_{e} about a forced equilibrium. Specifically, consider the perturbations

$$\Delta U(t) \approx U(t) - U_0, \tag{2.24}$$

$$\Delta W(t) \approx W(t) - W_0, \tag{2.25}$$

$$\Delta \theta(t) \approx \theta(t) - \theta_0, \tag{2.26}$$

$$\Delta X_{\mathrm{a}}(t) \approx X_{\mathrm{a}}(t) - X_{\mathrm{a},0}, \tag{2.27}$$

$$\Delta X_{\mathrm{T}}(t) \approx X_{\mathrm{T}}(t) - X_{\mathrm{T},0}, \tag{2.28}$$

$$\Delta Z_{\mathrm{a}}(t) \approx Z_{\mathrm{a}}(t) - Z_{\mathrm{a},0}, \tag{2.29}$$

$$\Delta Z_{\mathrm{T}}(t) \approx Z_{\mathrm{T}}(t) - Z_{\mathrm{T},0}, \tag{2.30}$$

$$\Delta M(t) \approx M(t) - M_0, \tag{2.31}$$

$$\Delta \delta_{\mathrm{e}}(t) \approx \delta_{\mathrm{e}}(t) - \delta_{\mathrm{e},0}, \tag{2.32}$$

where subscript zero indicates a value at a forced equilibrium.

Evaluating (2.23) under the equilibrium condition $\dot{U} \equiv \dot{W} \equiv \dot{Q} \equiv \dot{\theta} \equiv \dot{Z} \equiv 0$ implies that

$$0 = X_{\mathrm{a},0} + X_{\mathrm{T},0} - mg \sin\theta_0, \tag{2.33}$$

$$0 = Z_{\mathrm{a},0} + Z_{\mathrm{T},0} + mg \cos\theta_0, \tag{2.34}$$

$$0 = M_0, \tag{2.35}$$

$$0 = Q, \tag{2.36}$$

$$0 = -U_0 \sin\theta_0 + W_0 \cos\theta_0. \tag{2.37}$$

Next, substituting (2.24)–(2.32) into (2.23) yields

$$\begin{aligned} m\Delta\dot{U} &= X_{\mathrm{a},0} + \Delta X_{\mathrm{a}} + X_{\mathrm{T},0} + \Delta X_{\mathrm{T}} - mg(\cos\theta_0)(\sin\Delta\theta) \\ &\quad - mg(\sin\theta_0)(\cos\Delta\theta) - mW_0Q - m\Delta WQ, \end{aligned} \tag{2.38}$$

$$\begin{aligned} m\Delta\dot{W} &= Z_{\mathrm{a},0} + \Delta Z_{\mathrm{a}} + Z_{\mathrm{T},0} + \Delta Z_{\mathrm{T}} + mg(\cos\theta_0)(\cos\Delta\theta) \\ &\quad - mg(\sin\theta_0)(\sin\Delta\theta) + mU_0Q + m\Delta UQ, \end{aligned} \tag{2.39}$$

$$I_{\mathrm{yy}}\dot{Q} = M_0 + \Delta M, \tag{2.40}$$

$$\Delta\dot{\theta} = Q, \tag{2.41}$$

$$\begin{aligned} \dot{Z} &= -[(\sin\theta_0)(\cos\Delta\theta) + (\cos\theta_0)(\sin\Delta\theta)](U_0 + \Delta U) \\ &\quad + [(\cos\theta_0)(\cos\Delta\theta) - (\sin\theta_0)(\sin\Delta\theta)](W_0 + \Delta W). \end{aligned} \tag{2.42}$$

Using the equilibrium conditions (2.33)–(2.37), small angle approximations, and eliminating higher-order terms, (2.38)–(2.42) becomes

$$\begin{bmatrix} m\Delta\dot{U} \\ m\Delta\dot{W} \\ I_{\mathrm{yy}}\dot{Q} \\ \Delta\dot{\theta} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \Delta X_{\mathrm{a}} + \Delta X_{\mathrm{T}} - mg(\cos\theta_0)\Delta\theta - mW_0 Q \\ \Delta Z_{\mathrm{a}} + \Delta Z_{\mathrm{T}} - mg(\sin\theta_0)\Delta\theta + mU_0 Q \\ \Delta M \\ Q \\ -(\sin\theta_0)\Delta U + (\cos\theta_0)\Delta W - (U_0\cos\theta_0 + W_0\sin\theta_0)\Delta\theta \end{bmatrix}. \tag{2.43}$$

Next, we model the perturbations to aerodynamic forces and moments using the first-order Taylor-series approximations

$$\Delta X_{\mathrm{a}} = \left.\frac{\partial X_{\mathrm{a}}}{\partial U}\right|_0 \Delta U + \left.\frac{\partial X_{\mathrm{a}}}{\partial W}\right|_0 \Delta W + \left.\frac{\partial X_{\mathrm{a}}}{\partial Q}\right|_0 Q + \left.\frac{\partial X_{\mathrm{a}}}{\partial \delta_{\mathrm{e}}}\right|_0 \Delta\delta_{\mathrm{e}}, \tag{2.44}$$

$$\Delta Z_{\mathrm{a}} = \left.\frac{\partial Z_{\mathrm{a}}}{\partial U}\right|_0 \Delta U + \left.\frac{\partial Z_{\mathrm{a}}}{\partial W}\right|_0 \Delta W + \left.\frac{\partial Z_{\mathrm{a}}}{\partial Q}\right|_0 Q + \left.\frac{\partial Z_{\mathrm{a}}}{\partial \delta_{\mathrm{e}}}\right|_0 \Delta\delta_{\mathrm{e}}, \tag{2.45}$$

$$\Delta M = \left.\frac{\partial M}{\partial U}\right|_0 \Delta U + \left.\frac{\partial M}{\partial W}\right|_0 \Delta W + \left.\frac{\partial M}{\partial Q}\right|_0 Q + \left.\frac{\partial M}{\partial \delta_{\mathrm{e}}}\right|_0 \Delta\delta_{\mathrm{e}}, \tag{2.46}$$

where subscript zero indicates a forced equilibrium. Finally, substituting (2.44)–(2.46) into (2.43) implies that

$$\begin{bmatrix} \Delta\dot{U} \\ \Delta\dot{W} \\ \dot{Q} \\ \Delta\dot{\theta} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\left.\frac{\partial X_{\mathrm{a}}}{\partial U}\right|_0 & \frac{1}{m}\left.\frac{\partial X_{\mathrm{a}}}{\partial W}\right|_0 & \frac{1}{m}\left.\frac{\partial X_{\mathrm{a}}}{\partial Q}\right|_0 - W_0 & -g\cos\theta_0 & 0 \\ \frac{1}{m}\left.\frac{\partial Z_{\mathrm{a}}}{\partial U}\right|_0 & \frac{1}{m}\left.\frac{\partial Z_{\mathrm{a}}}{\partial W}\right|_0 & \frac{1}{m}\left.\frac{\partial Z_{\mathrm{a}}}{\partial Q}\right|_0 + U_0 & -g\sin\theta_0 & 0 \\ \frac{1}{I_{\mathrm{yy}}}\left.\frac{\partial M}{\partial U}\right|_0 & \frac{1}{I_{\mathrm{yy}}}\left.\frac{\partial M}{\partial W}\right|_0 & \frac{1}{I_{\mathrm{yy}}}\left.\frac{\partial M}{\partial Q}\right|_0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\sin\theta_0 & \cos\theta_0 & 0 & -U_0\cos\theta_0 - W_0\sin\theta_0 & 0 \end{bmatrix} \begin{bmatrix} \Delta U \\ \Delta W \\ Q \\ \Delta\theta \\ Z \end{bmatrix}$$

$$+\begin{bmatrix} \frac{1}{m}\frac{\partial X_{\mathrm{a}}}{\partial \delta_{\mathrm{e}}}\Big|_0 & \frac{1}{m} & 0 \\ \frac{1}{m}\frac{\partial Z_{\mathrm{a}}}{\partial \delta_{\mathrm{e}}}\Big|_0 & 0 & \frac{1}{m} \\ \frac{1}{I_{\mathrm{yy}}}\frac{\partial M}{\partial \delta_{\mathrm{e}}}\Big|_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_{\mathrm{e}} \\ \Delta X_{\mathrm{T}} \\ \Delta Z_{\mathrm{T}} \end{bmatrix}. \tag{2.47}$$

Chapter 3 Background: Filtered Dynamic Inversion

In this chapter, we review the filtered-dynamic-inversion controller given in [13, 14]. Filtered dynamic inversion is a high-parameter-stabilizing controller for systems that are multiple-input multiple-output (MIMO), linear time invariant (LTI), and minimum phase (i.e., invariant zeros contained in the open-left-half complex plane). The controller requires limited model information, specifically, the plant's relative degree and first nonzero Markov parameter. In Section 3.2, we augment the filtered-dynamic-inversion controller with an anti-windup strategy.

3.1 Review of Filtered Dynamic Inversion

In this section, we present filtered dynamic inversion (FDI) and review the closed-loop stability and performance properties, which are given in [13].

3.1.1 Problem Formulation

Consider the MIMO LTI system

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t), \tag{3.1}$$

$$y(t) = Cx(t), \tag{3.2}$$

where $t \geq 0$, $x(0) \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$, $u(t) \in \mathbb{R}^m$ is the control, $y(t) \in \mathbb{R}^m$ is the output, $w(t) \in \mathbb{R}^n$ is an unmeasured disturbance, and the triple (A, B, C) is controllable and observable. Define the relative degree d as the smallest integer i such that the ith Markov parameter $H_i \overset{\triangle}{=} CA^{i-1}B$ is nonzero. We assume the disturbance w is d-times differentiable, and w, $\dot{w}$, ..., $w^{(d)}$ are bounded. We make the following assumptions:

(3.A1) If $\lambda \in \mathbb{C}$ and $\det \begin{bmatrix} \lambda I_n - A & B \\ C & 0_{m\times m} \end{bmatrix} = 0$, then Re $\lambda < 0$.

(3.A2) d is known.

(3.A3) H_d is known.

Assumption (3.A1) states that the invariant zeros of (A, B, C) are contained in the open-left-half complex plane. Assumption (3.A3) is invoked for clarity of presentation; however, [13] demonstrates that FDI is robust to uncertainty in H_d. The system (3.1) and (3.2) is otherwise unknown.

Let $\mathbf{p} = \mathrm{d}/\mathrm{d}t$ denote the differential operator, and consider the $m \times m$ polynomial matrices

$$\alpha_{\mathrm{m}}(\mathbf{p}) = \mathbf{p}^d I_m + \mathbf{p}^{d-1}\alpha_{d-1} + \cdots + \mathbf{p}\alpha_1 + \alpha_0,$$
$$\beta_{\mathrm{m}}(\mathbf{p}) = \mathbf{p}^d \beta_d + \mathbf{p}^{d-1}\beta_{d-1} + \cdots + \mathbf{p}\beta_1 + \beta_0,$$

where α_0, ..., $\alpha_{d-1} \in \mathbb{R}^{m\times m}$; β_0, ..., $\beta_d \in \mathbb{R}^{m\times m}$; and if $\lambda \in \mathbb{C}$ and $\det \alpha_{\mathrm{m}}(\lambda) = 0$, then Re $\lambda < 0$. Next, consider the reference model

$$\alpha_{\mathrm{m}}(\mathbf{p}) y_{\mathrm{m}}(t) = \beta_{\mathrm{m}}(\mathbf{p}) r(t),$$

where $t \geq 0$; $r(t) \in \mathbb{R}^m$ is the reference-model command, which is d-times differentiable and where r, $\dot{r}$, ..., $r^{(d)}$ are bounded; $y_{\mathrm{m}}(t) \in \mathbb{R}^m$ is the reference-model output and the initial condition is given by $y_{\mathrm{m}}(0)$, ..., $y_{\mathrm{m}}^{(d-1)}(0)$ and $r(0)$, ..., $r^{(d-1)}(0)$. We define the performance $z(t) \overset{\triangle}{=} y(t) - y_{\mathrm{m}}(t)$, which is the command-following error.

3.1.2 Filtered Dynamic Inversion

Define the dynamic-inversion control

$$u_* \overset{\triangle}{=} -H_d^{-1}\left[CA^d x - \beta_d r^{(d)} + \sum_{i=0}^{d-1} CA^{d-1-i} w^{(i)} - \beta_i r^{(i)} + \alpha_i y^{(i)}\right], \tag{3.3}$$

which is not implementable because u_* depends on the full state x, the disturbance w, and the plant parameters A, B, and C.

The origin of the closed-loop system (3.1) and (3.2) with $u(t) \equiv u_*(t)$, $r(t) \equiv 0$, and $w(t) \equiv 0$ is globally asymptotically stable [13, Lemma 1]. Moreover, the closed-loop system (3.1) and (3.2) with $u(t) \equiv u_*(t)$ yields a command-following error that tends to zero asymptotically, that is, $\lim_{t\to\infty} z(t) = 0$.

Let $\eta_k(s)$ be a parameter-dependent polynomial, that is, a polynomial in s over the reals whose coefficients are functions of a real parameter k. Furthermore, let $\eta_k(s)$ be monic with degree $\rho \geq d$. Thus, $\eta_k(s)$ can be written as

$$\eta_k(s) = s^\rho + \eta_{\rho-1,k}s^{\rho-1} + \cdots + \eta_{1,k}s + \eta_{0,k},$$

where for all $k \in [0,\infty)$, $\eta_{0,k}$, ..., $\eta_{\rho-1,k} \in \mathbb{R}$. Define the parameter-dependent polynomial

$$\bar{\eta}_k(s) \overset{\triangle}{=} s^{\rho-1} + \eta_{\rho-1,k}s^{\rho-2} + \cdots + \eta_{2,k}s + \eta_{1,k}.$$

We impose the following conditions on $\eta_k(s)$:

(3.C1) There exists $k_0 > 0$, such that for all $k > k_0$, $\eta_k(s)$ is Hurwitz.

(3.C2) For all $\epsilon > 0$, there exists $k_\epsilon > k_0$ such that for all $k > k_\epsilon$,

$$\sup_{\omega\in\mathbb{R}} \left|\frac{\bar{\eta}_k(j\omega)}{\eta_k(j\omega)}\right| < \epsilon.$$

Note that $\eta_k(s) = (s + k)^\rho$ satisfies (3.C1) and (3.C2). See [13] for other choices of $\eta_k(s)$ that satisfy (3.C1) and (3.C2).

Consider the control u that satisfies

$$\eta_k(\mathbf{p})u = \eta_k(0)u_*, \tag{3.4}$$

where u_* is given by (3.3). Taking the dth derivative of (3.2) and using (3.1) yields

$$y^{(d)} = H_d u + CA^d x + \sum_{i=0}^{d-1} CA^{d-1-i} w^{(i)}. \tag{3.5}$$

To express (3.4) as an implementable control, it follows from (3.5) and (3.3) that

$$\begin{aligned} u_* &= -H_d^{-1}\left[y^{(d)} - H_d u - \beta_d r^{(d)} + \sum_{i=0}^{d-1} \alpha_i y^{(i)} - \beta_i r^{(i)}\right] \\ &= u - H_d^{-1}[\alpha_{\mathrm{m}}(\mathbf{p})y - \beta_{\mathrm{m}}(\mathbf{p})r]. \end{aligned} \tag{3.6}$$

Combining (3.4) and (3.6) yields the FDI controller

$$\mathbf{p}\bar{\eta}_k(\mathbf{p})u(t) = \eta_{0,k} H_d^{-1}[\beta_{\mathrm{m}}(\mathbf{p})r(t) - \alpha_{\mathrm{m}}(\mathbf{p})y(t)]. \tag{3.7}$$

For all $k > k_0$, let the FDI controller have the ρm-order state-space realization

$$\dot{x}_{\mathrm{c}} = A_{\mathrm{c}} x_{\mathrm{c}} + B_{\mathrm{c}} y + E_{\mathrm{c}} r, \tag{3.8}$$

$$u = C_{\mathrm{c}} x_{\mathrm{c}} + D_{\mathrm{c}} y + F_{\mathrm{c}} r, \tag{3.9}$$

where $x_{\mathrm{c}}(t) \in \mathbb{R}^{\rho m}$, and for all $k > k_0$, $A_{\mathrm{c}} \in \mathbb{R}^{\rho m \times \rho m}$, $B_{\mathrm{c}} \in \mathbb{R}^{\rho m \times m}$, $C_{\mathrm{c}} \in \mathbb{R}^{m \times \rho m}$, $D_{\mathrm{c}} \in \mathbb{R}^{m \times m}$, $E_{\mathrm{c}} \in \mathbb{R}^{\rho m \times m}$, and $F_{\mathrm{c}} \in \mathbb{R}^{m \times m}$. To see that such a realization exists, consider the block-observable realization given in [13]. The closed-loop system (3.1), (3.2), (3.8), and (3.9) is given by

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{\rm c} \end{bmatrix} = \tilde{A} \begin{bmatrix} x \\ x_{\rm c} \end{bmatrix} + \begin{bmatrix} BF_{\rm c} \\ E_{\rm c} \end{bmatrix} r + \begin{bmatrix} I_n \\ 0_{\rho m \times n} \end{bmatrix} w, \tag{3.10}$$

$$y = \begin{bmatrix} C & 0_{m\times \rho m} \end{bmatrix} \begin{bmatrix} x \\ x_{\rm c} \end{bmatrix}, \tag{3.11}$$

where

$$\tilde{A} \triangleq \begin{bmatrix} A + BD_{\rm c}B & BC_{\rm c} \\ B_{\rm c}C & A_{\rm c} \end{bmatrix}. \tag{3.12}$$

The following result from [13] characterizes the stability and performance of the closed-loop system.

Theorem 3.1. *Consider the closed-loop system (3.10)–(3.12) that consists of (3.1), (3.2), (3.8), and (3.9). Assume (3.A1)–(3.A3) are satisfied, and assume $\eta_k(s)$ satisfies (3.C1) and (3.C2). Then, the following statements hold:*

(i) There exists $k_s > k_0$ such that for all $k > k_{\rm s}$, $\tilde{A}$ is asymptotically stable.

(ii) For all $\delta > 0$, there exists $k_\delta > k_{\rm s}$ such that for all $k > k_\delta$,

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T z^{\rm T}(t) z(t) {\rm d}t < \delta.$$

Part (i) of Theorem 3.1 states that FDI is a high-parameter-stabilizing controller, specifically, there is a minimum parameter k above which $\tilde{A}$ is asymptotically stable. Part (ii) of Theorem 3.1 states that the average power of the performance is made arbitrarily small by a sufficiently large parameter k.

3.2 Filtered Dynamic Inversion with Anti Windup

We first specialize the FDI controller to single-input single-output (SISO) systems. Next, we augment the SISO FDI controller, which has integral action, with an anti-windup strategy.

We now specialize the FDI controller to the case where $m = 1$, $\rho > d$, and $\alpha_{\rm m}(s) = \beta_{\rm m}(s)$. In this case, (3.7) becomes

$$\mathbf{p}\bar{\eta}_k(\mathbf{p})u(t) = \frac{\eta_{0,k}}{H_d}\alpha_{\rm m}(\mathbf{p})e(t),$$

where $e(t) \overset{\triangle}{=} r(t) - y(t)$. Thus, the transfer function from e to u is

$$G_{\rm c}(s) = \frac{\eta_{0,k}\alpha_{\rm m}(s)}{H_d s\bar{\eta}_k(s)},$$

which has integral action, that is, a pole at zero.

We use partial fraction expansion to separate the pole at zero from the poles at the roots of $\bar{\eta}_k(s)$. Specifically, $G_{\rm c}$ can be expressed as

$$G_{\rm c}(s) = \frac{\eta_{0,k}}{H_d}\left(\frac{k_{\rm I}}{s} + \frac{c_{d-1}s^{d-1} + c_{d-2}s^{d-2} + \cdots + c_1 s + c_0}{\bar{\eta}_k(s)}\right), \tag{3.13}$$

where

$$k_{\rm I} = \frac{\alpha_{\rm m}(0)}{\bar{\eta}_k(0)}, \tag{3.14}$$

and, for $i = 0, 1, \cdots, d-1$,

$$c_i = \alpha_{i+1} - k_{\rm I}\eta_{i+2,k}. \tag{3.15}$$

Next, it follows from (3.13)–(3.15) that $G_{\rm c}$ has the state space realization (3.8) and

(3.9) where

$$A_{\rm c} = \begin{bmatrix} -\eta_{\rho-1,k} & -\eta_{\rho-2,k} & \cdots & -\eta_{2,k} & -\eta_{1,k} & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}, \quad E_{\rm c} = \begin{bmatrix} 1 \\ 0_{1\times\rho-2} \\ k_{\rm I} \end{bmatrix}, \tag{3.16}$$

$$B_{\rm c} = -E_{\rm c}, \qquad C_{\rm c} = \frac{\eta_{0,k}}{H_d}\left[0_{1\times\rho-d-1}\ c_{d-1} \quad c_{d-2} \quad \cdots \quad c_0 \quad 1\right], \tag{3.17}$$

$$D_{\rm c} = 0, \qquad F_{\rm c} = 0. \tag{3.18}$$

We now augment the FDI controller (3.8), (3.9), and (3.16)–(3.18) with an anti-windup strategy to mitigate integrator windup effects that can occur due to control saturation.

Let $u_{\rm sat}(t)$ denote $u(t)$ after it has been saturated between a lower limit $u_{\rm min} \in \mathbb{R}$ and an upper limit $u_{\rm max} > u_{\rm min}$. Specifically, $u_{\rm sat}(t) \triangleq {\rm sat}(u(t))$, where

$${\rm sat}(u) \triangleq \begin{cases} u_{\rm min}, & u \le u_{\rm min}, \\ u, & u_{\rm min} < u < u_{\rm max}, \\ u_{\rm max}, & u \ge u_{\rm max}. \end{cases} \tag{3.19}$$

To mitigate the effect of saturation, we use the back-calculation anti-windup method [18–20], which is shown in Figure 3.1. The FDI controller augmented with back-calculation is given by

$$\dot{x}_{\rm c} = A_{\rm c}x_{\rm c} + E_{\rm c}e + \begin{bmatrix} 0_{\rho-1} \\ -K_{\rm b} \end{bmatrix} (u - u_{\rm sat}), \tag{3.20}$$

$$u = C_{\rm c}x_{\rm c}, \tag{3.21}$$

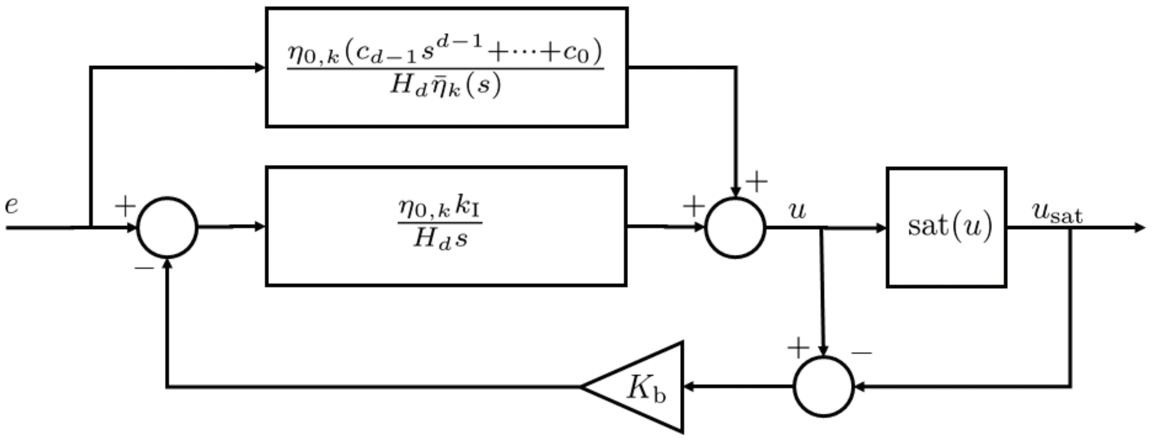

$\frac{\eta_{0,k}(c_{d-1}s^{d-1}+\cdots+c_0)}{H_d\bar{\eta}_k(s)}$
e
$+$
$-$
$\frac{\eta_{0,k}k_{\rm I}}{H_d s}$
$+$
$+$
u
$\mathrm{sat}(u)$
$u_{\rm sat}$
$+$
$-$
$K_{\rm b}$

Figure 3.1: *Filtered Dynamic Inversion with Back-Calculation Anti-Windup Strategy.* To mitigate integrator windup, back calculation uses feedback to augment the error signal, which is sent to the integrator. Specifically, the input to the integrator is $e - K_{\rm b}(u - u_{\rm sat})$, instead of e for FDI without back calculation.

where $A_{\rm c}$, $E_{\rm c}$, and $C_{\rm c}$ are given by (3.16)–(3.18).

3.2.1 Numerical Examples

We now provide numerical examples of the FDI controller with and without back calculation. Consider the serially connected two-mass structure shown in Figure 3.2 where $u_{\rm sat}$ is a control force on the second mass. The equations of motion for the system shown in Figure 3.2 are

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \ddot{q} + \begin{bmatrix} b_1 + b_2 & -b_2 \\ -b_2 & b_2 \end{bmatrix} \dot{q} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} q = \begin{bmatrix} 0 \\ u_{\rm sat} \end{bmatrix}, \qquad (3.22)$$

where $q \overset{\triangle}{=} [q_1 \quad q_2]^{\rm T}$. The model parameters are arbitrary positive numbers. For this section, the masses are $m_1 = 1$ kg and $m_2 = 1$ kg, the damping coefficients are $b_1 = 2$ kg/s and $b_2 = 0.5$ kg/s, and the spring constants are $k_1 = 3$ kg/s^2 and $k_2 = 1$ kg/s^2.

Let $y = q_2$ be the measurement. The transfer function from $u_{\rm sat}$ to y is given by

$$G(s) = \frac{s^2 + 2.5s + 4}{s^4 + 3s^3 + 6s^2 + 3.5s + 3},$$

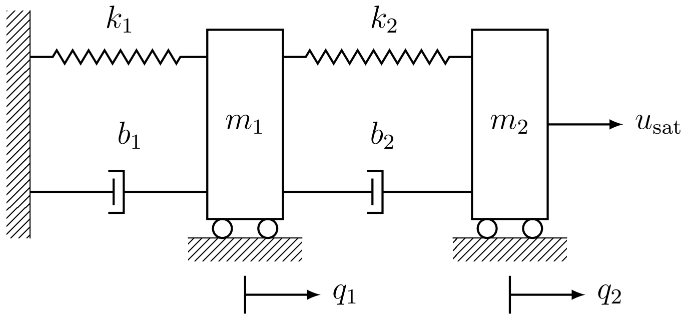


Figure 3.2: *Serially Connected Two-Mass Structure.* The transfer function G from $u_{\rm sat}$ to $y = q_2$ is fourth-order and relative degree two.

which implies that the relative degree is $d = 2$, and the first nonzero Markov parameter is $H_d = 1$. Furthermore, the zeros of G are contained in the open-left-half complex plane. Thus, (3.22) satisfies assumptions (3.A1)–(3.A3). The control objective is to have the output $y(t)$ asymptotically follow the reference $r(t) = 0.5 \sin t$. The disturbance $w(t)$ is zero.

Example 3.1. *No saturation on $u_{\rm sat}$ and FDI without back calculation.* Consider the SISO FDI controller (3.8), (3.9), (3.16)–(3.18), where $\alpha_{\rm m}(s) = \beta_{\rm m}(s) = s^2+2.8s+4$ and $\eta_k(s) = (s + k)^3$, which satisfies (3.C1) and (3.C2). Let $k = 20$, and let $u_{\rm sat} = {\rm sat}(u)$, where $u_{\rm min} = -\infty$ and $u_{\rm max} = \infty$, that is, the input is unsaturated, or equivalently $u_{\rm sat} = u$. Figure 3.3 shows the time history of y, r, e, and u. After an initial transient, the error e is small. In steady-state, the maximum value of the control signal u is approximately 0.27. △

Example 3.2. *Saturation on $u_{\rm sat}$ and FDI without back calculation.* Consider the SISO FDI controller (3.8), (3.9), (3.16)–(3.18), where $\eta_k(s)$, k, and $\alpha_{\rm m}(s)$ are the same as in Example 3.1. However, let $u_{\rm sat} = {\rm sat}(u)$, where $u_{\rm min} = -0.2$ and $u_{\rm max} = 0.2$. Figure 3.4 shows the time history of y, r, e, u, and $u_{\rm sat}$. The error e is large compared to the error in Example 3.1, which does not have saturation. The control u is large compared to $u_{\rm sat}$, and the output y lags the command r. △

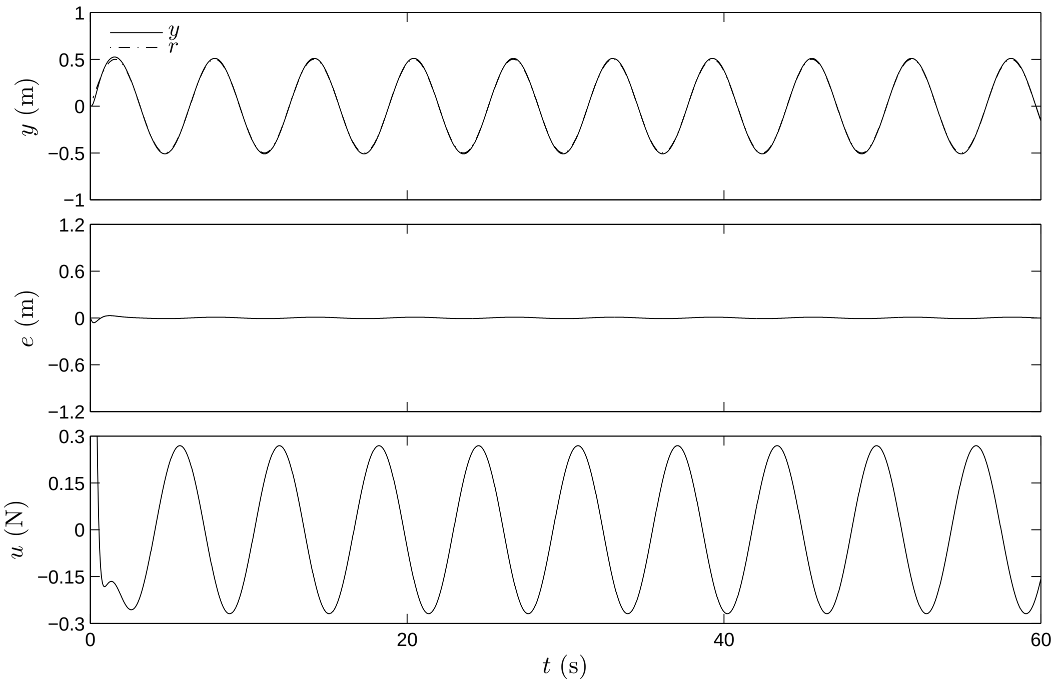


Figure 3.3: *Response without Saturation.* After an initial transient, the FDI controller makes the command-following error $e = r - y$ small.

Example 3.3. *Saturation on $u_{\rm sat}$ and FDI with back calculation.* Consider the SISO FDI controller (3.16)–(3.18), (3.20) and (3.21), where $\eta_k(s)$, k, and $\alpha_{\rm m}(s)$, $u_{\rm min}$, and $u_{\rm max}$ are the same as in Example 3.1. Let $K_{\rm b} = 10$. Figure 3.5 shows the time history of y, r, e, u, and $u_{\rm sat}$. The output y follows the command r better than in Example 3.2. The controller output u does not exceed the input $u_{\rm sat}$ by as much as in Example 3.2. $\triangle$

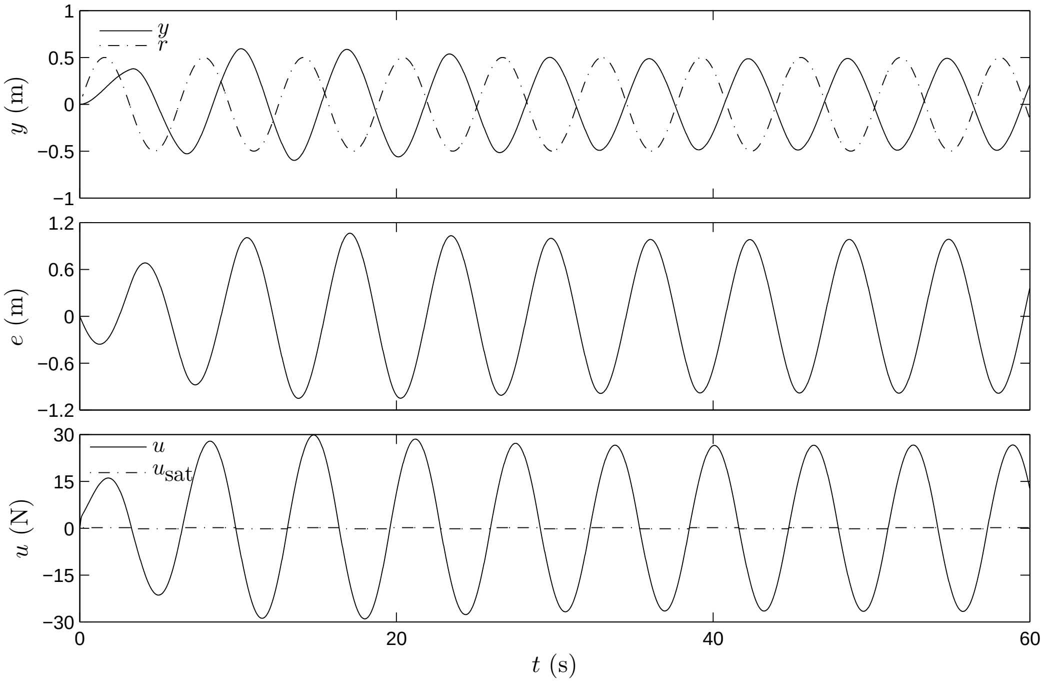


Figure 3.4: *Response with Saturation without Anti-windup.* With saturation on the controller's output, the controller is less able to follow the command than in Example 3.1, where there was no saturation. The controller's output u is large compared to the saturated value u_{sat}, and the output y lags the command r.

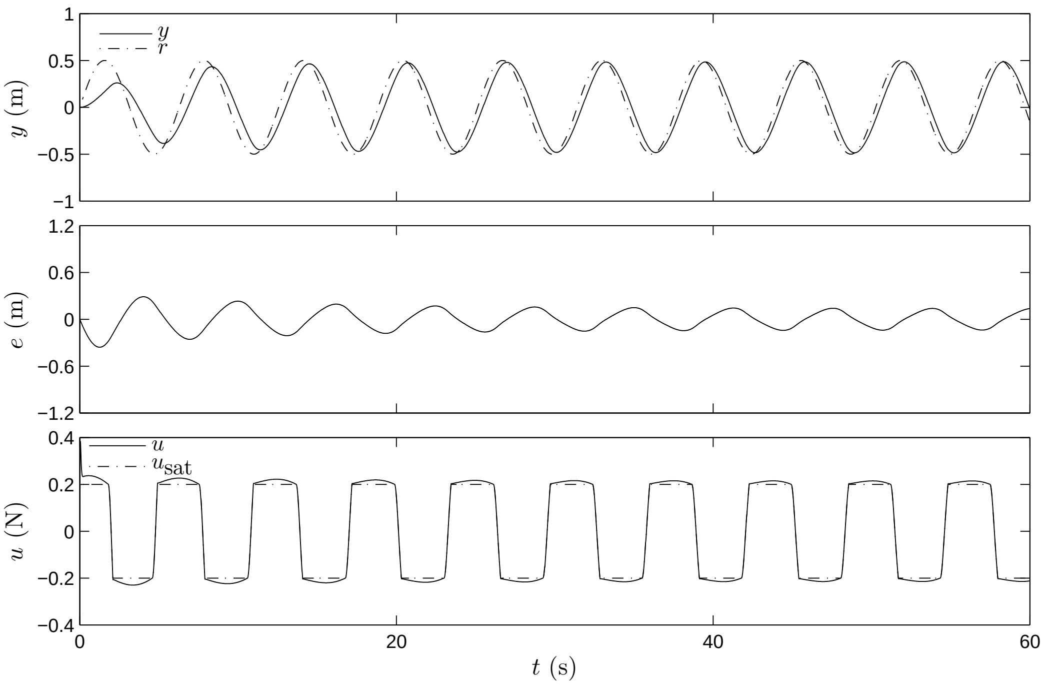


Figure 3.5: *Response with Saturation and Anti-windup.* If the FDI controller is augmented with back calculation, then the controller output u does not exceed the saturation value u_{sat} as much as in Example 3.2. The command following is improved relative to Example 3.2.

Chapter 4 Test Platform: The AeroWorks EDGE 540T

In this chapter, we describe the AeroWorks EDGE 540T fixed-wing aircraft that is the experimental testbed for the FDI controller. We compute the model parameters of the test platform required by (2.17)–(2.22) using Athena Vortex Lattice (AVL) software [21].

4.1 Airframe Description

The AeroWorks EDGE 540T (the "EDGE") is the mid-winged, aerobatic, remote-controlled aircraft, shown in Figure 4.1. The EDGE has a straight, tapered wing with span $b_{\rm r} = 1.52$ m, mean chord length $c_{\rm r} = 0.2975$ m, and planform area $S_{\rm r} = 0.4534$ m^2. The leading edge of the horizontal stabilizer is located 0.806 m from the leading edge of the wing. The EDGE has mass $m = 4.48$ kg at takeoff. The center of gravity is located 69.9 mm aft of the wing's leading edge, in accordance with the manufacturer's specifications [22]. For propulsion, the plane is fitted with an Electrifly RimFire 0.80 brushless outboard electric motor, which is rated for 1300 W constant output. The motor is mounted along the $\hat{\imath}_{\rm B}$ axis, and we assume that the thrust force acts along the $\hat{\imath}_{\rm B}$ axis. The electric motor draws power from two 8S lithium-polymer batteries, each with capacity 5000 mAh. This setup is capable of nine minutes of flight. The principle moments of inertia were measured using the experimental procedure in [23]; they are $I_{\rm xx} = 0.1778$ kg-m^2, $I_{\rm yy} = 0.3287$ kg-m^2, and $I_{\rm zz} = 0.4231$ kg-m^2.

We use the low-rate control deflections recommended by the manufacturer. Thus, the elevator is limited to a 15° deflection. The elevator servo is a Hitec HS-645MG

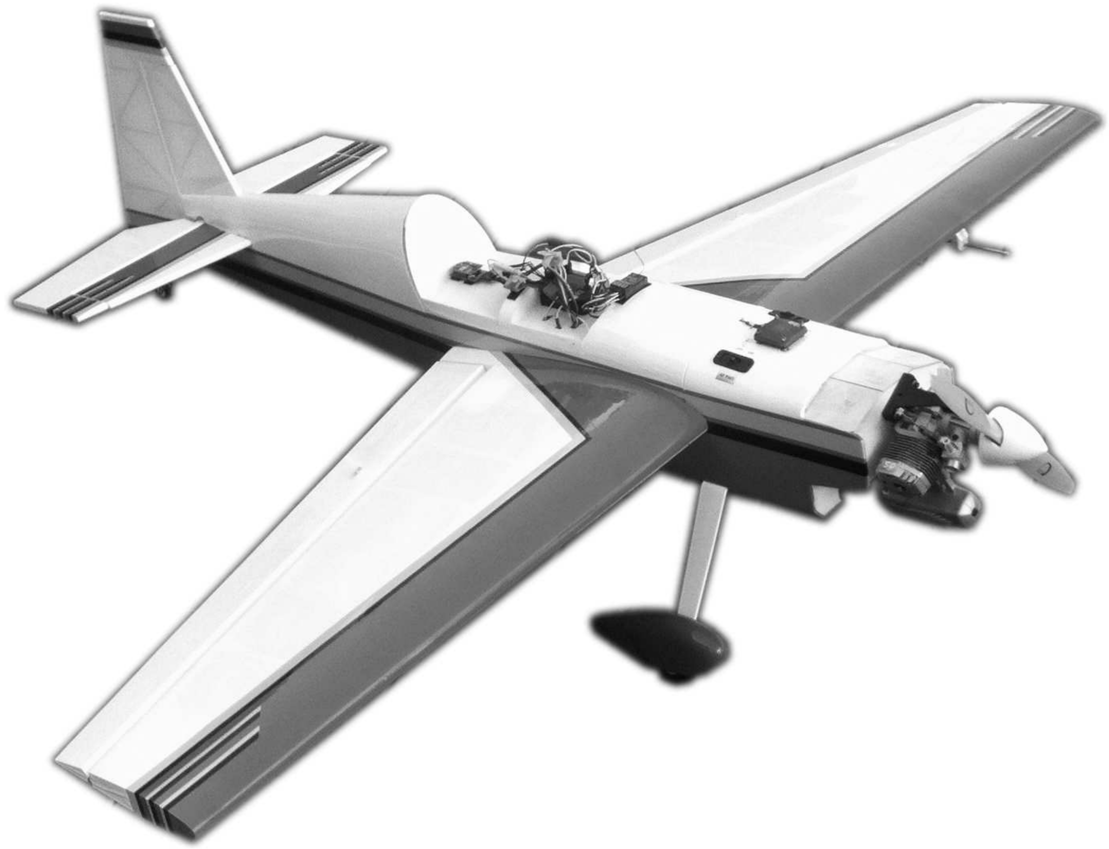


Figure 4.1: *The AeroWorks EDGE 540T.* The EDGE is a mid-winged, aerobatic, remote-controlled aircraft with a 1.52 m wingspan.

High-Torque 2BB Metal Gear servo, which has a maximum angular speed of 300°/s.

We equip the EDGE with an Ardupilot Mega 2.5, which is an open-source autopilot based on the Arduino computing platform [24]. The Ardupilot features an Invensense MPU-6000 six-axis accelerometer and gryoscope, Measurement Specialties MS5611-01BA03 barometer, Honeywell HMC5883L-TR magnetometer, and uBlox LEA-6H GPS system. In addition, a Pitot-static probe and pressure transducer provide airspeed sensing. The pressure transducer is a Freescale Semiconductor MPXV7002, which has a ± 2 kPa range, which roughly corresponds to a 55 m/s stagnation velocity.

The Ardupilot runs a software packaged called Arduplane, which is open-source.

The altitude h, latitude, longitude, airspeed U_r, Euler angles ϕ, θ, ψ, and angular rates P, Q, R are available for feedback.

4.2 Generating Model Parameters from Athena Vortex Lattice

We use AVL [21] to model the EDGE and generate aerodynamic model parameters. AVL uses the vortex lattice method to estimate the aerodynamic pressure distribution, and therefore aerodynamic force and moment, on a lifting body [25].

We use AVL in two different ways. First, we use AVL to estimate the first-order Taylor series expansion of the aerodynamic force and moment about a variety of airspeeds, angles of attack, and sideslip angles. Thus, for the nonlinear model (2.3), (2.4), (2.8), (2.9), and (2.17)–(2.22), we use AVL to build lookup tables, which are necessary to estimate the aerodynamic forces and moments (2.17)–(2.22). As inputs to AVL, we use six airspeeds $V_\mathrm{T} = \{10, 15, 20, 25, 30, 35\}$ m/s, thirteen angles of attack $\alpha = \{-30, -20, -12, -6, -4, -2, 0, 2, 4, 6, 12, 20, 30\}^\circ$, and seven sideslip angles $\beta = \{-30, -12, -4, 0, 4, 12, 30\}^\circ$. We use these inputs in a batch process to estimate 33 model parameters at 524 combinations of airspeed, angle of attack, and sideslip angle. The AVL estimates are dimensionless, and they are related to their dimensional quantities by

$$\begin{aligned}
CX_\mathrm{tot} &\approx \frac{1}{qS_\mathrm{r}} X_{\mathrm{a},0}, & CZ_\mathrm{tot} &\approx \frac{1}{qS_\mathrm{r}} Z_{\mathrm{a},0}, & CM_\mathrm{tot} &\approx \frac{1}{qS_\mathrm{r}c_\mathrm{r}} M_0,\\
CX_Q &\approx \frac{2U_0}{qS_\mathrm{r}c_\mathrm{r}} \frac{\partial X_\mathrm{a}}{\partial Q}, & CZ_Q &\approx \frac{2U_0}{qS_\mathrm{r}c_\mathrm{r}} \frac{\partial Z_\mathrm{a}}{\partial Q}, & CM_Q &\approx \frac{2U_0}{qS_\mathrm{r}c_\mathrm{r}^2} \frac{\partial M}{\partial Q},\\
CX_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}} \frac{\partial X_\mathrm{a}}{\partial \delta_\mathrm{e}}, & CZ_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}} \frac{\partial Z_\mathrm{a}}{\partial \delta_\mathrm{e}}, & CM_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}c_\mathrm{r}} \frac{\partial M}{\partial \delta_\mathrm{e}},\\
CX_{\delta_\mathrm{r}} &\approx \frac{1}{qS_\mathrm{r}} \frac{\partial X_\mathrm{a}}{\partial \delta_\mathrm{r}}, & CZ_{\delta_\mathrm{r}} &\approx \frac{1}{qS_\mathrm{r}} \frac{\partial Z_\mathrm{a}}{\partial \delta_\mathrm{r}}, & CM_{\delta_\mathrm{r}} &\approx \frac{1}{qS_\mathrm{r}c_\mathrm{r}} \frac{\partial M}{\partial \delta_\mathrm{r}},\\
CX_{\delta_\mathrm{a}} &\approx \frac{1}{qS_\mathrm{r}} \frac{\partial X_\mathrm{a}}{\partial \delta_\mathrm{a}}, & CZ_{\delta_\mathrm{a}} &\approx \frac{1}{qS_\mathrm{r}} \frac{\partial Z_\mathrm{a}}{\partial \delta_\mathrm{a}}, & CM_{\delta_\mathrm{a}} &\approx \frac{1}{qS_\mathrm{r}c_\mathrm{r}} \frac{\partial M}{\partial \delta_\mathrm{a}},\\
CY_\mathrm{tot} &\approx \frac{1}{qS_\mathrm{r}} Y_{\mathrm{a},0}, & CL_\mathrm{tot} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}} L_0, & CN_\mathrm{tot} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}} N_0,
\end{aligned}$$

$$\begin{aligned}
CY_P &\approx \frac{2U_0}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial Y_\mathrm{a}}{\partial P}, & CL_P &\approx \frac{2U_0}{qS_\mathrm{r}b_\mathrm{r}^2}\frac{\partial L}{\partial P}, & CN_P &\approx \frac{2U_0}{qS_\mathrm{r}b_\mathrm{r}^2}\frac{\partial N}{\partial P},\\
CY_R &\approx \frac{2U_0}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial Y_\mathrm{a}}{\partial R}, & CL_R &\approx \frac{2U_0}{qS_\mathrm{r}b_\mathrm{r}^2}\frac{\partial L}{\partial R}, & CN_R &\approx \frac{2U_0}{qS_\mathrm{r}b_\mathrm{r}^2}\frac{\partial N}{\partial R},\\
CY_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}}\frac{\partial Y_\mathrm{a}}{\partial \delta_\mathrm{e}}, & CL_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial L}{\partial \delta_\mathrm{e}}, & CN_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial N}{\partial \delta_\mathrm{e}},\\
CY_{\delta_\mathrm{r}} &\approx \frac{1}{qS_\mathrm{r}}\frac{\partial Y_\mathrm{a}}{\partial \delta_\mathrm{r}}, & CL_{\delta_\mathrm{r}} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial L}{\partial \delta_\mathrm{r}}, & CN_{\delta_\mathrm{r}} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial N}{\partial \delta_\mathrm{r}},\\
CY_{\delta_\mathrm{a}} &\approx \frac{1}{qS_\mathrm{r}}\frac{\partial Y_\mathrm{a}}{\partial \delta_\mathrm{a}}, & CL_{\delta_\mathrm{a}} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial L}{\partial \delta_\mathrm{a}}, & CN_{\delta_\mathrm{a}} &\approx \frac{1}{qS_\mathrm{r}b_\mathrm{r}}\frac{\partial N}{\partial \delta_\mathrm{a}},
\end{aligned}$$

where $q \overset{\triangle}{=} \frac{1}{2}\rho_\mathrm{a}V_\mathrm{T}^2$, ρ_a is air density, and the AVL estimate appears on the left-hand side of each approximation. The dimensionless parameters are shown as functions of angle of attack α and sideslip angle β for $V_\mathrm{T} = 20$ m/s in Appendix A.

AVL is also used to estimate steady level flight equilibria and estimate the stability derivatives we use in the linearized model. The AVL estimates are dimensionless and appear on the left-hand side of each of the approximations

$$\begin{aligned}
CX_U &\approx \frac{U_0}{qS_\mathrm{r}}\frac{\partial X_\mathrm{a}}{\partial U}, & CZ_U &\approx \frac{U_0}{qS_\mathrm{r}}\frac{\partial Z_\mathrm{a}}{\partial U}, & CM_U &\approx \frac{U_0}{qS_\mathrm{r}c_\mathrm{r}}\frac{\partial M}{\partial U},\\
CX_W &\approx \frac{U_0}{qS_\mathrm{r}}\frac{\partial X_\mathrm{a}}{\partial W}, & CZ_W &\approx \frac{U_0}{qS_\mathrm{r}}\frac{\partial Z_\mathrm{a}}{\partial W}, & CM_W &\approx \frac{U_0}{qS_\mathrm{r}c_\mathrm{r}}\frac{\partial M}{\partial W},\\
CX_Q &\approx \frac{2U_0}{qS_\mathrm{r}c_\mathrm{r}}\frac{\partial X_\mathrm{a}}{\partial Q}, & CZ_Q &\approx \frac{2U_0}{qS_\mathrm{r}c_\mathrm{r}}\frac{\partial Z_\mathrm{a}}{\partial Q}, & CM_Q &\approx \frac{2U_0}{qS_\mathrm{r}c_\mathrm{r}^2}\frac{\partial M}{\partial Q},\\
CX_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}}\frac{\partial X_\mathrm{a}}{\partial \delta_\mathrm{e}}, & CZ_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}}\frac{\partial Z_\mathrm{a}}{\partial \delta_\mathrm{e}}, & CM_{\delta_\mathrm{e}} &\approx \frac{1}{qS_\mathrm{r}c_\mathrm{r}}\frac{\partial M}{\partial \delta_\mathrm{e}},
\end{aligned}$$

which are tabulated in Appendix B for the steady, constant-altitude, wings-level forced equilibrium with $V_\mathrm{T} = 20$ m/s.

Chapter 5 Continuous-Time Filtered Dynamic Inversion with Nonlinear Aircraft Model

5.1 Longitudinal Autopilot

For autonomous longitudinal flight, we require a system, shown schematically in Figure 5.1, that uses mission information, GPS measurements, and other feedback signals to generate throttle and elevator commands. This system is often described in terms of three subsystems: navigation, guidance, and control [26]. The navigation system uses mission information and GPS measurements to generate an altitude command h_{d}. The guidance system uses the altitude command h_{d} and measured altitude $h \overset{\triangle}{=} -Z$ to generate a speed command U_{d} and a pitch command θ_{d}. The speed control system uses the measured speed U_{r} and commanded speed U_{d} to generate a throttle command. Finally, the pitch control system uses the measured pitch θ and commanded pitch angle θ_{d} to generate a servo command u_{e} for the elevator.

We are interested in improving the altitude-command-following behavior of the aircraft. Let $\mathcal{P}_h(t_1, t_0)$ denote the average power of altitude error over the finite time interval $[t_0, t_1]$, that is,

$$\mathcal{P}_h(t_1, t_0) \overset{\triangle}{=} \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} [h_{\mathrm{d}}(\tau) - h(\tau)]^2 \mathrm{d}\tau.$$

Moreover, define the average power of the altitude error over $[t_0, t_1]$,

$$\mathcal{P}_\theta(t_1, t_0) \overset{\triangle}{=} \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} [\theta_{\mathrm{d}}(\tau) - \theta(\tau)]^2 \mathrm{d}\tau.$$

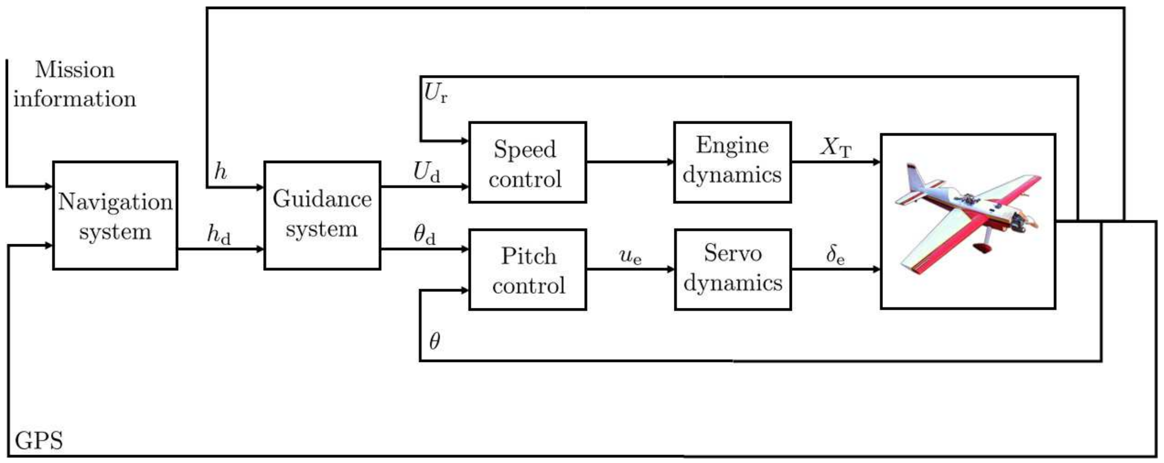


Figure 5.1: *Longitudinal Navigation, Guidance, and Control Systems.* The pitch and speed controllers form an inner loop for the guidance system, which uses the altitude error to generate a pitch command.

We propose improving altitude performance by implementing the FDI controller as the pitch controller. The main result of [13] implies that implementing the filtered dynamic inversion controller as the pitch controller allows us to make $\lim_{t_1 \to \infty} \mathcal{P}_\theta(t_1, t_0)$ arbitrarily small.

In the following sections, we describe the control loops.

5.1.1 Speed Control Inner Loop

For autonomous flight, we require a closed-loop speed controller. The EDGE is equipped with a Pitot probe and pressure transducer that measure U_r (approximately). We develop a closed-loop speed controller for use in simulation. Recall the thrust force acts solely in the $\hat{\imath}_\mathrm{B}$ direction, which implies that $Z_\mathrm{T} \equiv 0$. Since we lack a model of the throttle-to-speed dynamics, we design the $\hat{\imath}_\mathrm{B}$ thrust force X_T directly. Thus, the controller cannot be implemented on the EDGE, because the EDGE speed controller has percent throttle as its output. However, we assume that in practice we can design a control with the same closed-loop properties.

For speed control, we use a proportional-integral (PI) controller,

$$X_{\mathrm{T}}(t) = k_{\mathrm{T,P}}(U_{\mathrm{d}}(t) - U_{\mathrm{r}}(t)) + k_{\mathrm{T,I}} \int_0^t [U_{\mathrm{d}}(\tau) - U_{\mathrm{r}}(\tau)]\mathrm{d}\tau + k_{\mathrm{T,0}}, \tag{5.1}$$

where $k_{\mathrm{T,P}} \in \mathbb{R}$ is a proportional gain, $k_{\mathrm{T,I}} \in \mathbb{R}$ is an integral gain, and $k_{\mathrm{T,0}} \in \mathbb{R}$ is a constant bias.

5.1.2 Linearized Elevator-to-Pitch Dynamics with Speed Control

We now implement the speed controller (5.1) with the linearized longitudinal dynamics (2.47). We assume that $\vec{W} \equiv 0$, which implies that $U_{\mathrm{r}} \equiv U$. We also assume that $k_{\mathrm{T,0}} = X_{\mathrm{T,0}}$. Let $\Delta U_{\mathrm{d}}(t) \overset{\triangle}{=} U_{\mathrm{d}}(t) - U_{\mathrm{d,0}}$, where $U_{\mathrm{d,0}} = U_0$, and (5.1) becomes

$$\Delta X_{\mathrm{T}}(t) = k_{\mathrm{T,P}}(\Delta U_{\mathrm{d}}(t) - \Delta U(t)) + k_{\mathrm{T,I}} \int_0^t [\Delta U_{\mathrm{d}}(\tau) - \Delta U(\tau)]\mathrm{d}\tau. \tag{5.2}$$

Substituting (5.2) into (2.47) yields

$$\dot{x} = Ax + B_1 \Delta\delta_{\mathrm{e}} + B_2 \Delta U_{\mathrm{d}}, \tag{5.3}$$

where

$$x \overset{\triangle}{=} [\Delta U \quad \Delta W \quad Q \quad \Delta\theta \quad Z \quad x_{\mathrm{T,I}}]^{\mathrm{T}}, \tag{5.4}$$

$$A \overset{\triangle}{=} \begin{bmatrix} \frac{1}{m}\left(\frac{\partial X_{\mathrm{a}}}{\partial U}\Big|_0 - k_{\mathrm{T,P}}\right) & \frac{1}{m}\frac{\partial X_{\mathrm{a}}}{\partial W}\Big|_0 & \frac{1}{m}\frac{\partial X_{\mathrm{a}}}{\partial Q}\Big|_0 - W_0 & -g\cos\theta_0 & 0 & \frac{1}{m}k_{\mathrm{T,I}} \\ \frac{1}{m}\frac{\partial Z_{\mathrm{a}}}{\partial U}\Big|_0 & \frac{1}{m}\frac{\partial Z_{\mathrm{a}}}{\partial W}\Big|_0 & \frac{1}{m}\frac{\partial Z_{\mathrm{a}}}{\partial Q}\Big|_0 + U_0 & -g\sin\theta_0 & 0 & 0 \\ \frac{1}{I_{\mathrm{yy}}}\frac{\partial M}{\partial U}\Big|_0 & \frac{1}{I_{\mathrm{yy}}}\frac{\partial M}{\partial W}\Big|_0 & \frac{1}{I_{\mathrm{yy}}}\frac{\partial M}{\partial Q}\Big|_0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\sin\theta_0 & \cos\theta_0 & 0 & -U_0\cos\theta_0 - W_0\sin\theta_0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{5.5}$$

$$B_1 \triangleq \begin{bmatrix} \frac{1}{m}\frac{\partial X_\mathrm{a}}{\partial \delta_\mathrm{e}}\Big|_0 & \frac{1}{m}\frac{\partial Z_\mathrm{a}}{\partial \delta_\mathrm{e}}\Big|_0 & \frac{1}{I_\mathrm{yy}}\frac{\partial M}{\partial \delta_\mathrm{e}}\Big|_0 & 0 & 0 & 0 \end{bmatrix}^\mathrm{T}, \tag{5.6}$$

$$B_2 \triangleq \begin{bmatrix} \frac{1}{m}k_\mathrm{T,P} & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\mathrm{T}, \tag{5.7}$$

and $x_\mathrm{T,I}$ is the integrator state of the controller (5.2).

Now, we compute the linearized longitudinal dynamics of the aircraft model presented in Chapter 4. We verify that the linearized elevator-command-to-pitch dynamics satisfy the FDI assumptions (3.A1)–(3.A3). First, recall that we desire near-constant airspeed during measurement. Thus, assume constant airspeed command, that is, $\Delta U_\mathrm{d} \equiv 0$.

To approximate the servo dynamics, let δ_e satisfy

$$\tau_\mathrm{e}\dot{\delta}_\mathrm{e} + \delta_\mathrm{e} = u_\mathrm{e},$$

where u_e is the commanded elevator deflection, and τ_e is a time constant associated with the elevator servo. Define $\Delta u_\mathrm{e}(t) \triangleq u_\mathrm{e}(t) - u_\mathrm{e,0}$, where $u_\mathrm{e,0} = \delta_\mathrm{e,0}$. Thus,

$$\tau_\mathrm{e}\Delta\dot{\delta}_\mathrm{e} + \Delta\delta_\mathrm{e} = \Delta u_\mathrm{e}. \tag{5.8}$$

We approximate τ_e using the manufacturer's quoted servo rate limit of 300°/s. Note from (5.8) that if $\Delta u_\mathrm{e}(t)$ is the unit step function, then $\Delta\delta_\mathrm{e}(t) = 1 - e^{-t/\tau_\mathrm{e}}$. Thus, the rate of $\Delta\delta_\mathrm{e}(t)$ at $t = 0^+$ is $1/\tau_\mathrm{e}$. To approximate τ_e, we set this initial rate $1/\tau_\mathrm{e} = 300^\circ$/s, which gives $\tau_\mathrm{e} \approx 0.2$ s.

Next, we compute the linearized longitudinal dynamics about the trim condition given in Appendix B. Based on simulations, we choose $K_\mathrm{T,P} = 2$ and $K_\mathrm{T,I} = 0.5$. It follows from (5.3)–(5.6) and (5.8) that the transfer function from the elevator input perturbation Δu_{e} to pitch perturbation $\Delta\theta$ is given by

$$G_{\theta u_{\mathrm{e}}}(s) \overset{\triangle}{=} \frac{\mathcal{L}\{\Delta\theta(t)\}}{\mathcal{L}\{\Delta u_{\mathrm{e}}(t)\}} = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{s^6 + a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}, \tag{5.9}$$

where the coefficients b_3, ..., b_0, a_5, ..., a_0 are given in Table 5.1.

Table 5.1: *Transfer Function Parameters Used in (5.9).*

b_3	-636.6
b_2	-3472
b_1	-1729
b_0	-355.7
a_5	17.29
a_4	188
a_3	690.4
a_2	321.9
a_1	170.5
a_0	-2.48

The roots of the numerator polynomial of $G_{\theta u_{\mathrm{e}}}$ are in the open-left-half complex plane, which implies that $G_{\theta u_{\mathrm{e}}}$ is minimum phase. Thus, (3.A1) is satisfied. The transfer function $G_{\theta u_{\mathrm{e}}}$ is relative degree $d = 3$; knowledge of which satisfies (3.A2). Using (5.3)–(5.6) and (5.8), it follows that the first nonzero Markov parameter is

$$H_d = b_3 = \frac{1}{I_{\mathrm{yy}}\tau_{\mathrm{e}}} \frac{\partial M}{\partial \delta_{\mathrm{e}}} \approx \frac{\rho_{\mathrm{a}} V_{\mathrm{T}}^2 S_{\mathrm{r}} c_{\mathrm{r}} (CM_{\delta_{\mathrm{e}}}|_0)}{2 I_{\mathrm{yy}}\tau_{\mathrm{e}}}; \tag{5.10}$$

knowledge of which satisfies (3.A3).

5.1.3 FDI Pitch Control Inner Loop

We design an FDI controller, which uses pitch error $\tilde{\theta}(t) \overset{\triangle}{=} \theta_{\mathrm{d}}(t) - \theta(t)$ to generate the commanded elevator deflection $u_{\mathrm{e}}(t)$. To design the FDI controller, we must choose a controller order ρ, parameter-dependent polynomial $\eta_k(s)$, and reference model $\beta_{\mathrm{m}}(s)/\alpha_{\mathrm{m}}(s)$. As in Section 3.2, we choose $\rho > d$ and $\alpha_{\mathrm{m}}(s) = \beta_{\mathrm{m}}(s)$. Next, let $\rho = 4$ and $\eta_k(s) = (s+k)^4$. This choice of $\eta_k(s)$ satisfies (3.C1) and (3.C2). We know from experience and experimentation that overdamped reference model dynamics are desirable. Thus, we let $\alpha_{\mathrm{m}}(s) = (s+4)(s+6)(s+8)$.

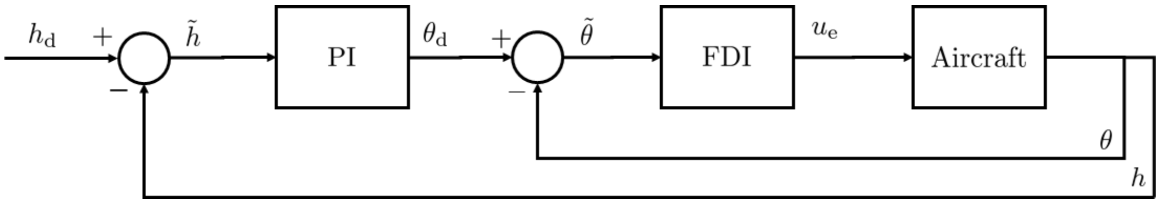


Figure 5.2: *Altitude Autopilot with FDI as Inner Pitch Control Loop.* The FDI controller uses pitch error $\tilde{\theta}$ to generate an elevator deflection command u_{e}.

We compute H_d using (5.10), with measured values for S_{r}, c_{r} and I_{yy}, a standard value for air density, $\rho_{\mathrm{a}} = 1.22$ kg/m^3, an estimate from AVL for $CM_{\delta_{\mathrm{e}}}|_0$, and $\tau_{\mathrm{e}} = 0.2$ s (as computed in Section 5.1.2). However, we note that, in general, V_{T} is a function of time. Corollary 1 in [13] shows that assumption (3.A3) can be replaced with the weaker assumptions that $\mathrm{sgn}(H_d)$ is known and an upper-bound $\bar{H}_d$ on the magnitude of H_d is known. Although the numerical and physical experiments are performed at $V_{\mathrm{T}} = 20$ m/s, we compute H_d from (5.10) under the assumption that $V_{\mathrm{T}} = 25$ m/s. In this case, H_d is $-1,005$. Thus, $\mathrm{sgn}(H_d) = -1$, and we let $\bar{H}_d = 1,005$.

Thus, the FDI controller is

$$G_{\mathrm{FDI}}(s) \triangleq \frac{k^4(s+4)(s+6)(s+8)}{\bar{H}_d(s^4 + 4ks^3 + 6k^2s^2 + 4k^3s)}. \tag{5.11}$$

The frequency response of the FDI pitch controller from pitch error to elevator command is shown in Figure 5.4.

5.1.4 PI Pitch Control Inner Loop

We design a PI controller, which is a standard flight controller that can be used to evaluate the performance of the FDI controller. The PI controller has the form

$$G_{\mathrm{PI}}(s) \triangleq \frac{K(s - z_{\mathrm{c}})}{s}, \tag{5.12}$$

where the gain $K \in \mathbb{R}$ and controller zero $z_{\mathrm{c}} \in \mathbb{R}$ must be chosen.

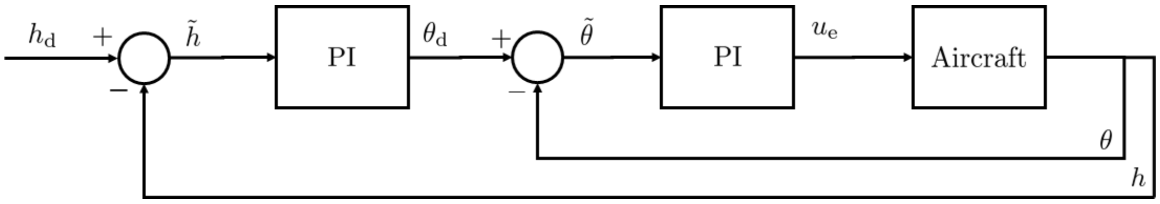


Figure 5.3: *Altitude Autopilot with PI as Inner Pitch Control Loop.* The PI controller uses pitch error $\tilde{\theta}$ to generate an elevator deflection command u_{e}.

Classical root locus design shows that for a relative degree three system, at high gain, two poles will diverge into the open-right-half complex plane. For the transfer function $G_{\theta u_{\mathrm{e}}}$ given by (5.9), choosing z_{c} negative and near the origin places the asymptote center in the open-left-half complex plane. If $z_{\mathrm{c}} = -0.5$, then all closed-loop poles are in the open-left-half complex plane for K on the interval $(-2.20, -0.005)$. Thus, we let $z_{\mathrm{c}} = -0.2$ and $K = -0.5$.

The frequency response of the PI pitch controller from pitch error to elevator command is shown in Figure 5.4.

5.1.5 Altitude Error Outer Loop

To generate the pitch command, we use an altitude error outer loop around the pitch controller. This outer loop uses altitude error to generate the pitch command

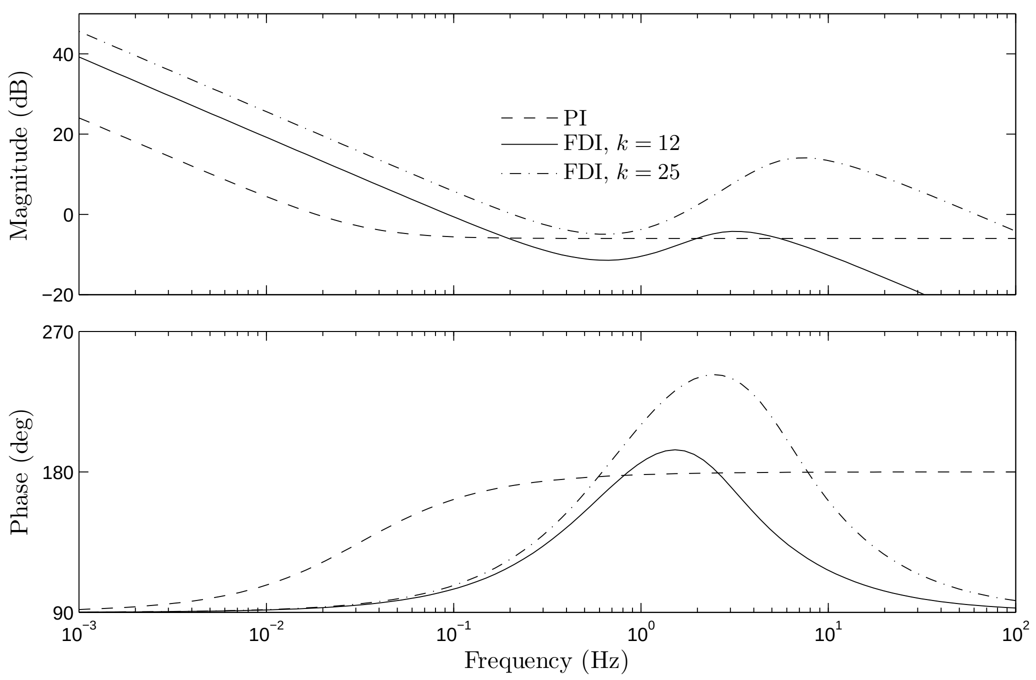


Figure 5.4: *Pitch Controller Frequency Response.* We show the frequency response of each pitch controller from pitch error to elevator command. Both the PI and FDI controllers have infinite gain at low frequency. However, the FDI rolls off at high frequency while the PI has finite gain at high frequency.

$\theta_{\rm d}$. We use a PI controller

$$\theta_{\rm d}(t) = k_{\rm h,P}\tilde{h}(t) + k_{\rm h,I}\int_0^t \tilde{h}(\tau){\rm d}\tau, \tag{5.13}$$

where $k_{\rm h,P}$ is proportional gain, $k_{\rm h,I}$ is integral gain, $\tilde{h}(t) \overset{\triangle}{=} h_{\rm d}(t) - h(t)$ is in meters, and $\theta_{\rm d}$ is in degrees. Based on simulation, we choose the gains $k_{\rm h,P} = 1.6$ and $k_{\rm h,I} = 0.2$.

5.2 Numerical Simulations

We simulate the nonlinear aircraft dynamics (2.3), (2.4), (2.8), (2.9), and (2.17)–(2.22) with the autopilot from Section 5.1. We compare the performance of the FDI pitch controller (5.11) to the PI pitch controller (5.12). For these controllers, we compare the magnitude of the frequency response from wind disturbance to pitch error $\tilde{\theta}$ and altitude error $\tilde{h}$, the finite-time average power of the pitch error $\mathcal{P}_\theta(t_1, t_0)$, and the finite-time average power of the altitude error $\mathcal{P}_h(t_1, t_0)$.

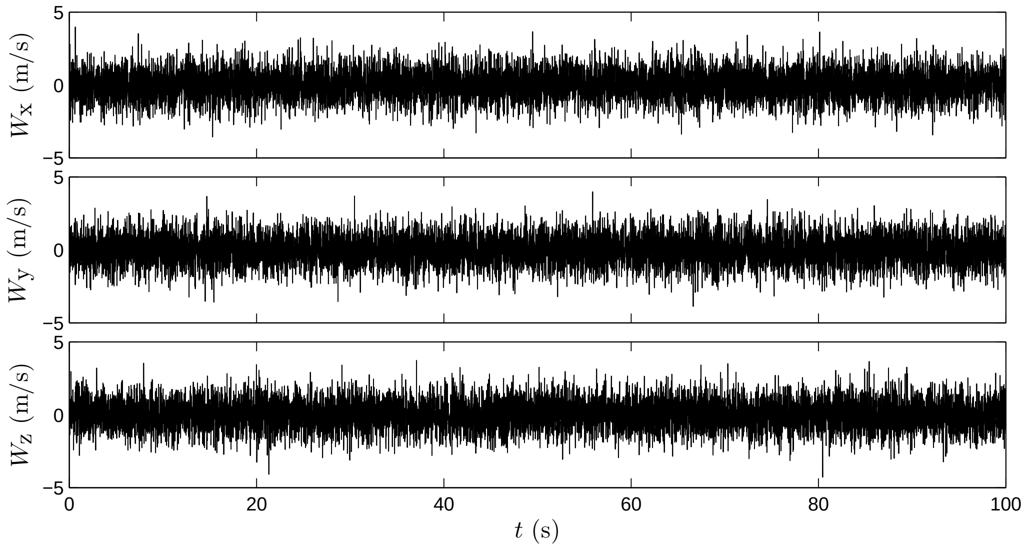


Figure 5.5: *White-Noise Wind.* For the purpose of simulation, we generate a time series realization of band-limited, zero-mean, unit-variance Gaussian white noise.

For all of the following simulations, we use the model parameters of the EDGE test platform. Thus, we let the mass $m = 4.48$ kg; moments of inertia $I_\mathrm{xx} = 0.1778$ kg-m^2, $I_\mathrm{yy} = 0.3287$ kg-m^2, and $I_\mathrm{zz} = 0.4231$ kg-m^2; reference span $b_\mathrm{r} = 1.52$ m, chord length $c_\mathrm{r} = 0.2975$ m, and planform area $S_\mathrm{r} = 0.4534$ m^2. The AVL estimates of the 1st-order Taylor series approximations of the aerodynamic forces and moments are shown graphically in Appendix A.

To maintain wings-level flight in the presence of a disturbance, we implement a roll-to-aileron PI controller, whose transfer function is

$$\frac{k_{\phi,\mathrm{P}} s + k_{\phi,\mathrm{I}}}{s}, \tag{5.14}$$

where $k_{\phi,\mathrm{P}} = 0.5$ and $k_{\phi,\mathrm{I}} = 1$. Note that this roll control allows us to focus on the longitudinal dynamics.

In each simulation, the initial heading is $\psi(0) = 0$, which implies that the plane is initially traveling in the $\hat{\imath}_\mathrm{I}$ direction. Although we do not implement a heading control, in all simulations, the plane's heading remains nearly constant.

To model the wind disturbance on the aircraft, we generate two stochastic, three-dimensional realizations of the wind in the inertial frame F_I. The first is a realization of band-limited, zero-mean, unit-variance Gaussian white noise, shown in Figure 5.5. The second is a zero-mean, unit-variance random sequence whose power spectra follow the Kolmogorov $\kappa^{-5/3}$ law [27] from 0.005 s^{-1} to 10 s^{-1}. The realization in the time domain is computed using an inverse fast Fourier transform and is shown in Figure 5.6. We remark that this is only a model for turbulence and is used as a disturbance with frequency content more representative of the wind than Gaussian white noise.

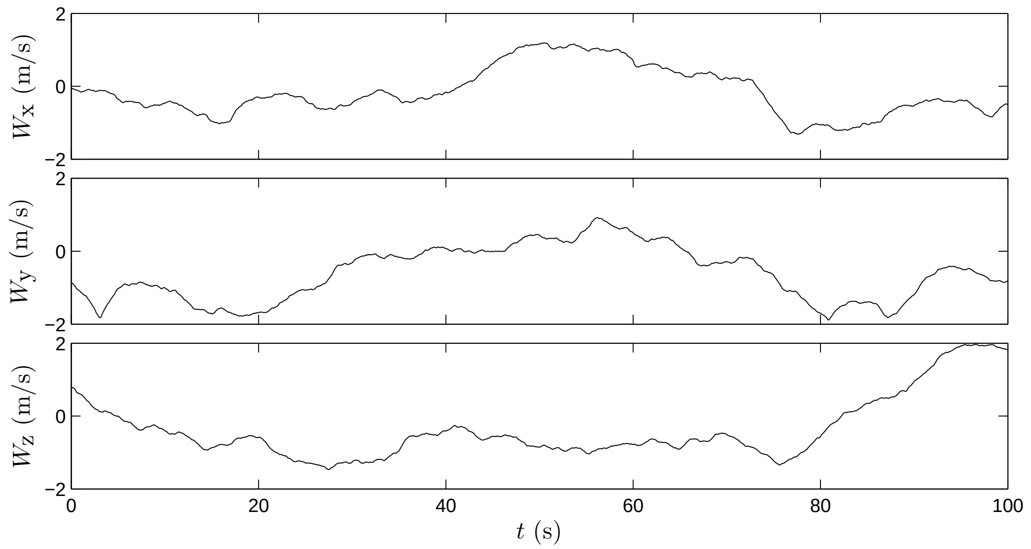


Figure 5.6: *Turbulent Wind.* For the purpose of simulating turbulent wind, we generate a zero-mean, unit-variance random sequence whose power spectra follow the Kolmogorov $\kappa^{-5/3}$ law.

Example 5.1. *Open-loop.* In this example, the roll-to-aileron controller (5.14) is the only control implemented. The airspeed-to-throttle static gain is $k_{\mathrm{T},0} = 7.658$ N, the elevator deflection is $\delta_{\mathrm{e},0} = -3.937°$, and the rudder and aileron deflections are $\delta_{\mathrm{r},0} = \delta_{\mathrm{a},0} = 0$, which are the steady-state values at the forced equilibrium where $V_{\mathrm{T}} = 20$ m/s. The initial conditions are the forced equilibrium values, that is $\theta(0) = \theta_0 = 5.532°$, $U(0) = U_0 = 20$, $W(0) = W_0 = U_0 \tan(\theta_0)$, $V(0) = P(0) = Q(0) = R(0) = \phi(0) = \psi(0) = 0$. We use the white-noise wind shown in Figure 5.5.

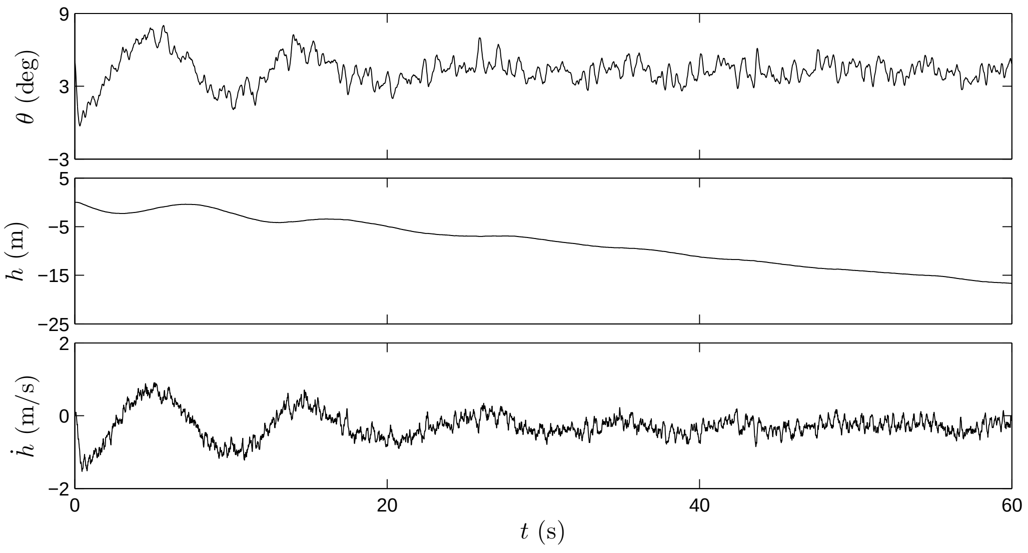


Figure 5.7: *Open-Loop Response.* Although the simulation is begun at the forced equilibrium with $V_{\mathrm{T}} = 20$ m/s, the plane diverges from the equilibrium and begins falling.

Time histories of the pitch θ, altitude h, and altitude derivative $\dot{h}$ are shown in Figure 5.7. We note that the transfer function $G_{\theta u_{\mathrm{e}}}$, given by (5.9), has a single real pole in the open-right-half complex plane. Although we begin the simulation at a forced equilibrium, the white-noise wind disturbance causes the aircraft to move away from the forced equilibrium. Moreover, since this forced equilibrium is unstable but the unstable pole of the linearization is near the origin, the response in Figure 5.7 diverges slowly. △

For the following examples, we implement the roll-to-aileron controller (5.14), the airspeed controller (5.1), the altitude error outer loop controller (5.13), and compare the performance of the FDI controller (5.11) to the performance of the PI controller (5.12). The controller parameters for (5.14) and (5.11)–(5.13) are the same as those given in Section 5.1.

Example 5.2. *Average power of pitch error and average power of altitude error as functions of k.* We simulate level flight by letting the altitude command $h_{\rm d}(t) \equiv 0$. We use the white-noise wind shown in Figure 5.5 as the disturbance. We simulate the FDI controller at different values of k and compute $\mathcal{P}_\theta(t_1, t_0)$ and $\mathcal{P}_h(t_1, t_0)$, where $t_0 = 20$ s and $t_1 = 100$ s. The results are shown in Figure 5.8.

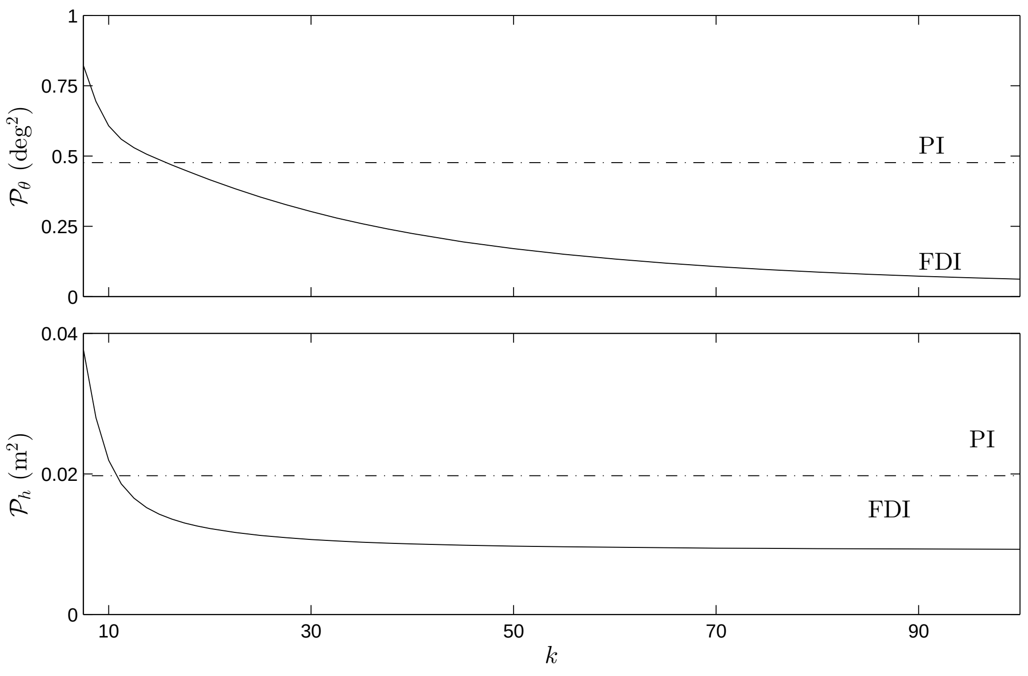


Figure 5.8: *Average Powers of the Performances as Functions of k.* As the FDI parameter k increases, $\mathcal{P}_\theta(t_1, t_0)$ decreases, becoming smaller than the average power of pitch error for the PI controller at approximately $k = 15$. The average power of altitude error becomes smaller than that of the PI at approximately $k = 12$ but appears to approach a nonzero value as k increases.

This example illustrates part (ii) of Theorem 3.1, which states that $\lim_{t_1 \to \infty} \mathcal{P}_\theta(t_1, t_0)$

is arbitrarily small for sufficiently large k. Similarly, the average power of altitude error reduces as k increases, but it appears to approach a nonzero value. $\triangle$

Example 5.3. *Frequency response with white-noise disturbance.* In this example, we simulate level flight by letting the altitude command $h_{\rm d}(t) \equiv 0$. We use the white-noise wind shown in Figure 5.5 as the disturbance. We compute the frequency response of the aircraft to the disturbance. We use the discrete fast Fourier transform on the simulation time series and take the quotient of the output and disturbance to approximate the transfer function from disturbance to output.

We compute the frequency response from each of the three wind components to pitch error $\tilde{\theta}$, altitude error $\tilde{h}$, and altitude derivative $\dot{h}$, as shown in Figures 5.9, 5.10, and 5.11.

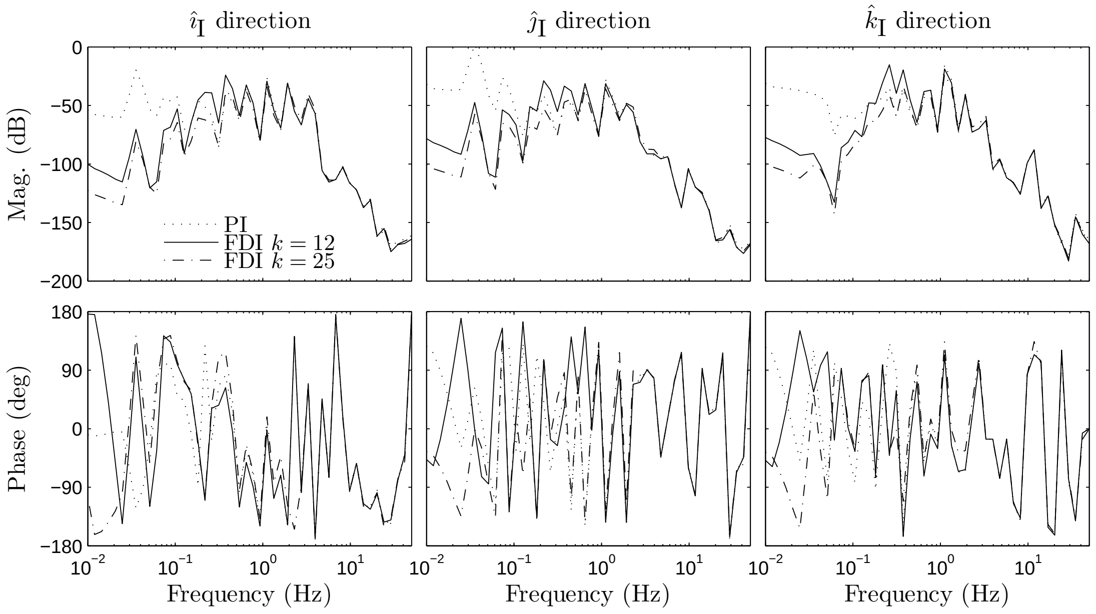


Figure 5.9: *Pitch Error Frequency Response.* The FDI controller makes the magnitude of the frequency response from disturbance to pitch error smaller than that of the PI controller, particularly at low frequency.

The advantage of the FDI controller is most pronounced in the frequency response from disturbance to pitch error, where the disturbance is more attenuated at low frequency by the FDI controller than by the PI controller. In the frequency responses

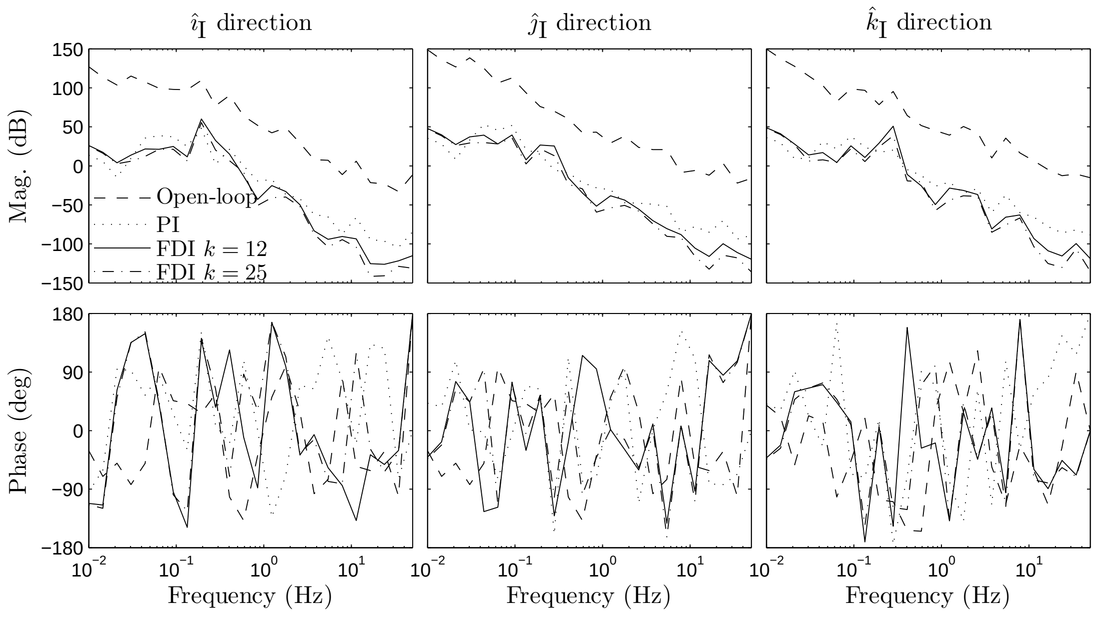


Figure 5.10: *Altitude Frequency Response.* The FDI controller makes the magnitude of the frequency response from disturbance to altitude error smaller than that of the PI controller at many frequencies.

to altitude error and altitude derivative, the magnitude of the FDI response is lower than that of the PI response at many frequencies but not all. △

Example 5.4. *Closed-loop response with turbulent wind.* We simulate level flight by letting the altitude command $h_{\rm d}(t) \equiv 0$. We use the turbulent wind shown in Figure 5.6. The FDI parameter is $k = 25$. The time histories of pitch, pitch command, pitch error, altitude, altitude derivative, and elevator deflection are shown in Figure 5.12.

For the FDI controller, $\mathcal{P}_\theta(t_1, t_0) = 0.00983$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 0.907$ m^2, where $t_0 = 20$ s and $t_1 = 100$ s. For the PI controller, $\mathcal{P}_\theta(t_1, t_0) = 0.104$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 0.960$ m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 10.6. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 1.058. This result is similar to the result of Example 5.2, that is, the FDI shows a large improvement in average power of pitch error and a smaller improvement in average power of altitude error. △

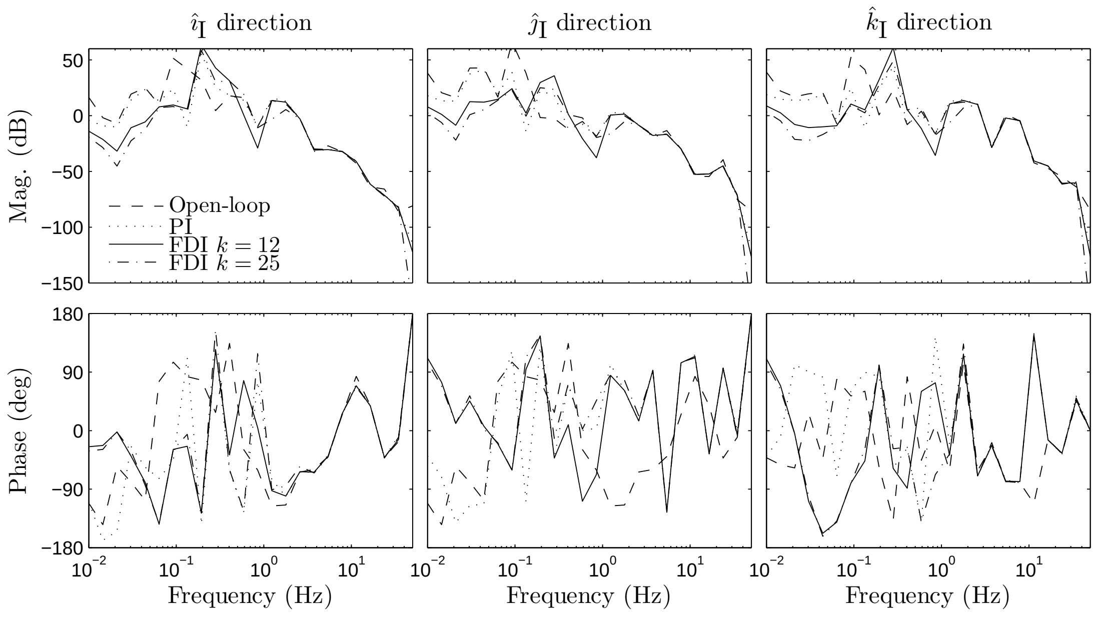


Figure 5.11: *Altitude Derivative Frequency Response.* For the controller parameters we choose, the FDI controller makes the magnitude of the frequency response from disturbance to altitude error derivative smaller than that of the PI controller at many frequencies.

Example 5.5. *Filtered-step altitude command with turbulent wind.* We let the altitude command $h_{\mathrm{d}}(t)$ be a series of steps with amplitude five, filtered through the first-order, unity-DC-gain, low-pass filter $2/(s+2)$. We use the turbulent wind shown in Figure 5.6. The FDI parameter is $k = 25$. The time histories of pitch, pitch command, pitch error, altitude, altitude derivative, and elevator deflection are shown in Figure 5.13.

For the FDI controller, $\mathcal{P}_\theta(t_1, t_0) = 0.427$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 5.69$ m^2, where $t_0 = 20$ s and $t_1 = 100$ s. For the PI controller, $\mathcal{P}_\theta(t_1, t_0) = 1.642$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 7.56$ m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 3.84. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 1.33. In this example, the FDI shows an improvement in both average power of pitch error and average power of altitude error. $\triangle$

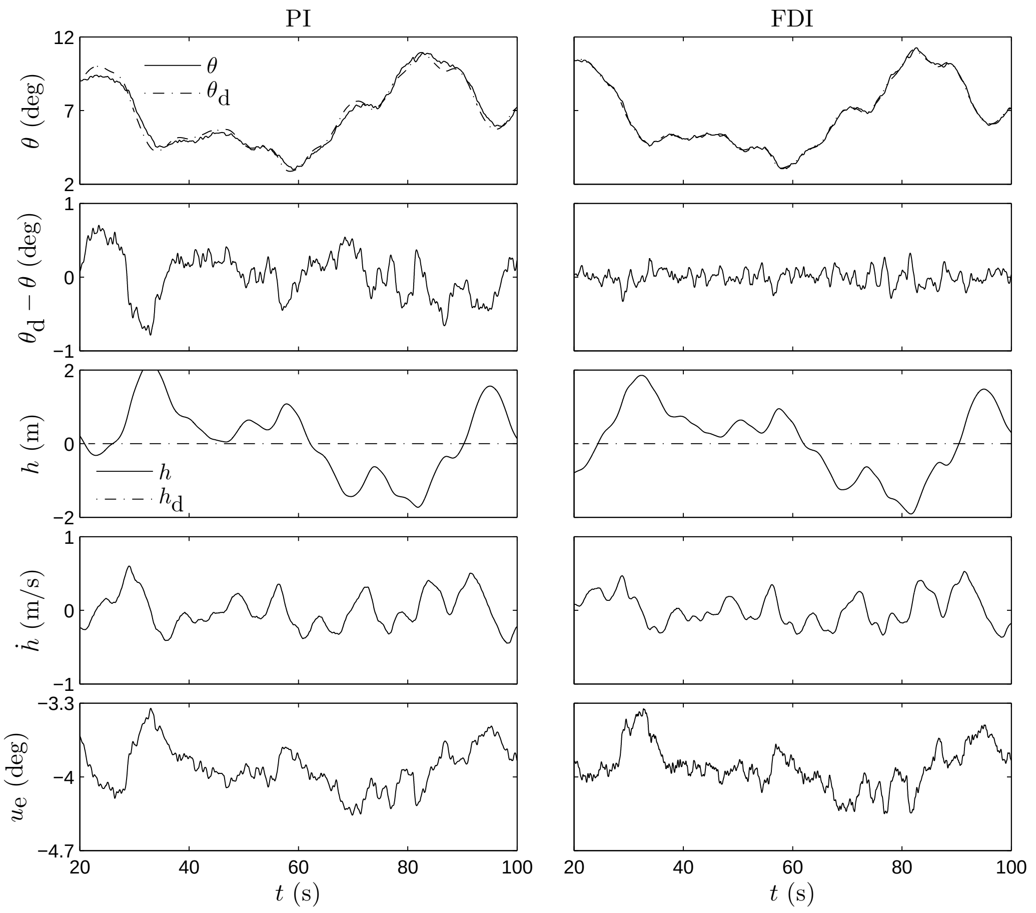


Figure 5.12: *Closed-Loop Responses with Zero Altitude Command.* The FDI controller exhibits better pitch command following than the PI controller. This has the effect of reducing altitude errors.

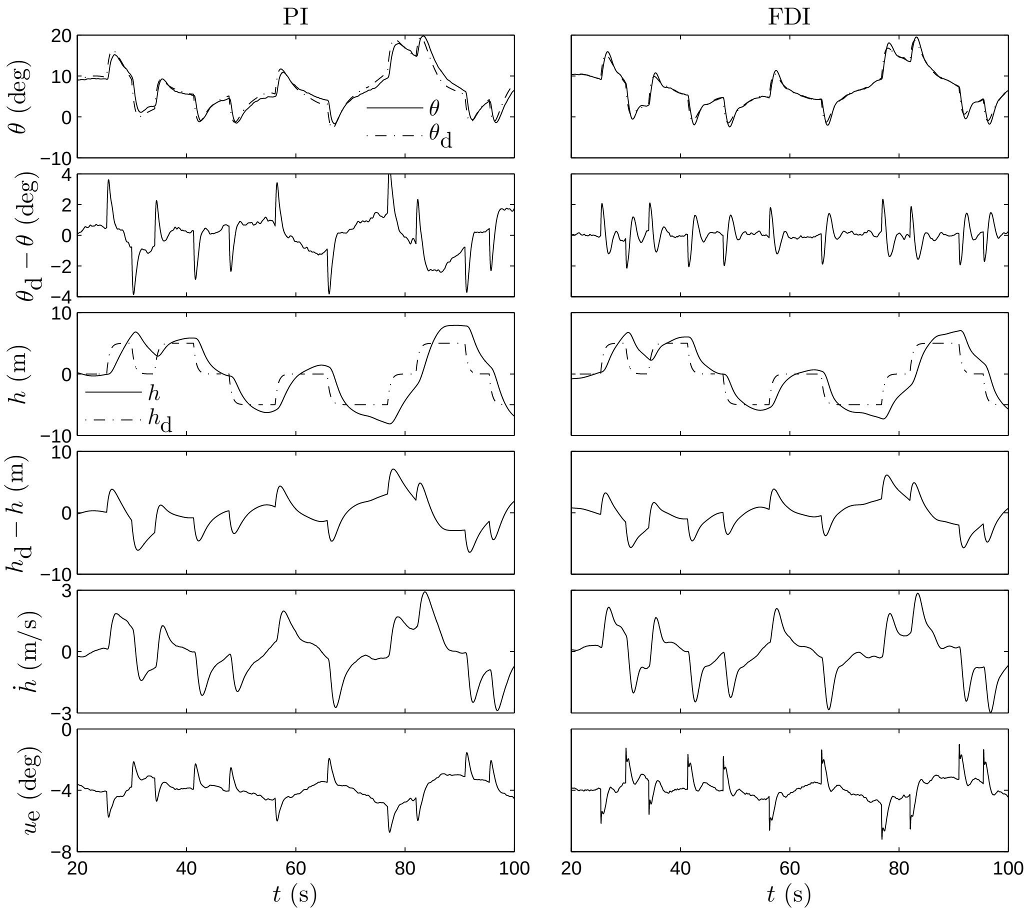


Figure 5.13: *Closed-Loop Responses with Step Altitude Command.* The FDI controller follows the pitch command better than the PI does, which has the effect of decreasing the average power of the altitude error.

Chapter 6 Discrete-Time Filtered Dynamic Inversion with Nonlinear Aircraft Model

In this chapter, we discretize the FDI and PI pitch controllers from Chapter 5. We augment the discrete-time FDI and PI controllers with an anti-windup strategy. Finally, we present simulation results of the continuous-time nonlinear aircraft dynamics with the discrete-time longitudinal altitude autopilot in feedback.

6.1 Discrete-Time Controllers

To implement the FDI and PI pitch controllers on digital hardware, we discretize the control laws (5.11)–(5.13). For each controller, we assume a zero-order hold on the input and a uniform sampling time $T_{\rm s}$.

6.1.1 Discrete-Time FDI Pitch Control Inner Loop

The continuous-time FDI pitch controller (5.11) has the state-space realization

$$\begin{aligned}\dot{x}_{\rm c}(t) &= A_{\rm c}x_{\rm c}(t) + B_{\rm c}\tilde{\theta}(t),\\ \dot{x}_{\rm I}(t) &= k_{\rm I}\tilde{\theta}(t),\\ u_{\rm e}(t) &= C_{\rm c}x_{\rm c}(t) + k^4\bar{H}_d^{-1}x_{\rm I}(t),\end{aligned}$$

where $x_\mathrm{c}(t) \in \mathbb{R}^3$, $x_\mathrm{I}(t) \in \mathbb{R}$, and

$$A_\mathrm{c} = \begin{bmatrix} -4k & -6k^2 & -4k^3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_\mathrm{c} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$C_\mathrm{c} = k^4 \bar{H}_d^{-1} \begin{bmatrix} 1 & 18 & 104 \end{bmatrix}, \quad k_\mathrm{I} = 48k^{-3}.$$

Next, note from the FDI formulation that $k > 0$. Discretizing the controller with a zero-order hold on the input yields

$$x_\mathrm{d}[i+1] = A_\mathrm{d} x_\mathrm{d}[i] + B_\mathrm{d} \tilde{\theta}[i],$$

$$u_\mathrm{e}[i] = C_\mathrm{d} x_\mathrm{d}[i],$$

where $x_\mathrm{d}[i] \in \mathbb{R}^4$, $i = 0, 1, 2, ...$, and

$$A_\mathrm{d} = \left[\begin{array}{ccc:c} & & & 0 \\ & \exp A_\mathrm{c} T_\mathrm{s} & & 0 \\ & & & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right], \quad B_\mathrm{d} = \begin{bmatrix} A_\mathrm{c}^{-1}(\exp A_\mathrm{c} T_\mathrm{s} - I_3) B_\mathrm{c} \\ k_\mathrm{I} T_\mathrm{s} \end{bmatrix}, \tag{6.1}$$

$$C_\mathrm{d} = \begin{bmatrix} C_\mathrm{c} & k^4 \bar{H}_d^{-1} \end{bmatrix}. \tag{6.2}$$

Next, we augment the FDI controller with an anti-windup scheme suited for digital hardware. We choose to use the integrator clamping method [28,29]. In the integrator clamping strategy, integration is conditional on the resulting controller output lying in the saturation bounds. Thus, the discrete-time FDI controller with anti-windup is given by

$$x'_\mathrm{d}[i+1] = A_\mathrm{d} x'_\mathrm{d}[i] + B'_\mathrm{d} \tilde{\theta}[i], \tag{6.3}$$

$$u_{\mathrm{e}}[i] = C_{\mathrm{d}} x'_{\mathrm{d}}[i], \tag{6.4}$$

where A_{d} and C_{d} are given by (6.1) and (6.2), and

$$B'_{\mathrm{d}} = \begin{cases} \left[A_{\mathrm{c}}^{-1}(\exp A_{\mathrm{c}} T_{\mathrm{s}} - I_3) B_{\mathrm{c}} \quad k_{\mathrm{I}} T_{\mathrm{s}}\right]^{\mathrm{T}}, & u_{\min} < C_{\mathrm{d}}(A_{\mathrm{d}} x'_{\mathrm{d}}[i] + B_{\mathrm{d}} \tilde{\theta}[i]) < u_{\max}, \\ \left[A_{\mathrm{c}}^{-1}(\exp A_{\mathrm{c}} T_{\mathrm{s}} - I_3) B_{\mathrm{c}} \quad 0\right]^{\mathrm{T}}, & \text{otherwise.} \end{cases} \tag{6.5}$$

6.1.2 Discrete-Time PI Pitch Control Inner Loop

The continuous-time PI pitch controller (5.12) has the state-space realization

$$\begin{aligned} \dot{x}_{\mathrm{c}}(t) &= -K z_{\mathrm{c}} \tilde{\theta}(t), \\ u_{\mathrm{e}}(t) &= K \tilde{\theta}(t) + x_{\mathrm{c}}(t). \end{aligned}$$

Discretizing the controller with a zero-order hold on the input yields

$$\begin{aligned} x_{\mathrm{d}}[i+1] &= x_{\mathrm{d}}[i] - K z_{\mathrm{c}} T_{\mathrm{s}} \tilde{\theta}[i], \\ u_{\mathrm{e}}[i] &= x_{\mathrm{d}}[i] + K \tilde{\theta}[i]. \end{aligned}$$

The PI controller is augmented with the integrator clamping anti-windup strategy, which yields

$$x'_{\mathrm{d}}[i+1] = \begin{cases} x'_{\mathrm{d}}[i] - K z_{\mathrm{c}} T_{\mathrm{s}} \tilde{\theta}[i], & u_{\min} < x'_{\mathrm{d}}[i] - K z_{\mathrm{c}} T_{\mathrm{s}} \tilde{\theta}[i] + K \tilde{\theta}[i] < u_{\max}, \\ x'_{\mathrm{d}}[i], & \text{otherwise,} \end{cases} \tag{6.6}$$

$$u_{\mathrm{e}}[i] = x'_{\mathrm{d}}[i] + K \tilde{\theta}[i]. \tag{6.7}$$

6.1.3 Discrete-Time Altitude Error Outer Loop

The continuous-time altitude error outer loop PI controller (5.13) has the state-space realization

$$\begin{aligned}\dot{x}_{\mathrm{c}}(t) &= k_{\mathrm{h,I}}\tilde{h}(t),\\ \theta_{\mathrm{d}}(t) &= k_{\mathrm{h,P}}\tilde{h}(t) + x_{\mathrm{c}}(t).\end{aligned}$$

Discretizing the controller with a zero-order hold on the input yields

$$\begin{aligned}x_{\mathrm{d}}[i+1] &= x_{\mathrm{d}}[i] + k_{\mathrm{h,I}}T_{\mathrm{s}}\tilde{h}[i],\\ \theta_{\mathrm{d}}[i] &= x_{\mathrm{d}}[i] + k_{\mathrm{h,P}}\tilde{h}[i].\end{aligned}$$

The altitude outer loop PI controller is augmented with the integrator clamping anti-windup strategy, which yields

$$x'_{\mathrm{d}}[i+1] = \begin{cases} x'_{\mathrm{d}}[i] + k_{\mathrm{h,I}}T_{\mathrm{s}}\tilde{\theta}[i], & \theta_{\mathrm{d,min}} < x'_{\mathrm{d}}[i] + k_{\mathrm{h,I}}T_{\mathrm{s}}\tilde{h}[i] + k_{\mathrm{h,P}}\tilde{h}[i] < \theta_{\mathrm{d,max}},\\ x'_{\mathrm{d}}[i], & \text{otherwise,}\end{cases} \tag{6.8}$$

$$u_{\mathrm{e}}[i] = x'_{\mathrm{d}}[i] + k_{\mathrm{h,P}}\tilde{h}[i]. \tag{6.9}$$

6.2 Numerical Simulations

In this section, we repeat Examples 5.2, 5.4, and 5.5 with the discrete-time FDI controller (6.1)–(6.5) and the discrete-time PI controller (6.6) and (6.7). We use the continuous-time nonlinear aircraft dynamics (2.3), (2.4), (2.8), (2.9), and (2.17)–(2.22), and the same model parameters as in Section 5.2. The purpose of these simulations is to ensure the performance of the discrete-time controllers is comparable to that of the continuous-time controllers. Thus, for the following examples, we implement the roll-to-aileron controller (5.14) and airspeed controller (5.1) in continuous time, and compare the performance of the discrete-time FDI controller (6.1)–(6.5) and the discrete-time PI controller (6.6) and (6.7). All controller parameters are the same as those used in Section 5.2. The sampling time is $T_\mathrm{s} = 0.02$ s, which is the sampling time of the controller hardware used on the test platform.

Example 6.1. *Average power of pitch error and average power of altitude error as functions of k.* We repeat the simulations of Example 5.2 with the discretized FDI controller and discretized PI controller. The results are shown in Figure 6.1.

As in Example 5.2, over a range of k, the average power of pitch error and the average power of altitude error decrease for increasing k, going below those of the PI controller. However, the average power of pitch error does not decrease as rapidly as in the continuous-time case, although the average power of altitude error appears to be very similar to that in the continuous-time case. Additionally, once k is increased above 40, the closed loop becomes unstable. $\triangle$

Example 6.2. *Closed-loop response with turbulent wind.* We repeat Example 5.4 with the discretized FDI and PI controllers. We simulate level flight by letting the altitude command $h_\mathrm{d}[i] \equiv 0$. We use the turbulent wind shown in Figure 5.6. We choose the FDI parameter $k = 25$. The time histories of pitch, pitch command, pitch error, altitude, altitude derivative, and elevator deflection are shown in Figure 6.2.

For the FDI controller, $\mathcal{P}_\theta(t_1, t_0) = 0.0105$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 0.904$ m^2, where $t_0 = 20$ s and $t_1 = 100$ s. For the PI controller, $\mathcal{P}_\theta(t_1, t_0) = 0.0781$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 0.889$ m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 7.41. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 0.984. This simulation result is surprising in that the average power of pitch error is improved but the average power of altitude error is made worse. This result shows that the relative improve-

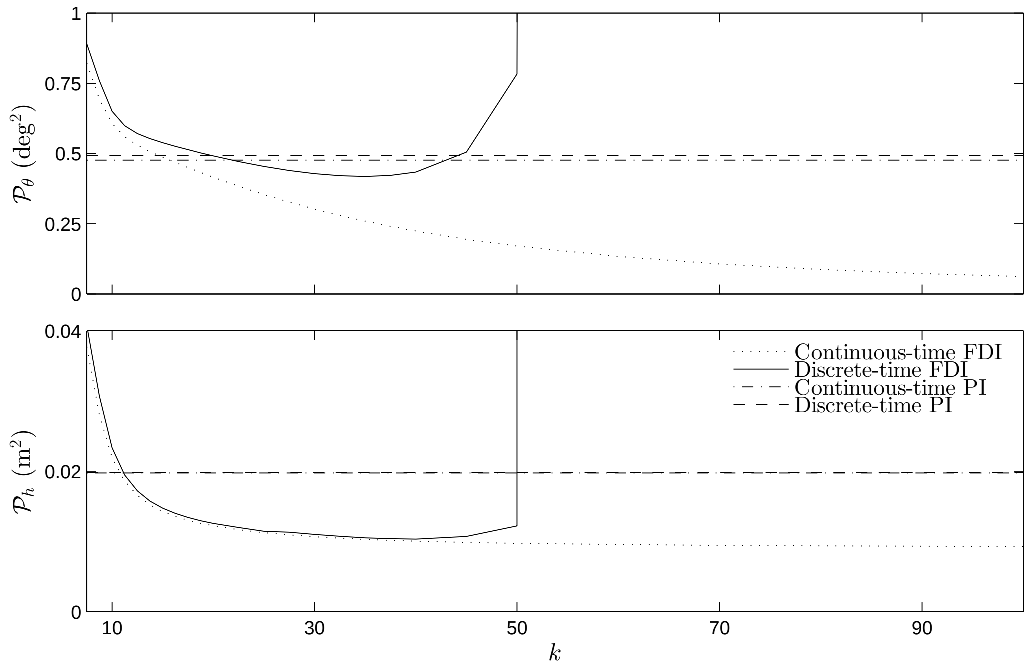


Figure 6.1: *Average Powerof the Performances as Functions of* k*.* As in Example 5.2, there is a range of k where the average power of pitch error and the average power of altitude error are better than those of the discrete-time PI control simulation. However, unlike the continuous-time case, the FDI-controlled closed loop becomes unstable as k increases past 40.

ments are sensitive to the frequency content of the disturbance, as this example uses the model turbulent wind and Example 6.1 uses the white-noise wind. △

Example 6.3. *Filtered-step altitude command with turbulent wind.* We repeat Example 5.5 with the discretized FDI and PI controllers. We let the altitude command $h_{\rm d}[i]$ be a series of steps with amplitude five, filtered through the first-order, unity-DC-gain, low-pass filter with transfer function $2/(s+2)$. We use the turbulent wind shown in Figure 5.6. We choose the FDI parameter $k = 25$. The time histories of pitch, pitch command, pitch error, altitude, altitude derivative, and elevator deflection are shown in Figure 6.3.

For the FDI controller, $\mathcal{P}_\theta(t_1, t_0) = 0.421$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 5.91$ m^2, where $t_0 =$

20 s and $t_1 = 100$ s. For the PI controller, $\mathcal{P}_\theta(t_1, t_0) = 1.31$ deg^2 and $\mathcal{P}_h(t_1, t_0) = 8.15$ m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 3.11. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 1.38. This example is similar to the continuous-time case. The large, fast changes in altitude command appear to favor the FDI controller. $\triangle$

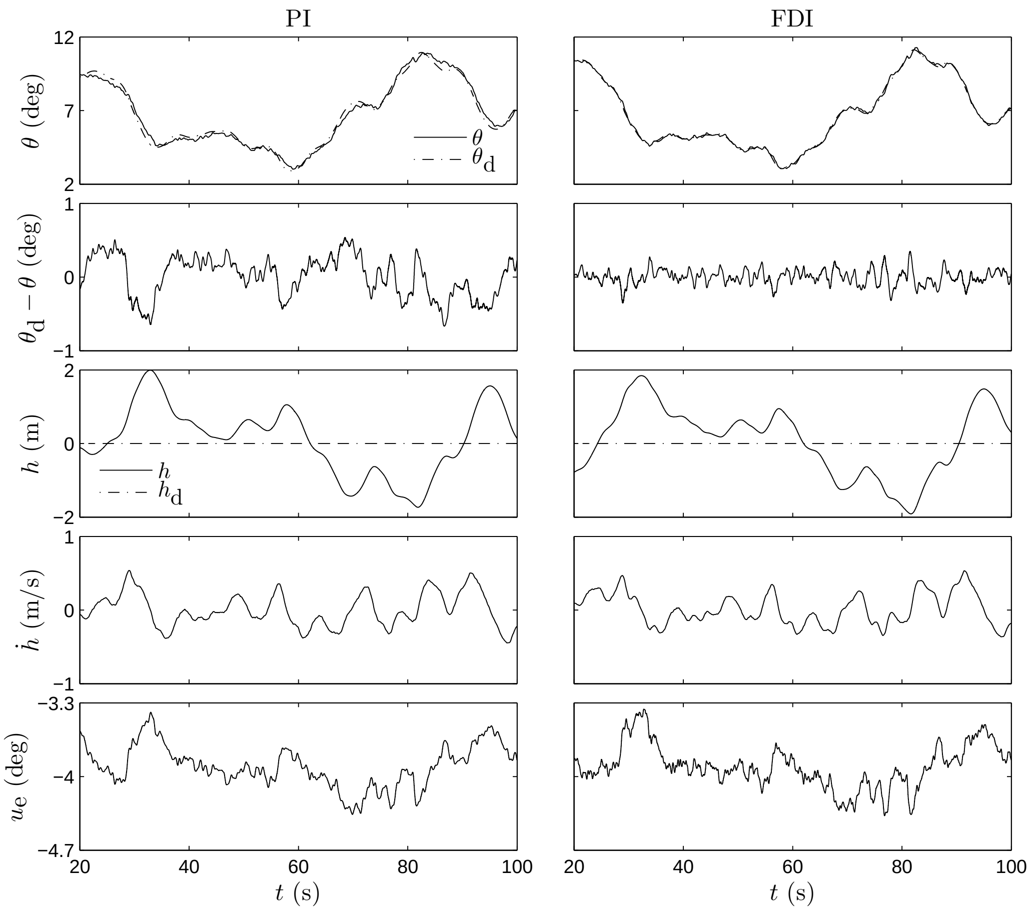


Figure 6.2: *Closed-Loop Responses with Zero Altitude Command.* The FDI controller exhibits better pitch command-following ability than does the PI. In this case, however, this does not have the effect of decreasing the average power of altitude error. This runs contrary to both the discrete-time white-noise disturbance case and the continuous-time model turbulent wind disturbance case.

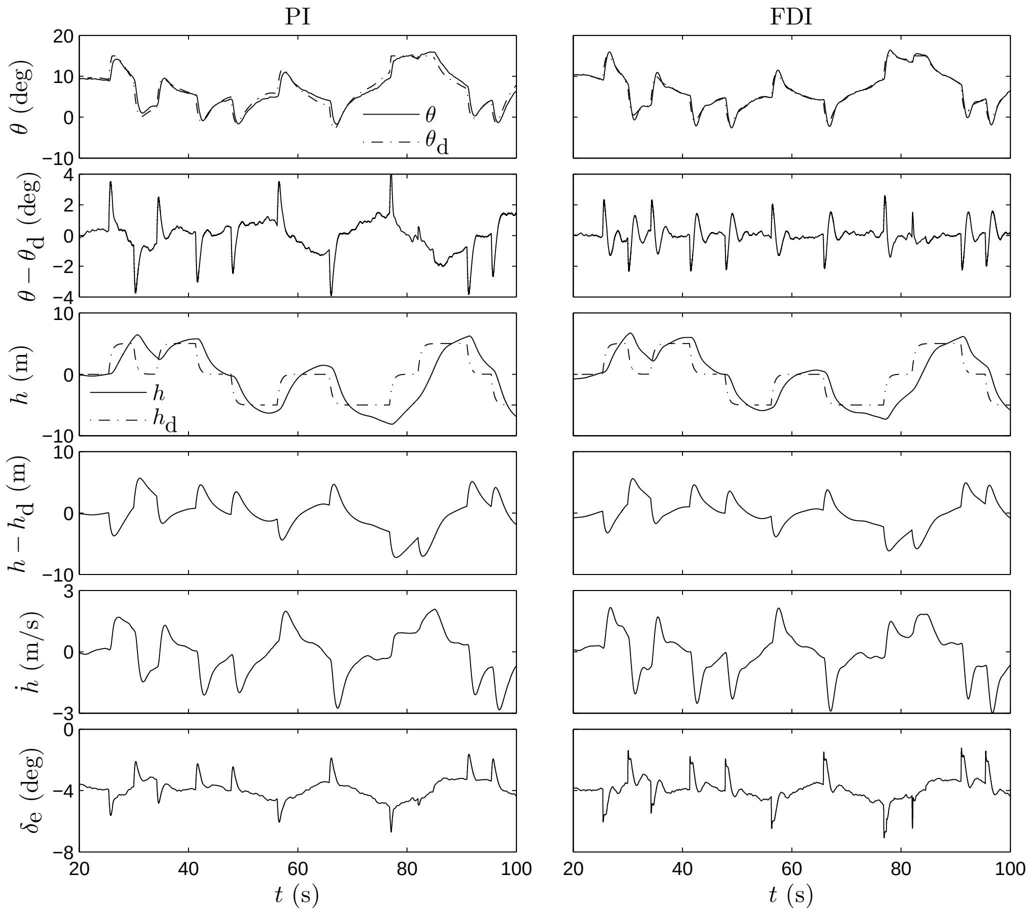


Figure 6.3: *Closed-Loop Responses with Step Altitude Command.* The FDI controller follows the pitch command better than the PI does. Unlike the previous example, the average power of altitude error is smaller for the FDI case than the PI case. The result is nearly the same as in the continuous-time case.

Chapter 7 Experimental Description

7.1 Airframe

We used the EDGE 540T airplane described in Chapter 4 to implement the FDI controller and evaluate its performance relative to the PI controller. To gather data, we made use of the Ardupilot's onboard datalogging capability. The Arduplane firmware can be altered to log an arbitrary signal at rates less than or equal to 50 Hz. Thus, we altered the Arduplane firmware to log altitude h, altitude error $h_{\mathrm{d}} - h$, commanded elevator deflection u_{e}, pitch command θ_{d}, and pitch θ at 50 Hz. In addition, we logged the plane's distance from its next waypoint.

The plane was equipped with an XBee transceiver. A PC laptop functioned as a groundstation using Mission Planner v1.3.1 software and was also equipped with an XBee transceiver. Thus, the ground crew was able to monitor position, attitude, and airspeed information while the plane was in the air. Additionally, the ground crew were able to change predefined tunable parameters while the plane was in the air.

7.2 Autopilot and Implementation

To implement the pitch controllers described in Chapters 5 and 6, we altered the pitch controller module of the Arduplane v2.74b source code, which is distributed on Google Code [30]. The control laws of the pitch controller are contained in the library files `AP_PitchController.h` and `AP_PitchController.cpp`.

The pitch controller defines several user-tunable parameters that can be changed from the ground while the plane is in the air. These are the proportional gain K;

the integral gain $-Kz_\mathrm{c}$; the estimated upper-bound on the first non-zero Markov parameter used in the FDI controller $\bar{H}_d$; a boolean d_c to select whether the PI or FDI pitch control is active; and a scaling term $\delta_{\mathrm{e,max}}$. The proportional gain K and integral gain $-Kz_\mathrm{c}$ are used in the PI control strategy. Both K and $\bar{H}_d$ are assumed to be negative. The controller selector d_c is 0 for the PI controller and 1 for the FDI controller. Finally, the scaling term $\delta_{\mathrm{e,max}}$ is the maximum deflection for the elevator. The scaling term is necessary because the Ardupilot hardware sends pulse widths as outputs to the servos. The pulse width values are calibrated before flight, and, when using angles in the code, the Arduplane software assumes that the maximum and minimum pulse widths correspond to a $\pm 45^\circ$ deflection at the control surface. To correct this assumption and therefore obtain accurate measurements of the control surface deflections, we multiply the input to the controllers by $45^\circ/\delta_{\mathrm{e,max}}$. Otherwise, gains, the leading Markov parameter $\bar{H}_d$, and elevator deflections would not be equivalent between the simulations and the experiment.

Finally, for convenience, and to ensure accuracy in transferring the control strategy from MATLAB to C, we created a MATLAB script that computes the control law (6.1)–(6.5), and writes the members of the A_d, B_d, and C_d matrices to a header file, `FDI_params.h`. This script is contained in Appendix C.

The function `get_servo_out` is responsible for inputting a pitch error and outputting a servo setting. It is called at every time step (i.e. at a rate of 50 Hz), after the altitude control outer loop has chosen a pitch command. Based on the value of d_c, it computes the PI or FDI strategy.

The function `reset_I` is run whenever there is a change in flight mode. For the PI, it resets the integrator state to zero, and, for the FDI, it resets the state vector to $0_{4\times 1}$.

The coded pitch controller is given in Appendices D–F, where Appendix F is the output of the script given in Appendix C for $k = 25$.

7.3 Flight Location and Flight Path

Through the University of Kentucky's partnership with the Lexington Model Airplane Club (LMAC), we had access to a property that hobbyists use to fly remote-controlled airplanes. The field features a paved runway that is approximately 200 m. The runway is oriented WSW and ENE, to match the predominant wind direction in the area during the warmer months. The field is shown in Figure 7.1.

The goal of the experimental flights was to evaluate the altitude and pitch command following ability of the FDI controller compared to a PI controller. Thus, we designed a flight path that we repeated several times. We also attempted to minimize the amount of time the plane is in the air in order to leave a margin on battery life.

The clockwise flight path is shown in Figure 7.2, where the waypoints A–H are all 100 m above a constant reference ground level. We define a lap as starting and ending at point A. The flight plan is as follows:

- Takeoff: The plane takes off going WSW under manual control.
- Experiment beginning: Under manual control, the plane gains altitude and turns clockwise. The pilot lines the plane up approximately with the front straightaway at approximately 100 m altitude. The plane is switched into automatic control before reaching waypoint A.
- First lap: We exclude the first lap for the purpose of measurement.
- Second and third laps: It is assumed that after the first lap under automatic control, any initial condition response has subsided. Thus, we use the second and third laps for measurement, beginning and ending at waypoint A.
- Fourth lap: On the back straightaway of the fourth lap, we change d_c from the ground to select the other pitch controller. We do not use the fourth lap for measurement.

Table 7.1: *Summary of Experimental Flights.*

Flight	Takeoff time	Wind (mph)[1]	Temp. (F)[1]	Barometric Pressure (in)[1]	Controller Order	k
1	12:14 PM	8.1 WSW	68.0	30.05	FDI then PI	25
2	12:42 PM	8.1 WSW	68.0	30.05	PI then FDI	30
3	2:01 PM	13.8 W	66.0	30.02	FDI then PI	30
4	3:12 PM	8.1 WSW	66.9	29.99	PI then FDI	12

- Fifth and sixth laps: We use the fifth and sixth laps for measurement, beginning and ending at waypoint A.
- Landing: Once the plane has completed the sixth lap at waypoint A, the pilot takes manual control, continues flying clockwise, and lands the plane on the runway going WSW.

When the mission was complete, we downloaded the onboard log from the Ardupilot. In processing the data, we used the distance-to-waypoint signal to decide where to truncate each measurement with respect to time.

7.4 Experiment Execution

The experiment presented in this work was executed Tuesday May 27th, 2014 at the LMAC field. The weather was clear and sunny. We conducted four flights, which are summarized in Table 7.1.

For the experiment, we use the same pitch and altitude controller parameters as in the simulations and vary the value of k. Specifically, we let the FDI parameter-dependent polynomial $\eta_k(s) = (s+k)^4$; FDI reference model $\alpha_{\rm m}(s) = \beta_{\rm m}(s) = (s+4)(s+6)(s+8)$; estimated upper-bound on the magnitude of the leading Markov parameter used in the FDI controller $\bar{H}_d = 1,005$, where we know $\operatorname{sgn}(H_d) = -1$; PI proportional gain $K = -0.5$; PI zero $z_{\rm c} = -0.2$; altitude controller proportional gain

[1]We list the weather conditions measured at Blue Grass Airport (KLEX) [31], which is approximately 20 miles west of the LMAC field.

$k_{\mathrm{h,P}} = 1.6$; and the altitude controller integral gain $k_{\mathrm{h,I}} = 0.2$.

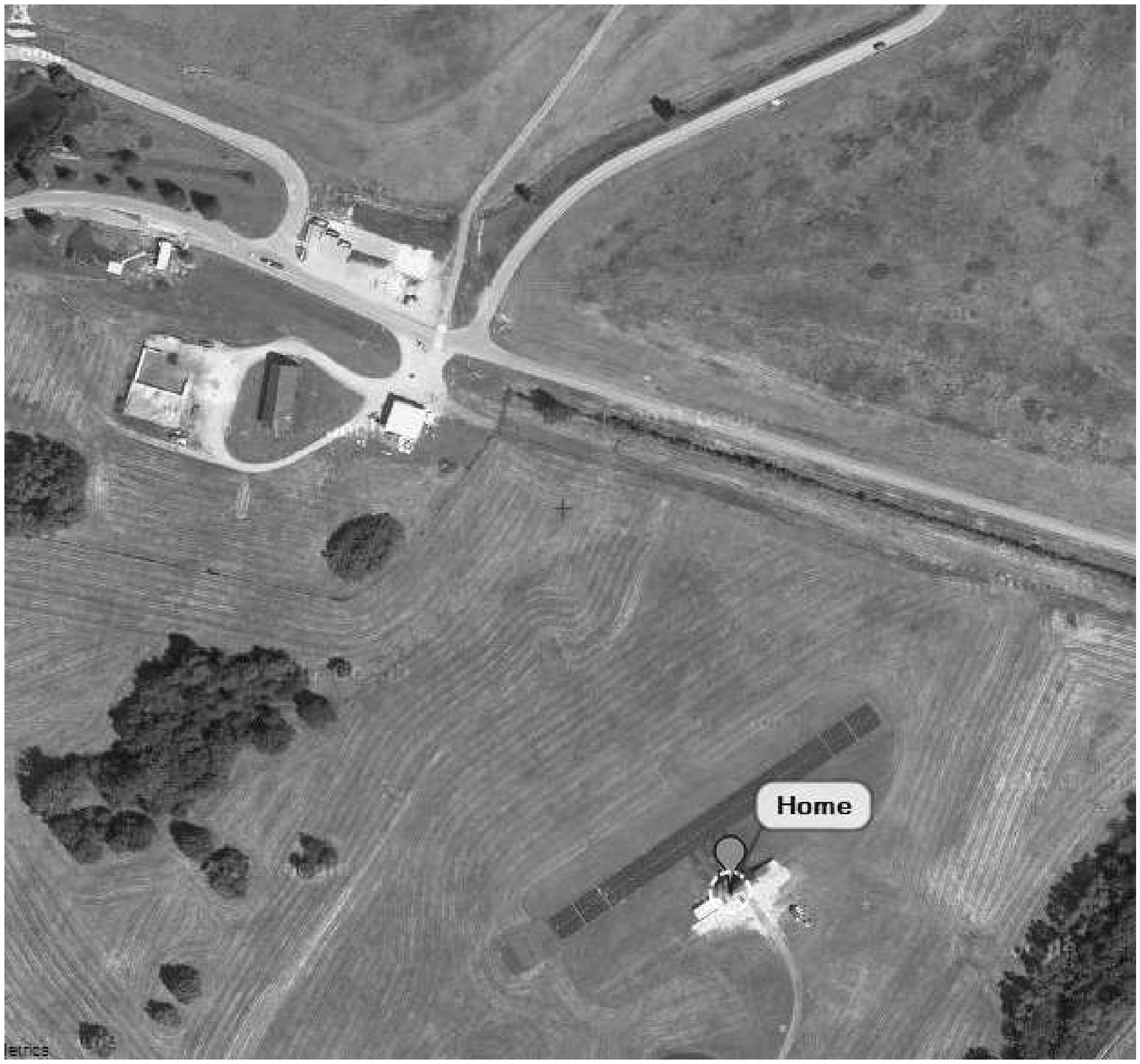


Figure 7.1: *Lexington Model Airplane Club.* Shown is a screenshot from the Mission Planner software. The Lexington Model Airplane Club field is located on the east side of Lexington, KY. The field features a 200 meter runway oriented WSW and ENE. The "Home" label indicates the GPS coordinates where the Ardupilot calibrates its zero altitude.

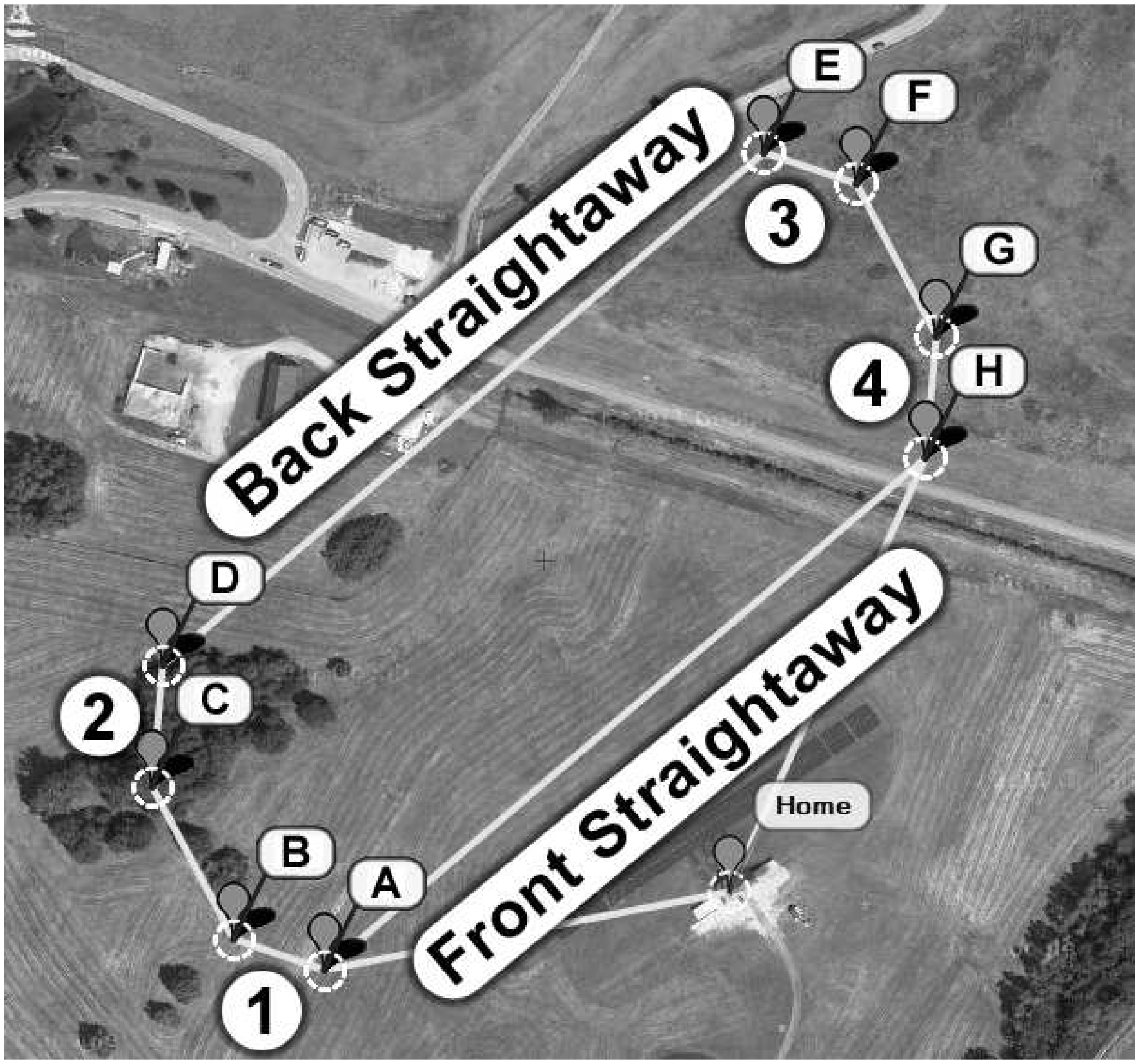


Figure 7.2: *Experimental Flight Path.* The experimental flightpath is designed resembling a clockwise oval racetrack. The flight path is designed such that the plane is flying straight and level for as long as possible without commanding any turns that may be outside of the plane's envelope in automatic control.

Chapter 8 Experimental Results

In this chapter, we present results from the experiment described in Chapter 7. We summarize the main result of the experiment, and discuss differences between the simulation and experiment.

8.1 Flight Results

8.1.1 Flight 1: FDI Controller with k=25

The first flight was conducted following the procedure described in Section 7.3. The FDI controller was used as the pitch controller first, then control was switched to the PI controller. Time histories of the pitch angle θ, pitch error $\theta_{\mathrm{d}} - \theta$, altitude h, altitude derivative $\dot{h}$, and elevator servo input u_{e} from the measurement portions of Flight 1 are shown in Figure 8.1. The discrete Fourier transforms of the pitch error and altitude error are shown in Figure 8.2, where the signals have been padded with zeros to make their length a power of two.

For the FDI controller, the average power of pitch error was 2.62 deg^2 and the average power of altitude error was 3.93 m^2. For the PI controller, the average power of pitch error was 13.71 deg^2 and the average power of altitude error was 10.14 m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 2.62. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 3.93.

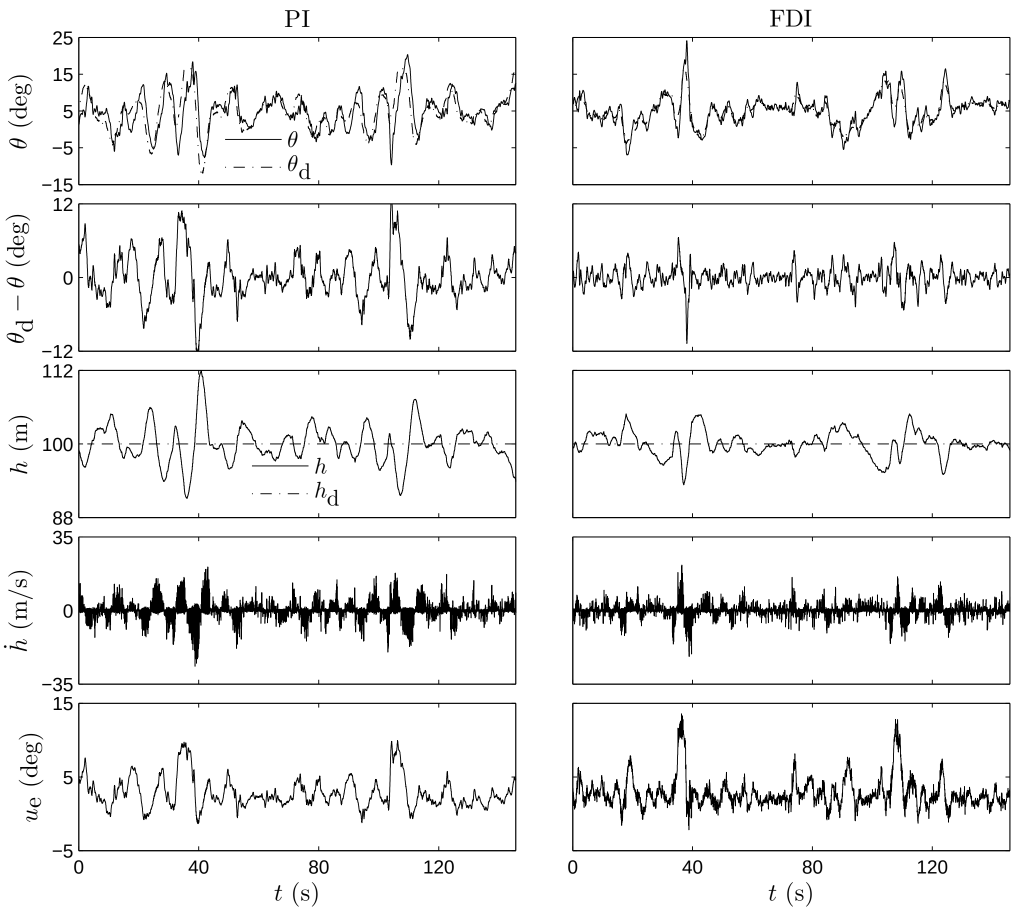


Figure 8.1: *Time Histories for Flight 1.* For this flight, $k = 25$. For the FDI case, pitch and altitude errors are visibly reduced while the control signal has larger peaks and faster rate.

8.1.2 Flight 2: FDI Controller with k=30

The second flight was conducted following the procedure described in Section 7.3. The PI controller was used as the pitch controller first, then control was switched to the FDI controller. Time histories of the pitch angle θ, pitch error $\theta_{\mathrm{d}} - \theta$, altitude h, altitude derivative $\dot{h}$, and elevator servo input u_{e} from the measurement portions of Flight 2 are shown in Figure 8.3. The discrete Fourier transforms of the pitch error and altitude error are shown in Figure 8.4, where the signals have been padded with zeros to make their length a power of two.

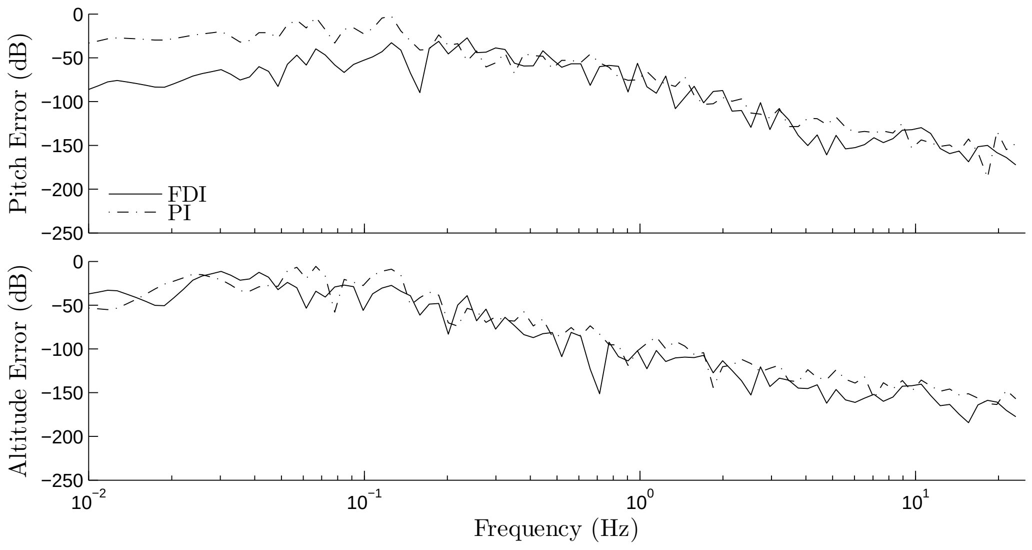


Figure 8.2: *Frequency Content for Flight 1.* For this flight, $k = 25$. For the FDI case, the frequency content below 0.2 Hz of the pitch error is smaller than that in the PI case. However, the frequency content of the altitude error is slightly smaller almost everywhere.

For the FDI controller, the average power of pitch error was 1.38 deg^2 and the average power of altitude error was 3.60 m^2. For the PI controller, the average power of pitch error was 10.90 deg^2 and the average power of altitude error was 14.47 m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 10.52. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 3.03.

8.1.3 Flight 3: FDI Controller with k=30

The third flight was conducted following the procedure described in Section 7.3. The FDI controller was used as the pitch controller first, then control was switched to the PI controller. Time histories of the pitch angle θ, pitch error $\theta_{\text{d}} - \theta$, altitude h, altitude derivative $\dot{h}$, and elevator servo input u_{e}, from the measurement portions of Flight 3 are shown in Figure 8.5. The discrete Fourier transforms of the pitch error

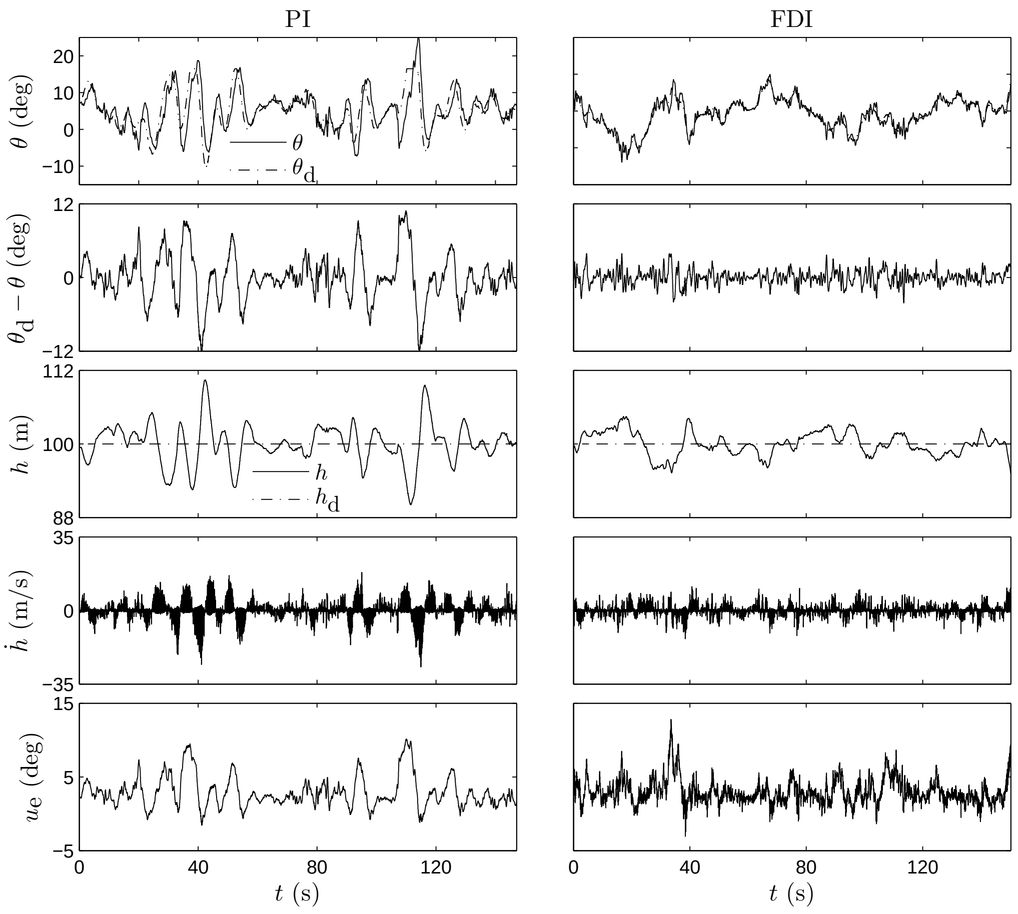


Figure 8.3: *Time Histories for Flight 2.* For this flight, $k = 30$. For the FDI case, pitch and altitude errors are visibly reduced while the control signal has larger peaks and faster rate.

and altitude error are shown in Figure 8.6, where the signals have been padded with zeros to make their length a power of two.

For the FDI controller, the average power of pitch error was 1.09 deg^2 and the average power of altitude error was 2.47 m^2. For the PI controller, the average power of pitch error was 7.09 deg^2 and the average power of altitude error was 11.16 m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 10.21. The ratio of the PI average power of altitude error to the FDI average power of altitude error is 2.87.

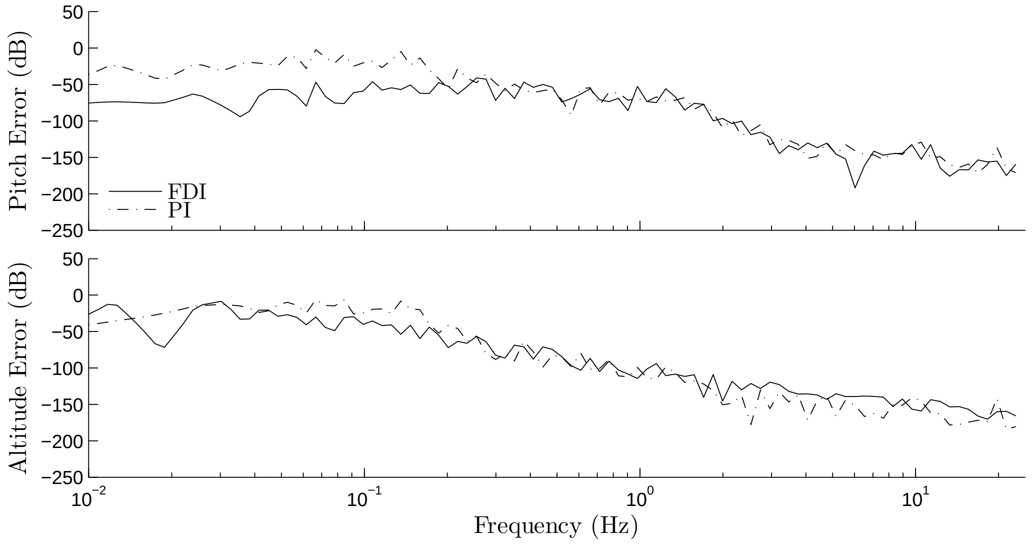


Figure 8.4: *Frequency Content for Flight 2.* For this flight, $k = 30$. For the FDI case, the frequency content below 0.2 Hz of the pitch error is smaller than that in the PI case. However, the frequency content of the altitude error is slightly smaller almost everywhere.

8.1.4 Flight 4: FDI Controller with k=12

The fourth flight was conducted following the procedure described in Section 7.3. The PI controller was used as the pitch controller first, then control was switched to the FDI controller. Time histories of the pitch angle θ, pitch error $\theta_\mathrm{d} - \theta$, altitude h, altitude derivative $\dot{h}$, and elevator servo input u_e, from the measurement portions of Flight 4 are shown in Figure 8.7. The discrete Fourier transforms of the pitch error and altitude error are shown in Figure 8.8, where the signals have been padded with zeros to make their length a power of two.

For the FDI controller, the average power of pitch error was 40.09 deg^2 and the average power of altitude error was 7.97 m^2. For the PI controller, the average power of pitch error was 8.05 deg^2 and the average power of altitude error was 5.60 m^2. Thus, the ratio of the PI average power of pitch error to the FDI average power of pitch error is 0.20. The ratio of the PI average power of altitude error to the FDI

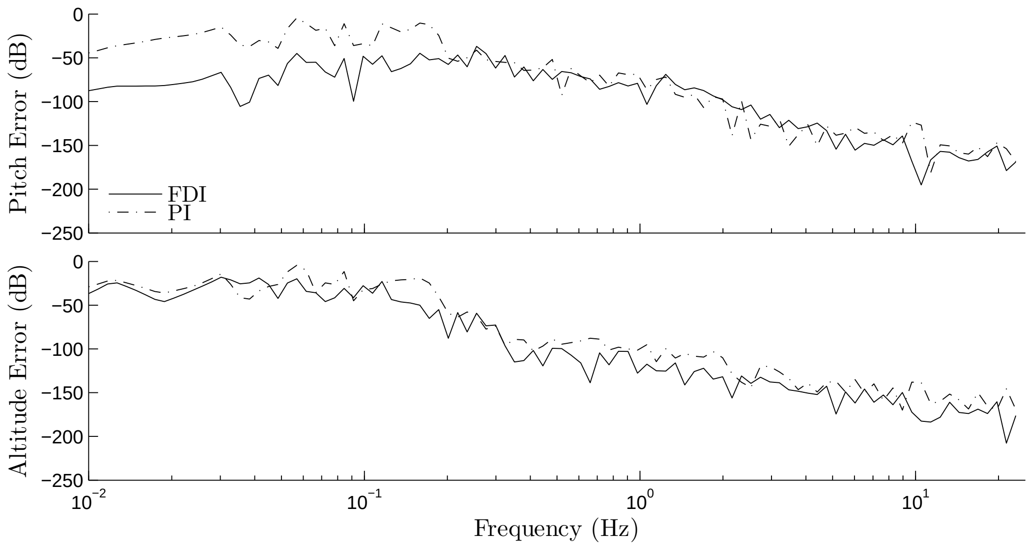


Figure 8.5: *Time Histories for Flight 3.* For this flight, $k = 30$. For the FDI case, pitch and altitude errors are visibly reduced and the control signal has larger peaks and faster rate.

Table 8.1: *Summary of Experimental Results.*

k	Flight	PI $\mathcal{P}_\theta$ (deg^2)	PI $\mathcal{P}_h$ (m^2)	FDI $\mathcal{P}_\theta$ (deg^2)	FDI $\mathcal{P}_h$ (m^2)	$\frac{\text{PI } \mathcal{P}_\theta}{\text{FDI } \mathcal{P}_\theta}$	$\frac{\text{PI } \mathcal{P}_h}{\text{FDI } \mathcal{P}_h}$
12	4	8.05	5.60	40.09	7.97	0.20	0.70
25	1	13.71	10.14	2.62	3.93	5.23	2.58
30	2	14.47	10.90	1.38	3.60	10.52	3.03
30	3	11.16	7.09	1.09	2.47	10.21	2.87

average power of altitude error is 0.70.

8.2 Discussion

Table 8.1 shows the average powers of performance from each flight, arranged in order of increasing k. In analyzing the results, we assume that the wind does not change appreciably during a single flight but may have changed between flights. Thus, to compare the FDI and PI, we examine the ratio between the average powers of performance. Recall our goal is to make the average powers of performance of pitch

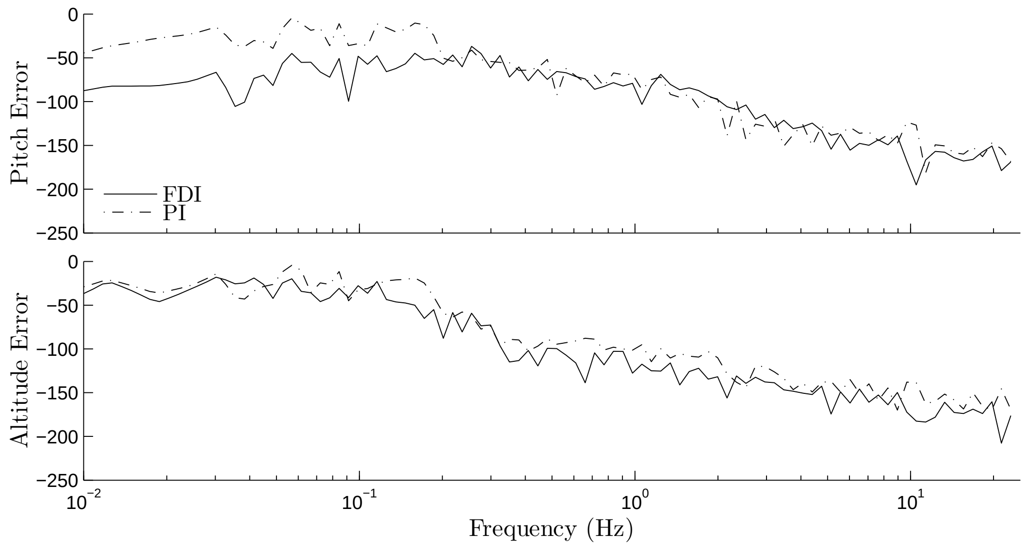


Figure 8.6: *Frequency Content for Flight 3.* For this flight, $k = 30$. For the FDI case, the low frequency content of the pitch error is much smaller than that in the PI case. However, the frequency content of the altitude error is slightly smaller almost everywhere.

and altitude error for the FDI controller smaller than those of the PI controller. Thus, we would like to make the ratio of the PI controller's average powers of a pitch and altitude error large. Table 8.1 shows an increase in these ratios with increasing k. Additionally, when we repeated the experiment for $k = 30$, the ratios of average power of pitch error were 3.0% percent different from each other, and the ratios of average power of altitude error were 5.4% different from each other. Thus, we argue that we have accomplished our goal of improving the command-following and disturbance-rejection properties of the altitude autopilot compared to a baseline PI controller.

Additionally, Figures 8.2, 8.4, and 8.6 show the frequency bands where the FDI improves pitch and altitude errors. Pitch errors are improved at low frequency while altitude errors are slightly improved at most frequencies.

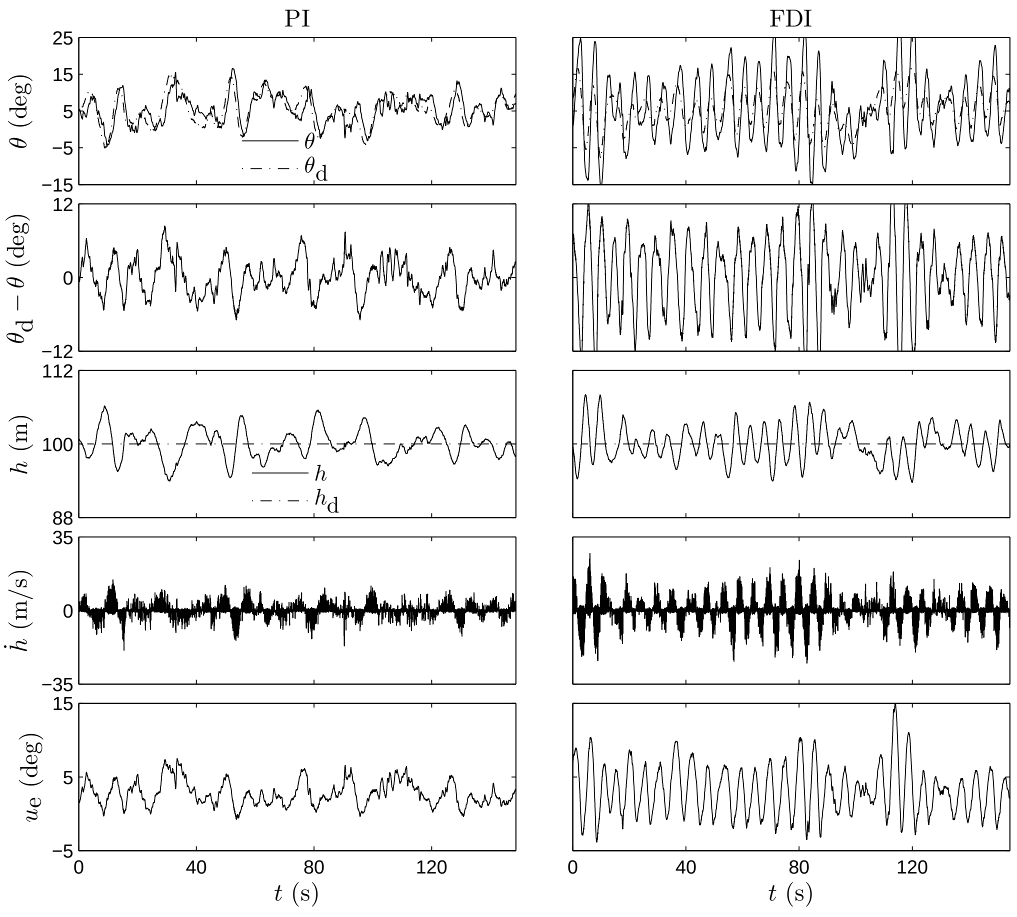


Figure 8.7: *Time Histories for Flight 4.* For this flight, $k = 12$. For the FDI case, the system is clearly at the edge of instability. Neither average power of pitch error or of altitude error was improved over the PI.

8.3 Discrepancies Between Simulation and Experiment

The ratios of powers of performance for the experiment appear to favor the FDI controller more than simulation would have suggested. First, we discuss several possible sources of variance that we believe were well-controlled during the experiment.

A possible source of variation in comparing the experimental results with the simulation results is airspeed error. However, in processing the flight data, we compute the average power of airspeed error for each measurement segment. For the PI con-

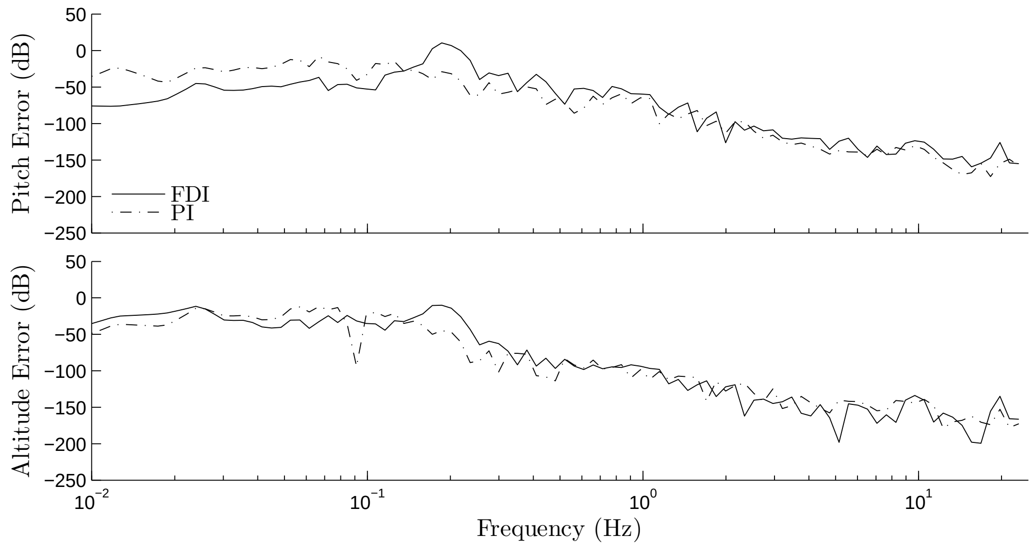


Figure 8.8: *Frequency Content for Flight 4.* For this flight, $k = 12$. For the FDI case, the low frequency content of the pitch error is slightly smaller than that in the PI case, however there is a large bump in the FDI around 0.2 Hz and the FDI is slightly larger at high frequency. The frequency content of the altitude error is slightly larger almost everywhere.

troller, the average powers of airspeed error range from 0.76 to 1.63 m^2/s^2. For the FDI controller, the average powers of airspeed error range from 0.28 to 0.72 m^2/s^2, where the largest was during flight 1, where $k = 25$.

Another possible source of variation that we argue is well-controlled is initial condition response in beginning the experiment and in switching controllers mid-flight. The presence of errors at the beginning of each measurement segment could skew the finite time powers of performance of pitch and altitude error. We made an effort to control any initial condition response by disregarding laps one and four. We argue on the basis of the time series presented in Figures 8.1, 8.3, 8.5, and 8.7 that the errors present at the beginning of each measurement are no larger than those seen during the two laps of measurement.

Next, we pose several possible explanations for the discrepancies between the simulation results and experimental results.

The most obvious difference between the simulations and the experiment is the flight path. The simulations did not implement a heading control, only a roll angle controller. In the simulations, the roll angle was controlled to zero, i.e. level flight. In the experiment, a navigation loop gave a roll command based on heading error. Thus, if the FDI controller were superior when the plane was not level, this would tend to favor the FDI in the experiment more than in the simulations.

Although airspeed errors were well-controlled during measurement, we cannot confirm equivalence between the airspeed controller we designed for simulation and the airspeed controller we used in the experiment. The airspeed dynamics are coupled with the pitch and altitude dynamics. Thus, differences in the airspeed controllers could be manifest in the difference between simulation and experiment.

For the simulation results, we use models of the wind, which may be dissimilar to the wind encountered during the experiment. As shown in Figures 5.9 and 5.10, the FDI and PI controllers suppress disturbances differently at different frequencies. In the case of altitude error, the FDI is not superior at every frequency. It is possible that the frequency content of the wind during the experiment favored the FDI controller.

Finally, differences between the estimated value of $\bar{H}_d$ and the actual value during experiment could cause changes in the average powers of performance of the FDI controller. Underestimating $\bar{H}_d$ can change the high-parameter-stabilizing nature of the FDI controller, while overestimating $\bar{H}_d$ can change the stability properties of the FDI controller for low values of k.

Chapter 9 Conclusions and Future Work

In this thesis, we implemented the FDI controller as the pitch controller in an altitude-hold autopilot. We designed and simulated the continuous- and discrete-time FDI controller with anti-windup on a nonlinear aircraft model. We showed numerically that the average power of altitude error improves (relative to a classical PI controller) for FDI parameter values in a range that could be implemented on digital hardware.

We conducted a flight experiment comparing the FDI controller to a classical PI controller. The experimental results showed that the FDI controller performed better than the PI controller at certain values of the parameter.

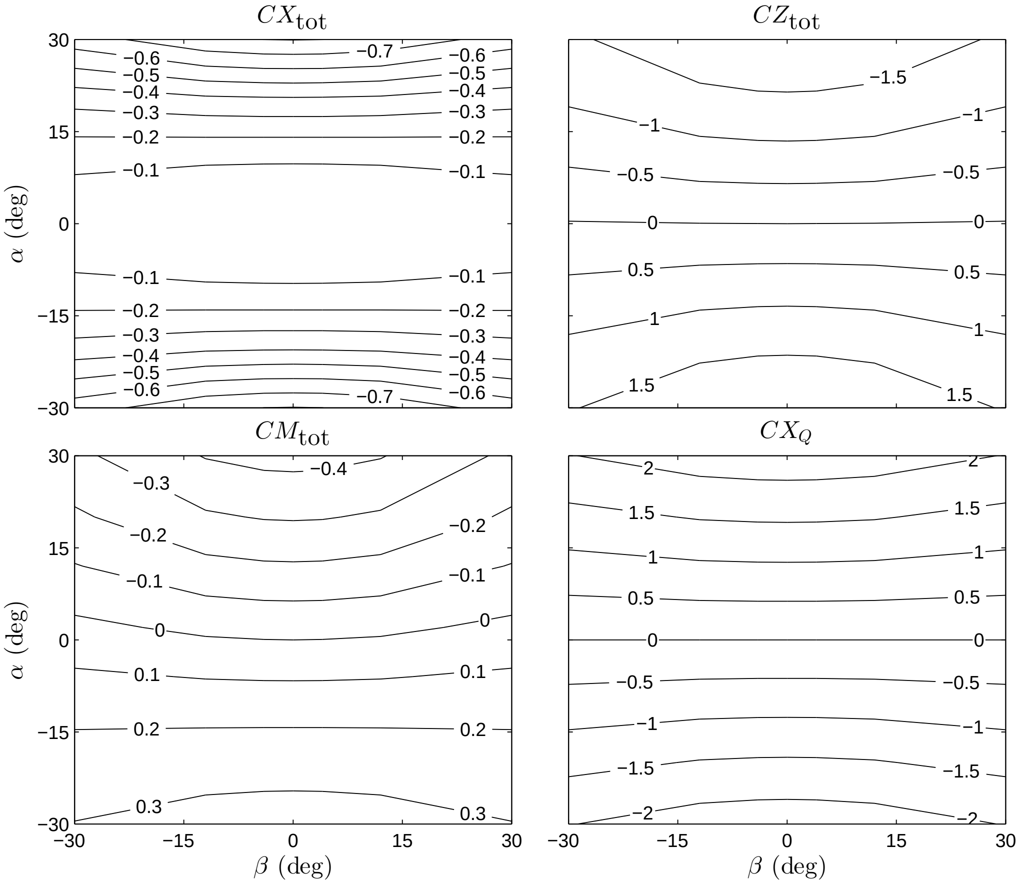
Future Work

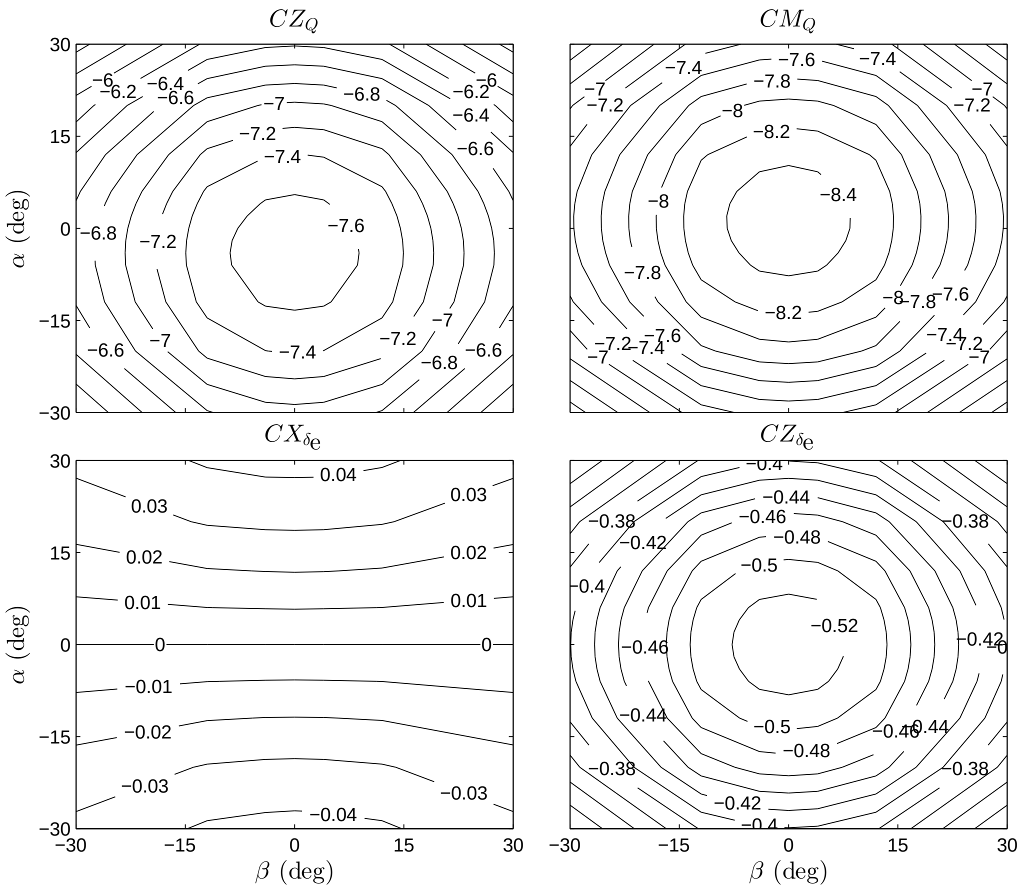
We suggest several ways of moving forward with the work presented in this thesis. First, we must acknowledge that the UAV will probably be flown in limited airspace, and thus will need to turn. We suggest altering the simulation to include the default navigation (i.e. way-point tracking) controller present on the autopilot. If more precise control is desired when the plane is turning, then MIMO filtered dynamic inversion is viewed as an option.

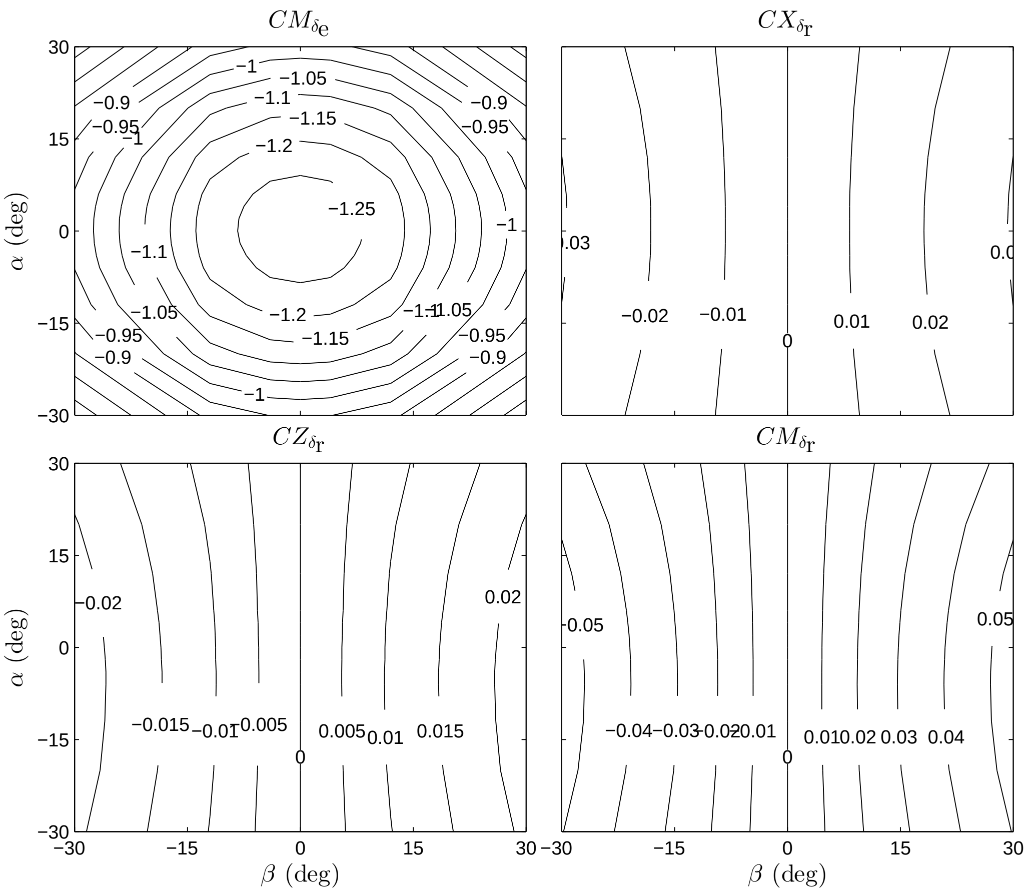
Secondly, we suggest the controller be tested more thoroughly. One experiment that would strengthen this thesis is tuning the PI controller to some optimal gains. Another is commanding the altitude ramps that will be crucial to way-point tracking during instrumented turbulence measurement.

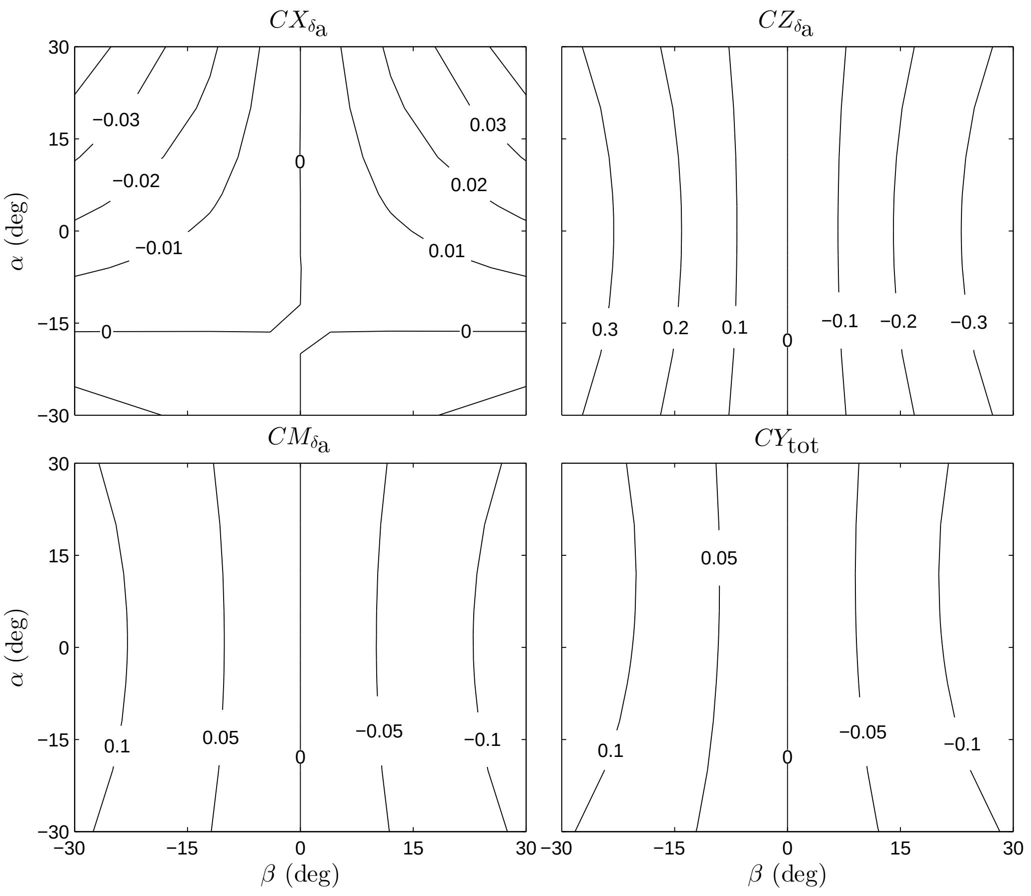
Appendices

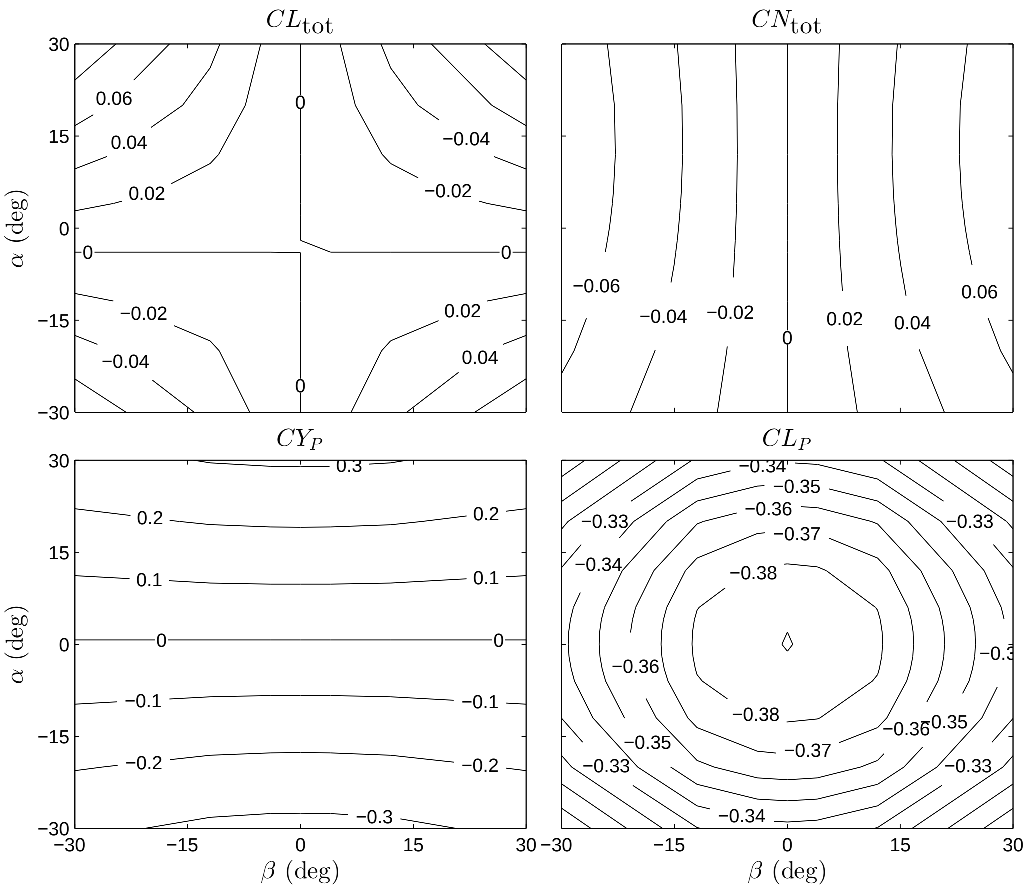
A Aerodynamic Model Parameters

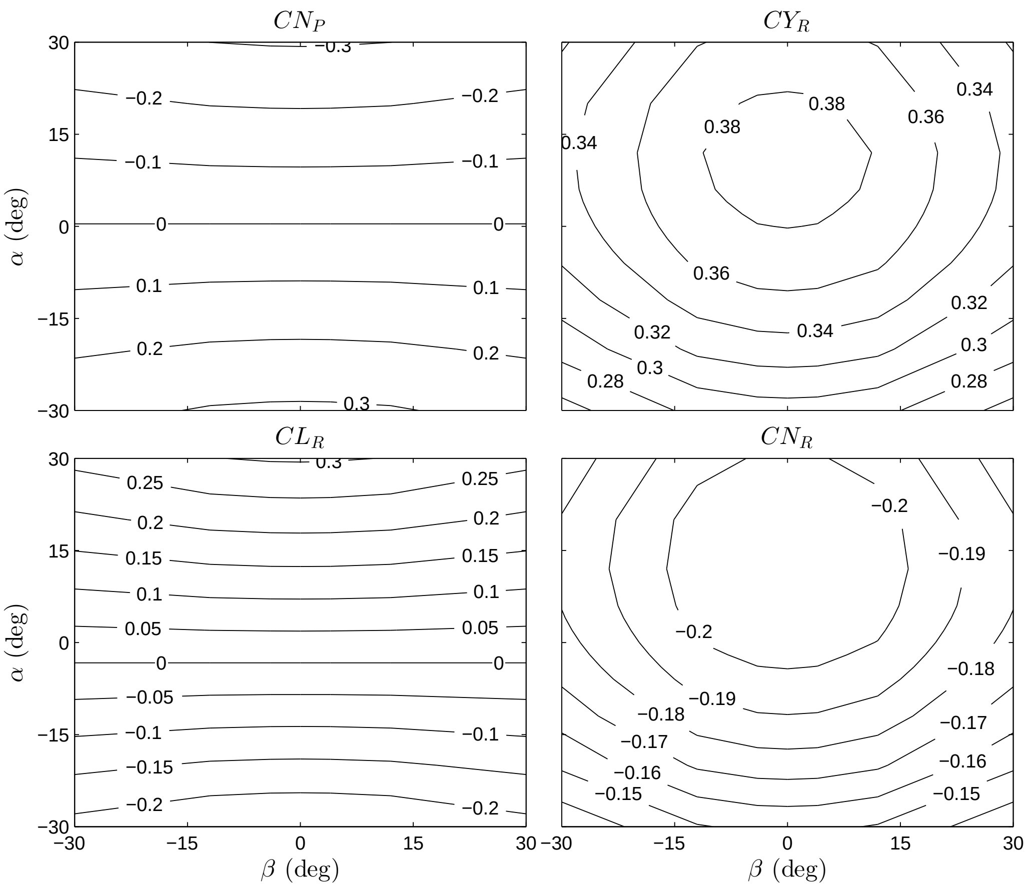


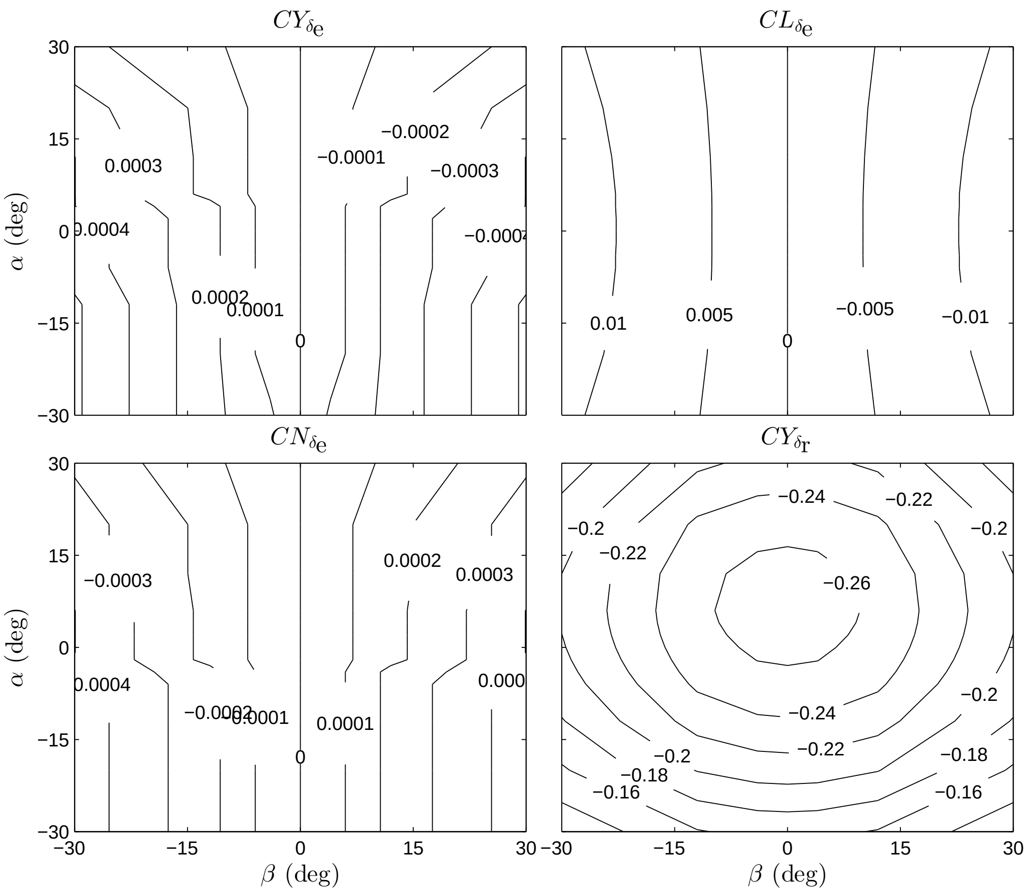

CZ_Q
CM_Q
CX_{δ_e}
CZ_{δ_e}
α (deg)
β (deg)
30
15
0
−15
−30
−6
−6.2
−6.4
−6.6
−6.8
−7
−7.2
−7.4
−7.6
−7.8
−8
−8.2
−8.4
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04
−0.38
−0.4
−0.42
−0.44
−0.46
−0.48
−0.5
−0.52

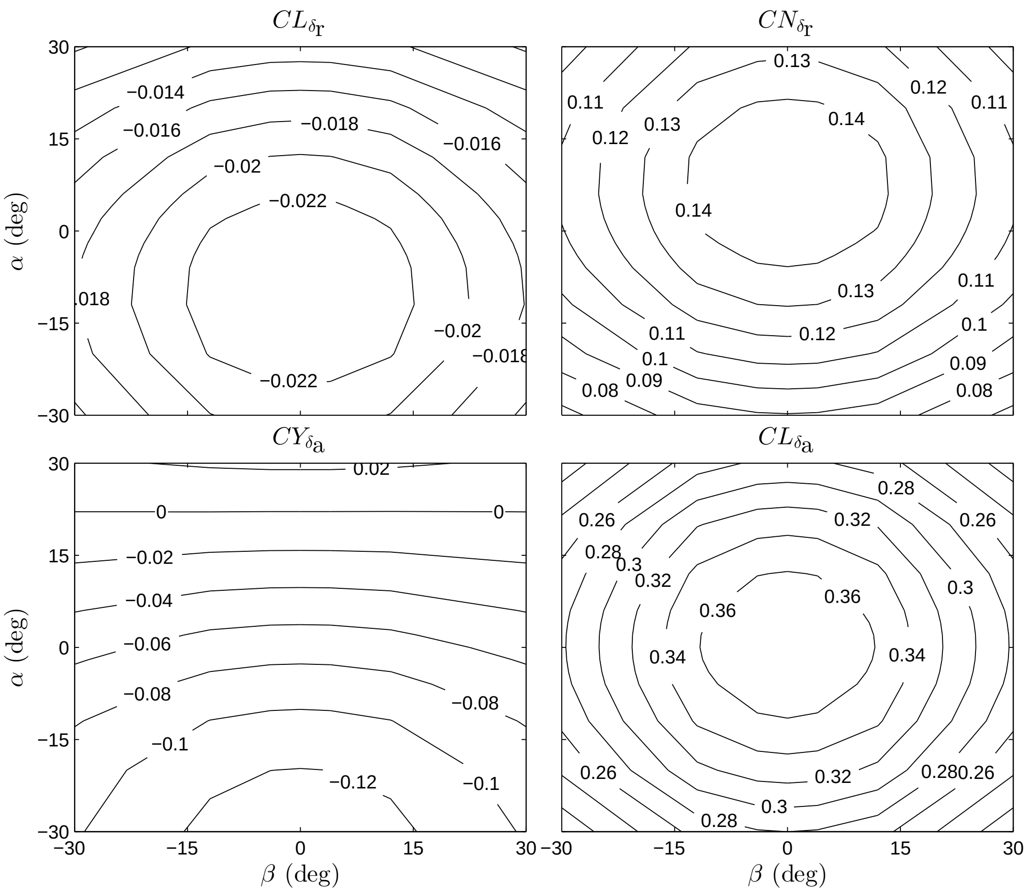

CM_{δ_e}
CX_{δ_r}
CZ_{δ_r}
CM_{δ_r}
α (deg)
β (deg)
−1.25
−1.2
−1.15
−1.1
−1.05
−1
−0.95
−0.9
−0.02
−0.01
0
0.01
0.02
−0.015
−0.005
0.005
0.015
−0.05
−0.04
−0.03
0.03
0.04
0.05


CX_{δ_a}
CZ_{δ_a}
CM_{δ_a}
CY_{tot}
−0.03
−0.02
−0.01
0
0.01
0.02
0.03
0.3
0.2
0.1
−0.1
−0.2
−0.3
0.05
−0.05
α (deg)
β (deg)
30
15
0
−15
−30

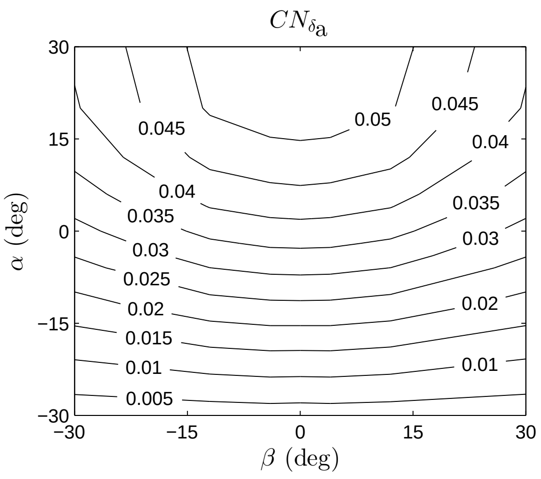

CL_{tot}
CN_{tot}
CY_P
CL_P
α (deg)
β (deg)
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
0.3
0.2
0.1
−0.1
−0.2
−0.3
−0.33
−0.34
−0.35
−0.36
−0.37
−0.38
30
15
−15
−30


CN_P
CY_R
CL_R
CN_R
α (deg)
β (deg)
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.38
0.36
0.34
0.32
0.3
0.28
0.25
0.15
0.05
−0.05
−0.15
−0.2
−0.19
−0.18
−0.17
−0.16
−0.15


$CY_{\delta e}$
$CL_{\delta e}$
$CN_{\delta e}$
$CY_{\delta r}$
α (deg)
β (deg)
30
15
0
−15
−30
0.0003
0.0004
0.0002
0.0001
0
−0.0001
−0.0002
−0.0003
−0.0004
0.01
0.005
−0.005
−0.01
−0.26
−0.24
−0.22
−0.2
−0.18
−0.16
0.0001
0.0002
0.0003



CL_{δ_r}
CN_{δ_r}
CY_{δ_a}
CL_{δ_a}
α (deg)
β (deg)
−0.014
−0.016
−0.018
−0.02
−0.022
0.08
0.09
0.1
0.11
0.12
0.13
0.14
0.02
0
−0.02
−0.04
−0.06
−0.08
−0.1
−0.12
0.26
0.28
0.3
0.32
0.34
0.36



CN_{δ_a}
30
15
0
−15
−30
−30
−15
0
15
30
α (deg)
β (deg)
0.05
0.045
0.045
0.04
0.04
0.035
0.035
0.03
0.03
0.025
0.02
0.02
0.015
0.01
0.01
0.005

B Model Parameters Used in Linearization

Trim Condition			
m	4.48 kg	I_{yy}	0.3287 kg·m^2
U_0	19.90 m/s	W_0	1.95 m/s
θ_0	5.60°	$\delta_{\text{e},0}$	-3.94°
Partial Derivatives			
Dimensionless		Dimensional	
CX_U	0	$\frac{\partial X_{\text{a}}}{\partial U}$	0 kg/s
CZ_U	-0.3947	$\frac{\partial Z_{\text{a}}}{\partial U}$	-2.0242 kg/s
CM_U	0.0016	$\frac{\partial M}{\partial U}$	0.0026 kg·m/s
CX_W	0.6174	$\frac{\partial X_{\text{a}}}{\partial W}$	3.4316 kg/s
CZ_W	-4.4178	$\frac{\partial Z_{\text{a}}}{\partial W}$	-24.554 kg/s
CM_W	-0.9232	$\frac{\partial M}{\partial W}$	-1.5266 kg·m/s
CX_Q	0.4035	$\frac{\partial X_{\text{a}}}{\partial Q}$	0.3336 kg·m/s
CZ_Q	-7.5992	$\frac{\partial Z_{\text{a}}}{\partial Q}$	-6.2827 kg·m/s
CM_Q	-8.5003	$\frac{\partial M}{\partial Q}$	-2.0907 kg·m^2/s
$CX_{\delta_{\text{e}}}$	0.0187	$\frac{\partial X_{\text{a}}}{\partial \delta_{\text{e}}}$	2.0727 N
$CZ_{\delta_{\text{e}}}$	-0.5265	$\frac{\partial Z_{\text{a}}}{\partial \delta_{\text{e}}}$	-58.246 N
$CM_{\delta_{\text{e}}}$	-1.2716	$\frac{\partial M}{\partial \delta_{\text{e}}}$	-41.852 N·m

C MATLAB Script to Generate FDI Controller Matrices

```
clear all


k=25;

H_d = -1005;
```

```
eta = poly([-k -k -k -k]);
eta_0 = eta(end);
eta_bar = eta(1:length(eta)-1);

alpha = poly([-4 -6 -8]);

G_ue_n = eta_0/H_d*alpha;
G_ue_d = [eta_bar 0];

FDI_K_I = polyval(alpha,0)/polyval(eta_bar,0);
c = alpha-FDI_K_I*eta_bar;
c = c(1:length(c)-1);

A = [-G_ue_d(2:end); ...
     1 0 0 0;
     0 1 0 0;
     0 0 0 0];

B = [1 0 0 FDI_K_I]';

C = eta_0*[c 1];

A_trunc = A(1:3,1:3);
B_trunc = B(1:3);
C_trunc = C(1:3);

A_d = expm(A_trunc*0.02);
```

```
B_d = inv(A_trunc)*(expm(A_d-eye(3)))*B_trunc*3E4;
C_d = C/3E4;
C_d_w_H_d = C_d/H_d;

int_B = FDI_K_I * eta_0 * 0.02;

out_file = fopen('FDI_params.h', 'w');

fprintf(out_file, '#define FDI_A11 %4.10f \n', A_d(1,1));
fprintf(out_file, '#define FDI_A12 %4.10f \n', A_d(1,2));
fprintf(out_file, '#define FDI_A13 %4.10f \n', A_d(1,3));
fprintf(out_file, '#define FDI_A21 %4.10f \n', A_d(2,1));
fprintf(out_file, '#define FDI_A22 %4.10f \n', A_d(2,2));
fprintf(out_file, '#define FDI_A23 %4.10f \n', A_d(2,3));
fprintf(out_file, '#define FDI_A31 %4.10f \n', A_d(3,1));
fprintf(out_file, '#define FDI_A32 %4.10f \n', A_d(3,2));
fprintf(out_file, '#define FDI_A33 %4.10f \n', A_d(3,3));

fprintf(out_file, '#define FDI_B1 %4.10f \n', B_d(1));
fprintf(out_file, '#define FDI_B2 %4.10f \n', B_d(2));
fprintf(out_file, '#define FDI_B3 %4.10f \n', B_d(3));

fprintf(out_file, '#define FDI_C1 %4.10f \n', C_d(1));
fprintf(out_file, '#define FDI_C2 %4.10f \n', C_d(2));
fprintf(out_file, '#define FDI_C3 %4.10f \n', C_d(3));

fprintf(out_file, '#define FDI_BI %4.10f \n', int_B);
```

```
fclose(out_file);
```

D AP_PitchController.h

```
// -*- tab-width: 4; Mode: C++; c-basic-offset: 4; indent-tabs-mode: nil -*-

#ifndef __AP_PITCH_CONTROLLER_H__
#define __AP_PITCH_CONTROLLER_H__

#include <AP_AHRS.h>
#include <AP_Common.h>
#include <math.h> // for fabs()

class AP_PitchController {
public:
AP_PitchController() {
AP_Param::setup_object_defaults(this, var_info);
}

void set_ahrs(AP_AHRS *ahrs) { _ahrs = ahrs; }

int32_t get_servo_out(int32_t angle_err, float scaler = 1.0,
bool stabilize = false, int16_t aspd_min = 0, int16_t aspd_max = 0);

void reset_I();

int16_t which_controller();
static const struct AP_Param::GroupInfo var_info[];

private:
AP_Float _K_P;
AP_Float _K_I;
AP_Float _roll_ff;
AP_Float _H_d;
AP_Float _ele_servo_max;
AP_Int16 _pi_or_fdi;

uint32_t _last_t;
float _last_out;

float _integrator;

float _x1;
float _x2;
float _x3;
float _x4;

float _ele_out;

AP_AHRS *_ahrs;

};
```

```
#endif // __AP_PITCH_CONTROLLER_H__
```

E AP_PitchController.cpp

```
// -*- tab-width: 4; Mode: C++; c-basic-offset: 4; indent-tabs-mode: nil -*-

// Initial Code by Jon Challinger
//  Modified by Paul Riseborough
// This library is free software; you can redistribute it and / or
// modify it under the terms of the GNU Lesser General Public
// License as published by the Free Software Foundation; either
// version 2.1 of the License, or (at your option) any later version.

#include <AP_Math.h>
#include <AP_HAL.h>
#include <AP_Common.h>
#include "AP_PitchController.h"
#include "FDI_params.h"

extern const AP_HAL::HAL& hal;
const AP_Param::GroupInfo AP_PitchController::var_info[] PROGMEM = {
AP_GROUPINFO("P",        0, AP_PitchController, _K_P,          0.4f),
AP_GROUPINFO("I",        1, AP_PitchController, _K_I,        0.0f),
AP_GROUPINFO("RLL",      2, AP_PitchController, _roll_ff,        1.0f),
AP_GROUPINFO("CONT",      3, AP_PitchController, _pi_or_fdi,        0),
AP_GROUPINFO("H_D",      4, AP_PitchController, _H_d,        0),
AP_GROUPINFO("TRVL",      5, AP_PitchController, _ele_servo_max,        0),
AP_GROUPEND
```

```
};


int32_t AP_PitchController::get_servo_out(int32_t angle_err,
      float scaler, bool stabilize,
      int16_t aspd_min, int16_t aspd_max)
{

angle_err = angle_err * (float)4500/_ele_servo_max;


float aspeed;
float rate_offset;
float bank_angle = _ahrs->roll;
bool inverted = false;
uint32_t tnow = hal.scheduler->millis();
uint32_t dt = tnow - _last_t;


if (_last_t == 0 || dt > 1000) {
dt = 0;
}
_last_t = tnow;


if(_ahrs == NULL) return 0;
float delta_time    = (float)dt * 0.001f;


if (fabsf(bank_angle) < radians(90)) {
    bank_angle = constrain_float(bank_angle,-radians(80),radians(80));
} else {
```

```
inverted = true;
if (bank_angle > 0.0f) {
bank_angle = constrain_float(bank_angle,radians(100),radians(180));
} else {
bank_angle = constrain_float(bank_angle,-radians(180),-radians(100));
}
}
if (!_ahrs->airspeed_estimate(&aspeed)) {
        aspeed = 0.5f*(float(aspd_min) + float(aspd_max));
}

if (_pi_or_fdi==0) {
float _integrator_delta = delta_time*(float)angle_err*_K_I;
if (((float)angle_err*_K_P+_integrator+_integrator_delta)>4500 ||
    ((float)angle_err*_K_P+_integrator+_integrator_delta)<-4500) {
_integrator_delta = 0;
}

_integrator = _integrator + _integrator_delta;

_ele_out = constrain_float(((float)angle_err*_K_P+_integrator),-4500,4500);
} else {

float _last_x1 = _x1;
float _last_x2 = _x2;
float _last_x3 = _x3;
float _last_x4 = _x4;
```

```
_x1 = FDI_A11 * _last_x1 + FDI_A12 * _last_x2 + FDI_A13 * _last_x3 +
      FDI_B1 * (float)angle_err;
_x2 = FDI_A21 * _last_x1 + FDI_A22 * _last_x2 + FDI_A23 * _last_x3 +
      FDI_B2 * (float)angle_err;
_x3 = FDI_A31 * _last_x1 + FDI_A32 * _last_x2 + FDI_A33 * _last_x3 +
      FDI_B3 * (float)angle_err;
_x4 = _x4 + FDI_BI * angle_err;

_ele_out = (FDI_C1 * _x1 + FDI_C2 * _x2 + FDI_C3 * _x3 + _x4)/_H_d;

if ((_ele_out>4500) || (_ele_out<-4500)) {
_x4 = _last_x4;
}

_ele_out = (FDI_C1 * _x1 + FDI_C2 * _x2 + FDI_C3 * _x3 + _x4)/_H_d;
_ele_out = constrain_float(_ele_out,-4500,4500);
}

return _ele_out;
}

void AP_PitchController::reset_I()
{
_integrator = 0.0f;

_x1 = 0.0f;
```

```
_x2 = 0.0f;

_x3 = 0.0f;

_x4 = 0.0f;

}


int16_t AP_PitchController::which_controller()

{

return _pi_or_fdi;

}
```

F FDI_params.h

```
#define FDI_A11 -0.0873081361

#define FDI_A12 -30.9794433151

#define FDI_A13 -411.0032226106

#define FDI_A21 0.0065760516

#define FDI_A22 0.5702970201

#define FDI_A23 -6.3192499584

#define FDI_A31 0.0001011080

#define FDI_A32 0.0166868515

#define FDI_A33 0.9494520176

#define FDI_B1 82.4009478931

#define FDI_B2 2.6624002385

#define FDI_B3 -0.4275192984

#define FDI_C1 12.9808333333

#define FDI_C2 230.3750000000
```

```
#define FDI_C3 1204.1666666667

#define FDI_BI 24.0000000000
```

Bibliography

[1] R. K. Pachauri, A. Reisinger, *et al.*, "Contribution of working groups i, ii and iii to the fourth assessment report of the intergovernmental panel on climate change," *Climate change 2007, synthesis report*, 2007.

[2] T. Foken, *Micrometeorology.* Springer, 2008.

[3] M. Metzger and H. Holmes, "Time scales in the unstable atmospheric surface layer," *Boundary-layer meteorology*, vol. 126, no. 1, pp. 29–50, 2008.

[4] G. Trevino and E. L. Andreas, "On reynolds averaging of turbulence time series," *Boundary-layer meteorology*, vol. 128, no. 2, pp. 303–311, 2008.

[5] D. H. Lenschow and W. B. Johnson Jr, "Concurrent airplane and balloon measurements of atmospheric boundary-layer structure over a forest," *Journal of Applied Meteorology*, vol. 7, no. 1, pp. 79–89, 1968.

[6] W. M. Angevine, S. Avery, and G. Kok, "Virtual heat flux measurements from a boundary-layer profiler-rass compared to aircraft measurements," *Journal of Applied Meteorology*, vol. 32, no. 12, pp. 1901–1907, 1993.

[7] C. R. Philbrick, "Raman lidar measurements of atmospheric properties," in *SPIE's International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, pp. 922–931, International Society for Optics and Photonics, 1994.

[8] W. L. Eberhard, R. E. Cupp, and K. R. Healy, “Doppler lidar measurement of profiles of turbulence and momentum flux,” *Journal of Atmospheric and Oceanic Technology*, vol. 6, no. 5, pp. 809–819, 1989.

[9] V. Matvev, U. Dayan, I. Tass, and M. Peleg, “Atmospheric sulfur flux rates to and from israel,” *Science of the total environment*, vol. 291, no. 1, pp. 143–154, 2002.

[10] A. van den Kroonenberg, T. Martin, M. Buschmann, J. Bange, and P. Vorsmann, “Measuring the wind vector using the autonomous mini aerial vehicle M2AV,” *Journal of Atmospheric and Oceanic Technology*, vol. 25, no. 11, pp. 1969–1982, 2008.

[11] S. Mayer, M. Jonassen, A. Sandvik, and J. Reuder, “Atmospheric profiling with the UAS SUMO: a new perspective for the evaluation of fine-scale atmospheric models,” *Meteorology and Atmospheric Physics*, vol. 116, no. 1–2, pp. 15–26, 2012.

[12] R. M. Thomas, K. Lehmann, H. Nguyen, D. L. Jackson, D. Wolfe, and V. Ramanathan, “Measurement of turbulent water vapor fluxes using a lightweight unmanned aerial vehicle system,” *Atmos. Meas. Tech*, vol. 5, pp. 5529–5568, 2012.

[13] J. B. Hoagg and T. M. Seigler, “Filtered-dynamic-inversion control for unknown minimum-phase systems with unknown-and-unmeasured disturbances,” *International Journal of Control*, vol. 86, no. 3, pp. 449–468, 2013.

[14] T. M. Seigler and J. B. Hoagg, “Filtered dynamic inversion for vibration control of structures with uncertainty,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 4, p. 041017, 2013.

[15] B. Etkin, *Dynamics of Atmospheric Flight.* Dover Publications, 2000.

[16] W. F. Phillips, *Mechanics of Flight.* John Wiley & Sons, 2010.

[17] R. C. Nelson, *Flight Stability and Automatic Control.* McGraw-Hill, New York, 1996.

[18] K. Johanastrom and L. Rundqwist, "Integrator windup and how to avoid it," in *American Control Conference, 1989*, pp. 1693–1698, IEEE, 1989.

[19] B. Wittenmark, "Integrators, nonlinearities, and anti-reset windup for different control structures," in *American Control Conference, 1989*, pp. 1679–1683, IEEE, 1989.

[20] Y. Peng, D. Vrancic, and R. Hanus, "Anti-windup, bumpless, and conditioned transfer techniques for PID controllers," *Control Systems, IEEE*, vol. 16, no. 4, pp. 48–57, 1996.

[21] M. Drela and H. Youngren, "AVL 3.30 user primer." `http://web.mit.edu/drela/Public/web/avl/avl_doc.txt`, 2010.

[22] AeroWorks, *.46-.61 EDGE 540T ARF Assembly Manual.*

[23] W. Gracey, "The experimental determination of the moments of inertia of airplanes by a simplified compound-pendulum method," Tech. Rep. 1629, Washington, DC, 1948.

[24] 3d Robotics, "Arduplane — fixed-wing aircraft UAV." `http://plane.ardupilot.com/`, 2014.

[25] V. M. Falkner, "The calculation of aerodynamic loading on surfaces of any shape," Tech. Rep. 1910, 1943.

[26] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation.* Wiley New York, 2003.

[27] S. B. Pope, *Turbulent flows.* Cambridge university press, 2000.

[28] S. Hodel and C. Hall, "Variable-structure PID control to prevent integrator windup," in *IEEE Transactions on Industrial Electronics*, vol. 2, pp. 442 – 451, IEEE, 2001.

[29] M. Tharayil and A. Alleyne, "A generalized PID error governing scheme for SMART/SBLI control," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 1, pp. 346–351, IEEE, 2002.

[30] "Downloads-ardupilot mega." https://code.google.com/p/ardupilot-mega/downloads/list, 2013. Accessed: 7-11-2014.

[31] Weather Underground, "History," *http://www.wunderground.com/history/.*

Vita

Jon Mullen has a Bachelor of Science degree from the University of Kentucky in Mechanical Engineering.