Instrumentation and Control of a Ducted Fan Unmanned Aerial Vehicle in Hover Mode

Benjamin P. Straub

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Daniel J. Stilwell, Chair

Craig A. Woolsey

A.A. (Louis) Beex

August 5, 2016

Blacksburg, Virginia

Keywords: Unmanned Aerial Vehicle, UAV, Control, Ducted Fan

Instrumentation and Control of a Ducted Fan Unmanned Aerial Vehicle in Hover Mode

Benjamin P. Straub

(ABSTRACT)

Unmanned aerial vehicles (UAVs) are increasingly being used for both military and commercial applications to replace more costly and dangerous manned operations. Vehicles with vertical take-off and landing (VTOL) and hovering capabilities are of interest for functions such as surveillance and inspection where the ability to hold the position of the vehicle is desired. Ducted fan vehicles are of particular interest because of their high efficiency per unit diameter when compared to the more commonly seen multirotor vehicles. This makes ducted fan UAVs very well suited for size-constrained missions such as indoor inspection or urban reconnaissance. However, the advantages of ducted fans come at the cost of complex nonlinear dynamics which present challenging modeling and control problems.

This thesis provides a detailed discussion of the instrumentation, modeling, and control of a ducted fan UAV. The dynamic model of the UAV is computed from a simplified parametric model. Unknown parameters of the model are found from system identification based on flight data. Synthesis of a linear state feedback controller based on this model is discussed, and it is demonstrated in hardware that this controller can effectively stabilize the vehicle.

Instrumentation and Control of a Ducted Fan Unmanned Aerial Vehicle in Hover Mode

Benjamin P. Straub

(GENERAL AUDIENCE ABSTRACT)

Unmanned aerial vehicles (UAVs) are increasingly being used for both military and commercial applications to replace more costly and dangerous manned operations. Vehicles with vertical take-off and landing (VTOL) and hovering capabilities are of interest for functions such as surveillance and inspection where the ability to hold the position of the vehicle is desired. Ducted fan vehicles are of particular interest because of their high efficiency per unit diameter when compared to the more commonly seen multirotor vehicles. This makes ducted fan UAVs very well suited for size-constrained missions such as indoor inspection or urban reconnaissance. However, the advantages of ducted fans come at the cost of complex dynamics which present challenging modeling and control problems.

This thesis provides a detailed discussion of the instrumentation, modeling, and control of a ducted fan UAV. The dynamic model of the UAV is computed from a simplified parametric model. Unknown parameters of the model are found from system identification based on flight data. Using this parametric model, development of a linear controller that uses feedback from the vehicle's state is discussed, and it is demonstrated in hardware that this controller can effectively stabilize the vehicle.

Acknowledgments

I owe many thanks to all those who have helped me through my time in graduate school.

First, I thank my advisor Dr. Daniel Stilwell. He has been a wonderful advisor throughout my time as a graduate research assistant, and I don't know how I would have finished without his help and support. Likewise, I would like to thank the rest of my thesis committee, Dr. Louis Beex and Dr. Craig Woolsey, for their support.

I want to thank my labmates at the Autonomous Systems and Controls Lab for their help as well as their comic relief. They've been a joy to work with. I especially want to thank my officemate Rami Jabari, who risked life and limb to help me test my ducted fan.

I owe many thanks to my family. Without the love, support, and guidance of my parents, I never would have gotten to Virginia Tech in the first place. I especially want to thank my amazing wife, Kayla, who has been incredibly supportive, helpful, and encouraging. And not once did she rub in the fact that she finished her thesis months before me.

Most importantly, I give thanks to God. My life has been incredibly blessed, and I owe all of my success to Him.

Contents

List of Figures

List of Tables

List of Acronyms

ADC	analog to digital converter
AHRS	attitude and heading reference system
AROD	airborne remotely operated device
BLDC	brushless direct current
CFD	computational fluid dynamic
DARPA	Defense Advanced Research Projects Agency
EDF	electric ducted fan
ESC	electronic speed controller
LQR	linear-quadratic regulator
MAV	micro air vehicle
OAV	organic air vehicle
ODE	ordinary differential equation
PPM	pulse position modulation
PWM	pulse width modulation
RC	radio-controlled

RMS root mean square

RPM revolutions per minute

UART universal asynchronous receiver/transmitter

UAV unmanned aerial vehicle

VPE vector processing engine

VTOL vertical take-off and landing

List of Symbols

$\boldsymbol{a}_b = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} \end{bmatrix}^\mathsf{T}$ body-frame linear acceleration vector

$\boldsymbol{a}_i$ inertial-frame linear acceleration vector

$\boldsymbol{A}$, $\boldsymbol{B}$ linear system state matrix and input matrix

$\boldsymbol{B}_a$, $\boldsymbol{B}_s$ input matrix, with the input defined as $\boldsymbol{\nu}_a$ and $\boldsymbol{\nu}_s$, respectively

C_r, C_s, C_{cs}, C_{aero} constants used in the equations for the moments $\boldsymbol{\tau}$

c_{gm}, $c_g s$ constants used in the input transformation function $g(\cdot)$

$\boldsymbol{g}(\boldsymbol{u}) = \begin{bmatrix} g(u_{s1}) & g(u_{s2}) & g(u_{s3}) & g(u_{s4}) \end{bmatrix}^\mathsf{T}$ function relating the input servo angles $\boldsymbol{u}$ to the effective force produced by each flap

$\tilde{\boldsymbol{g}}(\boldsymbol{u})$ modified form of $\boldsymbol{g}(\boldsymbol{u})$

$\boldsymbol{g}_a(\boldsymbol{u}) = \begin{bmatrix} g_\phi(\boldsymbol{u}) & g_\theta(\boldsymbol{u}) & g_\psi(\boldsymbol{u}) \end{bmatrix}^\mathsf{T}$ the effects of the input servo angles on the aircraft roll, pitch, and yaw

h_{sonar}, h_{IR}, h_{baro} altitude measurement from the ultrasonic distance sensor, IR distance

sensor, and barometer, respectively

$\ddot{h}_{acc} = a_{zi}$ altitudinal acceleration determined from the accelerometer

h_{est}, $\dot{h}_{est}$ complementary filtered altitude estimate and altitudinal velocity estimate, respectively

h_{meas} altitude measurement estimate determined from h_{sonar}, h_{IR}, and h_{baro}

I_x, I_y, I_z moments of inertia

$J(\cdot)$ cost function minimized by the LQR problem

$J_\psi(\cdot)$, $J_{\phi\theta}(\cdot)$ cost functions for the yaw model fitting and roll/pitch model fitting

k discrete-time index

k_ψ, $k_{\psi 0}$, $k_{\psi d}$, $\tilde{k}_\psi$, $\tilde{k}_{\psi 0}$ parameters used in the parametric yaw model

k_p, k_{cs}, $\tilde{k}_{cs}$, k_{aero}, k_g, k_d parameters used in the parametric roll/pitch model

$\boldsymbol{K}$ linear state feedback gain matrix

$\boldsymbol{K}_a$, $\boldsymbol{K}_s$ linear state feedback gain matrix used to yield $\boldsymbol{\nu}_a$ and $\boldsymbol{\nu}_s$, respectively

$\boldsymbol{L}_b^i(\boldsymbol{\theta})$ matrix that transforms body angular rates $\boldsymbol{\omega}$ into Euler angle rates $\dot{\boldsymbol{\theta}}$

L, M, N moments about the x_b, y_b, and z_b axes

$\mathbf{m}_b$ magnetometer readings, or magnetic flux density, in the body frame

$\mathbf{m}_{level}$ magnetometer readings, or magnetic flux density, in an inertially level frame

p, q, r body angular rates about the x_b, y_b, and z_b axes

p_{sim}, q_{sim}, r_{sim} simulated body angular rates

$\boldsymbol{\mathcal{R}}_{\alpha}^{\beta}$ rotation matrix transforming from some frame α to some frame β

Δt timestep between samples or iterations

t time

$^{\mathsf{T}}$ transpose

$\boldsymbol{T}_u$ matrix that transforms the servo-centric effects $\boldsymbol{g}(\boldsymbol{u})$ or $\boldsymbol{\nu}_s$ to the aircraft-centric effects $\boldsymbol{g}_a(\boldsymbol{u})$ or $\boldsymbol{\nu}_a$

$\boldsymbol{T}_{inv}$ pseudo-inverse of $\boldsymbol{T}_u$

u, v, w body linear velocities along the x_b, y_b, and z_b axes

$\boldsymbol{u} = \begin{bmatrix} u_{s1} & u_{s2} & u_{s3} & u_{s4} \end{bmatrix}^{\mathsf{T}}$ servo angle input vector

$\boldsymbol{v}_{f,b} = \begin{bmatrix} v_{f,xb} & v_{f,yb} \end{bmatrix}^{\mathsf{T}}$ body velocities projected onto the inertial x-y plane, as reported by the optical flow sensor

$\boldsymbol{v}_{f,i} = \begin{bmatrix} v_{f,xi} & v_{f,yi} \end{bmatrix}^{\mathsf{T}}$ the horizontal velocity $\boldsymbol{v}_{f,b}$ rotated into the inertial frame

$\boldsymbol{v}_{c,b} = \begin{bmatrix} v_{c,xb} & v_{c,yb} \end{bmatrix}^{\mathsf{T}}$ the horizontal velocity complementary filter estimate $\boldsymbol{v}_{c,i}$ as a projection from the body frame onto the inertial x-y plane

$\boldsymbol{v}_{c,i} = \begin{bmatrix} v_{c,xi} & v_{c,yi} \end{bmatrix}^{\mathsf{T}}$ horizontal velocities in the inertial frame, as estimated by the velocity complementary filter

V_{ind} induced air velocity inside the duct

x_b, y_b, z_b body axes in the forward, right, and down directions of the vehicle

$\boldsymbol{x} = \begin{bmatrix} \phi & \theta & \psi & p & q & r \end{bmatrix}^\mathsf{T}$ state vector for the linearized attitude model

$\boldsymbol{x}_0$ initial value of the state vector at time $t = 0$

$\boldsymbol{x}_{eq}$ equilibrium state vector

δ_{s1}, δ_{s2}, δ_{s3}, δ_{s4} servo input biasing terms used in $\boldsymbol{g}(\boldsymbol{u})$

$\boldsymbol{\theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\mathsf{T}$ Euler angles describing the attitude

$\boldsymbol{\theta}_{sim} = \begin{bmatrix} \phi_{sim} & \theta_{sim} & \psi_{sim} \end{bmatrix}^\mathsf{T}$ simulated attitude Euler angles

$\boldsymbol{\nu}_s = \begin{bmatrix} \nu_{s1} & \nu_{s2} & \nu_{s3} & \nu_{s4} \end{bmatrix}^\mathsf{T}$ the effective inputs for each flap, equal to $\tilde{\boldsymbol{g}}(\boldsymbol{u})$

$\boldsymbol{\nu}_a = \begin{bmatrix} \nu_\phi & \nu_\theta & \nu_\psi \end{bmatrix}^\mathsf{T}$ the effective inputs on the aircraft roll, pitch, and yaw

$\boldsymbol{\tau} = \begin{bmatrix} L & M & N \end{bmatrix}^\mathsf{T}$ vector of moments on the vehicle

$\boldsymbol{\tau}_{rotor}$, $\boldsymbol{\tau}_{static}$, $\boldsymbol{\tau}_{gyro}$, $\boldsymbol{\tau}_{cs}$, $\boldsymbol{\tau}_{aero}$ components of the overall moment vector $\boldsymbol{\tau}$

$\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^\mathsf{T}$ vector of body angular rates

ω_p propeller angular speed

ω_{pe} equilibrium propeller angular speed

Chapter 1

Introduction

Ducted fan unmanned aerial vehicles (UAVs) have several unique advantages over multi-rotor or fixed wing UAVs. However, ducted fan vehicles present challenges in terms of control, as they tend to have complex nonlinear dynamics that can be difficult to identify and model. This thesis details the instrumentation, modeling, and control of an electric ducted fan (EDF) UAV in hover mode.

Unmanned aerial vehicles (UAVs) have been traditionally developed for, funded by, and used for military applications [1]. However, as the design, instrumentation, and control of UAVs become less expensive, smaller, and more reliable, UAVs are being considered for more applications as a way of replacing difficult and costly manned aircraft or terrestrial operations. Some applications require UAVs to fly in tight enclosed environments, where size is a major constraint. Many times, this requires a vehicle capable of vertical takeoff and

landing (VTOL) and a hovering mode so that the vehicle can takeoff and hold its position in size-limited spaces. Because a single-rotor vehicle such as an EDF is more efficient per unit diameter than a similarly-sized multi-rotor vehicle, single-rotor vehicles have a distinct advantage in size-limited environments [2].

1.1 Motivation

The hover capability of VTOL UAVs makes them well-suited for inspection applications while the small size and efficiency of ducted fan vehicles make them an ideal choice for indoor and enclosed missions. As such, the use of ducted fan UAVs is particularly well suited for indoor inspection missions, especially where ground vehicles are not viable and where manned inspection is difficult, expensive, or dangerous. For example, the regular inspection of the tanks and voids of a Navy ship for corrosion is a lengthy, expensive, and risky process for manned inspection [3], and it is an ideal UAV application. Another use-case for VTOL ducted fan UAVs is surveillance in buildings, debris, or urban environments after a disaster. In 2011, after the Fukushima Daiichi nuclear disaster, the Honeywell T-Hawk ducted fan UAV – a platform originally used in Afghanistan by US and British militaries for detection of improvised explosive devices – was used to explore and gather critical data of the Fukushima power plant [4]. This is a case where manned surveillance would have been too dangerous and where the space-limited conditions made the ducted fan an ideal choice over other UAVs.

1.2 Background

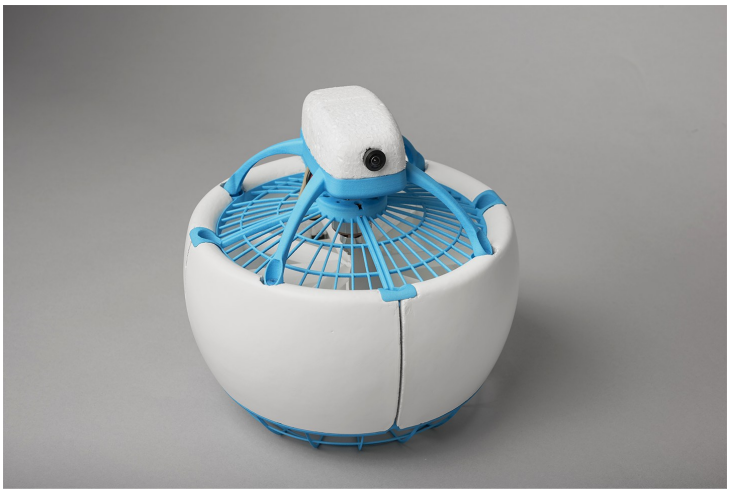
A large amount of work has been done on ducted fan UAVs in the past decade. One driver of research on VTOL ducted fan vehicles is the Defense Advanced Research Projects Agency (DARPA). Among other UAV programs, DARPA began funding the Organic Air Vehicle (OAV) program in 2001 to specifically encourage development of small unmanned ducted fan air vehicles for military use [5]. DARPA funding led to the development of several ducted fan UAVs, including the Honeywell T-Hawk Micro Air Vehicle (MAV), shown in Figure 1.1a, and the Aurora Goldeneye 80, shown in Figure 1.1b. Another example of a ducted fan UAV is the Airborne Remotely Operated Device (AROD) in Figure 1.1c developed by Sandia National Laboratories. While these examples are all powered by a gasoline engine, there are also several electric ducted fan vehicles as well. The EDF is even making its way into the hobbyist recreational market, in the case of the Fleye, shown in Figure 1.1d.

Ducted fan UAVs have been used in the military as reconnaissance, search, and scouting vehicles. Their relatively small size makes them portable and their VTOL capability allows them to be launched in small areas as well as to land in remote locations for extended surveillance of an area. The efficiency of the duct gives them a high mission endurance compared to multi-rotor vehicles, while also protecting people from the propeller and protecting the propeller from obstacles, as found in an urban setting. However, these same qualities have made ducted fan UAVs useful in other applications, such as the Fukushima disaster

(a) Honeywell T-Hawk MAV[1]

(b) Aurora Goldeneye 80[2]

(c) Sandia National Labs AROD[3]



(d) Fleye[4]

Figure 1.1: Example ducted fan vehicles

surveillance mission previously discussed [4].

[1]https://aerospace.honeywell.com/blogs/2015/november/it-is-a-bird-it-is-a-plane-it-is-a-drone

[2]http://defense-update.com/products/g/goldeneye80.htm

[3]http://www.public.navy.mil/spawar/Pacific/Robotics/Pages/AROD.aspx

[4]http://gofleye.com/

1.3 Related Work

Modeling of a ducted fan vehicle is covered in the work by Avanzini et al. [6], which uses computational fluid dynamics (CFD) analysis and wind-tunnel testing to develop a model, and then looks at the stability and the effect of uncertainty in the model. Wang [7] provides a simple model for the ducted fan dynamics and discusses some experiments for identifying the parameters in the model. A nonlinear model of the RMIT University ducted fan vehicle is developed by Zhao [8] and Zhao et al. [9] using primarily CFD analysis and wind-tunnel testing, followed by development and simulation of a linear state feedback controller. Building on this, Banazadeh and Emami [10] use simulation to examine the effectiveness of two different controller designs on the RMIT vehicle model: a linear state feedback controller and a model predictive controller. They found that the model predictive controller performed well, but that their linear controller was unable to stabilize the system. In [11], Arunesh-waran, et al., examine controller design for a ducted fan UAV and use simulation to test a nonlinear backstepping controller that uses an adaptive neural network to estimate model uncertainties. Similarly, Johnson and Turbe [12] also present the use of a neural network adaptive model to characterize the Georgia Tech GTSpy VTOL ducted fan UAV. Like [8, 9] and [10], the work of Pflimlin et al. [13] also provides details of the development of an attitude controller using linearized dynamics; it also presents a demonstration of the performance of the controller in hardware on the HoverEye ducted fan from Bertin Technologies. Franz, Milam, and Hauser [14] develop a receding horizon Lyapunov-based model predictive controller for the CalTech ducted fan and compare its performance to a gain-scheduled

linear-quadratic regulator (LQR) controller, concluding that the receding horizon controller performs better in terms of step response and disturbance rejection.

While much of the modeling research focuses on CFD and wind-tunnel data, Franz and Hauser [15] present an approach more similar to the identification method used in this thesis; Franz and Hauser attempt to determine some of the unknown parameters in the model by doing a least squares minimization search over the parameter space using flight data. This was tested using the controller developed in [14] and showed a clear improvement in performance.

In a less traditional approach to ducted fan vehicles, Xu, et al., [16] look at modeling of a novel two-duct tandem ducted fan vehicle. Other papers present novel actuation methods, such as [2] which provides an examination of the use of synthetic jets around the duct to induce moments in the vehicle.

1.4 Contribution

This thesis presents a method of modeling and characterizing ducted fan UAV hover mode dynamics using empirical data and physics principles, without the need for expensive wind-tunnel testing or CFD analysis. This is done using data from a relatively inexpensive sensor suite, which is also presented in this thesis. Finally, it is shown that a LQR-based state feedback controller based on the linearized empirical model stabilizes the attitude dynamics of the vehicle.

1.5 Organization

This thesis covers the background notation and coordinate frame definitions in Chapter 2. Chapter 3 discusses the hardware of the vehicle that was used for this research. Chapter 4 describes the sensor conditioning and state estimation used on the vehicle. Modeling and control are covered in Chapters 5 and 6, respectively. The thesis is concluded in Chapter 7.

Chapter 2

Notation

This chapter details the notation and coordinate frame definitions that will be used for the remainder of the thesis. Standard notation is used throughout. In general, scalar quantities are represented using italicized lowercase letters; vectors are represented by bold lowercase letters; and matrices are represented by bold uppercase letters.

2.1 Coordinate Frames

The inertial or Earth frame used in this thesis is a North-East-Down frame. Variables, vectors, and axes in the inertial frame are denoted with the subscript i. The angular displacement from the inertial frame to the body frame is described using the Euler angles: roll (ϕ), pitch (θ), and yaw (ψ). Specifically, this thesis defines this rotation using 3-2-1 Euler angles. These angles define a yaw ψ about the initial inertial z-axis (z), followed by a pitch

θ about the newly rotated y-axis (y'), followed by a roll ϕ about the new x-axis (x'').

The orientation of the body frame on the vehicle is depicted in Figure 2.1. Variables, vectors, and axes in the body frame are denoted with the subscript b. The z_b axis points down, the x_b axis points forward out of the designated front of the vehicle, and the y_b axis points out the right side to complete the right-handed coordinate system. The angular rates about the x_b, y_b, and z_b axes are denoted as p, q, and r, respectively. The linear velocities in the direction of the x_b, y_b, and z_b axes are denoted as u, v, and w, respectively.

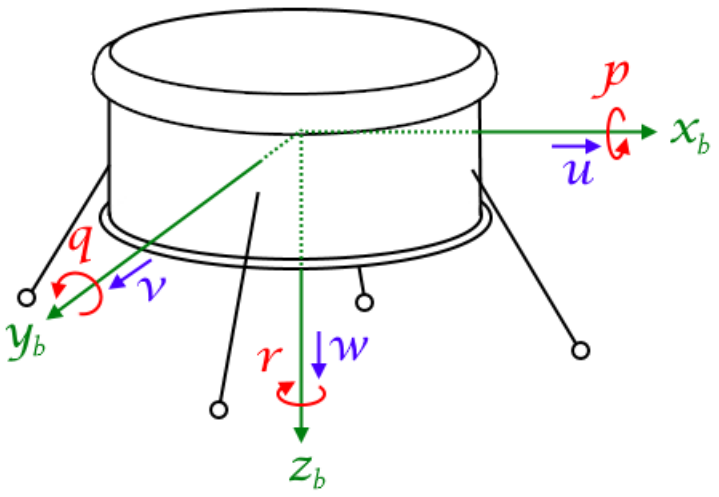


Figure 2.1: The body frame is illustrated with body axes x_b, y_b, and z_b; body angular rates p, q, and r; and body linear velocities u, v, and w.

2.2 Coordinate Transformations

To change a vector $\boldsymbol{\eta}_i$ in the inertial frame to a vector $\boldsymbol{\eta}_b$ in the body frame, a rotation matrix $\boldsymbol{\mathcal{R}}_i^b(\boldsymbol{\theta})$ may be used.

$$\boldsymbol{\eta}_b = \boldsymbol{\mathcal{R}}_i^b(\boldsymbol{\theta})\, \boldsymbol{\eta}_i \tag{2.1}$$

where $\boldsymbol{\theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\mathsf{T}$ is the collection of Euler angles. Under our definition of the Euler angles, the rotation matrix $\boldsymbol{\mathcal{R}}_i^b(\boldsymbol{\theta})$ is

$$\boldsymbol{\mathcal{R}}_i^b(\boldsymbol{\theta}) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2.2}$$

A full derivation of $\boldsymbol{\mathcal{R}}_i^b$ is found in [17].

The matrix $\boldsymbol{\mathcal{R}}_i^b$ is of the special orthogonal group of order 3, $SO(3)$, meaning it satisfies

$$\boldsymbol{\mathcal{R}}_i^b[\boldsymbol{\mathcal{R}}_i^b]^\mathsf{T} = \boldsymbol{I}$$

and

$$\det\left(\boldsymbol{\mathcal{R}}_i^b\right) = 1$$

where $[\cdot]^\mathsf{T}$ indicates a transpose operation and $\det(\cdot)$ indicates the matrix determinant. Therefore, the inverse of the rotation matrix is equal to the transpose. The transforma-

tion of $\boldsymbol{\eta}_b$ from the body frame back to $\boldsymbol{\eta}_i$ in the inertial frame can be written as

$$\boldsymbol{\eta}_i = [\boldsymbol{\mathcal{R}}_b^i]^{-1}\boldsymbol{\eta}_b = [\boldsymbol{\mathcal{R}}_i^b]^{\mathsf{T}}\boldsymbol{\eta}_b \tag{2.3}$$

It is sometimes useful to determine the Euler angles from a given rotation matrix,

$$\phi = \text{atan2}\left(\mathcal{R}_{2,3},\ \mathcal{R}_{3,3}\right) \tag{2.4a}$$

$$\theta = \sin^{-1}\left(-\mathcal{R}_{1,3}\right) \tag{2.4b}$$

$$\psi = \text{atan2}\left(\mathcal{R}_{1,2},\ \mathcal{R}_{1,1}\right) \tag{2.4c}$$

where the notation $\mathcal{R}_{m,n}$ indicates the element at row m column n of the given rotation matrix $\boldsymbol{\mathcal{R}}$, and $\text{atan2}(y, x)$ is the two-argument arctangent function.

The angular rates of the Euler angles are not equal to the body angular rates, such as what would be obtained from a gyroscope. They are instead related by a transformation matrix $\boldsymbol{L}_b^i(\boldsymbol{\theta})$,

$$\dot{\boldsymbol{\theta}} = \boldsymbol{L}_b^i(\boldsymbol{\theta})\boldsymbol{\omega} \tag{2.5}$$

where $\dot{\boldsymbol{\theta}} = \begin{bmatrix}\dot{\phi} & \dot{\theta} & \dot{\psi}\end{bmatrix}^{\mathsf{T}}$ is the vector of Euler angle rates and $\boldsymbol{\omega} = \begin{bmatrix}p & q & r\end{bmatrix}^{\mathsf{T}}$ is the vector of body angular rates. The matrix $\boldsymbol{L}_b^i(\boldsymbol{\theta})$ is not a rotation matrix, and therefore $[\boldsymbol{L}_b^i(\boldsymbol{\theta})]^{-1} \neq$

$[\boldsymbol{L}_b^i(\boldsymbol{\theta})]^\mathsf{T}$. In the defined coordinate system,

$$\boldsymbol{L}_b^i(\boldsymbol{\theta}) = \frac{1}{\cos\theta}\begin{bmatrix} \cos\theta & \sin\phi\sin\theta & \cos\phi\sin\theta \\ 0 & \cos\phi\cos\theta & -\sin\phi\cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2.6}$$

$$= \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{2.7}$$

The inverse relationship is

$$\boldsymbol{\omega} = [\boldsymbol{L}_b^i(\boldsymbol{\theta})]^{-1}\dot{\boldsymbol{\theta}} \tag{2.8}$$

where

$$[\boldsymbol{L}_b^i(\boldsymbol{\theta})]^{-1} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.9}$$

A full derivation of $\boldsymbol{L}_b^i(\boldsymbol{\theta})$ and $[\boldsymbol{L}_b^i(\boldsymbol{\theta})]^{-1}$ can be found in [17].

Chapter 3

Hardware

This chapter covers the physical hardware of the EDF vehicle that was used for this thesis. The physical chassis and actuators are discussed first, followed by the electronics, including sensors and controllers. A picture of the vehicle is shown in Figure 3.1.

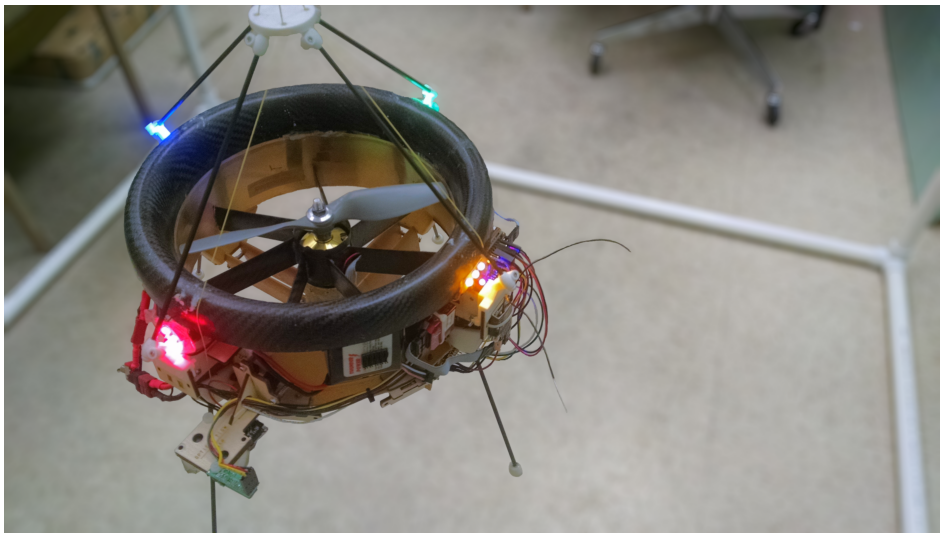


Figure 3.1: Image of the ducted fan UAV used in this thesis

3.1 Chassis

The EDF chassis was constructed by AVID LLC, based on their EDF-8 vehicle. In the work done for this thesis, none of AVID's original electronics or sensors were used, except for the motor and the batteries. All other electronics were stripped off, and only the chassis was used.

The EDF chassis is primarily composed of a single two-bladed propeller supported inside an eight inch duct. The propeller / rotor is driven by a motor contained inside a cylindrical center enclosure that is held in place in the center of the duct by seven struts extending between the central enclosure and the inner edge of the duct. These support struts also serve a double purpose as static veins to reduce swirling airflow in the duct and to counteract the yaw moment induced by the spinning propeller. A picture showing the static veins and the propeller inside the duct is shown in Figure 3.2.

Located underneath the bottom opening of the duct are four pairs of flaps, which can be actuated by the four servos located underneath the motor in the center body. These flaps act as control surfaces that can redirect some of the airflow in order to create roll, pitch, and yaw moments for controlling the vehicle. In each pair of flaps, one flap is directly attached to the servo shaft and the other companion flap is connected to the driven flap by a mechanical linkage. A close-up image of the control surfaces and their linkage is shown in Figure 3.3.

Four legs are attached to the front, back, left, and right sides of the outside of the duct. These legs support the vehicle when not in flight and also give the vehicle some initial

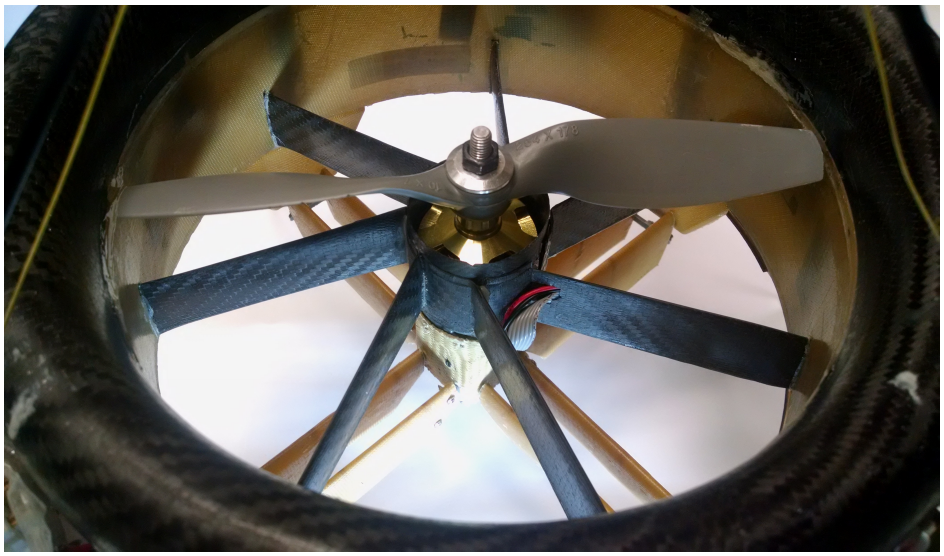


Figure 3.2: Image of the static veins and the propeller inside the duct of the vehicle

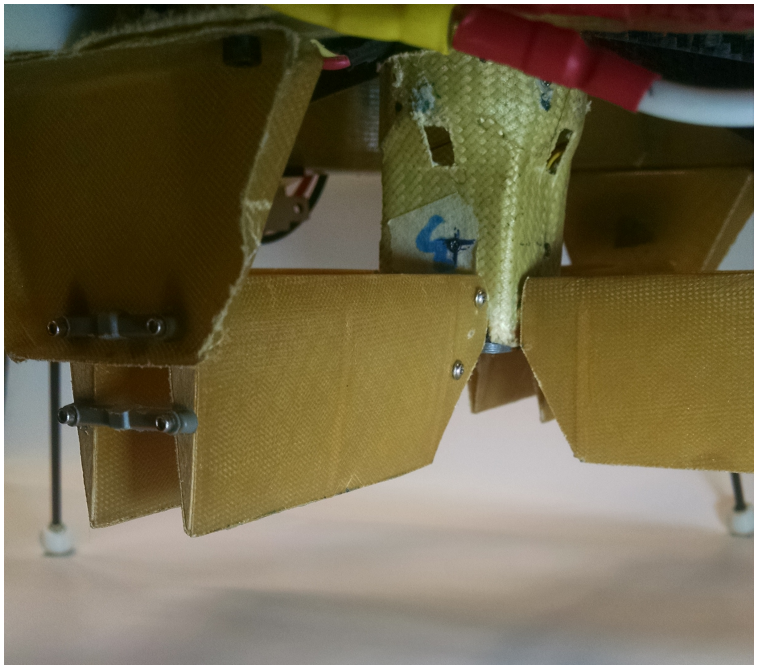


Figure 3.3: Image of the servo-driven control surfaces located at the bottom of the duct

clearance off the ground to reduce ground effects during takeoff.

Mounting surfaces were added to the outside of the duct, which were not present on the original AVID EDF-8 chassis. An additional mounting surface is supported about 4 inches above the top of the fan by carbon fiber rods and 3D-printed plastic brackets. This top mounting point is intended to support a small payload sensor, such as a camera for inspection.

3.2 Actuators

The EDF vehicle is actuated by two means: the primary motor that spins the rotor to provide thrust, and the servo motors that move the bottom control flaps to induce moments on the vehicle.

3.2.1 Primary Motor

The primary motor is housed in the central cylinder, and its shaft is connected directly to the rotor. As the motor spins, air is forced through the duct, providing thrust.

The vehicle uses a brushless direct current (BLDC) outrunner motor. BLDC outrunner motors are commonly used in electric radio-controlled (RC) aircraft. Specifically, this vehicle uses the Scorpion SII-3008-1220KV brushless outrunner motor, which is a fourteen pole motor with a maximum continuous current rating of 32A. For the three-cell batteries used

in the system, this results in a maximum continuous power rating of 360W.

3.2.2 Servo Motors

Each pair of control flaps is actuated by a servo motor. The four servo motors are housed inside the center body below the primary motor. In order to fit in the small space available in the center body, the ultra nano form factor of servos was used. Specifically, the vehicle uses four Hitec HS-5035HD digital servos. These replaced the original Hitec HS-35HD analog servos used on the original AVID EDF-8 chassis. The use of digital servos allows the center point and range of motion to be trimmed on the motor. Additionally, digital servos can provide a smaller deadband region than analog servos at the cost of slightly increased power consumption, which gives the vehicle higher control resolution. The servos are commanded with the standard servo-style pulse-width modulation (PWM) signals; a pulse width of 1000μs sets the servo position to one end of its positional range while a pulse of 2000μs sets its position to the other end of its range.

3.3 Electronics

This section covers the primary electronics on the EDF vehicle. This includes the micro-controller boards used for control of the vehicle as well as the various sensor systems.

3.3.1 Primary Control Board

The primary control board is a microcontroller board that performs the main control calculations. This is a Flip32+ board, which is based around the STM32F103 microcontroller. The STM32F103 is a 32-bit ARM Cortex-M3 microcontroller clocked at 72MHz. Additionally, the board contains a few built-in sensors, including a barometer for determining altitude. The microcontroller runs a custom version of the open source CleanFlight software, which provides the framework in which the control calculations are performed while handling all of the ancillary functions such as receiving the RC input and reading the on-board sensors.

In addition to determining control values, the primary control board also handles several other functions. It reads in the RC controller input which allows manual outer loop control from a user. It also handles driving the actuator signals, including the servo commands and the commands to the electronic speed controller (ESC) used to drive the primary motor.

3.3.2 Auxiliary Sensor Board

The auxiliary sensor board is another microcontroller board that handles the majority of the sensor reading and signal filtering. This board is the Teensy 3.1, which is based around the MK20DX256VLH7 microcontroller. The MK20DX256VLH7 is a 32-bit ARM Cortex-M4 clocked at 72MHz.

Because the primary control board has a limited number of ports, the auxiliary sensor board is needed to connect to the various sensors. The sensor board reads these sensors,

performs several filtering and transformation calculations, then sends all of the sensor data to the control board over a single universal asynchronous receiver/transmitter (UART) serial interface.

3.3.3 Attitude and Heading Reference System

The attitude and heading reference system (AHRS) is a sensor that provides a real-time estimate of the attitude and heading, which can be given as the 3-2-1 Euler angles described in Section 2.1. It also provides body angular rate readings and linear body accelerations.

The specific AHRS used on the EDF vehicle is the VectorNav VN-100 Rugged. The VN-100 is an integrated package consisting of an accelerometer, gyroscope, and magnetometer, along with some computing power. It uses the sensor readings in a Kalman filter to estimate the attitude and heading.

3.3.4 Optical Flow

The optical flow sensor uses a small camera to capture successive images of the ground. By comparing the movement of features between frames and by knowing the height of the camera above the ground, the sensor can estimate the horizontal velocity of the vehicle.

The optical flow sensor on the EDF vehicle is the PixHawk PX4Flow. This sensor includes an on-board ultrasonic distance sensor for measuring the height above the ground. The sensor uses this height in its estimate calculations, and also provides the reading to

the control board. The PX4Flow also includes a gyroscope for obtaining rough estimates of attitude, which allows the sensor to compensate for the vehicle tilt in the horizontal velocity estimate.

The optical flow sensor can be sensitive to the environment in which it is used. It can have difficulty producing accurate estimates in rough terrain. It can also give poor estimates when over a very low-contrast surface, such as a plain tile floor, due to the lack of visual features to track in the images.

The PX4Flow sensor is connected to the auxiliary sensor board through an I2C interface. The horizontal velocity estimates and ultrasonic height measurement are queried every 10ms.

3.3.5 Infrared Distance Sensor

The EDF vehicle can support several infrared distance sensors for determining the proximity to walls, obstacles, the floor, or the ceiling. The only configuration specifically used in this thesis is as an additional altitude sensor, although several experiments were done using these sensors for wall avoidance.

The specific sensor being used on the vehicle is the Sharp GP2Y0A60SZ. This device has a sensing range of 10cm to 150cm. The sensor output consists of an analog voltage, which is read by an analog-to-digital converter (ADC) on the auxiliary sensor board. The sensor board is programmed to read eight of these sensors at once, sampling every 10ms.

3.3.6 Electronic Speed Controller

The electronic speed controller (ESC) receives PWM commands from the control board and drives the high-power BLDC motor. The ESC being used is the Phoenix Edge Lite 50 from Castle Creations. In addition to driving the motor, this ESC can be configured to provide feedback of the rotational speed of the motor. It does this by sending a brief pulse to the control board each time a motor pole is electrically commutated. By measuring the frequency of pulses and dividing by the number of poles in the motor, the speed of the motor can be determined in revolutions per minute (RPM).

3.3.7 Radio Control Receiver

The vehicle uses an RC receiver to allow a user to have manual outer loop control of the vehicle via an RC transmitter. The receiver on the vehicle is the FrSky D4R-II, and the hand-held transmitter is the FrSky X9D Taranis Plus. This receiver-transmitter pair communicates over 2.4GHz. The receiver is read through a pulse position modulation (PPM) signal by the primary control board.

3.3.8 Data Logger

Although not involved in the control of the EDF vehicle, the data logger is crucial to the analysis and modeling of the vehicle. The data logger used is the Sparkfun OpenLog, which receives input over a UART interface and logs it to a microSD card. This connects to the

sensor board, which sends all of the sensor information to be logged every 10ms.

3.3.9 Electronics Summary

The high-level connection diagram of the sensors, controllers, and other electronics is shown in Figure 3.4. Additionally, the controller boards are summarized in Table 3.1 and the sensors are summarized in Table 3.2.

Table 3.1: Microcontrollers used on the EDF vehicle

	Board	**Microcontroller**	**Architecture**
Primary Control Board	Flip32+	STM32F103	32-bit ARM Cortex-M3 @ 72MHz
Auxiliary Sensor Board	Teensy 3.1	MK20DX256VLH7	32-bit ARM Cortex-M4 @ 72MHz

Table 3.2: Sensors used on the EDF vehicle

Sensor	**Model**	**Interface**	**Measurements / Estimates**
AHRS	VectorNav VN-100 Rugged	UART	Attitude (ϕ, θ, ψ) Body angular rates (p, q, r) Body linear acceleration ($\dot{u}$, $\dot{v}$, $\dot{w}$) Body magnetic flux density (m_{bx}, m_{by}, m_{bz})
Optical Flow	PixHawk PX4Flow	I2C	Body horizontal velocity ($v_{f,xb}$, $v_{f,yb}$) Sonar altitude (h_{sonar})
IR Distance Sensors	GP2Y0A60SZ	Analog	IR altitude (h_{IR})
ESC Feedback	Phoenix Edge Lite 50	Pulse Frequency	Propeller angular rate (ω_{prop})

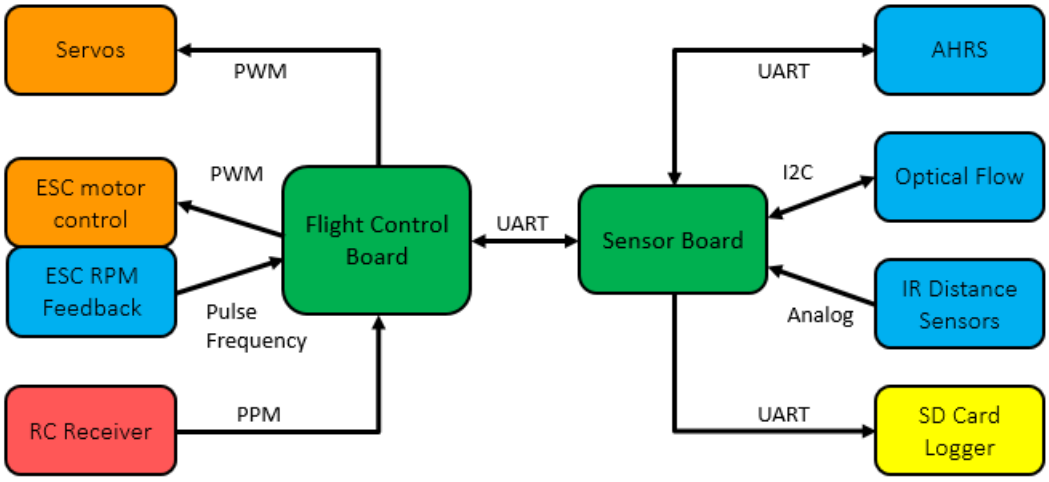


Figure 3.4: This shows how the various electronics on the EDF vehicle are connected. Each connection is labeled with the type of interface used, and the connections are drawn with arrows indicating the direction of communication.

Chapter 4

Sensor Conditioning and State Estimation

This chapter details the steps taken to improve the performance of the sensors along with the processing that was done to obtain accurate state estimates for the vehicle. Many of the sensors initially gave very inaccurate and noisy readings, due to a combination of vehicle vibration, sensor configuration, and inherent limitations in the sensors themselves. Special care had to be taken in the mounting and configuration of the sensors as well as in the filtering and processing of the sensor data.

4.1 AHRS Vibration

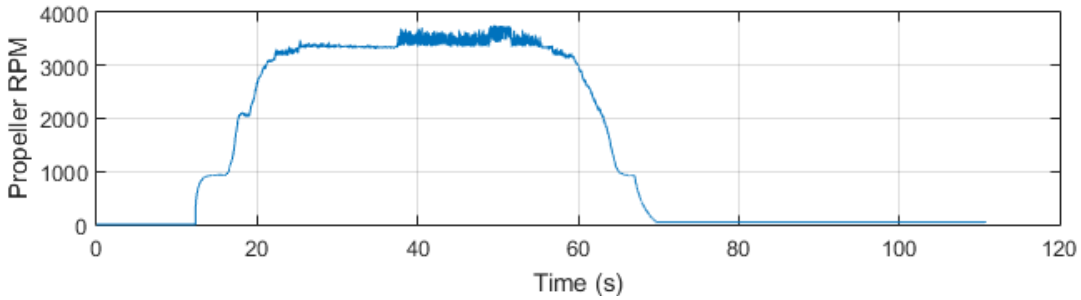
Perhaps the single biggest issue with the initial sensor suite of the EDF vehicle was the high amount of vibrational noise experienced by the AHRS. Initially, the sensor was rigidly mounted to the outer frame of the vehicle; this allowed the full vibrational force coming from the motor and propeller to be transferred to the sensor.

Figure 4.1 demonstrates some of the issues seen when using the AHRS in its default configuration in the presence of vibration induced by the prop motor. The test shown in Figure 4.1 was performed with the vehicle held to the ground to prevent it from moving or rotating. Under this setup, the roll, pitch, and yaw angles should remain at zero degrees. Additionally, the angular rates (p, q, and r) and linear accelerations ($\dot{u}$, $\dot{v}$, and $\dot{w}$) should be zero. However, as the angular speed of the propeller increases, the amplitude of high frequency noise that appears in the angular rate and linear acceleration measurements increases. The amplitude of the noise is up to 60 degrees per second in the angular rate measurements and 1g in the acceleration measurements. Figure 4.1 also shows that the Euler angle estimates generated by the AHRS steadily drift away from their true values of zero degrees while the motor is on, and only slowly drift back after the motor is turned off. This test shows a drift on the roll estimate of nearly 10 degrees in 20 seconds, and even after the motor is turned off, it takes significant time to recenter back to zero. Additionally, the yaw estimate also drifts away from zero, but unlike the roll and pitch errors, the yaw estimate does not correct itself after the vibration is removed. Altogether, the sensor errors seen here would be a significant

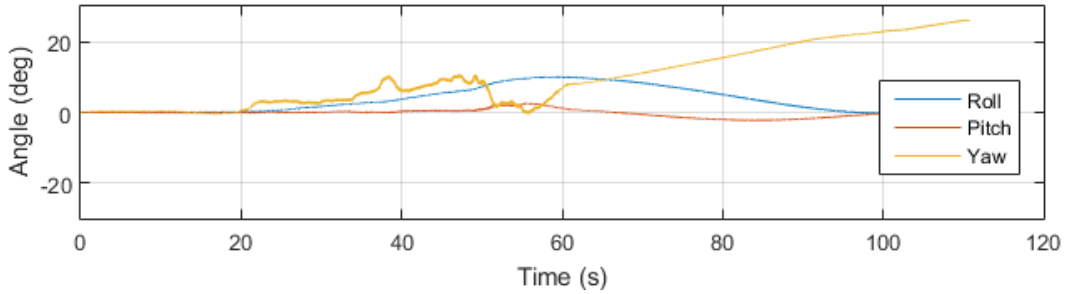
problem for any attitude control system.

The noise seen in the angular rate and linear acceleration measurements in Figure 4.1 is directly caused by the vibration of the sensor. To look more closely at the spectrum of the noise, a test was run in which the propeller speed was held at approximately 4000 RPM using joystick throttle control. The power spectral density of the accelerometer data is then estimated using a Welch periodogram, and the result is shown in Figure 4.2. Research on propeller-induced vibration has shown that for a well-balanced two-bladed propeller, the majority of the vibration spectral power occurs at the second harmonic of the propeller rate [18]. For the experiment done in Figure 4.2, a speed of 4000 RPM is equivalent to 66.7 cycles per second. The second harmonic is then at 133.3Hz. The AHRS samples at 200Hz, which would cause the 133.3Hz power to alias to 66.7Hz. The AHRS performs some low-pass filtering and reports the data at a decimated rate of 100Hz, causing further aliasing of the frequency of interest to 33.3Hz. Inspection of Figure 4.2 confirms that the observed noise power is indeed at approximately 33.3Hz.

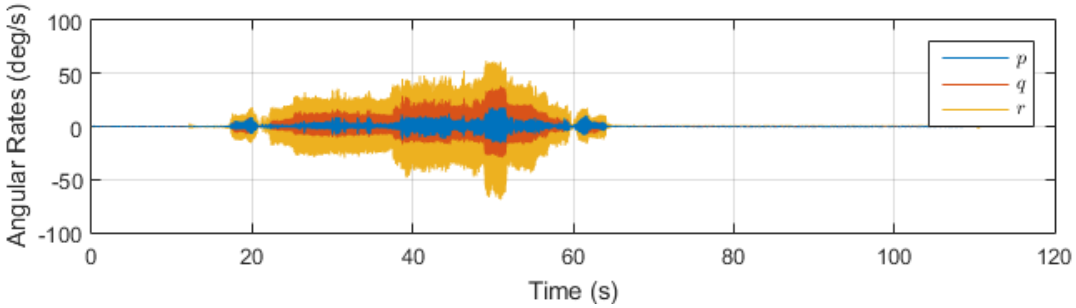
To explain the significant drift of the Euler angles, it is necessary to explain some of the inner workings of the sensor. As mentioned in Section 3.3.3, the AHRS uses an EKF to estimate the attitude. It uses the gyroscope's angular rate readings to perform the prediction step of the EKF. It then uses the accelerometer readings to infer the direction of gravity and the magnetometer data to determine a heading, which are used in the update step of the EKF. During these steps, a certain noise level or uncertainty is assumed for each reading and is used in the EKF calculations. The VN-100 sensor employs a proprietary algorithm,



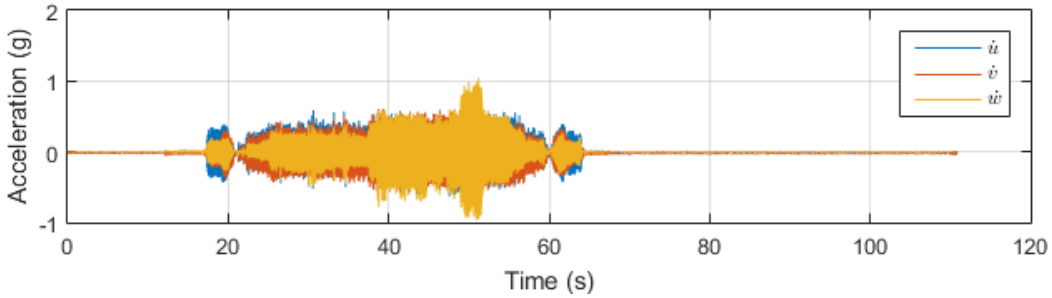
(a) Propeller speed measurement



(b) Attitude Euler angle estimates



(c) Body angular rate measurements



(d) Acceleration measurements

Figure 4.1: Test showing measurements for the propeller speed (a), attitude Euler angles (b), body angular rates (c), and acceleration (d) while the vehicle is held in a stationary position on the ground

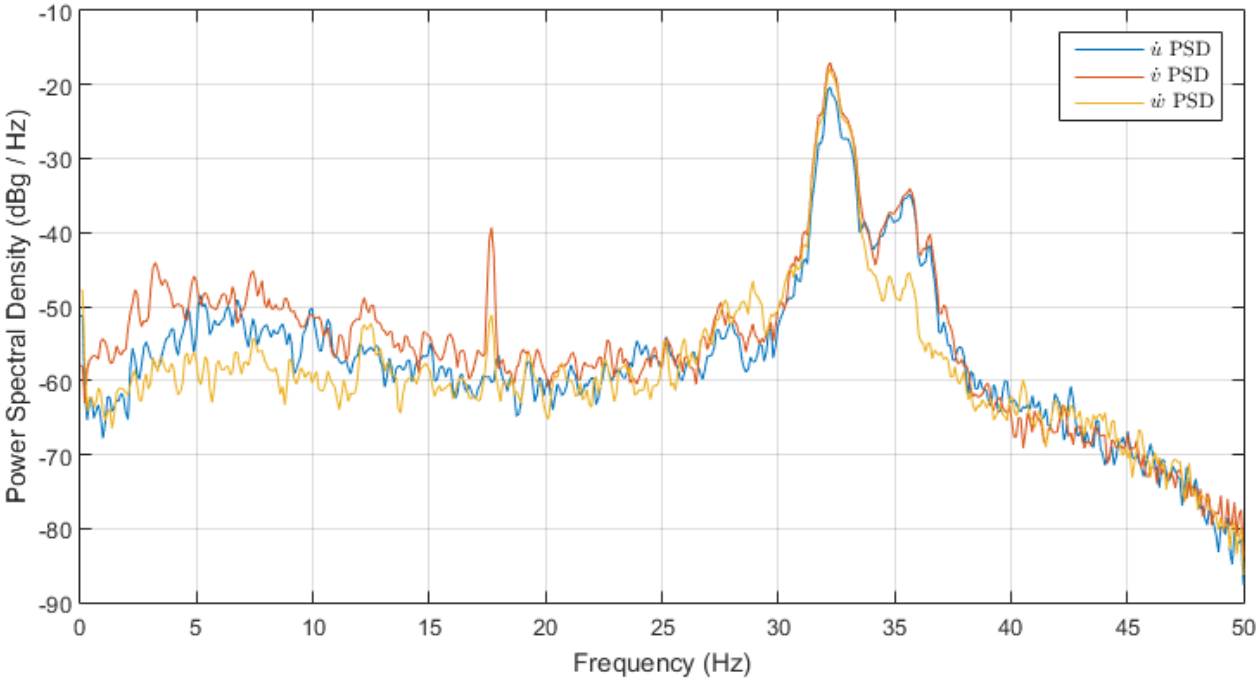


Figure 4.2: Power spectral density estimate of the vibrations in the acceleration readings while motor speed is held at 4000 RPM

.

which VectorNav refers to as the vector processing engine (VPE) that will adaptively filter the sensor measurements and tune the measurement uncertainties in the EKF [19]. The VPE tuning performs very poorly in the vibrational profile experienced on the EDF vehicle, which significantly contributed to the high drift on the attitude seen in Figure 4.1.

4.1.1 AHRS Settings Configuration

The first step taken towards addressing the issue of vibrational issues in the AHRS is changing the sensor's internal settings. The VN-100 allows you to disable the uncertainty tuning in the VPE; disabling the VPE showed immediate improvement in the attitude esti-

mate.

Using additional settings, changes were made to the lengths of internal moving average filters that are applied to the angular rate and linear acceleration readings. Increasing the filter size lowers the filter bandwidth, which in turn reduces the high frequency noise that was being seen on the angular rate and linear acceleration readings being reported by the AHRS. This comes at the cost of a slight increase in the group delay of the signals. The window size was increased from a default of 4 samples (at an internal sampling rate of 200Hz) to 16 samples, which is an increase in delay from 10ms to 40ms.

4.1.2 AHRS Vibration Isolation Mounting

To reduce the amount of vibration being transferred from the chassis into the sensor, passive vibration isolating mounting is used. Applying a small amount of closed-cell neoprene foam between the AHRS and the chassis dampens the mechanical vibration. Additionally, a piece of a Moongel drum damper pad is loosely sandwiched between the sensor and the chassis. This is a material that is intended to be used by musicians for damping the vibration of drum heads, but has found use in hobbyist UAV builds for isolation mounting.

To compare the effectiveness of the different mounting options – no isolation, foam isolation, or foam and damper gel isolation – another set of tests was performed by holding the vehicle on the ground while maintaining a propeller speed of 4000 RPM. The root mean square (RMS) acceleration of each body axis and overall magnitude are shown in Table 4.1.

Table 4.1: RMS accelerations using different sensor mounting schemes

	x-axis, $\dot{u}_{RMS}$ (g)	y-axis, $\dot{v}_{RMS}$ (g)	z-axis, $\dot{w}_{RMS}$ (g)	magnitude, $[\|\|\dot{\boldsymbol{v}}\|\|_2]_{RMS}$ (g)
No isolation	0.0813	0.1173	0.1047	0.1770
Foam only	0.0556	0.1155	0.1203	0.1758
Foam and damper gel	0.0492	0.0460	0.1227	0.1400

4.1.3 Other AHRS Conditioning

Because the sensor is not mounted in the same orientation as the body axes, all measurements from the AHRS need to be appropriately rotated. For the vector measurements such as body angular rates, linear acceleration, and magnetometer readings, this is simply done by multiplying by the rotation matrix $\boldsymbol{\mathcal{R}}_a^b$. The matrix $\boldsymbol{\mathcal{R}}_a^b$ defines the rotation from the AHRS mounting orientation to the defined body orientation, similar to the transformations described in Section 2.2. To transform the Euler angles from the AHRS frame, the angles are converted to a rotation matrix, pre-multiplied by the AHRS-to-chassis rotation, and then converted back to Euler angles. This is denoted

$$\boldsymbol{\mathcal{R}}_i^b = \boldsymbol{\mathcal{R}}_a^b \, \boldsymbol{\mathcal{R}}_i^a(\phi_a, \theta_a, \psi_a) \tag{4.1}$$

where ϕ_a, θ_a, and ψ_a are the Euler angles given by the AHRS. The matrix $\boldsymbol{\mathcal{R}}_a^b$ is the AHRS-to-body rotation, and is determined by carefully measuring the mounting angle of the AHRS. Lastly, the desired body Euler angles are found from $\boldsymbol{\mathcal{R}}_i^b$ using (2.4).

After vibration isolation mounting the AHRS and disabling the VPE, the roll and pitch estimates experience minimal drift. However, the yaw estimate still experiences high amounts of sporadic drift. To fix this issue, the yaw estimate from the AHRS is not used; instead, the magnetometer data is used directly to determine the yaw of the vehicle. To do this, the body magnetometer readings are first rotated into an inertially level frame by multiplying by a rotation matrix composed of the vehicle's roll and pitch and a yaw of zero,

$$\mathbf{m}_{level} = \boldsymbol{\mathcal{R}}\left(\phi, \theta, 0\right)\ \mathbf{m}_b \tag{4.2}$$

This level frame is related to the inertial frame by a single rotation about the z axis, and the angle of this rotation is the yaw of the vehicle. These level-frame readings are then filtered with a low-pass filter to reduce noise. The direction of the filtered $(m_{level,x}, m_{level,y})$ then points north, and the yaw is then

$$\psi = -\text{atan2}\left(m_{level,y},\ m_{level,x}\right) \tag{4.3}$$

4.2 Optical Flow Conditioning

Like the AHRS, the optical flow sensor also suffered from vibration issues, as well as just poor sensor performance. The biggest problem was with the original sensor firmware. In the original firmware, the sensor would reject most of its velocity estimates when operating too low to the ground or when the ground surface was very low contrast. When an estimate was

rejected, the PX4Flow reported zero velocity.

To correct some of these issues, an open-source custom firmware was obtained from [20] and flashed onto the optical flow board. This firmware performs much better than the original for the low altitude and low contrast conditions in which the EDF vehicle was tested. Additionally, this firmware never rejects estimates, meaning that a velocity estimate is always being reported by the optical flow sensor, even if the confidence in the estimate is low.

To address the vibration issues, the optical flow was mounted using techniques similar to those used in the AHRS mounting in Section 4.1.2. Instead of mounting directly to the side of the chassis, the optical flow was mounted to an angle bracket attached to the side of the chassis, which presented two mounting points to apply vibration isolation. Again, closed-cell foam and drum damper pads were employed to reduce vibration transfer.

4.3 Velocity Estimate Complementary Filter

While the adjustments described in Section 4.2 significantly improved the optical flow sensor performance, the velocity estimates produced by the sensor were still somewhat noisy and prone to error. To address this issue, a complementary filter is employed to fuse the optical flow velocity estimates with the accelerometer data provided from the AHRS to produce a smoother, more accurate estimate than the optical flow data or integrated accelerometer data could provide alone.

The acceleration data is obtained in the body frame and has the gravity component already removed by the AHRS, leaving only the true body acceleration $\boldsymbol{a}_b = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} \end{bmatrix}^T$. The current attitude estimate $\boldsymbol{\theta}$ is used to convert this measurement to the inertial acceleration,

$$\boldsymbol{a}_i = \boldsymbol{\mathcal{R}}_b^i(\boldsymbol{\theta})\, \boldsymbol{a}_b \tag{4.4}$$

where $\boldsymbol{a}_i$ is the true acceleration in the inertial frame.

The optical flow sensor provides a velocity estimate which accounts for the current roll and pitch. The velocities that it reports are the estimated body u and body v velocities projected onto the inertial x-y plane. In other words, it accounts for the roll angle ϕ and pitch angle θ, but not the yaw angle ψ. The transformation

$$\begin{bmatrix} v_{f,xi} \\ v_{f,yi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} v_{f,xb} \\ v_{f,yb} \end{bmatrix} \tag{4.5}$$

where ψ is the current yaw estimate, maps the body frame velocity estimates $\boldsymbol{v}_{f,b} = \begin{bmatrix} v_{f,xb} & v_{f,yb} \end{bmatrix}^\mathsf{T}$ from the optical flow sensor to inertial frame velocity estimates $\boldsymbol{v}_{f,i} = \begin{bmatrix} v_{f,xi} & v_{f,yi} \end{bmatrix}^\mathsf{T}$.

A complementary filter is used to integrate and combine the inertial frame acceleration data with the inertial frame optical flow velocity estimates. This filter is composed of two steps. The first step takes the previous filtered velocity estimate $\boldsymbol{v}_{c,i}[k-1|k-1]$ at timestep $k-1$ and adds the predicted change in velocity found by multiplying the current acceleration

$\boldsymbol{a}_i[k]$ with the size of the timestep Δt, denoted

$$\boldsymbol{v}_{c,i}[k|k-1] = \boldsymbol{v}_{c,i}[k-1|k-1] + \Delta t\ \boldsymbol{a}_i[k] \tag{4.6}$$

The optical flow sensor's estimate of the velocity is then combined with the prediction using a weighting gain ξ_v between 0 and 1, which can be chosen based on the relative confidence in the optical flow sensor versus the accelerometer data. In practice, this was chosen by trial-and-error to achieve an estimate that most closely matches the observed true velocity. The estimate at time k is then calculated as

$$\boldsymbol{v}_{c,i}[k|k] = \xi_v\ \boldsymbol{v}_{c,i}[k|k-1] + (1-\xi_v)\ \boldsymbol{v}_{f,i}[k] \tag{4.7}$$

$$= \boldsymbol{v}_{f,i}[k] + \xi_v\ (\boldsymbol{v}_{c,i}[k|k-1] - \boldsymbol{v}_{f,i}[k]) \tag{4.8}$$

The current velocity estimate in the inertial x and y directions is then $\boldsymbol{v}_{c,i}[k|k]$. It is of use to transform this estimate back into a body component projected onto the inertial $x-y$ plane (similar to how the optical flow velocity estimates are reported from the sensor). This can be done using

$$\begin{bmatrix} v_{c,xb} \\ v_{c,yb} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} v_{c,xi} \\ v_{c,yi} \end{bmatrix} \tag{4.9}$$

This complementary filtering process is performed entirely on the auxiliary sensor board before sending the data to the primary control board. Figure 4.3 shows an example result of

the velocity complementary filtering. It shows the plots of the original optical flow reading, the naive integration of accelerometer data, and the complementary filter fusion of the two sensors.

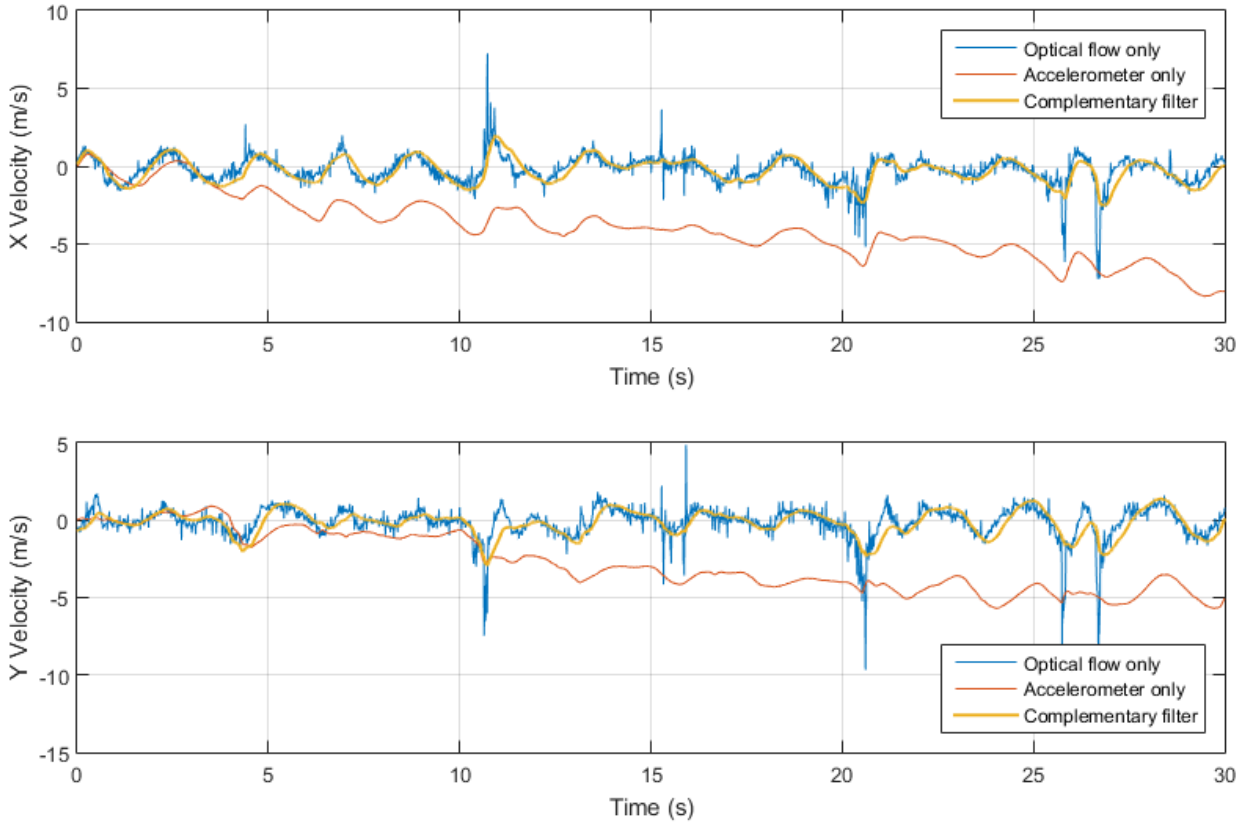


Figure 4.3: Two plots (for x and y directions) showing the raw optical flow velocities (blue), the naive integration of accelerations (red), and the complementary filter fusion of the two (yellow)

.

4.4 Altitude Estimation

On the EDF vehicle used in this research, altitude information can be obtained from four sensors:

- the sonar sensor on the optical flow board, h_{sonar}
- the downward-facing IR distance sensor, h_{IR}
- the barometer, h_{baro}, located on the primary control board
- the acceleration measurements from the AHRS, $a_{zi} = \ddot{h}_{acc}$

The IR sensor has an operating range between 10cm and 150cm above the ground. The sonar sensor's operating range is between 30cm and 500cm, although it performs best between 100cm and 300 cm. The sonar sensors also tend to be very noisy and have large erroneous spikes in the measurement. The barometric altitude sensor estimates the altitude by reading the local barometric pressure; it is not restricted to a certain range above the ground, but it has a much lower resolution and accuracy than the other two altitude sensors.

To select an altitude measurement h_{meas} from the three altitude measurement sensors, thresholding is done based on the current estimated altitude, h_{est}. Additionally, because the sonar measurement is prone to large spikes, h_{sonar} is either accepted or rejected by looking at the difference between the sensor reading and the estimated altitude. If the difference is more than 50cm, the sonar measurement is rejected. The full piecewise function for determining the altitude measurement is

$$h_{meas} = \begin{cases} h_{IR} & \text{if } h_{est} < 100\text{cm} \text{ and } h_{sonar} \textit{ accepted} \\ h_{IR} & \text{if } h_{est} < 150\text{cm} \text{ and } h_{sonar} \textit{ rejected} \\ h_{sonar} + (h_{IR} - h_{sonar}) \cdot \frac{150\text{cm} - h_{est}}{50\text{cm}} & \text{if } 100\text{cm} \leq h_{est} < 150\text{cm} \text{ and } h_{sonar} \textit{ accepted} \\ h_{sonar} & \text{if } 150\text{cm} \leq h_{est} < 200\text{cm} \text{ and } h_{sonar} \textit{ accepted} \\ h_{baro} + (h_{sonar} - h_{baro}) \cdot \frac{300\text{cm} - h_{est}}{100\text{cm}} & \text{if } 200\text{cm} \leq h_{est} < 300\text{cm} \text{ and } h_{sonar} \textit{ accepted} \\ h_{baro} & \text{if } h_{est} \geq 150\text{cm} \text{ and } h_{sonar} \textit{ rejected} \\ h_{baro} & \text{if } h_{est} \geq 300\text{cm} \end{cases} \tag{4.10}$$

The next altitude estimate is then calculated using a complementary filter, similar to the one used in Section 4.3. The prediction step of the filter is

$$h_{est}[k|k-1] = h_{est}[k-1|k-1] + \frac{1}{2}\ddot{h}_{acc}[k] \cdot \Delta t^2 + \dot{h}_{est}[k-1|k-1] \cdot \Delta t \tag{4.11}$$

where $\dot{h}_{est}[k-1|k-1]$ is the estimate of the z velocity from the previous timestep. The update equation for h_{est} is

$$h_{est}[k|k] = h_{meas}[k] + \xi_h \left(h_{est}[k|k-1] - h_{meas}[k]\right) \tag{4.12}$$

where ξ_h is some weighting gain chosen for the altitude complementary filter.

To get the z velocity estimate $\dot{h}_{est}$, another complementary filter is used. The prediction step is formulated as

$$\dot{h}_{est}[k|k-1] = \dot{h}_{est}[k-1|k-1] + \ddot{h}_{acc}[k] \cdot \Delta t \tag{4.13}$$

and the update step is

$$\dot{h}_{est}[k|k] = \frac{h_{meas}[k] - h_{meas}[k-1]}{\Delta t} + \xi_w \left(\dot{h}_{est}[k|k-1] - \frac{h_{meas}[k] - h_{meas}[k-1]}{\Delta t} \right) \tag{4.14}$$

where ξ_w is some weighting gain chosen for the altitudinal velocity complementary filter.

Using the combination of the two complementary filters for h_{est} and $\dot{h}_{est}$ gives a good estimate for not only the altitude but also the altitude velocity, which is useful for outer-loop control of the altitude.

The altitude complementary filter is performed on the primary control board. Figure 4.4 shows an example result of the altitude complementary filtering. It shows the raw sonar and IR sensor distance readings along with the overall complementary filtered altitude estimate which combines these readings with acceleration and barometer data. In this case, the test was performed at a relatively low altitude from the ground, and so the barometer data was never actually used. Naively double-integrating the accelerometer data quickly diverges, and so an altitude estimate derived purely from the acceleration is shown on the altitude plot. Figure 4.4 also shows the altitude velocity estimate that is obtained as part of the two-stage

complementary filtering for the altitude.

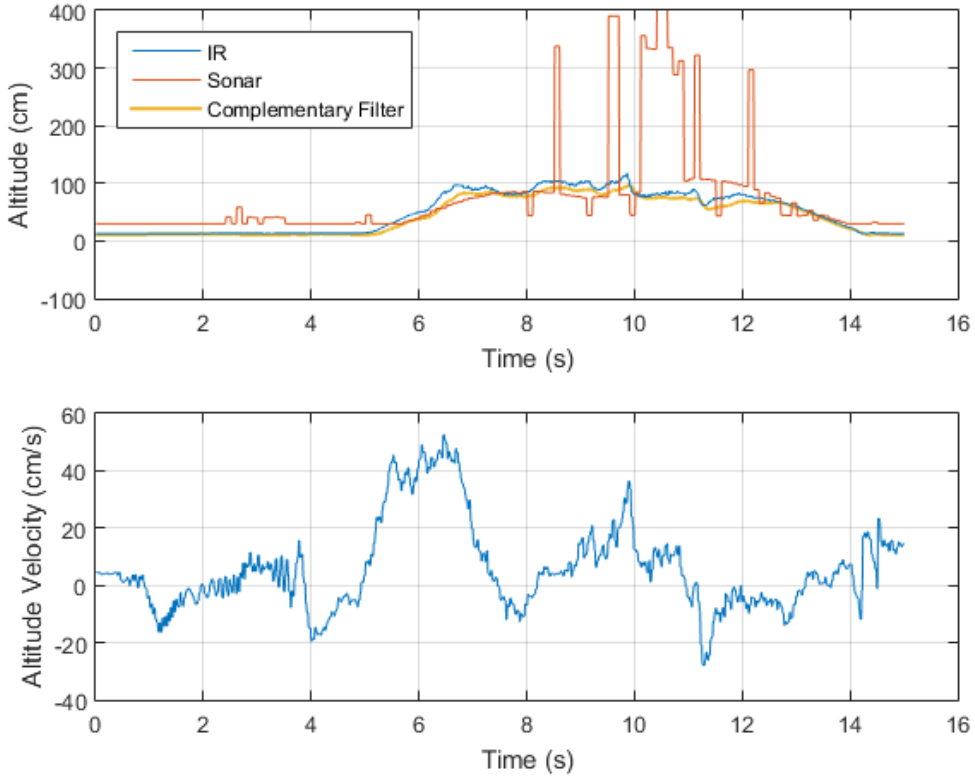


Figure 4.4: The top plot shows the raw IR sensor distance (blue), the raw sonar distance (red), and the complementary filter estimate for the altitude (yellow). The bottom plot shows the altitude velocity estimate from the complementary filter.

.

Chapter 5

Modeling

This chapter covers the details of the dynamic model of the ducted fan UAV in hover mode. This chapter briefly covers the general non-linear model, then simplifies the model into a parametric form. The parameters are then identified using system identification techniques.

5.1 The Aerodynamic Model

The starting model used in this thesis is based primarily on the work of [7], [8], and [10], and basic physics principles. Also, this thesis focuses first and foremost on the attitude dynamics, so the force and linear displacement equations will be ignored.

The model begins by reiterating (2.5), which relates the rates of the Euler angles to the body angular rates.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{L}_b^i(\boldsymbol{\theta}) \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$
$$= \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5.1}$$

The angular accelerations are related to the moments on the vehicle by

$$\begin{cases} I_x\dot{p} = L - qr(I_z - I_y) \\ I_y\dot{q} = M - rp(I_x - I_z) \\ I_z\dot{r} = N - pq(I_y - I_x) \end{cases} \tag{5.2}$$

where L, M, and N are the moments or torques about the x, y, and z axes, respectively, and I_x, I_y, and I_z are the moments of inertia. The moments can be broken into their individual components as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{rotor} + \boldsymbol{\tau}_{static} + \boldsymbol{\tau}_{gyro} + \boldsymbol{\tau}_{cs} + \boldsymbol{\tau}_{aero} \tag{5.3}$$

where $\boldsymbol{\tau} = \begin{bmatrix} L & M & N \end{bmatrix}^\mathsf{T}$ is the vector of moments, $\boldsymbol{\tau}_{rotor}$ is the moment generated by the spinning rotor, $\boldsymbol{\tau}_{static}$ is the moment from the static veins in the duct, $\boldsymbol{\tau}_{gyro}$ is the moment resulting from the gyroscopic reaction of pitching or rolling the spinning propeller, $\boldsymbol{\tau}_{cs}$ is the

moment induced by the control surfaces, and $\boldsymbol{\tau}_{aero}$ the moment resulting from lift and drag effects on the duct.

As the rotor spins on the z-axis to generate thrust, it also produces a torque $\boldsymbol{\tau}_{rotor}$ about the z-axis. This can be modeled as

$$\boldsymbol{\tau}_{rotor} = \begin{bmatrix} 0 & 0 & C_r\omega_p^2 \end{bmatrix}^\mathsf{T} \tag{5.4}$$

where ω_p is the angular speed of the propeller and C_r is a simplifying constant that depends on several factors including the rotor radius, the density of air, and the aerodynamic properties of the rotor itself. More detail can be found in [6] and [7].

The seven static veins or stators in the duct provide a moment counter to $\boldsymbol{\tau}_{rotor}$. This static vein moment $\boldsymbol{\tau}_{static}$ can be written as

$$\boldsymbol{\tau}_{static} = \begin{bmatrix} 0 & 0 & -7V_{ind}^2C_s \end{bmatrix}^\mathsf{T} \tag{5.5}$$

where V_{ind} is the induced air velocity inside the duct and C_s is a simplifying constant that combines the air density, effective lift area of the stators, the angles of the stators, the distance from the center of gravity on the x-y plane to the aerodynamic centers of the stators, and the aerodynamic properties of the stator [7].

The $\boldsymbol{\tau}_{gyro}$ term is the gyroscopic reaction of the propeller, which is rotating at a high

speed about the z-axis, undergoing a roll or pitch. This is modeled as

$$\boldsymbol{\tau}_{gyro} = \begin{bmatrix} -I_p\omega_p q & I_p\omega_p p & 0 \end{bmatrix}^\mathsf{T} \tag{5.6}$$

where I_p is the moment of inertia of the propeller.

The moments induced by actuation of the control surfaces, $\boldsymbol{\tau}_{cs}$, is very similar to the moment from the stators. However, $\boldsymbol{\tau}_{cs}$ depends on the input angles of the servo-controlled flaps, $\boldsymbol{u} = \begin{bmatrix} u_{s1} & u_{s2} & u_{s3} & u_{s4} \end{bmatrix}^\mathsf{T}$, where u_{s1}, u_{s2}, u_{s3}, and u_{s4} are the angles of the flaps on $+x$, $+y$, $-x$, and $-y$ axes of the vehicle, respectively. The control surface moments can be modeled

$$\boldsymbol{\tau}_{cs} = \begin{bmatrix} V_{ind}^2 C_{cs} \left(g(u_{s1}) + g(-u_{s3})\right) \\ V_{ind}^2 C_{cs} \left(g(u_{s2}) + g(-u_{s4})\right) \\ V_{ind}^2 C_{cs\psi} \left(g(u_{s1}) + g(u_{s2}) + g(u_{s3}) + g(u_{s4})\right) \end{bmatrix} \tag{5.7}$$

where C_{cs} is a constant very similar to C_s and $g(\cdot)$ is a function that relates the angle of the flap to the effective force/moment it generates. The constant $C_{cs\psi}$ is the similar to C_{cs}, with the difference being the distances from the flap aerodynamic centers to the vehicle center of gravity along the different axes. The function $g(\cdot)$ was originally assumed to be linear within the operating region of $\pm 30°$; however, tests showed that this was not the case. This is discussed in detail in Section 5.3.

In order to simplify both the modeling and control of the vehicle, the vector of servo control input functions $\boldsymbol{g}(\boldsymbol{u}) = \begin{bmatrix} g(u_{s1}) & g(u_{s2}) & g(u_{s3}) & g(u_{s4}) \end{bmatrix}^\mathsf{T}$ is transformed to a vector

of aircraft inputs denoted as $\boldsymbol{g}_a(\boldsymbol{u})$ to yield the average effect of the flaps for rolling, pitching, and yawing as

$$\boldsymbol{g}_a(\boldsymbol{u}) = \boldsymbol{T}_u \boldsymbol{g}(\boldsymbol{u}) \tag{5.8a}$$

$$\begin{bmatrix} g_\phi(\boldsymbol{u}) \\ g_\theta(\boldsymbol{u}) \\ g_\psi(\boldsymbol{u}) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & -0.5 & 0 \\ 0 & 0.5 & 0 & -0.5 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \begin{bmatrix} g(u_{s1}) \\ g(u_{s2}) \\ g(u_{s3}) \\ g(u_{s4}) \end{bmatrix} \tag{5.8b}$$

where $\boldsymbol{T}_u$ is the transformation matrix; $g_\phi(\boldsymbol{u})$ is the average roll effect of the flaps; $g_\theta(\boldsymbol{u})$ is the average pitch effect of the flaps; and $g_\psi(\boldsymbol{u})$ is the average yaw effect of the flaps. The vector collection of $g_\phi(\boldsymbol{u})$, $g_\theta(\boldsymbol{u})$, and $g_\psi(\boldsymbol{u})$ is $\boldsymbol{g}_a(\boldsymbol{u})$. Using these transformed inputs, (5.7) can be rewritten

$$\boldsymbol{\tau}_{cs} = \begin{bmatrix} 2V_{ind}^2 C_{cs} g_\phi(\boldsymbol{u}) \\ 2V_{ind}^2 C_{cs} g_\theta(\boldsymbol{u}) \\ 4V_{ind}^2 C_{cs\psi} g_\psi(\boldsymbol{u}) \end{bmatrix} \tag{5.9}$$

It is also useful to note the pseudo-inverse of $\boldsymbol{T}_u$, which is denoted as $\boldsymbol{T}_{inv}$. The pseudo-inverse is

$$\boldsymbol{T}_{inv} = \left(\boldsymbol{T}_u^\mathsf{T} \boldsymbol{T}_u\right)^{-1} \boldsymbol{T}_u^\mathsf{T} \tag{5.10}$$

which yields

$$\boldsymbol{T}_{inv} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \tag{5.11}$$

The pseudo-inverse has the useful properties

$$\boldsymbol{T}_u \boldsymbol{T}_{inv} \boldsymbol{T}_u = \boldsymbol{T}_u \tag{5.12}$$

$$\boldsymbol{T}_{inv} \boldsymbol{T}_u \boldsymbol{T}_{inv} = \boldsymbol{T}_{inv} \tag{5.13}$$

The last moment component being considered is the aerodynamic effect of the lip. As the vehicle moves forward, higher velocities correspond to higher airflow over the leading lip of the duct which creates a nose-up pitch. This is modeled in [6] by looking at the pitching moment coefficient versus the angle of attack and the ratio of velocity to relative induced air velocity in the duct. However, [6] also shows that the resulting nose-up moment induced by this effect is approximately linearly related to vehicle velocity at level hover-mode flight. As such, the aerodynamic moment $\boldsymbol{\tau}_{aero}$ is modeled as

$$\boldsymbol{\tau}_{aero} = \begin{bmatrix} C_{aero} u & C_{aero} v & 0 \end{bmatrix}^{\mathsf{T}} \tag{5.14}$$

where C_{aero} is the proportionality constant relating the body x-y velocity and the aerodynamic moment induced by the duct.

5.2 Simplified Parametric Model

The attitude model is next further simplified by reducing the equations to a minimal number of terms with a single parameterizing constant that accounts for all of the coefficients associated with that term. Additionally, the yaw dynamics are considered separately from the pitch and roll dynamics.

5.2.1 Yaw Parametric Model

The yaw dynamics are less complex and are therefore considered first. Because of the radial symmetry of the vehicle body, I_x and I_y are assumed to be equal, and thus the z-axis portion of (5.2) becomes

$$\dot{r} = \frac{N}{I_z} \tag{5.15}$$

Entering the non-zero moment terms in place of N yields

$$\dot{r} = \frac{1}{I_z}\left(C_r\omega_p^2 - 7V_{ind}^2 C_s + 4V_{ind}^2 C_{cs} g_\psi(\boldsymbol{u})\right) \tag{5.16}$$

However, the induced airspeed V_{ind} is related to the angular speed of the propeller ω_p by the advance ratio of the propeller and propeller diameter. So (5.16) can be parametrized

$$\dot{r} = k_\psi \omega_p^2 \left(g_\psi(\boldsymbol{u}) + k_{\psi 0}\right) \tag{5.17}$$

where k_ψ absorbs the moment of inertia and the scale of the effect of the propeller speed, and $k_{\psi 0}$ is the effect of the rotor and stators. The vehicle stators were designed to cancel out the yawing effect of the rotors, so ideally $k_{\psi 0}$ should be 0. However, any residual uncanceled yaw moment will be present in this term and can be compensated for by the control input $g_\psi(\boldsymbol{u})$. Lastly, a general damping term on r is added to help account for other unmodeled dynamics,

$$\dot{r} = k_\psi \omega_p^2 \left(g_\psi(\boldsymbol{u}) + k_{\psi 0} \right) - k_{\psi d} r \tag{5.18}$$

where $k_{\psi d}$ is the damping coefficient.

5.2.2 Roll and Pitch Parametric Model

The roll and pitch dynamics are now considered. The relevant portion of (5.2) can be divided through by the moments of inertia, yielding

$$\begin{cases} \dot{p} = \frac{L}{I_x} - qr\frac{I_z - I_y}{I_x} \\ \dot{q} = \frac{M}{I_y} + rp\frac{I_z - I_x}{I_y} \end{cases} \tag{5.19}$$

Inserting in the non-zero moment terms in place of L and M gives

$$\begin{cases} \dot{p} = \frac{1}{I_x}\left(-I_p \omega_p q + 2V_{ind}^2 C_{cs} g_\phi(\boldsymbol{u}) + C_{aero} u - qr(I_z - I_y) \right) \\ \dot{q} = \frac{1}{I_y}\left(I_p \omega_p p + 2V_{ind}^2 C_{cs} g_\theta(\boldsymbol{u}) + C_{aero} v + rp(I_z - I_x) \right) \end{cases} \tag{5.20}$$

Rewriting this parametrically with an additional term added for general angular rate damping to cover additional unmodeled effects yields the parametrized equation

$$\begin{cases} \dot{p} = -k_p\omega_p q + k_{cs}\omega_p^2 g_\phi(\boldsymbol{u}) + k_{aero}u - k_g qr - k_d p \\ \\ \dot{q} = k_p\omega_p p + k_{cs}\omega_p^2 g_\theta(\boldsymbol{u}) + k_{aero}v + k_g rp - k_d q \end{cases} \tag{5.21}$$

where moment of inertia is absorbed into all of the constants and k_p is the parameter on the propeller gyroscopic reaction term, k_{cs} is the parameter on the control surface term, k_{aero} is the parameter for the duct aerodynamic term, k_g is the parameter for the angular rate coupling, and k_d is the general angular rate damping term. This can be written more succinctly in vector form

$$\begin{bmatrix} \dot{p} \\ \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -k_d & -k_p\omega_p - k_g r \\ \\ k_p\omega_p + k_g r & -k_d \end{bmatrix} \begin{bmatrix} p \\ \\ q \end{bmatrix} + k_{cs}\omega_p^2 \begin{bmatrix} g_\phi(\boldsymbol{u}) \\ \\ g_\theta(\boldsymbol{u}) \end{bmatrix} + k_{aero} \begin{bmatrix} u \\ \\ v \end{bmatrix} \tag{5.22}$$

5.3 Input Function

The moments induced by the control surfaces were originally thought to be linearly related to the angle of the control surfaces. However, early attempts at modeling and control based on this assumption proved poor, especially when control angles were close to saturating at $\pm 30°$. After getting access to a six-axis force-torque gauge, the true relationship was measured empirically. This set of experiments was run with a constant propeller speed of

4500 RPM. Six experiments were run: a slow and fast sweeping version each for the roll inputs, $+u_{s1}$ and $-u_{s3}$; the pitch inputs, $+u_{s2}$ and $-u_{s4}$; and the yaw inputs, $+u_{s1}$, $+u_{s2}$, $+u_{s3}$, and $+u_{s4}$. In these tests, the relevant flaps were simultaneously swept back and forth between $-30°$ and $30°$ in a 10 second period for the slow test and a 1 second period for the fast test. The results are plotted in Figure 5.1. Note that the force-torque sensor could not be mounted at the center of gravity of the vehicle, and so instead of plotting the roll and pitch angles against the rolling and pitching moments around the center of gravity, they were plotted against the y-axis and x-axis forces on the sensor respectively. However, the rolling moment of the body is related to the x-axis force on the sensor by

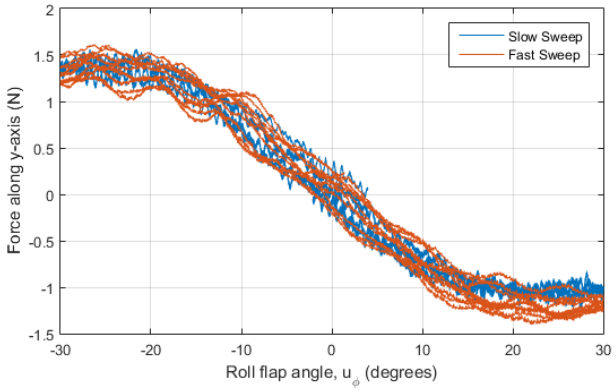
$$L_{body} = -d_f \; F_{x,sensor} \tag{5.23}$$

where L_{body} is the rolling moment of the body, $F_{x,sensor}$ is the x-axis force on the sensor, and d_f is the vertical distance between the aerodynamic center of the flaps and the center of gravity. Likewise,

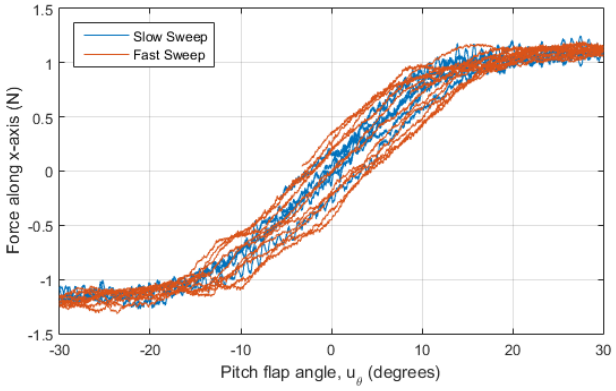
$$M_{body} = d_f \; F_{y,sensor} \tag{5.24}$$

where M_{body} is the pitching moment of the body and $F_{y,sensor}$ is the y-axis force on the sensor.

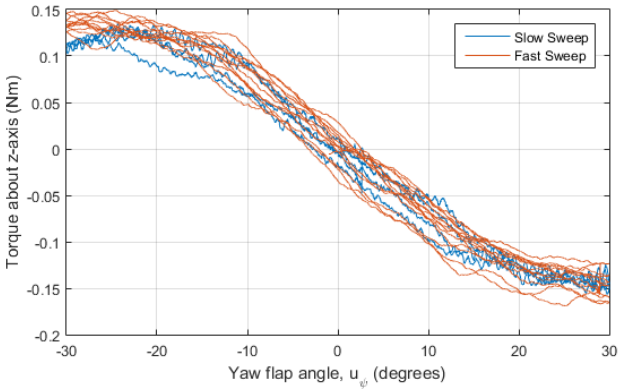
Based on visual inspection of the results in Figure 5.1, a sinusoidal function is used to approximate the relationship. This can capture the relatively linear section near the origin and the gradual saturation that peaks at about $\pm 25°$. This approximated relationship is



(a) Roll test results



(b) Pitch test results



(c) Yaw test results

Figure 5.1: Tests measuring the relationship between flap angle and the moment / force induced on the vehicle

written parametrically as

$$g(u_{\mathrm{x}}) = c_{gm} \sin\left(\frac{\pi/2}{c_{gs}}(u_{\mathrm{x}} - \delta_{\mathrm{x}})\right) + c_{gm} \sin\left(\frac{\pi/2}{c_{gs}}\delta_{\mathrm{x}}\right) \tag{5.25}$$

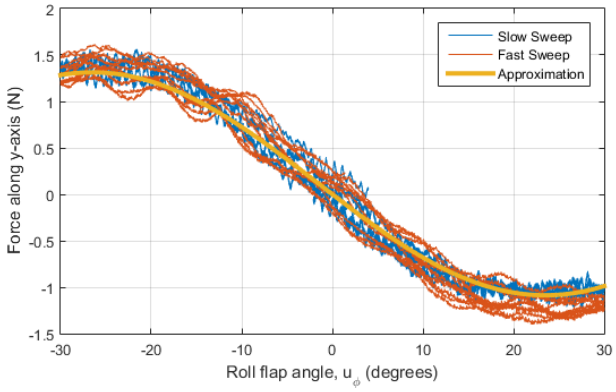
where c_{gm} is the magnitude scale factor, c_{gs} scales the input down by saturation peak, and δ_{x} is a biasing term that accounts for the flaps producing more force in one direction than the other due to imperfect symmetry in the individual flaps. The relationship seen in Figure 5.1

and approximated in (5.25) is consistent with standard models of lift in airfoils, especially as the angle of attack nears the stall angle [21]. The subscript x is in place of $s1$, $s2$, $s3$, or $s4$ for the four servos. Note that the δ_{x} term does not adjust the center value u_{x} that yields $g(u_{\mathrm{x}}) = 0$; the function $g(\cdot)$ still passes through the origin regardless of the value of δ_{x}. The function (5.25) assumes that the flaps have already been trimmed individually to be perfectly centered at zero degrees. Notice that while each axis has its own biasing term, the magnitude scale factor and saturation peak are assumed to be the same for all servo inputs, based on the symmetry of the layout of the flaps.

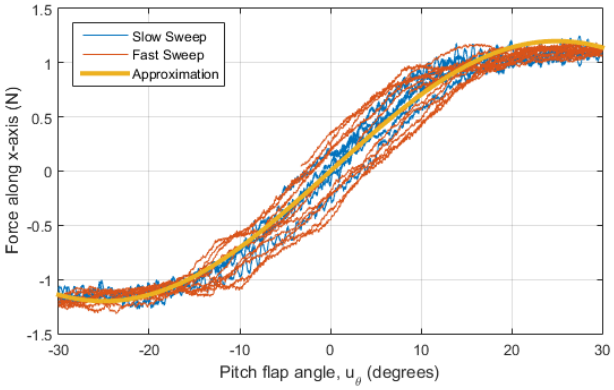
By inspection, the saturation peak value is selected to be 25°, and this is the value used throughout the remainder of this thesis. To test the fit of (5.25) to the force and torque data shown previously in Figure 5.1, magnitudes and bias values were manually picked. In Figure 5.2, the input function approximation is plotted on top of the previous test results.

The relationship in (5.25) can be modified slightly to make parameter identification simpler. First, the use of digital servos for the control of the flaps allows for the trimming of not only the center point, but also of the end points of the servo range. This means that the bias term δ_{x} can be effectively trimmed away. Second, the c_{gm} scale factor can be pulled out to be absorbed into the k_{cs} factor in (5.22) and the k_{ψ} factor in (5.18). The other modification is to multiply the terms in (5.25) by c_{gs}. This results in the transformed input being the same order of magnitude as the original input u_{x}. This yields

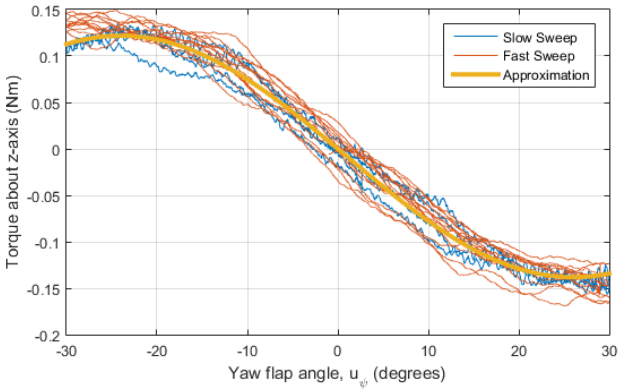
$$\nu_{\mathrm{x}} = \tilde{g}(u_{\mathrm{x}}) = c_{gs} \sin\left(\frac{\pi/2}{c_{gs}} u_{\mathrm{x}}\right) \tag{5.26}$$



(a) Roll input approximation



(b) Pitch input approximation



(c) Yaw input approximation

Figure 5.2: The approximation for the relationship between flap angle and the moment / force induced on the vehicle, shown on top of the recorded test data

where ν_{x} is the transformed input ν_{s1}, ν_{s2}, ν_{s3}, or ν_{s4}. As in (5.8), the vector $\boldsymbol{\nu}_s = \begin{bmatrix} \nu_{s1} & \nu_{s2} & \nu_{s3} & \nu_{s4} \end{bmatrix}^{\mathsf{T}}$ can be transformed into three aircraft inputs $\boldsymbol{\nu}_a = \begin{bmatrix} \nu_\phi & \nu_\theta & \nu_\psi \end{bmatrix}^{\mathsf{T}}$ as

$$\boldsymbol{\nu}_a = \boldsymbol{T}_u \boldsymbol{\nu}_s \tag{5.27}$$

The resulting changes in scale from $g(\cdot)$ to $\tilde{g}(\cdot)$ are absorbed into the k_{cs} factor in the

roll-pitch dynamics in (5.22) as

$$k_{cs}g_{\phi}(\boldsymbol{u}) = \tilde{k}_{cs}\tilde{g}_{\phi}(\boldsymbol{u}) = \tilde{k}_{cs}\left(0.5\tilde{g}(u_{s1}) - 0.5\tilde{g}(u_{s3})\right) \tag{5.28a}$$

$$k_{cs}g_{\theta}(\boldsymbol{u}) = \tilde{k}_{cs}\tilde{g}_{\theta}(\boldsymbol{u}) = \tilde{k}_{cs}\left(0.5\tilde{g}(u_{s2}) - 0.5\tilde{g}(u_{s4})\right) \tag{5.28b}$$

where $\tilde{k}_{cs}$ is the combination of k_{cs}, $1/c_{gm\phi}$, and c_{gs}. In a similar manner, the scale change is absorbed into k_{ψ} from (5.18) as

$$k_{\psi}\omega_p^2\left(g_{\psi}(\boldsymbol{u}) + k_{\psi 0}\right) = \tilde{k}_{\psi}\omega_p^2\left(\tilde{g}_{\psi}(\boldsymbol{u}) + \tilde{k}_{\psi 0}\right) \tag{5.29}$$

It is useful to have the inverse of $\tilde{g}(u_{\mathrm{x}})$,

$$u_{\mathrm{x}} = \tilde{g}^{-1}(\nu_{\mathrm{x}}) = \sin^{-1}\left(\frac{\nu_{\mathrm{x}}}{c_{gs}}\right)\frac{c_{gs}}{\pi/2} \tag{5.30}$$

where the x subscript stands for one of the four servos. This inverse relationship $\tilde{g}^{-1}(\nu_{\mathrm{x}})$ is valid under the condition

$$|\nu_{\mathrm{x}}| \leq c_{gs} \tag{5.31}$$

5.4 Model Parameter Identification Process

In order to determine the values of the parameters for the model that best fit the true dynamics of the vehicle, a cost function minimization search was performed on the param-

eters, first on the subset of parameters associated with the yaw portion of the model, then the subset associated with the roll and pitch portion of the model. This cost minimization compares the difference between flight data and simulation results.

The flight data is obtained by recording data while performing a test flight. The initial test flights occurred on a tether; however, the interaction with the tether corrupts the data and produces a less accurate model. When the model becomes more accurate, though, the controller tends to perform better, allowing the test flight to interact less with the tether, thus producing better flight data for modeling. This modeling and controller design loop is an iterative process, and is discussed in more detail in Section 6.4.

Simulation data is used in the cost minimization process. The simulation data is produced by using the `ode45` nonlinear ordinary differential equation (ODE) solver in MATLAB. The roll and pitch portion of the model are treated separately from the yaw portion during the parameter identification, and so these are simulated separately. During the yaw simulation, the roll angle ϕ, pitch angle θ, x-axis angular rate p, y-axis angular rate q, motor speed ω_p, and the control inputs $\boldsymbol{u}$ are interpolated from the flight data, while the yaw angle ψ and z-axis angular rate r are simulated. Likewise, when the roll and pitch simulation is run, the yaw angle ψ, z-axis angular rate r, motor speed ω_p, and control inputs $\boldsymbol{u}$ are interpolated from flight data while the roll angle ϕ, pitch angle θ, x-axis angular rate p, and y-axis angular rate q are simulated. In both cases, the simulation data is given only at the time instances that the logged flight data is sampled at, to allow for easier comparison.

The cost function being minimized for the yaw parameters is

$$J_\psi(\psi[k],\ r[k],\ \psi_{sim}[k],\ r_{sim}[k]) = \sum_{\kappa=k_i}^{k_f}(\psi[\kappa] - \psi_{sim}[\kappa])^2 + w_J \sum_{\kappa=k_i}^{k_f}(r[\kappa] - r_{sim}[\kappa])^2 \quad (5.32)$$

where k is the discrete-time index of the data, k_i is the beginning index, k_f is the final index, ψ_{sim} is the simulated yaw, r_{sim} is the simulated z-axis angular rate, and w_J is a weighting constant indicating the relative importance of the angular rate error minimization over the angle error minimization. Similarly, the cost function for the roll and pitch parameters is

$$\begin{aligned} J_{\phi\theta}(\phi[k],\ \theta[k],\ p[k],\ q[k],\ \phi_{sim}[k],\ \theta_{sim}[k],\ p_{sim}[k],\ q_{sim}[k]) = \\ \sum_{\kappa=k_i}^{k_f}\left[(\phi[\kappa] - \phi_{sim}[\kappa])^2 + (\theta[\kappa] - \theta_{sim}[\kappa])^2\right] \\ + w_J \sum_{\kappa=k_i}^{k_f}\left[(p[\kappa] - p_{sim}[\kappa])^2 + (q[\kappa] - q_{sim}[\kappa])^2\right] \end{aligned} \quad (5.33)$$

The weighting constant w_J accounts for the difference in magnitude of the units of the angle errors (degrees) and the units of the angular rate errors (degrees per second). Because the trajectories of the angles are derived almost completely from the angular rates, the angular rates were deemed to be more directly important in parameter identification, and so the weight w_J was weighted heavily in favor of the angular rates.

The cost functions are minimized using MATLAB's `fmincon` and `fminsearch`. In both of these functions, the cost is evaluated by simulating the set of ODEs for a given set of parameter values and then passing the simulation results into (5.32) or (5.33). The `fmincon`

function performs a constrained minimization and was configured to use the interior-point algorithm [22]. While `fmincon` does a good job of parameter optimization, it tends to quickly get stuck in local minima. In the parameter identification script, `fmincon` is called first and then the resulting parameters are passed in as the starting point for `fminsearch`. The `fminsearch` function uses the simplex search algorithm to minimize the cost function [23]. This function is able to break out of the local minima left by `fmincon`, but can also fail to reduce the value of the cost function for a large number of iterations without any improvement in the cost.

5.5 Model Parameter Identification Results

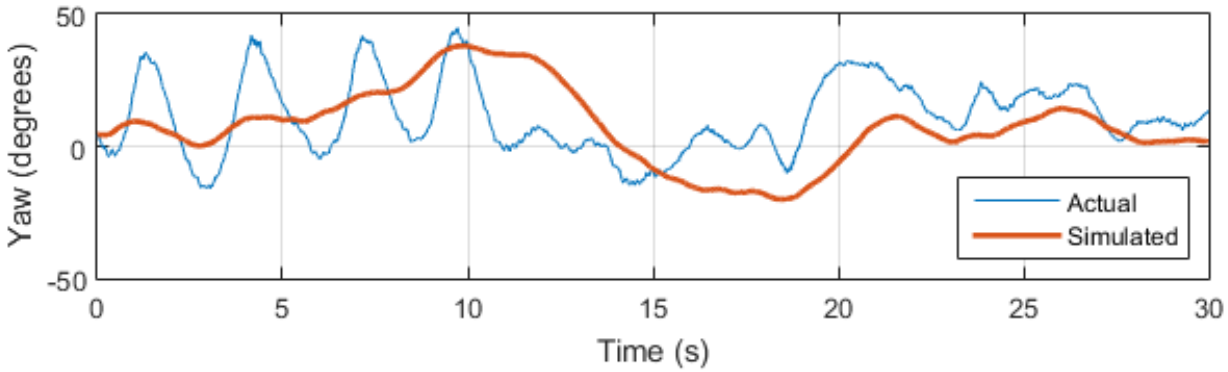
The parameter identification process is performed using flight data. For this process, a thirty second clip of data is selected on the criteria of having clean, glitchless data, minimal tether interaction, and minimal risk of wall or ground effect interaction. After performing the cost-minimizing parameter search, the system is simulated using the identified set of parameters.

The first set of parameters identified is the set associated with the yaw / z-axis portion of the model. The identified parameters are listed in Table 5.1. The simulation of the yaw angle and the z-axis angular rate are shown alongside the recorded data in Figure 5.3.

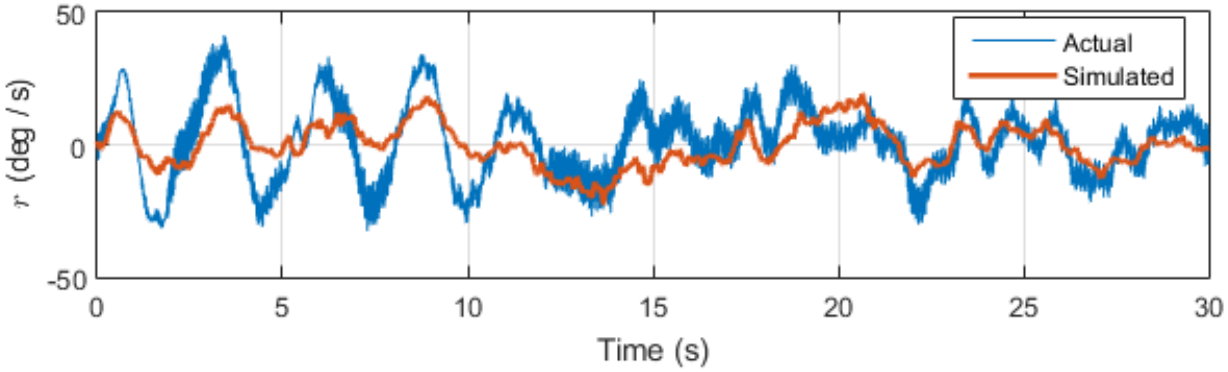
The simulation data for the identified yaw parameters appears visually to be a somewhat poor fit to the recorded data. The difference between the simulation and the data occurs

Table 5.1: Identified parameters of the yaw dynamics

Parameter	**Identified Value**
$\tilde{k}_{\psi}$	-2.26377e-4
$\tilde{k}_{\psi 0}$	-0.0125123
$k_{\psi d}$	25.8541



(a) Yaw angle simulation



(b) z-axis angular rate simulation

Figure 5.3: The recorded flight data for the yaw angle (a) and z-axis angular rate r (b) shown alongside the simulated yaw and z-axis angular rate results using the parametric yaw model and the identified parameters

for a number of reasons. First, the vehicle may experience effects from unmodeled dynamics such as ground effects. There could be imperfections or imbalances in the vehicle or the flaps. Another issue is that the recorded data contains some amount of error. Because of the poor yaw estimation by the AHRS, the yaw estimate had to be determined purely from the magnetometer data, as described in Section 4.1.3. However, this makes the yaw estimate sensitive to magnetic disturbances such as the structure of the building in which the test was performed and electromagnetic interference from the vehicle's motor. It is also worth noting that the identification of the yaw dynamics may be considered less important than the identification of the roll and pitch dynamics, as yaw control is less critical for flight stability than roll and pitch control.

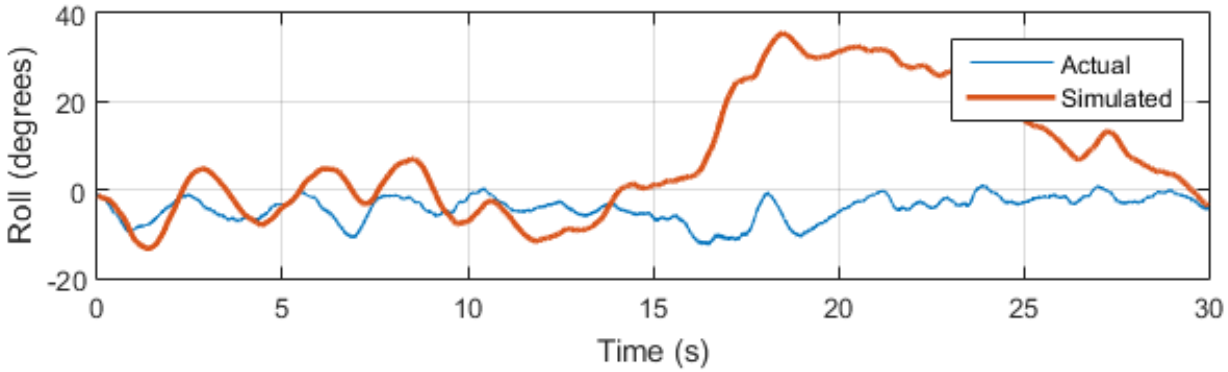
The roll and pitch dynamics are similarly simulated, and the best fitting set of parameters is identified that minimizes the cost function (5.33). The identified parameters are listed in Table 5.2. The simulated roll and pitch are shown alongside the recorded roll and pitch in Figure 5.4. Likewise, the simulated x-axis and y-axis angular rates are shown alongside the recorded angular rates in Figure 5.5.

The simulation results for the angular rates for the roll and pitch model seen in Figure 5.5 match fairly closely to the recorded data. It still suffers from imperfect parameter identification, presence of unmodeled effects, and the issue of errors in the actual recorded data. The matching in the angles in Figure 5.4 is worse, as errors in p and q become magnified due to the integration process. However, the dynamics described by the defined model and identified parameters are close to the true dynamics and provide a good basis for controller

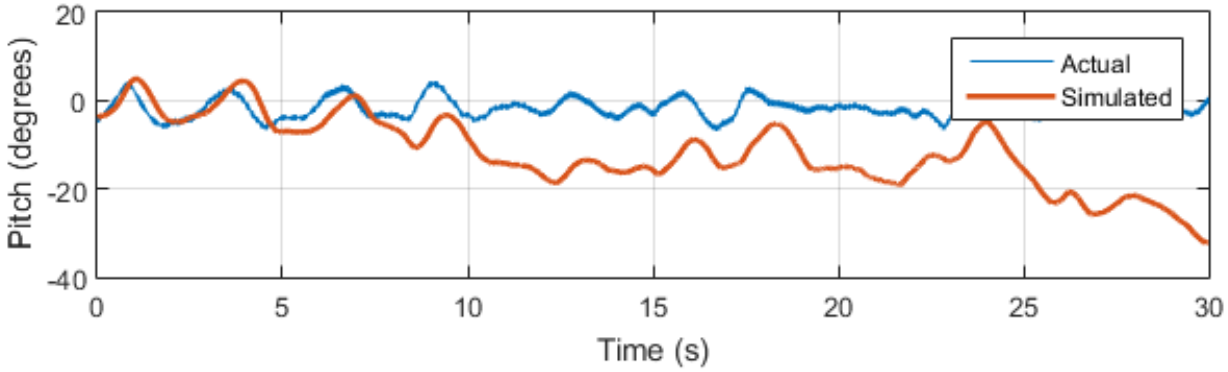
Table 5.2: Identified parameters of the roll and pitch dynamics

Parameter	**Identified Value**
k_p	7.83507e-3
$\tilde{k}_{cs}$	7.63713e-5
k_{aero}	7.95150e-3
k_g	4.44613e-4
k_d	11.6918

design. In Chapter 6, a controller is designed based on this model and is tested on the vehicle.

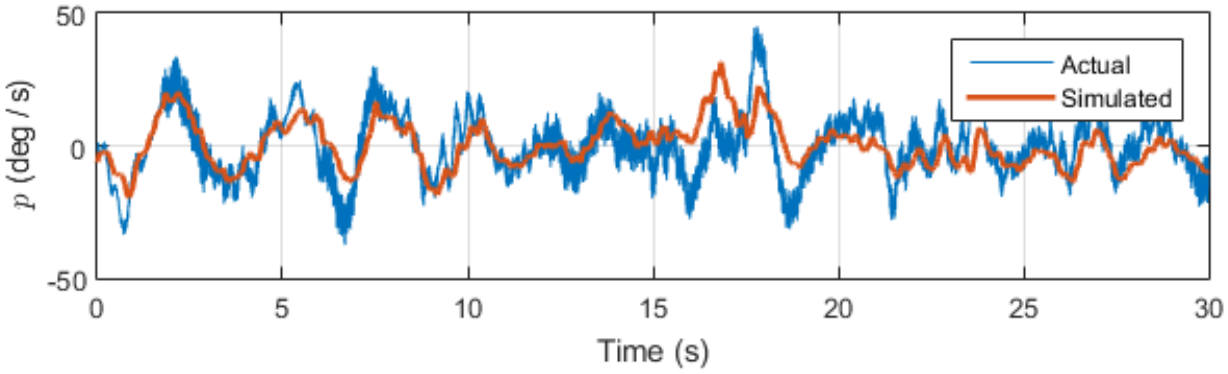


(a) Roll angle simulation

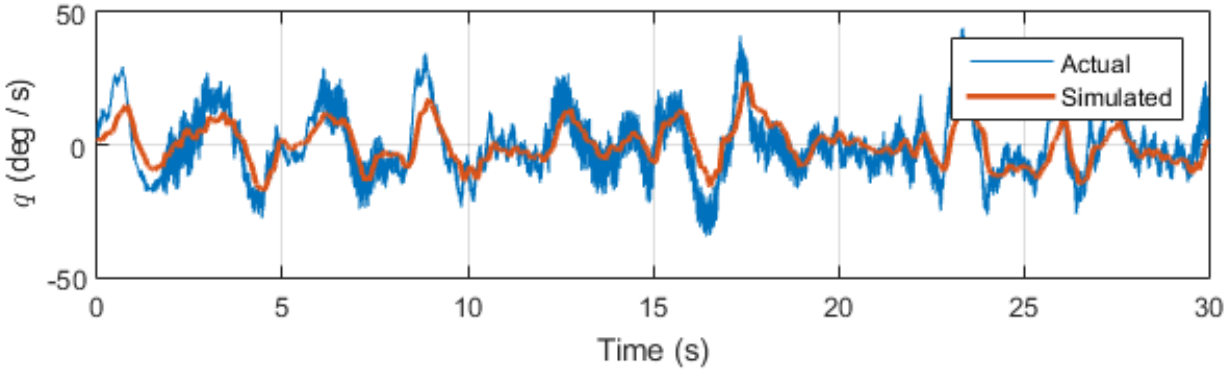


(b) Pitch angle simulation

Figure 5.4: The recorded flight data for the roll angle (a) and the pitch angle (b) shown alongside the simulated roll and pitch results using the parametric roll/pitch model and the identified parameters



(a) x-axis angular rate simulation



(b) y-axis angular rate simulation

Figure 5.5: The recorded flight data for the x-axis angular rate p (a) and the y-axis angular rate q (b) shown alongside the simulated x-axis and y-axis angular rate simulation results using the parametric roll/pitch model and the identified parameters

Chapter 6

Control

This chapter begins by briefly describing the proposed controller design method. It uses the modeling result from the previous chapter to design a state feedback controller for the EDF vehicle. It also covers the verification of the control system in physical flight tests.

6.1 Linear Quadratic Regulator Formulation

The linear quadratic regulator (LQR) design method uses a linear model and produces a matrix of state feedback gains that minimizes the weighted integration of a quadratic function of state error and control effort. We can write the linear state space model in the

form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{6.1a}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{6.1b}$$

where $\boldsymbol{x}(t)$ is the state vector, $\boldsymbol{u}(t)$ is the input vector, and $\boldsymbol{x}_0$ is the initial condition of the state vector.

In the general form of the LQR problem, the cost function being minimized is

$$J(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \frac{1}{2}\boldsymbol{x}^\mathsf{T}(t_f)\boldsymbol{H}\boldsymbol{x}(t_f) + \frac{1}{2}\int_0^{t_f} \boldsymbol{x}^\mathsf{T}(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau) + \boldsymbol{u}^\mathsf{T}(\tau)\boldsymbol{R}\boldsymbol{u}(\tau)d\tau \tag{6.2}$$

where t_f is the final time, $\boldsymbol{H}$ is a positive semi-definite weighting matrix on the final states, $\boldsymbol{Q}$ is a positive semi-definite weighting matrix on the state trajectory, and $\boldsymbol{R}$ is a positive semi-definite weighting matrix on the control inputs. The LQR problem finds the gain matrix $\boldsymbol{K}(t)$ that minimizes the cost J when the control input is

$$\boldsymbol{u}(t) = -\boldsymbol{K}(t)\boldsymbol{x}(t) \tag{6.3}$$

In many systems, where the run time is longer than the system transients, only the steady state solution of $\boldsymbol{K}(t)$ is desired. In this case, the cost function is

$$J(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \frac{1}{2}\int_0^{\infty} \boldsymbol{x}^\mathsf{T}(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau) + \boldsymbol{u}^\mathsf{T}(\tau)\boldsymbol{R}\boldsymbol{u}(\tau)d\tau \tag{6.4}$$

and the steady state $\boldsymbol{K}$ that minimizes this is

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{P} \tag{6.5}$$

where the matrix $\boldsymbol{P}$ is the solution to the algebraic Riccati equation

$$\boldsymbol{0} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathsf{T}}\boldsymbol{P} \tag{6.6}$$

A more complete derivation of the steady-state LQR problem and its solution can be found in [24].

6.2 Linearization

In order to use the LQR controller design method, a linear model of the form (6.1) is needed. The nonlinear model of the attitude dynamics of the EDF vehicle can be linearized by evaluating the Jacobian of the model at the hover-mode operating point.The full state vector $\boldsymbol{x}$ for the attitude dynamics is defined as $\begin{bmatrix}\phi & \theta & \psi & p & q & r\end{bmatrix}^{\mathsf{T}}$. The full attitude

model, composed of (5.1), (5.18), and (5.22), is

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}) = \left[\begin{array}{c|ccc} & 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ \mathbf{0}_{3\times3} & 0 & \cos\phi & -\sin\phi \\ & 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \\ \hline & -k_d & -k_p\omega_p & 0 \\ \mathbf{0}_{3\times3} & k_p\omega_p & -k_d & 0 \\ & 0 & 0 & -k_{\psi d} \end{array}\right] \begin{bmatrix} \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} + \left[\begin{array}{c} \mathbf{0}_{3\times1} \\ \hline -k_g qr \\ k_g rp \\ \tilde{k}_\psi \omega_p^2 \tilde{k}_{\psi 0} \end{array}\right] + \omega_p^2 \left[\begin{array}{c} \mathbf{0}_{3\times1} \\ \hline \tilde{k}_{cs}\tilde{g}_a(u_\phi) \\ \tilde{k}_{cs}\tilde{g}_a(u_\theta) \\ \tilde{k}_\psi \tilde{g}_a(u_\psi) \end{array}\right] + k_{aero} \left[\begin{array}{c} \mathbf{0}_{3\times1} \\ \hline u \\ v \\ 0 \end{array}\right] \tag{6.7}$$

where the notation $\mathbf{0}_{m\times n}$ indicates an m by n matrix of zeros. The Jacobian for this system is

$$\frac{d\boldsymbol{f}(\boldsymbol{x})}{d\boldsymbol{x}} = \left[\begin{array}{ccc:ccc} q\cos\phi\tan\theta - r\sin\phi\tan\theta & q\sin\phi\sec^2\theta + r\cos\phi\sec^2\theta & 0 & & & \\ -q\sin\phi - r\cos\phi & 0 & 0 & & & \\ q\cos\phi\sec\theta - r\sin\phi\sec\theta & q\sin\phi\sec\theta\tan\theta + r\cos\phi\sec\theta\tan\theta & 0 & & & \\ \hdashline 0 & 0 & 0 & 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & 0 & 0 & 0 & \cos\phi & -\sin\phi \\ 0 & 0 & 0 & 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \\ \hdashline & & & -k_d & -k_p\omega_p - k_g r & -k_g q \\ & & & k_p\omega_p + k_g r & -k_d & k_g p \\ & & & 0 & 0 & -k_{\psi d} \end{array}\right] \tag{6.8}$$

The hover-mode equilibrium point being used is

$$\boldsymbol{x}_{eq} = \begin{bmatrix} \phi_{eq} & \theta_{eq} & \psi_{eq} & p_{eq} & q_{eq} & r_{eq} \end{bmatrix}^{\mathsf{T}} = \mathbf{0}_{6\times 1} \tag{6.9}$$

Evaluating the Jacobian in (6.8) at the equilibrium point yields the linearized state matrix $\boldsymbol{A}$,

$$\boldsymbol{A} = \left.\frac{d\boldsymbol{f}(\boldsymbol{x})}{d\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{eq}} = \left[\begin{array}{ccc:ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 0 & -k_d & -k_p\omega_{pe} & 0 \\ 0 & 0 & 0 & k_p\omega_{pe} & -k_d & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_{\psi d} \end{array}\right] \tag{6.10}$$

where ω_{pe} is the equilibrium propeller speed of 4500 RPM or 471 radians per second.

When using the transformed input $\boldsymbol{\nu}_a = \begin{bmatrix} \nu_\phi & \nu_\theta & \nu_\psi \end{bmatrix}^\mathsf{T}$ described in Section 5.3 as the system input, the input matrix $\boldsymbol{B}_a$ is

$$\boldsymbol{B}_a = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hdashline \tilde{k}_{cs}\omega_{pe}^2 & 0 & 0 \\ 0 & \tilde{k}_{cs}\omega_{pe}^2 & 0 \\ 0 & 0 & \tilde{k}_{\psi}\omega_{pe}^2 \end{array}\right] \tag{6.11}$$

where the a subscript indicates that it is using the aircraft-centric input $\boldsymbol{\nu}_a$ as opposed to the servo-centric input $\boldsymbol{\nu}_s$. The equivalent servo-centric input matrix can be obtained using

the relationship in (5.27),

$$\boldsymbol{B}_a \boldsymbol{\nu}_a = \boldsymbol{B}_a \boldsymbol{T}_u \boldsymbol{\nu}_s = \boldsymbol{B}_s \boldsymbol{\nu}_s \tag{6.12}$$

$$\boldsymbol{B}_s = \boldsymbol{B}_a \boldsymbol{T}_u \tag{6.13}$$

With $\boldsymbol{A}$ and $\boldsymbol{B}_s$ available, the linear state equation

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_s \boldsymbol{\nu}_s \tag{6.14}$$

is used for the design of the linear state feedback controller. Inserting the values of the parameters identified in Section 5.5 into (6.10) and (6.11) gives the matrices

$$\boldsymbol{A} = \left[\begin{array}{ccc:ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 0 & -11.6918 & -3.69219 & 0 \\ 0 & 0 & 0 & 3.69219 & -11.6918 & 0 \\ 0 & 0 & 0 & 0 & 0 & -25.8541 \end{array}\right] \tag{6.15}$$

$$\boldsymbol{B}_a = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hdashline 16.9595 & 0 & 0 \\ 0 & 16.9595 & 0 \\ 0 & 0 & -50.2707 \end{array}\right] \tag{6.16}$$

It can be noted that this linearized system is controllable, as the controllability matrix composed of $\boldsymbol{A}$ and $\boldsymbol{B}_a$ has a full rank 6. Therefore, any linear feedback that asymptotically stabilizes this linearized system, such as the LQR controller to be proposed in Section 6.3, is also able to asymptotically stabilize the original nonlinear system in some local neighborhood around the equilibrium [25].

6.3 LQR Controller Synthesis and Implementation

The LQR controller synthesis is performed using MATLAB's `lqr` function. The weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ were initially selected as diagonal matrices whose elements are the inverse of the squares of typical desired values of states and control signals, as suggested in [26]. The $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices were then adjusted by trial and error to achieve the desired performance.

The matrices that are used in control synthesis are

$$\boldsymbol{Q} = \text{diag}\left(\left[\frac{1}{(\pi/10)^2} \quad \frac{1}{(\pi/10)^2} \quad \frac{1}{(\pi/6)^2} \quad \frac{1}{(\pi/4)^2} \quad \frac{1}{(\pi/4)^2} \quad \frac{1}{(\pi/2)^2}\right]\right) \tag{6.17}$$

where $\text{diag}(\boldsymbol{\alpha})$ indicates a diagonal matrix with a main diagonal of $\boldsymbol{\alpha}$, and

$$\boldsymbol{R} = \left(\frac{1}{30^2}\right)\boldsymbol{I}_{4\times 4} \tag{6.18}$$

where $\boldsymbol{I}_{4\times 4}$ is a 4 by 4 identity matrix.

Using $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices in (6.17-6.18) along with the final value of $\boldsymbol{A}$ in (6.15) and $\boldsymbol{B}_s$ obtained from the final value of $\boldsymbol{B}_a$ in (6.16), the state feedback gain matrix $\boldsymbol{K}_s$ is computed

$$\boldsymbol{K}_s = \left[\begin{array}{cccccc} 67.5216 & 0.541128 & -28.6479 & 26.4759 & 0 & -9.10824 \\ -0.541128 & 67.5216 & -28.6479 & 0 & 26.4759 & -9.10824 \\ -67.5216 & -0.541128 & -28.6479 & -26.4759 & 0 & -9.10824 \\ 0.541128 & -67.5216 & -28.6479 & 0 & -26.4759 & -9.10824 \end{array}\right] \tag{6.19}$$

Because of the symmetry of the system, this gain matrix can be transformed equivalently

back and forth between the servo-centric and aircraft-centric matrices

$$\boldsymbol{K}_a = \boldsymbol{T}_u \boldsymbol{K}_s \tag{6.20}$$

$$\boldsymbol{K}_s = \boldsymbol{T}_{inv} \boldsymbol{K}_a \tag{6.21}$$

noting that

$$\boldsymbol{K}_a = \boldsymbol{T}_u \boldsymbol{T}_{inv} \boldsymbol{K}_a \tag{6.22}$$

$$\boldsymbol{K}_s = \boldsymbol{T}_{inv} \boldsymbol{T}_u \boldsymbol{K}_s \tag{6.23}$$

The servo-centric or aircraft-centric control signal can be determined from the state vector using either matrix,

$$\boldsymbol{\nu}_a = -\boldsymbol{K}_a \boldsymbol{x} \tag{6.24}$$

$$\boldsymbol{\nu}_s = -\boldsymbol{K}_s \boldsymbol{x} \tag{6.25}$$

On the vehicle, the gain matrix $\boldsymbol{K}_a$ is used to find $\boldsymbol{\nu}_a$ which is then transformed into $\boldsymbol{\nu}_s = \boldsymbol{T}_{inv} \boldsymbol{\nu}_a$. Finally, $\boldsymbol{\nu}_a$ is transformed into the actual servo angle vector $\boldsymbol{u}_s$ using (5.30).

6.4 Flight Tests and Data Recording

In order to identify the model parameters as described in Section 5.4, in-flight data is needed. This is done by recording data from the vehicle during test flights to an SD card. The data recorded during test flights includes:

- Attitude estimates (ϕ, θ, ψ)
- Angular rate measurements (p, q, r)
- Acceleration measurements ($\dot{u}$, $\dot{v}$, $\dot{w}$)
- Propeller angular speed (ω_p)
- Commanded servo values (u_{s1}, u_{s2}, u_{s3}, u_{s4})
- Complementary filtered horizontal velocity estimates ($v_{c,xi}$, $v_{c,yi}$)
- Complementary filtered altitude and altitudinal velocity (h_{est}, $\dot{h}_{est}$)
- Raw altitude measurements (h_{IR}, h_{sonar})

The flights are also usually video recorded as well, and the video and data logs can be approximately synchronized. This helps to identify segments of data that should provide the best example of the model dynamics without undesired effects such as interaction with the flight tether that supports the vehicle during tests. Figure 6.1 shows an image of the vehicle being supported by a tether, which is typical for early flight tests.

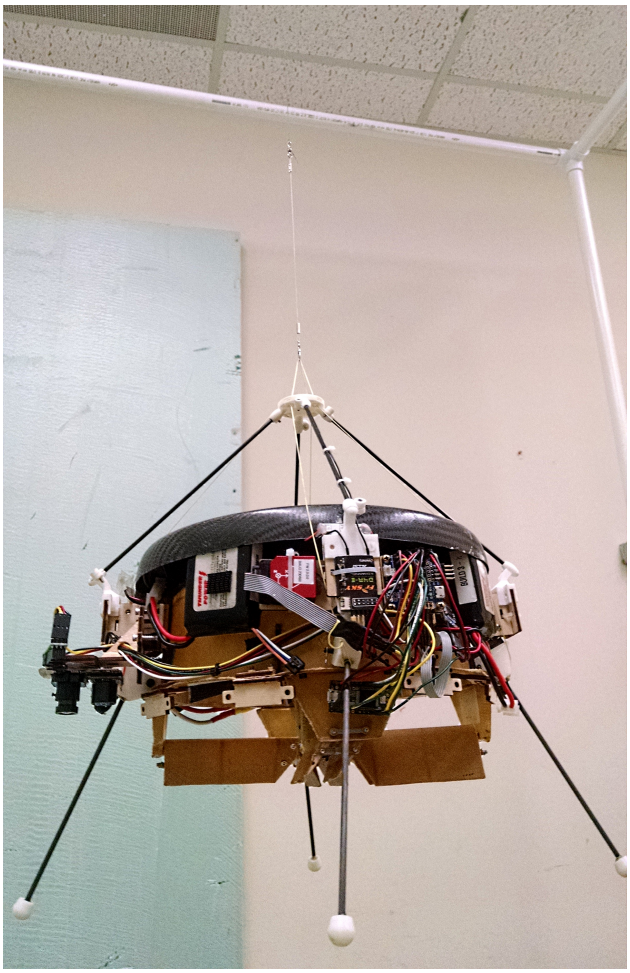


Figure 6.1: An image of the vehicle being supported by a tether for flight testing
.

As previously mentioned in Section 5.4, flight testing, modeling, and controller design is an iterative process. Better flight tests provide cleaner data for modeling and parameter identification. A more accurate model allows for the design of a better, more stable controller. A more stable controller then produces better flight tests. Initially, this process was begun by using a combination of physical measurements of the vehicle, and AVID's published linearized dynamics for the chassis [27]. However, because of the changes made to the vehicle such as

instrumentation and electronics changes, the linearized dynamics from AVID were no longer accurate for the vehicle. By performing multiple tests and making changes to the $\boldsymbol{Q}$ and $\boldsymbol{R}$ weighting matrices, marginally stable performance was achieved under support of the tether. The vehicle was very shaky, had highly chattering servo inputs, and had to have at least part of its weight supported by the tether; however, it was able to capture data that could then improve the model. This iterative test-model-control process was carried out in parallel to refinements in the model terms as well as the instrumentation improvements discussed in Chapter 4.

When tethered flights proved a controller to be reasonably stable, untethered flights could be performed with minimal risk to the vehicle. Untethered test flights were executed indoors in a safe and spacious open area. The data obtained from these tests proved to be the most useful for parameter identification, as there was no tether interaction to obscure the model. It was data from these flights that was used to perform the final parameter identification presented in Section 5.5.

6.5 Final Flight Test Results

Using the final controller design shown in Section 6.3, several full untethered flight test were performed. An image of the vehicle in flight is shown in Figure 6.2. Additionally, Figure 6.3 shows the roll, pitch, and yaw angles, the body angular rates, and the commanded servo angles recorded from the flight. This is plotted over the time frame of just after take-off to

just before landing.

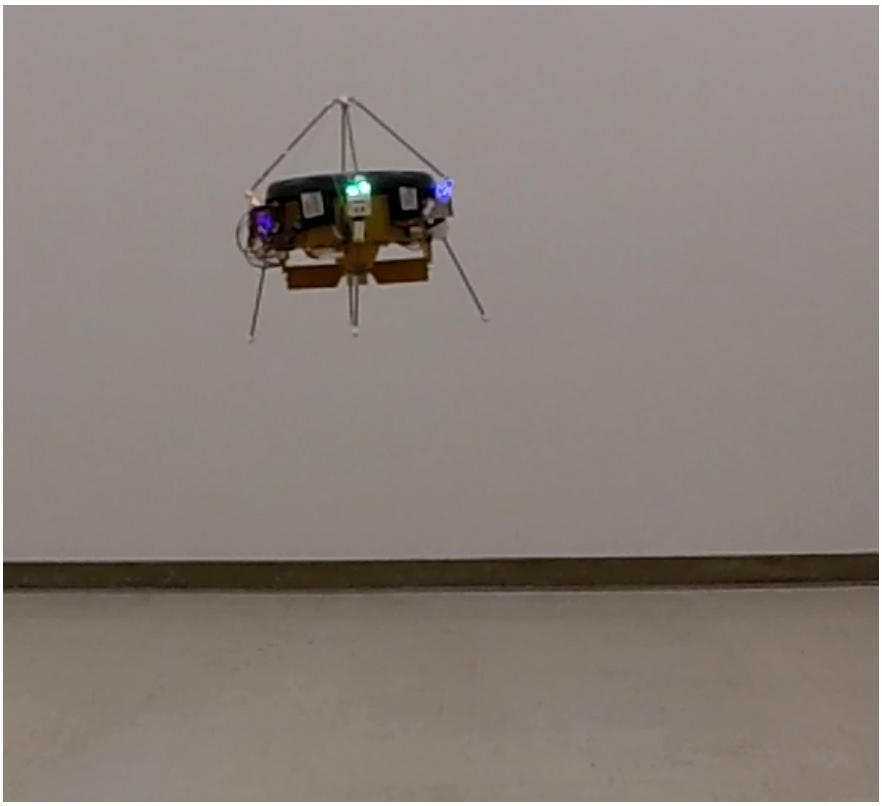
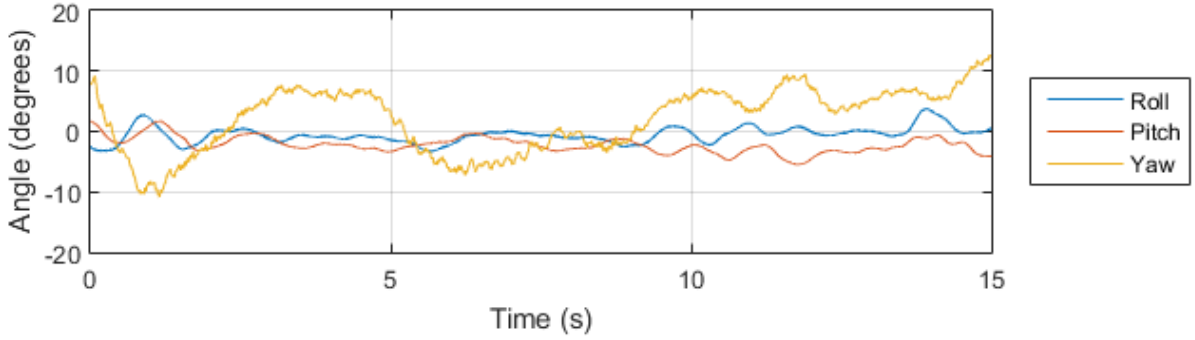


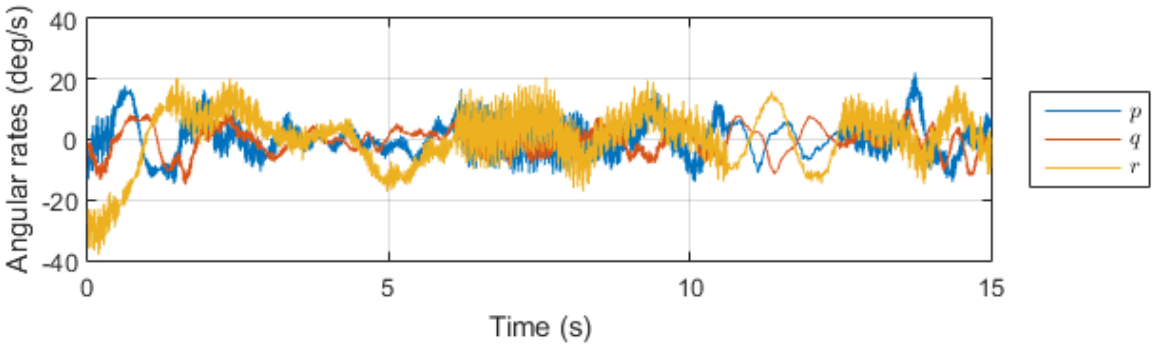
Figure 6.2: An image of the vehicle during a flight test

.

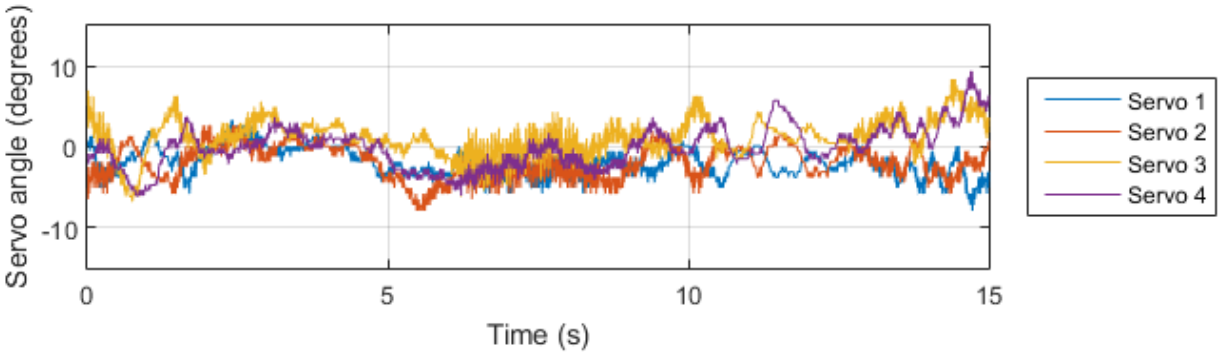
As seen in Figure 6.3, the estimated roll and pitch angles stay almost entirely within $\pm 5^\circ$ of the origin, with only one excursion of the pitch to -5.5°. The estimated yaw angle stays within about $\pm 10^\circ$. The angular rates all stay within about $\pm 20^\circ/\text{s}$, with the notable exception being the large z-axis turn rate at the beginning. This large turn rate occurs because the vehicle is initially slightly misaligned in the yaw and it is able to suddenly rotate towards zero yaw as soon as the vehicle lifts off the ground. The servo values shown in Figure 6.3c show that the control effort is nowhere near saturating to the full range of control authority at $\pm 25^\circ$, as identified in Section 5.3. Additionally, the plot shows that



(a) Roll, pitch, and yaw angles



(b) Body angular rates

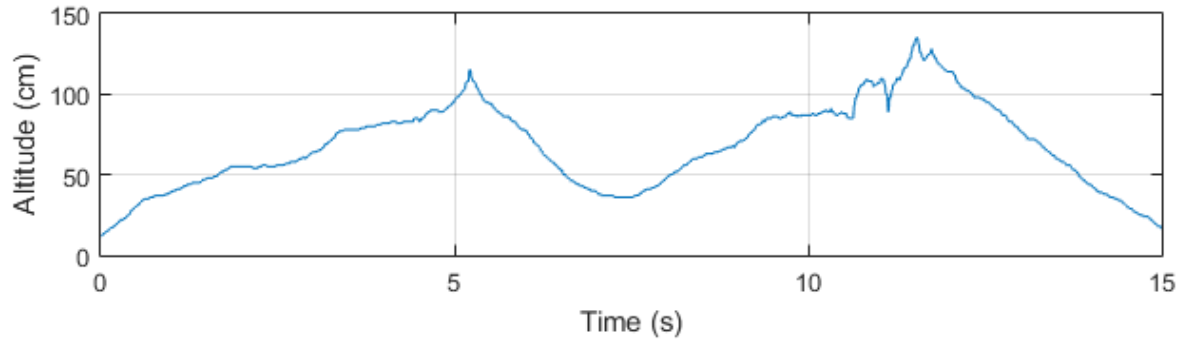


(c) Servo-controlled flap angles

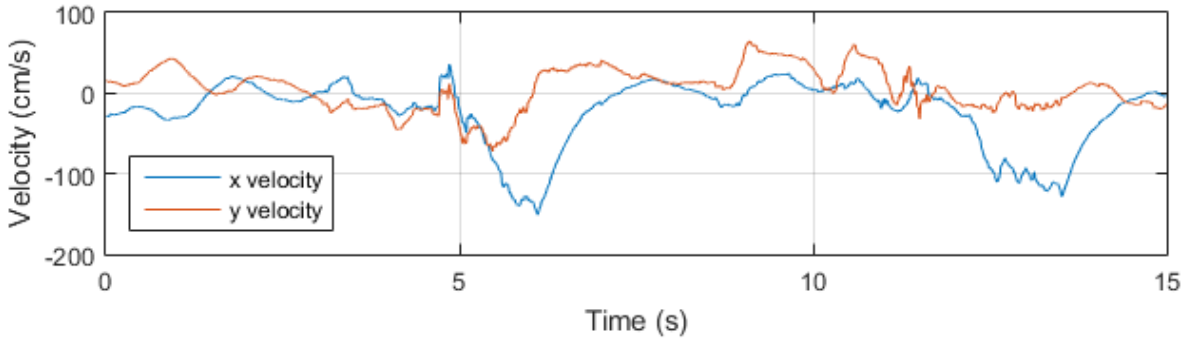
Figure 6.3: The attitude angles (a), angular rates (b), and servo angles (c) for the data recorded for the flight using the finalized controller

there is little of the undesirable chattering in the control signal, in the sense that the high bandwidth oscillations are low amplitude.

During this flight, a pilot is giving attitude and throttle commands to the vehicle through an RC transmitter. Because no outer loop control is enabled, the pilot must manually regulate the horizontal position of the vehicle by commanding roll and pitch angles and regulate the altitude of the vehicle by adjusting the throttle. For the same flight shown previously in Figure 6.3, the estimates from both the altitude complementary filter and horizontal velocity complementary filter are shown in Figure 6.4. While there is no ground truth available to which to compare these estimates, visual inspection of a video recording of the flight indicates that both the altitude and velocity estimates appear to be fairly accurate. With these estimates, outer loop control of altitude and velocity are possible, which would ease the burden on the pilot for remote operation and potentially build up to fully autonomous operation of the vehicle.



(a) Altitude estimate



(b) Horizontal velocity estimate

Figure 6.4: The complementary filtered altitude estimate (a) and complementary filtered horizontal velocity estimate (b) for the flight using the finalized controller

Chapter 7

Conclusions

This work presents an in-depth analysis of instrumentation for an electric ducted fan UAV, which is broadly applicable for any small electric UAV. It develops a simple parametric model for the ducted fan vehicle dynamics and demonstrates a method for identifying the parameters without the need for wind tunnel tests or CFD analysis. Finally, it demonstrates that the discussed parametric model and identified parameters can be used to develop a linear state feedback controller, and that this controller stabilizes the attitude of the vehicle in flight.

While this thesis discusses instrumentation and estimation of altitude and velocity, only attitude is considered in the modeling and control design process. In future work, altitude and velocity may be modeled and controlled through an outer loop controller. Another potential expansion of this work is the use of gain scheduling to improve performance away

from the assumed operating point, such as at varying propeller speeds or non-level flight.

One of the motivations cited for this work is the relative advantage of a ducted fan vehicle over other UAV designs for the size-constrained task of indoor inspection. While this thesis covers the modeling and control of a vehicle suited for this task, it does not address the issue of self-generated disturbances that result from prop-wash in an enclosed environment. In future work, the behavior of these disturbances may be examined. It is believed that these disturbances can be modeled as time-varying Markov chains, and that the work in [28] may be used in future research to implement a controller for optimal rejection of the self-generated disturbances.

Bibliography

[1] E. Darack, "A brief history of unmanned aircraft." Air & Space Smithsonian. http://www.airspacemag.com/photos/a-brief-history-of-unmanned-aircraft-174072843, 17 May 2011. [Online: accessed July 2016].

[2] O. J. Ohanian III, *Ducted Fan Aerodynamics and Modeling, with Applications of Steady and Synthetic Jet Flow Control*. PhD dissertation, Virginia Polytechnic Institute and State University, May 2011.

[3] E. Lemieux, A. Webb, K. Lucas, P. Slebodnick, M. Krupa, F. Martin, and E. Hogan, "Remote tank monitoring and inspection methods." U.S. Naval Research Laboratory. http://www.nrl.navy.mil/research/nrl-review/2002/chem-biochem-research/lemieux/, 2002. [Online: accessed July 2016].

[4] A. C. Madrigal, "Inside the drone missions to Fukushima." The Atlantic. http://www.theatlantic.com/technology/archive/2011/04/

inside-the-drone-missions-to-fukushima/237981/, 28 April 2011. [Online: accessed July 2016].

[5] DARPA, "FACT FILE: A Compendium of DARPA Programs." https://www.hsdl.org/?view&did=440746, August 2003. [Online: accessed July 2016].

[6] G. Avanzini, S. D'Angelo, and G. De Matteis, "Performance and stability of ducted-fan uninhabited aerial vehicle model," *Journal of Aircraft*, vol. 40, pp. 86–93, Jan 2003.

[7] Z. Wang, Z. Liu, N. Fan, and M. Guo, "Flight dynamics modeling of a small ducted fan aerial vehicle based on parameter identification," *Chinese Journal of Aeronautics*, vol. 26, no. 6, pp. 1439 – 1448, 2013.

[8] H. W. Zhao, "Development of a dynamic model of a ducted fan VTOL UAV," Master's thesis, RMIT University, August 2009.

[9] H. Zhao, C. Bil, and B. Yoon, "Ducted fan VTOL UAV simulation in preliminary design," Aviation Technology, Integration, and Operations (ATIO) Conferences, American Institute of Aeronautics and Astronautics, Sep 2009.

[10] A. Banazadeh and S. A. Emami, "Control effectiveness investigation of a ducted-fan aerial vehicle using model predictive controller," in *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pp. 532–537, Aug 2014.

[11] R. Aruneshwaran, J. Wang, S. Suresh, and T. K. Venugopalan, "Neural adaptive back stepping flight controller for a ducted fan uav," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, pp. 2370–2375, July 2012.

[12] E. N. Johnson and M. A. Turbe, "Modeling, control, and flight testing of a small-ducted fan aircraft," *Journal of guidance, control, and dynamics*, vol. 29, no. 4, pp. 769–779, 2006.

[13] J.-M. Pflimlin, P. Binetti, P. Soures, T. Hamel, and D. Trouchet, "Modeling and attitude control analysis of a ducted-fan micro aerial vehicle," *Control Engineering Practice*, vol. 18, no. 3, pp. 209 – 218, 2010.

[14] R. Franz, M. Milam, and J. Hauser, "Applied receding horizon control of the caltech ducted fan," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 5, pp. 3735–3740 vol.5, May 2002.

[15] R. Franz and J. Hauser, "Optimization based parameter identification of the caltech ducted fan," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 2697–2702 vol.3, June 2003.

[16] B. Xu, X. Wang, C. Xiang, Y. Ma, and W. Chen, "Modelling and hovering control of a novel multi-tandem ducted fan vehicle," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pp. 1346–1354, June 2015.

[17] H. A. Ardakani and T. J. Bridges, "Review of the 3-2-1 Euler angles: a yaw-pitch-roll sequence." University of Surrey. http://personal.maths.surrey.ac.uk/st/T.Bridges/SLOSH/3-2-1-Eulerangles.pdf, April 2010. [Online: accessed May 2016].

[18] S. Radkowski and P. Szulim, "Analysis of vibration of rotors in unmanned aircraft," in *Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On*, pp. 748–753, Sept 2014.

[19] VectorNav Technologies, *VN-100 User Manual*, 2014. Rev 2.05.

[20] L. Meier, P. Riseborough, *et al.*, "PX4Flow firmware." GitHub. https://github.com/priseborough/px4flow, 2014. [Online: accessed April 2016].

[21] M. Tooley, A. Filippone, T. Megson, M. V. Cook, P. Carpenter, E. Houghton, D. Wyatt, L. R. Jenkinson, J. Marchman, F. D. Florio, J. Watkinson, P. Gurfil, and H. Curtis, *Aerospace Engineering e-Mega Reference*, ch. 2, pp. 36–54. Butterworth-Heinemann, 2009.

[22] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[23] J. Lagarias, J. Reeds, M. Wright, and P. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[24] P. Dorato, C. T. Abdallah, and V. Cerone, *Linear Quadratic Control: An Introduction*, ch. 2, pp. 9–20. Prentice-Hall, 1995.

[25] A. Isidori, *Nonlinear Control Systems*, ch. 4, pp. 172–173. Springer-Verlag, 3 ed., 1995.

[26] J. B. Burl, *Linear Optimal Control: $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Methods*, ch. 6, pp. 194–195. Menlo Park, CA: Addison Wesley Longman, 1999.

[27] AVID LLC, "EDF-8 linearized dynamics," May 2013.

[28] C. C. Lutz, *Switched Markov Jump Linear Systems: Analysis and Control Synthesis*. PhD dissertation, Virginia Polytechnic Institute and State University, October 2014.