ABSTRACT

Title of thesis: OPEN LOOP SYSTEM IDENTIFICATION OF A MICRO QUADROTOR HELICOPTER FROM CLOSED LOOP DATA

Derek Scott Miller, Master of Science, 2011

Thesis directed by: Professor J. Sean Humbert
Department of Aerospace Engineering

Quadrotors are a favorite platform amongst academic researchers. Yet there is limited work present on the identification of Linear Time Invariant (LTI) state space models for quadrotors. This thesis focuses on the development of a 70 gram quadrotor with a maximum dimension of less than 6 inches. An MAV this small is extremely agile and in this case, dynamically unstable. Therefore it requires an active control scheme to stabilize the vehicle's attitude. To develop an effective controller using the vast array of available linear control tools, model knowledge is required. The scope of this thesis is to identify an LTI state space model of the 70 gram quadrotor mentioned about hover. Because the quadrotor is unstable, it cannot be flown without some form of feedback control present. To determine the bare airframe dynamics of a closed loop system, the feedback gains must be turned down as low as possible so as to not distort the natural response. Then the bare airframe dynamics can be calculated from the identified closed loop dynamics if the feedback gains and control law are known. Using the derived open loop system this thesis presents options for controller development for both a stabilizing inner loop, and a station keeping outer loop controller.

OPEN LOOP SYSTEM IDENTIFICATION OF A MICRO QUADROTOR HELICOPTER FROM CLOSED LOOP DATA

by

Derek Scott Miller

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2011

Advisory Committee:
Professor J. Sean Humbert Chair/Advisor
Professor Robert M. Sanner
Professor Inderjit Chopra

Acknowledgments

I'll do my best and attempt to fit all those to whom I owe my gratitude, however I know I am likely to overlook someone, and for that I apologize.

First I would like to thank Dr. Humbert for granting me the opportunity to study in his lab. Without the equipment available in the Autonomous Vehicle Lab, this process would have been vastly more difficult. Perhaps even more helpful than the facilities in the AVL are the fellow grad students. Thank you to Greg Gremillion for advice, assistance, and comic relief throughout the entire project. Thanks to Badri Ranganathan, if it weren't for your help I would probably still be trying to make sense of the firmware. Doug Szczublewski, thank you for all the LabVIEW advice and for continuously reminding me I'm a ginger. Joseph Conroy - even though you're no longer with the lab and were never directly attached to this project, the help you provided has been a major impetus behind the progress of this project (as with most projects in the AVL). And of course I owe a great deal of thanks to the rest of the lab. Steve Gerardi, Ken MacFarlane, Nick Kostreski, Renee Campbell - without all your help, I would likely be retaking a few courses. Specifically Steve, thank you for always reminding me that things could be worse. I could be you (just kidding, I'm jealous of your moustache, and I do like spaghetti).

I also must acknowledge several others outside of the lab. Paul Samuel, thank you for all the vehicle development and overall guidance throughout this project. I don't think the Microquad would be around without your electronics knowledge, soldering skills, and Solidworks prowess. I have to acknowledge Ankur Mehta at University of California, Berkeley too. Thank you for your coding expertise and your patience with my lack thereof. I would also like to thank the Army Research Lab MAST CTA for funding this project.

The faculty at the University of Maryland must be acknowledged as well. They are some of the most brilliant minds I have ever encountered, and I don't feel I could have received a better education elsewhere. More specifically thank you to Dr. Sanner and Dr. Chopra for agreeing to be on my committee. I must also thank Dr. Morelli. I wish I could have taken your system identification course, but thank you for fielding my questions regarding SIDPAC without any

obligation; your assistance was extraordinarily useful.

My family also deserves a great deal of thanks. Thank you for being so supportive throughout my whole life, and more recently for your understanding of my schedule, and my lack of calling as of late. I promise one day I'll be able to take vacations and days off. Thank you to my girlfriend throughout this process, Rachel. Thanks for being patient through this whole ordeal, and being understanding when I've been glued to my laptop. Also thanks to Chris Kennedy for offering a fresh pair of eyes. Your revisions are greatly appreciated. Finally I have to thank God for all the opportunities he has presented me which have finally led me to this point.

Table of Contents

List of Tables

List of Figures

Nomenclature

$\mathbf{A}$	Dynamics Matrix
$\mathbf{A}_{CL}$	Closed Loop Dynamics Matrix
A_c	Force Surface/Capture Area
$\mathbf{B}$	Controls Matrix
b	Motor Thrust Factor
$\mathbf{C}$	Sensor Matrix
$C_{L\alpha}$	Lift Curve Slope
d	Motor Torque Drag Factor
$J(\cdot)$	Cost Function
$l_{[\cdot]}$	State Moment Arm
L_1	Roll Moment Arm
L_2	Pitch Moment Arm
$\mathbf{M}$	Fisher Information Matrix
p	Body Roll Rate
q	Body Pitch Rate
$Q_{[\cdot]}$	Dynamic Pressure About State
r	Body Yaw Rate
$\mathbf{S}$	Sensitivity Matrix
$\vec{u}$	Control Signal Vector
u	Body x-axis Translational Rate
v	Body y-axis Translational Rate
w	Body z-axis Translational Rate
$\vec{x}$	Dynamic State Vector
$\vec{y}$	Output Vector
α	Aerodynamic Angle of Attack
θ	Pitch Angle
$\hat{\boldsymbol{\theta}}$	Parameter Estimate Vector
λ	Rotor Inflow Ratio
ρ	Density of Air
σ	Rotor Solidity
ϕ	Roll Angle
ψ	Yaw/Heading Angle
ω_i	ith Rotor Rotation Speed
ω_r	Residual RPM of Rotor Ensemble
Ω	Helicopter Rotor Rotation Speed

	Abbreviations
CW	Clockwise
CCW	Counter Clockwise
CG	Center of Gravity
COTS	Commercial Off The Shelf
DOF	Degrees of Freedom
ESC	Electronic Speed Controller
GPS	Global Positioning System
IGE	In Ground-Effect
IMU	Inertial Measurement Unit
LHP	Left Half Plane
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MAV	Micro Air Vehicle
MBC	Model-Based Controller
MEMS	Microelectromechanical Systems
MIMO	Multiple Input Multiple Output
OGE	Out of Ground-Effect
PD	Proportional, Differential
PID	Proportional, Integral, Differential
PWM	Pulse Width Modulation
RC	Remote Controlled
RFC	Reference Flight Condition
RHP	Right Half Plane
RP	Rapid Prototype
RPM	Rotations per Minute
SISO	Single Input Single Output
SMR	Single Main Rotor
SWaP	Size Weight and Power
UAV	Uninhabited Aerial Vehicle
VTOL	Vertical Take-off and Landing

Chapter 1

Introduction

Unmanned aerial vehicles have long been considered the future of military operations due to cost effectiveness, reconnaissance capabilities, and their ability to reduce civilian casualties on the battlefield. As advancements in technology allow for smaller and lighter electronics, we come ever closer to making that future a reality. With UAVs in the forefront, extensive research has been performed on them, bringing to light their various classes and prototypes. Of particular interest are micro aerial vehicles (MAVs) due to the vast majority of applications for which they are suited. Their inherent agility and vastly reduced size allow them to maneuver in tight spaces and makes them extremely stealthy, thus appealing to reconnaissance, surveillance, and tracking operations. Additionally, MAVs offer cost-effective solutions for radiation and chemical hazard detection, environment mapping, and hostile targeting. More importantly, MAVs will have the ability to assist soldiers in battle and possibly remove them from danger altogether.

Inhibiting MAV development though, is a myriad of obstacles. MAVs have low moments of inertia as a result of their small mass. Consequently, their dynamics are much quicker than those of larger aerial vehicles, resulting in a need for a high bandwidth. While advancing technology has granted smaller electronics, small electrical components have notoriously poor robustness to latency. Therefore loop closure rates, and delays can often bottleneck the performance of MAVs. These delays can create pilot induced oscillations (PIOs), effectively destabilizing the platform. This trait requires the vehicle have an active control system implemented. Also innate to MAVs is a lack of robustness to process disturbances such as wind gusts. It is this attribute which has fueled the research presented in this paper. In order to develop an active control scheme to combat these deficiencies in MAVs, a good mathematical model must be present to leverage the dynamics of the platform and enhance its performance.

Perhaps another layer deeper into the hierarchy of UAVs, are the MAVS that are vertical take-off and landing (VTOL) capable. The appeal of VTOL MAVs is the ability to hover. The majority of an MAV's potential applications require the capability of hovering. A comparison of several different MAV configurations can be seen in Table 1.1 from [1].

Configuration	Advantages	Disadvantages
Fixed-Wing	Simple Mechanics, Silent Operation	No Hovering
Single Rotor Heli	Good Controllability, Good Maneuverability	Complex Mechanics, Large Rotor and Tail Boom
Coaxial Rotor Heli	Simple Mechanics, Compactness	Complex Aerodynamics
Tandem Rotor Heli	Good Controllability, Simple Aerodynamics	Complex Mechanics, Large Size
Quadrotor	Good Maneuverability, Simple Mechanics, Increased Payload	High Energy Consumption, Large Size
Bird-Like (Ornithopter)	Good Maneuverability, Compactness, Low Energy Consumption	Complex Control, Complex Mechanics, No Hovering
Insect-Like (Flapping Wing)	Good Maneuverability, Compactness	Complex Control, Complex Mechanics

Table 1.1: MAV Configuration Comparison [1]

Amongst the configurations that are VTOL capable, there are many differences in common flight performance characteristics. Table 1.2 from [1] provides a cursory comparison of common qualitative flight performance metrics between those configurations from Table 1.1 that are capable of hovering.

	A[1]	B[2]	C[3]	D[4]	E[5]
Power Cost	2	2	2	1	3
Control Cost	1	4	2	3	1
Payload/Volume	2	4	3	3	1
Maneuverability	4	2	2	3	3
Mechanics Simplicity	1	3	1	4	1
Aerodynamics Complexity	1	1	1	4	1
Low Speed Flight	4	4	3	4	2
High Speed Flight	2	1	2	3	3
Miniaturization	2	4	2	3	4
Survivability	1	3	1	1	3
Stationary Flight	4	4	4	4	2
Total	24	32	23	33	24

Table 1.2: Hovering Flight Performance Comparison (1 = Bad, 4 = Very good) [1]

[1] A = Single Main Rotor
[2] B = Coaxial Rotor
[3] C = Tandem Rotor
[4] D = Quadrotor
[5] E = Insect-Like

From this analysis there are two clear candidates for a VTOL MAV testbed based on simplicity and performance. The coaxial helicopter and the quadrotor are the two with the best scores based on the criterion highlighted in Table 1.2. There are several other key differences between the two platforms. The quadrotor is dynamically unstable [1], while the coaxial helicopter is inherently stable [2]. Therefore a quadrotor must have an active controller to be pilotable. However it is this instability in the quadrotor that makes it more maneuverable and more desirable for controller design.

There are two configurations for a quadrotor, the most commonly used for MAVs is the "+" or cross configuration. In this scenario the vehicle uses the fore rotor and the aft rotor to create a pitching moment, while using the port and starboard rotors to create a rolling moment. In order to create a yawing moment the quadrotor must increase its two counterclockwise rotors while simultaneously decreasing its two clockwise rotors to yaw to the right (clockwise). A simplistic depiction of a cross configuration quadrotor's mechanics can be seen in Figure 1.1 where the relative rotation speed of the rotors is proportional to the thickness of the arrows.

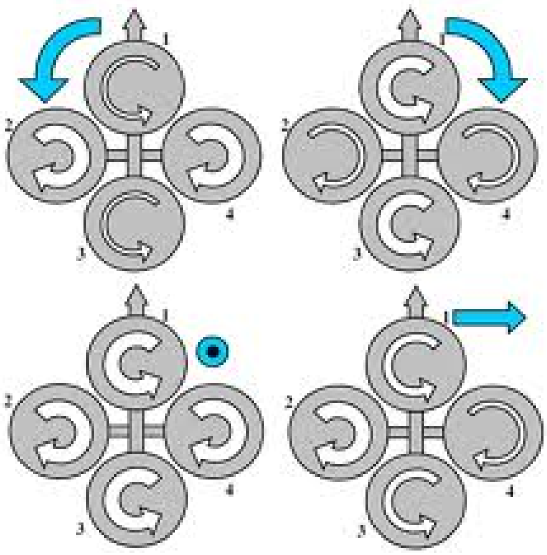


Figure 1.1: Mechanics of a Cross Configuration Quadrotor

The other quadrotor configuration less commonly used is the “x” configuration. The benefit of this arrangement over the cross configuration is that instead of only incorporating two out of four rotors in the pitch and roll motions the “x” configuration uses all four rotors, increasing the control authority of the vehicle, and therefore increasing the mobility. Naturally, this inclusion of all the rotors in each motion complicates the mixing of the rotors. To pitch forward, the front two rotors decrease their RPM, while the aft two increase their RPM to balance thrust, but create a pitching moment. The roll is done similarly, but with an imbalance in the RPM of the two starboard and two port rotors. Figure 1.2 shows the “x” configuration’s equivalent mechanics to that in Figure 1.1.

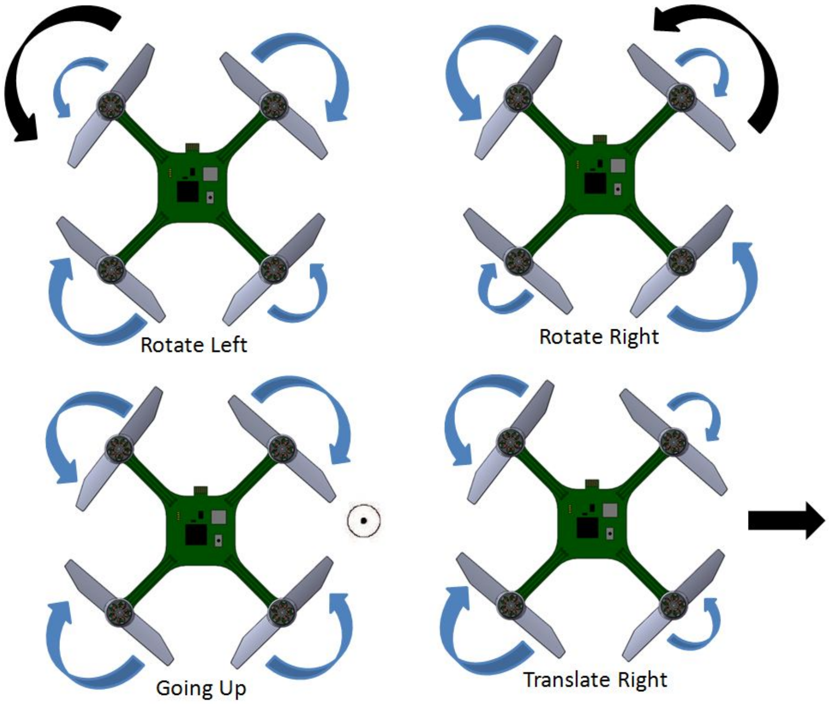


Figure 1.2: Mechanics of an “x” Configuration Quadrotor

It is for all the above reasons that the quadrotor is the choice for many labs as a research testbed. There is a larger payload capacity, allowing researchers to augment the vehicle with state of the art sensors and hardware. The simple mechanics allow for accurate modeling of the aircraft, and therefore powerful model-based controllers (MBCs) can be wielded to harness the agility and maneuverability of these dynamically unstable vehicles. There is also a simple structure to the vehicle itself, not calling for any delicate linkages. Another benefit the quadrotor has over other rotorcraft is that the rotors are small enough in diameter that blade flapping is negligible. As a corollary to the small size of the rotors, they have small moments of inertia. This has a role to play in the gyroscopic effects sometimes seen in other rotorcraft. With a small inertia of the rotor blade the gyroscopic effects can effectively be ignored as well. Therefore, a quadrotor is a good candidate for a VTOL MAV. As it is likely to be the easiest to obtain an accurate mathematical model, which is the keystone to implementing some of the more powerful controllers to realize the true potential of these MAVs.

1.1 Motivation

The popularity of MAVs has led to the development of a great deal of cutting edge technology to support these small vehicles. The state of the art MAVs have aggressive MBCs to enhance their performance. As the name suggests a model-based controller requires a model to synthesize the controller. System identification, the impetus behind the research presented in this thesis, is an effective method for estimating the model. There is a wealth of research done on modeling and identifying MAVs. Even a considerable effort has been distributed to the system identification of other rotorcraft and micro rotorcraft too.

Quadrotors are included in this modeling and identification work done on MAVS, as they are very popular in many different research fields for reasons mentioned in the previous section. Yet the modeling effort available in the work mentioned is not presented in the linear time invariant (LTI) state space format, and it is usually done by measuring and recording the innumerable parameters of the vehicles. Therefore, this research offers a unique approach (time domain system identification) to identifying a state space model for a micro scale quadrotor. The only other resource found at the time of this paper that presents a system identification procedure completed on a quadrotor platform is reference [3]. This project and paper differ too because the vehicle in focus is truly a micro-scale vehicle. This definition is thrown around a lot with a very loose definition of micro-scale. Some vehicles referred to as MAVs are often over 1 kg in mass. The quadrotor studied in this paper is an order of magnitude smaller, with a maximum mass of 70 g. A size that offers many advantages over larger vehicles, yet also imposes many new challenges still to be conquered.

When vehicles get to this scale, there are even more opportunities and applications because the vehicle has a very particular set of advantages over other platforms. The stealth of the vehicle increases. The maneuverability increases as well; along with the agility. The MAV can inspect small crevices where larger vehicles wouldn't fit. Another benefit that is heavily weighted for military applications is the low mass and small size. This creates the potential for these vehicles to be carried around by soldiers with limited equipment, and it is envisioned that they can be deployed and operated autonomously, minimizing the restrictions on the soldier.

While these advantages make MAVs on this scale (< 100 g) very attractive, there are still many obstacles to overcome to realize their full potential. One major disadvantage to vehicles this small is the available hardware. Whether it's a function of the power required for the electronics or the size themselves, the electronics available to small vehicles can induce latency; or if they are sensors, they are highly susceptible to measurement noise. This limitation in electronics poses a problem because the agility of these vehicles requires a high loop-closure rate. Therefore, the electronics themselves can destabilize a stable system by inducing phase shifts that manifest unstable oscillations. Another effect inherent to small MAVs is their lack of robustness to disturbances. Small gusts are capable of quickly destabilizing a vehicle of this size.

In order to combat these effects, a robust controller must be implemented. Yet in order to synthesize such a controller a model of the system must also be known, or at least obtainable. The work in this thesis presents a method for obtaining a bare airframe model of the system from data collected from closed loop flights. As mentioned, quadrotors are unstable and require feedback control for a human to be capable of piloting. Therefore, it is impossible to fly the vehicle open loop, and hence a bare airframe model cannot be attained from raw flight data. Upon determining this all-encompassing model, a controller can be formulated based upon performance and robustness criterion. This thesis will also provide several options for control along with simulated results for these algorithms.

Chapter 2

Quadrotor Modeling

The quadrotor is no different from most rotorcraft in the sense that it is a very nonlinear system despite its simple mechanics. It can be considered, in fact even more nonlinear than other rotorcraft, because there are four rotors where other rotorcraft have 1 or two rotors at most, thus increasing the complexity of the aerodynamics. It is also important to note that for MAVs the rotors are on a much smaller scale, which results in a very small Reynolds number. This attribute adds an extra complex element, as the Reynolds number may be up to three orders of magnitude smaller than those witnessed on full-scale helicopters. At this scale the flow fields experienced by the rotor are vastly different, and relatively unfamiliar when compared with that of a conventional helicopter. Flow visualization experiments have yielded results indicating a large conical wake at the center of the rotor, suggesting a more turbulent flow over the rotor than typically seen in the wake of larger rotors [4]. It's important to look at the fundamentals of how the quadrotor behaves, and develop the full nonlinear system in order to better comprehend the platform.

2.1 Dynamic Modeling

2.1.1 Kinematics

A full system model includes countless effects due to not just the aerodynamics but gravity, kinematics, and other effects such as gyroscopic effects. The difficulty in effectively modeling all these effects is that many of them are best represented in different coordinate frames. The two frames, which can be seen in Figure 2.1, used are the earth-fixed frame, sometimes called the gravity frame, or inertial frame, $\boldsymbol{G} = \{G_x, G_y, G_z\}$. The other frame is the body-fixed frame, the coordinate axes fixed to the center of gravity of the vehicle, $\boldsymbol{B} = \{\hat{e}_x, \hat{e}_y, \hat{e}_z\}$.

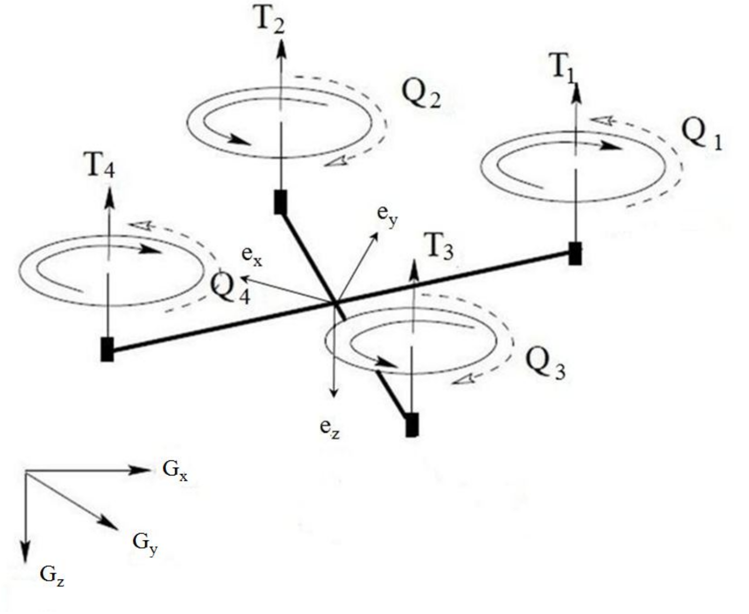


Figure 2.1: Micro Quadrotor Mechanics and Coordinate Frames

The gravity frame is useful because as its name suggests, the gravity force does not change, it is always acting along the G_z axis. This is also the frame that the location of the vehicle is best measured $[x, y, z]^T$. The translational velocities of the vehicle $[u, v, w]^T$ and the rotational velocities $[p, q, r]^T$ are typically reported in the body axes. In order to get from one frame to the other the three angles used are known as Euler angles $[\phi, \theta, \psi]^T$. In this model, the rotation sequence is known as 3-2-1, where the first rotation is about the $\hat{e}_z$ axis by $\psi°$, and the second rotation is about the $\hat{e}_y$ axis by $\theta°$, and finally the third rotation about the $\hat{e}_x$ axis by $\phi°$.

When going from one frame to the other, there is an explicit relationship depending on the Euler angles that can be represented as $\vec{r}_B = [T_{BG}]\vec{r}_G$. Here $\vec{r}_G$ is a position $[x, y, z]^T$ in the gravity axes, and $\vec{r}_B$ is in the body axes. The matrix $[T_{BG}]$ is known as a rotation matrix, a function of the Euler angles. A rotation matrix is an operator that maps one frame to another $[T_{BG}] : \boldsymbol{G} \rightarrow \boldsymbol{B}$. It is known that every orientation of $\boldsymbol{B}$ can be expressed with respect to $\boldsymbol{G}$ as an element of $SO(3)$, in other words $[T_{BG}] \in SO(3)$. The $SO(3)$ group is a lie subgroup of the more broad general linear group of invertible matrices. Being an element of this group grants several notable properties.

$$SO(3) = \left\{[T_{BG}] \in \mathbb{R}^{3x3}, [T_{BG}]^T[T_{BG}] = \mathbb{I}_3, Det([T_{BG}]) = +1\right\} \tag{2.1}$$

For the ψ, θ, ϕ (3-2-1) rotation sequence the rotation matrix to get the body axes with respect to the gravity axes is seen below in Equation 2.2 (from [5]).

$$[T_{BG}] = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2.2}$$

From the second property of the $SO(3)$ group it can be shown that the transpose of the rotation matrix is equivalent to the inverse. This leads to the relationship that the representation of the gravity frame with respect to the body frame is the transpose of the rotation matrix to represent the body frame with respect to the gravity frame, i.e. $[T_{GB}] = [T_{BG}]^T$. Using this relationship we obtain what are commonly known as the navigation equations, our first set of equations defining the motion of the vehicle.

$$\frac{dx}{dt} = (\cos\theta\cos\psi)u + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)v + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)w \tag{2.3}$$

$$\frac{dy}{dt} = (\cos\theta\sin\psi)u + (\sin\phi\sin\theta\sin\psi - \cos\phi\cos\psi)v + (\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)w \tag{2.4}$$

$$\frac{dz}{dt} = (-\sin\theta)u + (\sin\phi\cos\theta)v + (\cos\phi\cos\theta)w \tag{2.5}$$

The second set of equations is also from the attitude kinematics. The relationship between the Euler rates and the body rates, is derived through a geometric analysis of the process of going from one frame to the other in the 3-2-1 sequence. The derivation can be found in [6].

$$p = \dot{\phi} - \dot{\psi}\sin\theta \tag{2.6}$$

$$q = \dot{\theta}\cos\phi - \dot{\psi}\cos\theta\sin\phi \tag{2.7}$$

$$r = \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \tag{2.8}$$

2.1.2 Dynamics

In order to obtain the rest of the equations of motion and develop a dynamic model, it is necessary to start with Newton's laws. First for translational dynamics, the common formula of $F = ma$ can be more formally expressed as

$$\vec{F} =^F \frac{d}{dt}(\vec{p}^B). \tag{2.9}$$

Where $\vec{p}^B = m\vec{v}^B$ is the momentum of the airframe, and $\vec{F}$ is an expression of the total forces applied to the CG of the vehicle. There exists a very similar expression for the angular dynamics as well.

$$\vec{M}_G =^F \frac{d}{dt}(\vec{H}_G^B) \tag{2.10}$$

$\vec{M}_G$ are the total moments applied about the CG of the body. $\vec{H}_G^B = I_G^B \, {}^F\vec{\omega}^B$ is the angular momentum of the body. An important aspect to these two laws is that the derivatives are total derivatives of vectors in different frames. This leads to a number of effects best explained through Reynold's Transport Theorem (Equation 2.11).

$${}^F\frac{d}{dt}(\cdot) =^B \frac{d}{dt}(\cdot) +^F \vec{\omega}^B \times (\cdot) \tag{2.11}$$

This means that there are two contributions to the vector derivative. The first is a rate of change due to the movement of the body frame, and the second is due to the direction change from the rotation of the body frame.

When all the derivatives and cross products are carried out, the 6 equations of motion are derived in the body frame [7]. The force equilibrium equations from $\vec{F} =^F \frac{d}{dt}(m\vec{v}^B)$ where

$\vec{F} = [X, Y, Z]^T$ breaks down into 3 separate equations.

$$m(\dot{u} + qw - rv) + mg\sin\theta = X \tag{2.12}$$

$$m(\dot{v} + ru - pw) - mg\sin\phi\cos\theta = Y \tag{2.13}$$

$$m(\dot{w} + pv - qu) - mg\cos\phi\cos\theta = Z \tag{2.14}$$

The moment equilibrium equations from $\vec{M}_G =^F \frac{d}{dt}(I_G^B\, {}^F\vec{\omega}^B)$ are

$$I_{xx}\dot{p} - (I_{yy} - I_{zz})qr + I_{yz}(r^2 - q^2) - I_{xz}(pq + \dot{r}) + I_{xy}(pr - \dot{q}) = L \tag{2.15}$$

$$I_{yy}\dot{q} - (I_{zz} - I_{xx})pr + I_{xz}(p^2 - r^2) - I_{xy}(qr + \dot{p}) + I_{yz}(pq - \dot{r}) = M \tag{2.16}$$

$$I_{zz}\dot{r} - (I_{xx} - I_{yy})pq + I_{xy}(q^2 - p^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) = N \tag{2.17}$$

The difficulty in analyzing these equations is with the right hand side. The moments and forces are a combination of a host of various physical and aerodynamic effects.

The above 6 equations, along with the 6 equations from the kinematics are the basis for a full system model. However, for the remainder of this section, the focus is on the inner loop of the vehicle, and therefore the navigation equations will be ignored, until the outer loop is investigated. In order to obtain the inner loop model, the applied moments and forces must be elaborated. As the model to be derived is a hover model, the rotors are to be assumed out of ground effect, and therefore that is one less complex source.

The first source to account for these forces and moments is that of the rotors. The rotors provide a thrust along the $-\hat{e}_z$ axis, they also create pitch, roll, and yaw moments. From Blade Element Theory, presented in Chapter 3 of [4], both the thrust and the torque created by a rotor with a linearly twisted blade are proportional to the rotor rotation speed squared ($T, Q \propto \omega^2$). Here a simple constant will lump the rest of the parameters in the full equations, so $T_i = b\omega_i^2$ and $Q_i = d\omega_i^2$. These will be the physical controls, as these are the inputs to retain control of the vehicle.

$$Z_c = \delta_{thr} = -b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{2.18}$$

$$N_c = \delta_{yaw} = d(\omega_2^2 + \omega_3^2 - \omega_1^2 - \omega_4^2) \tag{2.19}$$

$$L_c = \delta_{lat} = L_1 b(\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2) \tag{2.20}$$

$$M_c = \delta_{lon} = L_2 b(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \tag{2.21}$$

As the rotors are spinning at speeds faster than 10,000 RPM, it's important to investigate gyroscopic torques due to the rotations in different frames. The gyroscopic moments from the Coriolis effect are found from Equation 2.22 ([8])

$${}^B[M_i]_{gyro} = \vec{I_r} \cdot \dot{\vec{\omega_i}} +^F \vec{\omega}^B \times (\vec{I_r} \cdot \dot{\vec{\omega_i}}) \tag{2.22}$$

Assuming the rotor is symmetric, and that the rotor rotates along the $\hat{e}_z$ axis then the inertia $\vec{I_r}$ is just the inertia about the rotation axis, $I_r = I_z$, then this simplifies down to a function of the residual RPM of the 4 rotors.

$$L_{gyro} = I_r q \omega_r \tag{2.23}$$

$$M_{gryo} = -I_r p \omega_r \tag{2.24}$$

$$N_{gyro} = I_r r \dot{\omega}_r \tag{2.25}$$

Where ω_r is the CCW residual RPM, $\omega_r = \omega_2 + \omega_3 - \omega_1 - \omega_4$.

The rest of the significant Forces and moments are a result of the aerodynamics of the rigid body. However, these are a result of complex flows across complicated geometries. It is therefore best to just treat them as a function of nondimensional force and moment coefficients $C_{[\cdot]}$, and dynamic pressure of the state inspected $Q_{[\cdot]}$ which tends to be a function of the state itself and the density of the air ρ. Therefore the aerodynamic force and moment contributions are

$$X_{aero} = Q_u A_c C_X \tag{2.26}$$

$$Y_{aero} = Q_v A_c C_Y \tag{2.27}$$

$$Z_{aero} = Q_w A_c C_Z \tag{2.28}$$

$$L_{aero} = Q_p A_c C_L l_p \tag{2.29}$$

$$M_{aero} = Q_q A_c C_M l_q \tag{2.30}$$

$$N_{aero} = Q_r A_c C_N l_r \tag{2.31}$$

Combining all these effects the full derived nonlinear inner loop dynamic model is

$$p = \dot{\phi} - \dot{\psi} \sin\theta \tag{2.32}$$

$$q = \dot{\theta} - \dot{\psi} \cos\theta \sin\phi \tag{2.33}$$

$$r = \dot{\psi} \cos\theta \cos\phi - \dot{\theta} \sin\phi \tag{2.34}$$

$$m(\dot{u} + qw - rv) + mg \sin\theta = -Q_u A_c C_X \tag{2.35}$$

$$m(\dot{v} + ru - pw) - mg \sin\phi \cos\theta = -Q_v A_c C_Y \tag{2.36}$$

$$m(\dot{w} + pv - qu) - mg \cos\phi \cos\theta = -Q_w A_c C_Z + \delta_{thr} \tag{2.37}$$

$$I_{xx}\dot{p} - (I_{yy} - I_{zz})qr + I_{yz}(r^2 - q^2) - I_{xz}(pq + \dot{r}) + I_{xy}(pr - \dot{q}) = -Q_p A_c C_L l_p + I_r q \omega_r + \delta_{lat} \tag{2.38}$$

$$I_{yy}\dot{q} - (I_{zz} - I_{xx})pr + I_{xz}(p^2 - r^2) - I_{xy}(qr + \dot{p}) + I_{yz}(pq - \dot{r}) = -Q_q A_c C_M l_q - I_r p \omega_r + \delta_{lon} \tag{2.39}$$

$$I_{zz}\dot{r} - (I_{xx} - I_{yy})pq + I_{xy}(q^2 - p^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) = -Q_r A_c C_N l_r + I_r r \dot{\omega}_r + \delta_{yaw} \tag{2.40}$$

While the quadrotor aircraft is actually a pretty simple platform when compared to other rotorcraft, this is still a very complex model with a large number of parameters that are difficult to attain. Fortunately blade flapping can be ignored as the rotors' radius is small enough that a semi-rigid material (like that used) won't bend significantly under the thrust created. This avoids a potentially much more complex model. Since there are four double-bladed rotors, the incorporation of blade flapping in the model could add another 8 states.

2.2 Linearized Model

The full nonlinear model is very useful, as it provides insight into the behavior of the vehicle. However, it is not as powerful as a linear model. There is an abundance of well-studied tools available to enhance a linear system. Therefore, the next step is to simplify the full nonlinear equations of motion and then to linearize it. There are many assumptions to be made that will accomplish this. The first of which simplifies the three moment equilibrium equations. It is common that the x-z plane of an aircraft be symmetrical or nearly symmetrical. Another benefit of a quadrotor is the symmetry of the planform about its planes. Therefore, the y-z plane is also assumed nearly symmetrical. This leads to negligible values for all inertia products I_{xy}, I_{xz}, and I_{yz}. Hence the moment equilibrium equations become

$$I_{xx}\dot{p} - (I_{yy} - I_{zz})qr = -Q_p A_c C_L l_p + I_r q\omega_r + \delta_{lat} \tag{2.41}$$

$$I_{yy}\dot{q} - (I_{zz} - I_{xx})pr = -Q_q A_c C_M l_q - I_r p\omega_r + \delta_{lon} \tag{2.42}$$

$$I_{zz}\dot{r} - (I_{xx} - I_{yy})pq = -Q_r A_c C_N l_r + I_r r\dot{\omega}_r + \delta_{yaw} \tag{2.43}$$

Next, recall that the model is of the quadrotor in hover. Therefore, the model will be a small perturbation model about hover. This means that all states will be represented as an equilibrium or trim value and a small perturbation from this trim value, for example $u = u_0 + \Delta u$. A small perturbation model allows two more simplifying assumptions to be made. First we can neglect products of disturbances, as they must be small for a *small* perturbation model, and a second order term of a small variable is negligible. Second, since this model is about hover, that means all the reference values are 0, except for the trim throttle command to balance the gravity force δ_{thr_0}. A third assumption is more of a simplification. The inertia of the rotors was found (through CAD modeling) to be $I_r = 4.2503 \cdot 10^{-7} kg \cdot m^2$, and the standard deviation of the rotor RPM when held at throttle required to hover is approximately 105.5 RPM. This means that at a worst case scenario $\omega_r = 44.192 rad/s$, and therefore the maximum gyroscopic moments are $L_{gryo_{max}} = 1.8783 \cdot 10^{-5} \Delta q \; N \cdot m$ and $M_{gryo_{max}} = 1.8783 \cdot 10^{-5} \Delta p \; N \cdot m$. In a small perturbation model this means that neither Δp nor Δq will be large enough to make the gyroscopic moments significant. Therefore the gyroscopic effects of the propellers can be neglected. Under these assumptions, the small perturbation model is

$$\Delta p = \Delta\dot{\phi} \tag{2.44}$$

$$\Delta q = \Delta\dot{\theta} \tag{2.45}$$

$$\Delta r = \Delta\dot{\psi} \tag{2.46}$$

$$m\Delta\dot{u} + mg\Delta\theta = \Delta X \tag{2.47}$$

$$m\Delta\dot{v} - mg\Delta\phi = \Delta Y \tag{2.48}$$

$$m\Delta\dot{w} - mg = \Delta Z + \delta_{thr_0} \tag{2.49}$$

$$I_{xx}\Delta\dot{p} = \Delta L \tag{2.50}$$

$$I_{yy}\Delta\dot{q} = \Delta M \tag{2.51}$$

$$I_{zz}\Delta\dot{r} = \Delta N \tag{2.52}$$

The disturbances in the forces and moments above must be expanded to account for the force or moment perturbations due to either the controls, or the aerodynamic effects. Since these will be expanded the control perturbation values ($\Delta\delta_{lon}, \Delta\delta_{lat}$, etc) are not included above as they will be accounted for in the expansion. The disturbances are best expanded via Taylor expansion [7]. In order to expand each force or moment perturbation, the states or controls that each depend on must be assumed. For instance for a quadrotor, the roll moment perturbation will depend on the aerodynamic damping of the roll rate, and the lateral translational velocity (airflow will increase across the two rotors leading the translation and not change across the two trailing), and of course the control input. Another, potential state affecting the roll moment may be the roll angle if the vehicle exhibits any natural stiffness, therefore it can be expanded as follows

$$\Delta L = \frac{\partial L}{\partial v}\Delta v + \frac{\partial L}{\partial p}\Delta p + \frac{\partial L}{\partial \phi}\Delta \phi + \frac{\partial L}{\partial \delta_{lat}}\Delta \delta_{lat} \tag{2.53}$$

The partial derivatives represent stability and control derivatives. The actual values that go into the state space matrices aren't exactly these. An example is $X_u = \frac{1}{m}\frac{\partial X}{\partial u}$.

Assuming near perfect symmetry of the quadrotor in terms of rotor and motor balancing and dimensions, the full model is actually quite decoupled. This is partly due to the lack of blade flapping, but also due to the simplicity of the mechanics of the vehicle. After all these assumptions, a structure for the dynamics and controls matrix can be determined [9]. From now on, the Δ will be dropped in front of each term, as the states are now to be considered deviations from the trim value. Therefore, the state vector is $\vec{x} = [u, v, w, p, q, r, \phi, \theta, \psi]^T$ and the control vector is $\vec{u} = [\delta_{lon}, \delta_{lat}, \delta_{yaw}, \delta_{thr}]^T$. This leads to a state space structure $\dot{\vec{x}} = A\vec{x} + B\vec{u}$ of the following dynamics and control matrices representing the bare airframe inner loop model dynamics

$$\dot{\vec{x}} = \mathbf{A}\vec{x} + \mathbf{B}\vec{u} \tag{2.54}$$

$$\dot{\vec{x}} = \begin{bmatrix} X_u & 0 & 0 & 0 & X_q & 0 & 0 & -g & 0 \\ 0 & Y_v & 0 & Y_p & 0 & 0 & g & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_v & 0 & L_p & 0 & 0 & L_\phi & 0 & 0 \\ M_u & 0 & 0 & 0 & M_q & 0 & 0 & M_\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Z_{thr} \\ 0 & L_{lat} & 0 & 0 \\ M_{lon} & 0 & 0 & 0 \\ 0 & 0 & N_{yaw} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{u} \tag{2.55}$$

The stability and control derivatives in these matrices are a complicated combination of numerous parameters not only dependent on the geometry of the vehicle, but also on the aerodynamic environment. These would be very difficult to identify accurately using traditional methods. In order to determine these parameters this paper will use a software package developed to identify these parameters using time-domain data. The software package used is known as SIDPAC (System

Identificatin Programs for Aircraft), a collection of MATLAB$^{\text{TM}}$ scripts and functions, programmed by Dr. Eugene Morelli at NASA Langley [10].

Chapter 3

Vehicle Development

3.1 Overview

As discussed in Chapter 1, there are many advantages of a quadrotor as a viable option for a VTOL capable MAV. The anticipation is that micro-scale vehicles will be able to fly autonomously and in uncertain and often times gusty environments. To be able to do so the platform must be agile, and have the control authority available to reject disturbances. However, to obtain the agility necessary a platform may often be unstable, and therefore require an active control system. The choice to develop a quadrotor over other vehicle designs was based on the simplicity of the platform. The choice of the "x" configuration was made to utilize all four motors in all motions, thus increasing control actuation, and allowing the vehicle to reject gust disturbances.

The Micro Autonomous Systems Technology (MAST) Collaborative Technology Alliance (CTA) requires a vehicle on the scale of 6 inches maximum dimension, that uses the Guidance and Inertial Navigation Assistant (GINA) mote developed at the University of California Berkely (seen in Figure 3.1) to provide sensing, communication, and control. While other more stable platforms are being investigated as well, it is thought that due to the extra available control authority, the quadrotor platform will be better suited to reject gust disturbances, which are notoriously troublesome to vehicles on this scale.

As with the development of any other vehicle the micro-scale quadrotor, more commonly known as the microquad has undergone many different vehicle design iterations. Each version incorporates a new design feature, or a different hardware option. Most components are typically COTS, which are selected based upon SWaP restrictions and chosen after trade studies comparing performance and robustness.

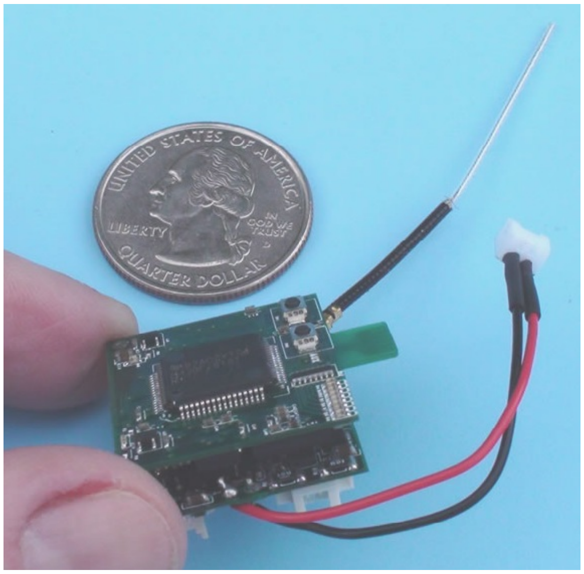


Figure 3.1: University of California Berkeley GINA Mote

3.2 1st Iteration

3.2.1 Design

The first vehicle design is a very minimalistic model. It was a first draft designed to be a proof of concept. The main objective was to demonstrate the vehicle could produce enough lift to take off and hover. The design can be seen in Figure 3.2, it is a very small vehicle that only consists of the airframe, a radio receiver, 4 ESCs and 4 motors with custom Rapid Prototyped (RP) rotors. All hardware was chosen empirically, based off of successful use and implementation

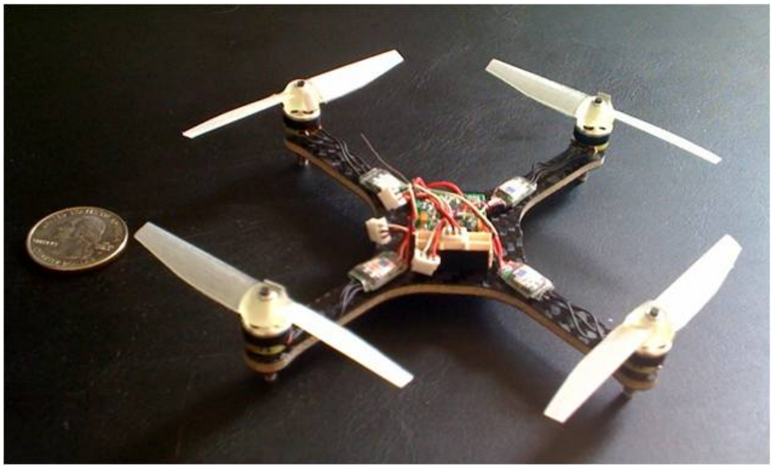


Figure 3.2: Microquad V1

in other similar projects. The ESCs were selected because they were the smallest option available (.25 grams each). The motors were picked contingent on their performance at an acceptable power level. The rotors were custom designed also as a mock-up of rotors from larger scale quadrotors, incorporating taper and twist in an attempt to improve efficiency.

3.2.2 Accomplishments

This version was developed in September 2009 and weighed only approximately 40 grams. In addition to verifying that the hardware can create enough lift, the 1^{st} version was also intended to confirm that the vehicle can be controlled with feedback. Since the design did not incorporate any sensors to provide the feedback, a Vicon$^{\text{TM}}$ visual motion tracking system (pictured in Figure 3.3) was used to provide feedback and a custom LabVIEW$^{\text{TM}}$ program was used to compute and send the commands back to the receiver, effectively controlling the vehicle.

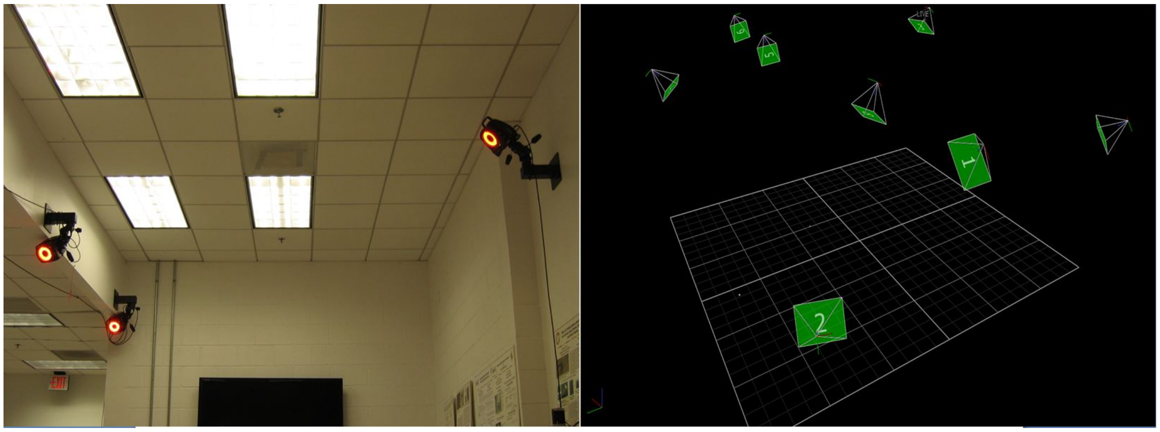


Figure 3.3: Vicon Motion Capture System Environment

The vehicle proved it could create sufficient thrust using the selected hardware. It could do so at approximately 60% commanded RPM, indicating that this design could handle an estimated payload of approximately 20 grams. The next step was to validate that the vehicle could be controlled using the Vicon$^{\text{TM}}$ and LabVIEW$^{\text{TM}}$ system. The yaw DOF was first tested as it was anticipated to be the slowest rotation, and therefore the easiest to control. The platform was secured to a yaw test stand and outfitted with Vicon$^{\text{TM}}$ markers in order to fit a model to the structure that Vicon$^{\text{TM}}$ can track. The microquad could successfully track a desired yaw angle, and reject disturbances, with only the measured yaw angle, using a simple PD control scheme. The proportional gain and differential gain were found iteratively.

After this was completed, the other two other rotational DOFs (pitch and roll) were investigated. The same procedure done with yaw was attempted. The vehicle was secured to a custom pitch and roll test stand and feedback control from Vicon$^{\mathrm{TM}}$ was attempted. However, even with the increased inertial damping added from the test stands designed to isolate roll and pitch motions, the motions could not be stabilized with the hardware integrated with the vehicle. It was apparent that the commands were being implemented with a noticeable delay. The cause of the latency was determined to be due to the ESC and motor combination. Upon testing using a hall effect sensor, magnet, and servo tester the latency between a commanded RPM change and the actual RPM change was found to be approximately 0.25 seconds. A delay of this magnitude is catastrophic for vehicles on this scale. Due to the small mass of the platform and the small dimensions, the moments of inertia are especially small. This subsequently creates very fast rotational accelerations. Therefore, within a quarter of a second, the vehicle could easily rotate far past an acceptable limit before a restoring moment created by the control is implemented. This limitation led to a new choice of hardware, implemented on the next iteration of the microquad.

3.3 2$^{\mathrm{nd}}$ Iteration

3.3.1 Design

The 2$^{\mathrm{nd}}$ design of the microquad, was incited by the requirement for new hardware, most specifically ESCs. The new ESCs were chosen based on their performance for Walkera$^{\mathrm{TM}}$ brand RC helicopters' tail rotor controllers. The tail rotor on these small RC helicopters needs to have a fine level of control with minimal latency. Therefore, these ESCs, known as Fleas were a good fit. The actual design of the airframe was changed too, seen in Figure 3.4. The rotors and motors were moved to the underside of the airframe. The idea behind this is that the wake of the rotors will be unobstructed, therefore increasing the efficiency of the rotors. Finally, the GINA 2.1 mote was also integrated to the frame. With the inclusion of the larger hardware, and implementation of the mote, this version was a fair amount heavier than the 1st version, weighing in at approximately 50 grams.

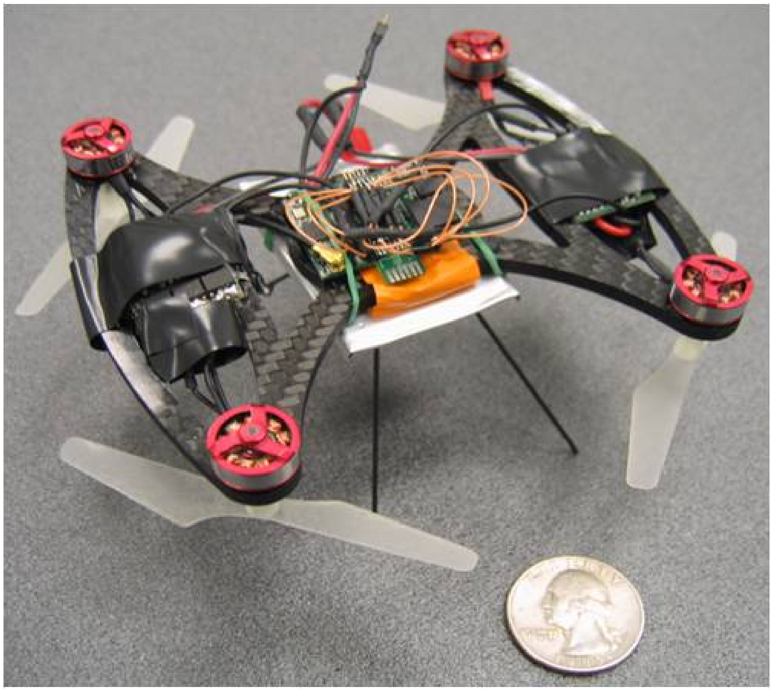


Figure 3.4: Microquad V2

3.3.2 Accomplishments

With the receipt of the mote, the development of the attitude stabilization code began with the 2nd version of the microquad. The first step was to establish successful communication with the Mote and a computer control station. This actually was a cumbersome exercise. The protocol was developed in Java$^{\mathrm{TM}}$ and had to be replicated for implementation in the LabVIEW$^{\mathrm{TM}}$ framework established. This involved determining the packet structure for communication between the GINA Mote and Berkeley's Basestation attached to a computer. This had to be an active method, as the joystick packets that were sent up to the Mote on the microquad had to be updated rapidly with new joystick commands.

After this was successfully achieved, the inner loop had to be closed. The inner loop is the alias that is commonly used to refer to the faster dynamics corresponding to the attitude of the vehicle, while the outer loop is used to refer to the slower translational dynamics. In order to close the inner loop the attitude has to be fed back to the control system. On smaller vehicles like the microquad this is typically done with a MEMS IMU package consisting of a three axis gyroscope chip that measures the rotation rates about the three body axes, and a three axis linear accelerometer chip which approximates the pitch and roll angles by measuring the gravitation acceleration along the x and y axes of the accelerometer. These two sensors were provided on the GINA 2.1 Mote that was integrated on the 2$^{\text{nd}}$ version of the microquad. The accelerometer chip in place was an STMicroelectronics$^{\text{TM}}$ LIS344ALH. The three axis gyro is actually a combination of two separate chips. The x and y axes rotation rates were measured with the InvenSense$^{\text{TM}}$ IDG-650 dual-axis gyroscope, while the z axis rotation rate is measured from an STMicroelectronics$^{\text{TM}}$ LISY300AL single axis gyroscope.

The IDG-650 chip works well at estimating the pitch and roll rates p and q respectively. The measurements are relatively clean and have minimal noise present. However, the other three necessary measurements are much more susceptible to sensor noise. The underlying cause is a common problem experienced with micro rotorcraft. The operation of the motor or motors creates substantial vibrations which manifest themselves as horizontal accelerations that corrupt the measurements of the accelerometer. The motor vibrations also degrade the measurement of the yaw rate r due to the cross coupling of the linear accelerations about the x-y plane of the vehicle.

The common solution to this problem is usually some method of filtering the measurements. However, as can be seen in Figure 3.5, the noise is considerable, too much to be filtered out with a common filter such as a Low Pass filter.

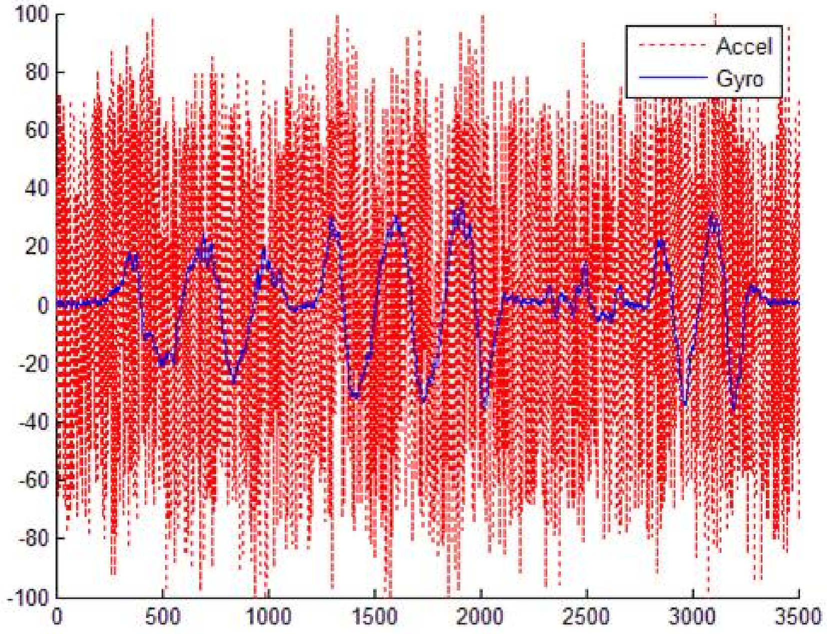


Figure 3.5: Measurement Noise from Motor Vibrations at 50% Commanded RPM

Therefore a different method to refine the measurements of the roll and pitch angles ϕ and θ respectively, had to be used. However, due to the limitations in processing power on vehicles of this size, the number of filtering options is also limited. On a fixed precision processor, like the Texas Instruments$^{\text{TM}}$ MSP430 microcontroller on the Mote, this also limits the precision of the filtered measurement. With the available sensors there are essentially three options. The first is to low pass filter the accelerometer measurements for the angles and use the gyro measurements for the rates. The second option is to use just the gyro, to use the direct measurements for the rates, and to integrate the rates to back out the angles. While this is clean and relatively free of noise, the consequence is the drift incurred from integrating the rate. The last is what is known as a complementary filter, where one sensor complements the other and the benefits of both are utilized. The qualitative process of it is that the integrated rate is passed through a high-pass filter, to remove the low frequency drift. While the accelerometer measurements are passed through a low pass filter to remove some of the high frequency noise [11]. These two separate filtered measurements are combined in an appropriately scaled manner to create a cleaner estimate of the roll and pitch angles ϕ and θ. While this sounds rather difficult, it is actually quite simple to implement in code as shown below.

```
angle=f*(angle+(gyro*dt)/sfgyro)+(1-f)*(accel/sfaccel);
```

Where f is the filter cutoff ratio f $\in [0, 1]$, and $sfaccel$ is the accelerometer scale factor, and $sfgyro$ is the gyro scale factor. The implementation of this on the measurements from Figure 3.5 is seen in Figure 3.6.

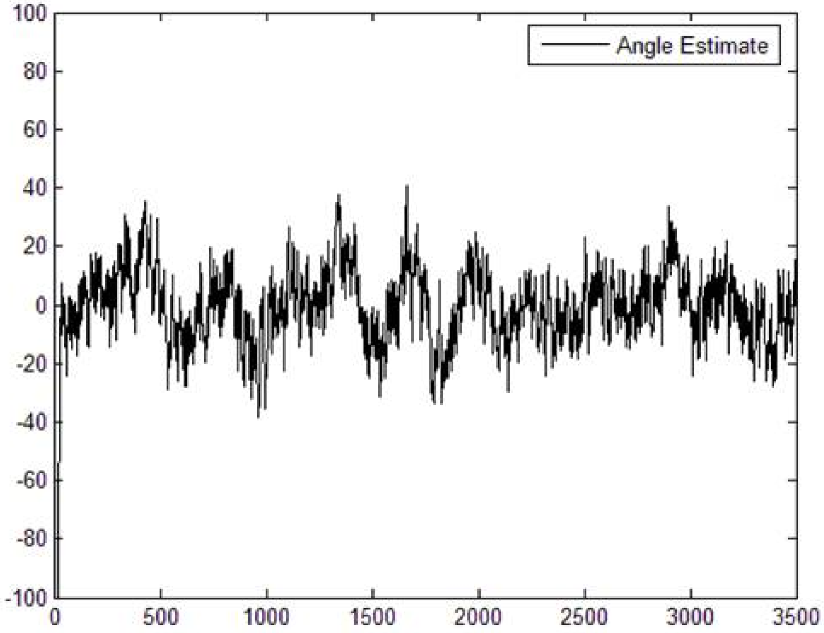


Figure 3.6: Complementary Filtered Angle Estimate at 50% Commanded RPM

This is a clear improvement over a low pass filter alone on the accelerometer, or a high pass filter alone on the gyro measurements. While this is a drastic improvement, the noise is still too abundant to effectively implement attitude feedback and close the inner loop. This led to searching for other solutions to the motor vibration problem. Since it was clear that filtering alone wasn't enough, mechanical isolation methods were investigated. This led to a new design and the 3$^{\text{rd}}$ version of the microquad.

3.4 3$^{\text{rd}}$ Iteration

3.4.1 Design

A major influence behind the form factor of the 3$^{\text{rd}}$ design of the microquad is reducing the magnitude of the motor vibration noise present in the accelerometer measurements. The modular design of the 3$^{\text{rd}}$ version (Figure 3.7) allows for the addition of vibration damping material in between the structure. The motors were mounted in such a way to avoid direct contact with the airframe. They were mounted with with an absorptive material preventing any contact of the motor, or the motor mount, or even the mounting screws with the airframe structure (Figure 3.8).

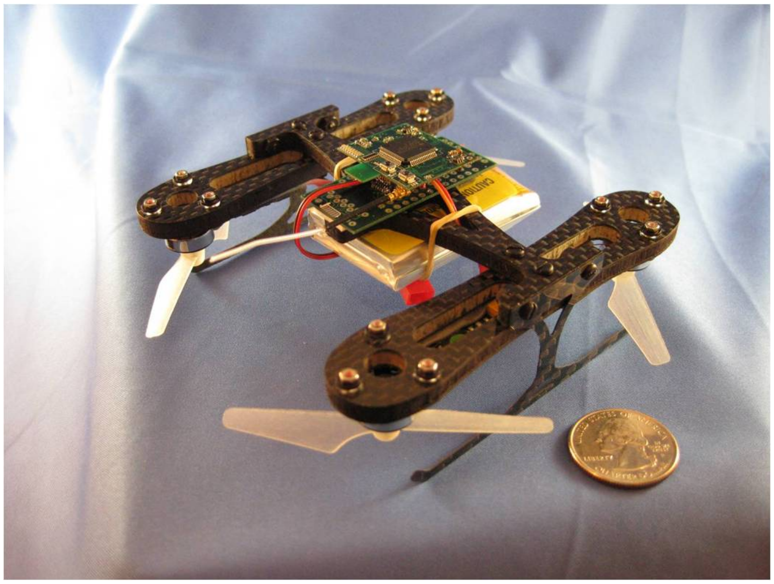


Figure 3.7: Microquad V3

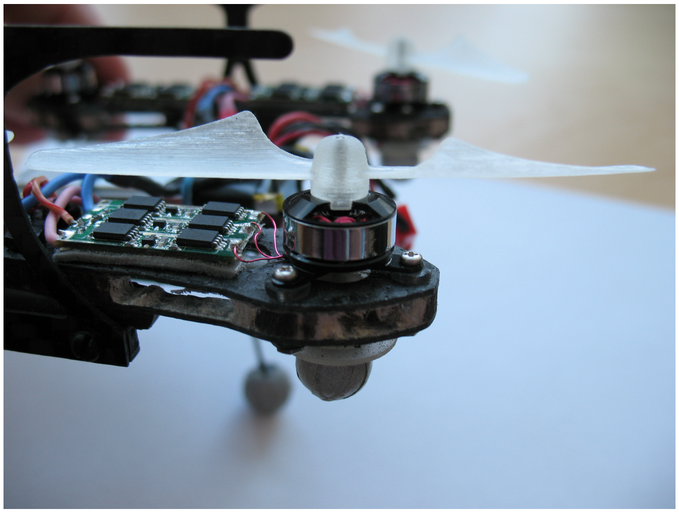


Figure 3.8: Mechanical Isolation of Motors from Airframe

As mentioned earlier, the modular design permits the introduction of vibration damping material between the two faces of the structure. The material used is a visco elastic polymer Sorbothane^{TM} . It is commonly used in gel insoles for reducing discomfort while walking. The inclusion of the Sorbothane^{TM} material is seen in Figure 3.9.

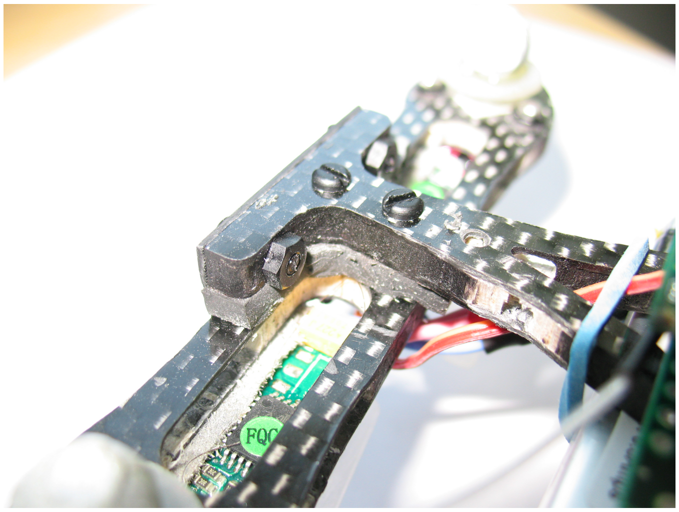


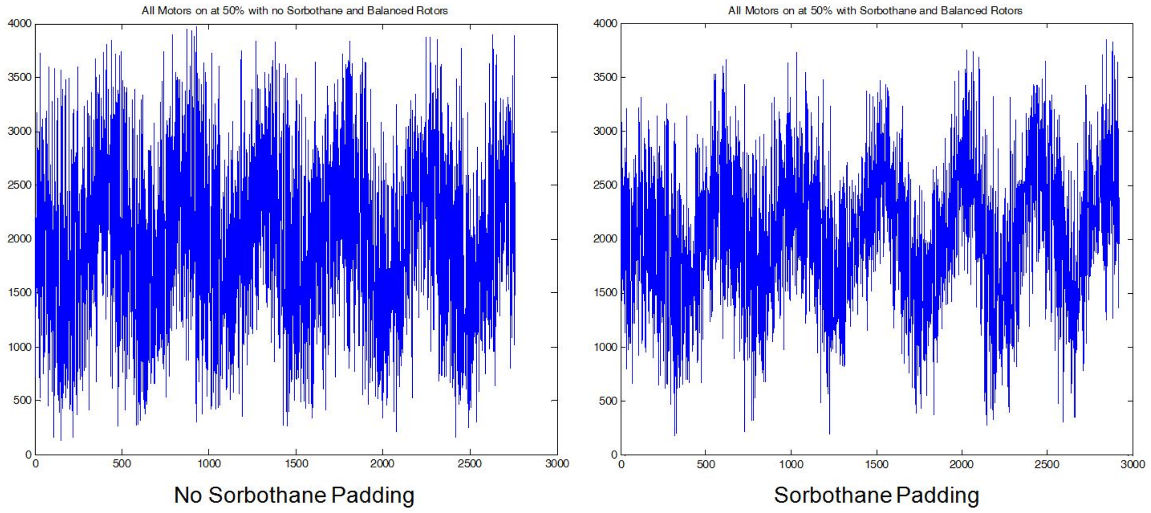
Figure 3.9: Inclusion of Sorbothane^{TM} Damping Material

Aside from the vibration solution attempts, there are several other noticeable differences in the 3rd version. The ESCs have changed again. They are another COTS component that is also used frequently in Walkera^{TM} brand RC helicopters' tail rotors. However, these are known to have a better resolution at higher RPMs. They are also, lighter and thinner, thus fitting the design better. Another aspect to the 3^{rd} version is that the Mote is connected to a breakout board. This allows much more robust connections to the ESCs and the battery, while reducing weight from extraneous wires. With the addition of all the vibration solutions the weight of the 3^{rd} version was the largest of the 5 different designs. It weighed in at over 75 grams with all the components installed.

3.4.2 Accomplishments

A large effort with this iteration of the microquad was placed toward solving the vibration problem. However, even without this solved, stable flights were possible using just the pitch and roll rate feedback. The method used was to select a large damping gain on the pitch and roll DOFs to significantly slow down these faster dynamics enough for a human pilot to control. This was consistently successful, but to be able to close the outer loop, the inner loop must be closed as well. Therefore, the pitch and roll angles must be measured cleanly and accurately.

While the implementation of the isolation mounts helped a little, it wasn't enough, even implementing the complementary filter, attitude feedback could not be achieved. The inclusion of the Sorbothane$^{\text{TM}}$ helped too, but again wasn't enough to allow feedback using the accelerometers. A comparison of accelerometer measurements with and without the addition of Sorbothane$^{\text{TM}}$ is shown in Figure 3.10.



No Sorbothane Padding

Sorbothane Padding

Figure 3.10: Reduction in Accelerometer Noise with Addition of Sorbothane^{TM}

While it is not apparent at a first glance, there is actually a reduction in the mean peek-to-peek variations by about 15%. While this is a significant improvement it is still not enough. This required further investigation into other possible answers to the vibration noise. One common cause of the vibrations is an imbalance in the rotors' mass distributions. As the rotor rotates, an imbalance in the mass creates an impulse transmitted to the frame at each rotation. In order to combat this, the rotors must be delicately balanced. The rotors used on this version of the microquad were printed using an RP machine. This led to an uncertain distribution of material across the blades. Therefore it was a useful endeavor to balance them. An example of a balanced rotor is visible in Figure 3.11.

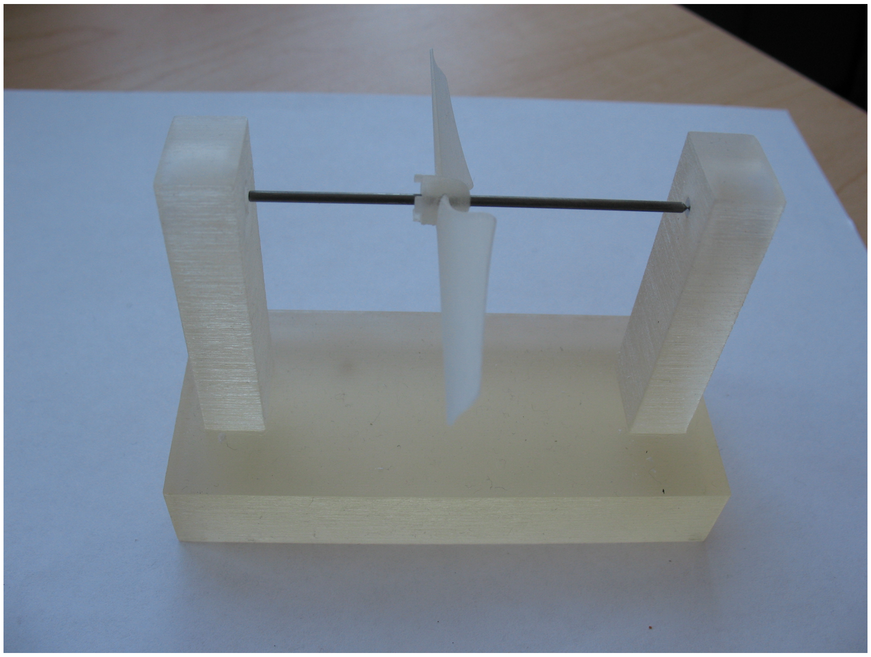


Figure 3.11: Balanced RP Rotor on Balancing Stand

This too had substantial results in reducing the magnitude of the vibrations present in the accelerometer measurements. Similarly to Figure 3.10, the results from balancing the rotors, shown in Figure 3.12, are difficult to see. Although it isn't obvious, there is a reduction in the mean peek-to-peek variations of approximately 13%. Compounding this with the Sorbothane^{TM} and the mechanical isolation mounts, a considerable improvement in the accelerometer measurements is

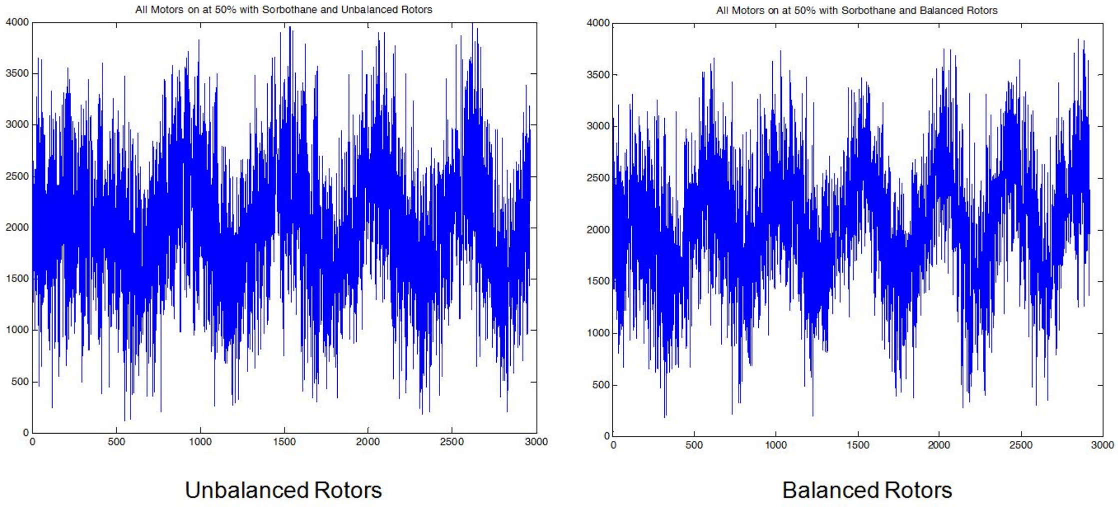


Figure 3.12: Reduction in Accelerometer Noise with Balanced Rotors

achieved. However, it is still not enough to allow implementation of proportional feedback of the roll and pitch angles ϕ and θ.

This notion led to the realization that the current accelerometer would not be suitable on the microquad. The STMicroelectronics^{TM} LIS344ALH is too sensitive to the vibrations of the motors. This is actually one of the reasons this particular chip was chosen by The University of California, Berkeley. As part of the MAST program, the Mote was intended to be able to measure the vibrations of a nearby footstep. Therefore a new Mote with a less sensitive accelerometer was required. Fortunately a different version of the GINA Mote was in development containing a new accelerometer. The GINA 2.2c Mote had a Kionix^{TM} KXSD9-1026 accelerometer, which is much more resilient to vibrations as seen in Figure 3.13.

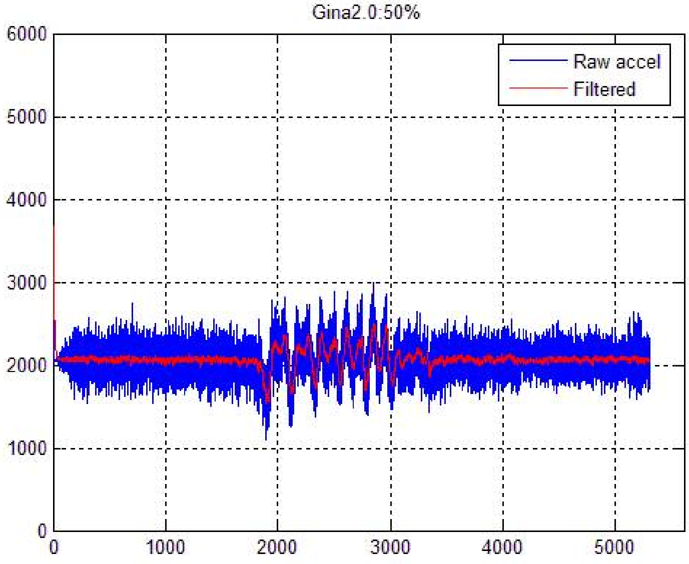


Figure 3.13: Kionix^{TM} KXSD9-1026 Accelerometer Measurements with Low Pass Filtered Signal

Therefore the solution to the vibration problem is to use the GINA 2.2c Mote with the Kionix^{TM} KXSD9-1026 accelerometer. If necessary, the Sorbothane^{TM} material can be included as well. The rotors also must be balanced to avoid significant attitude disturbances due to excessive accelerometer noise. On top of these methods, the measurements are to be filtered using a complementary filter. The arrival of the GINA 2.2c Mote coincided with the development of the next version of the microquad and therefore its progress will be documented in the next section.

3.5 4th Iteration

3.5.1 Design

The design of the 4th iteration of the microquad had a few notable differences from the previous design. However, its development was largely an effect of the reception of the new GINA 2.2c Mote and the destruction of the previous version as a result of a hard crash-landing. The new airframe was designed to be less massive, and more symmetrical than the last (see Figure 3.14). The flight ready vehicle was about 10 grams less massive than the 3rd version, weighing in at approximately 64 grams. This removed the modular design, and therefore removed the option of adding vibration damping material. This option turned out to be superfluous anyway, as the accelerometer on the new Mote was much less susceptible to the motor vibrations. The new Mote fit with the same breakout board that was used with the previous design. This allowed a simple transition hardware-wise to the new Mote. Along with the new Mote and the new airframe, the microquad was also outfitted with new rotors as seen in Figure 3.14. The new COTS

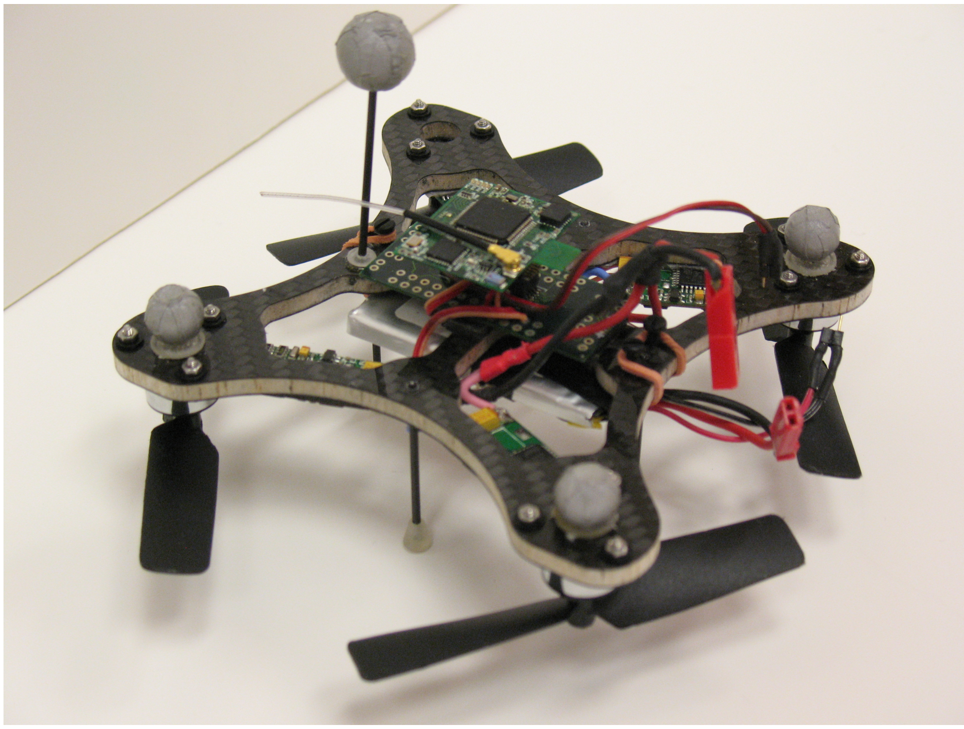


Figure 3.14: Microquad V4

rotors were a recently available product from GWS$^{\mathrm{TM}}$. Propellers of this size were not previously available in counter-rotating pairs, which are required for quadrotor configurations. With the availability of these rotors, they were tested and found to provide greater lift, albeit at a higher power consumption. They also were much stiffer than the RP material rotors printed from a 3D printing machine. This created less blade flapping, and more consistent flights. One aspect that was problematic was the manufacturing quality. They were unevenly molded, resulting in a very poorly balanced rotor. They were so poorly balanced that if the rotors were left unbalanced the vehicle would shake in flight, even without proportional attitude feedback implemented. However, balancing these was relatively easy and quite deterministic.

3.5.2 Accomplishments

Many of the most notable milestones of the microquad were achieved with the 4th iteration. A great deal of this success is due to the newest version of the Mote. The new sensors on the 2.2c Mote proved to be the key to completely closing the inner loop of the microquad. With a closed inner loop, the flights were consistently more stable, and much more controllable.

With the Kionix$^{\mathrm{TM}}$ KXSD9-1026 accelerometer and the InvenSense$^{\mathrm{TM}}$ ITG3200 gyroscope, clean estimates of the inner loop states $[\phi, \theta, p, q, r]^T$ could be obtained. The rates were clean and direct from the ITG3200. The angles, needed to be filtered and were done so using the complementary filter mentioned in § 3.3.2. While there was still some errant noise present in all these measurements, it was fast enough and small enough that it was not manifested in the actuation. The complementary filter allowed clean estimates without introducing phase lag. A problem characteristic to a low-pass filter, which is the other common option for filtering noisy signals on limited processing power. With the ability to consistently fly stable, controlled flights, the outer loop control was able to be investigated.

Using the Vicon$^{\mathrm{TM}}$ system, the outer loop states could be fed back as well. This allowed the attempt at implementing station keeping control algorithms on the microquad. The first outer loop DOF controlled using Vicon$^{\mathrm{TM}}$ feedback was the yaw or heading angle. Since this could not be measured by the sensors on board the Mote, it had to be done so using an external measurement system. Fortunately, the yaw rate was already fed back on the inner loop control scheme, and therefore only the yaw angle needed to be measured, and no unnecessary derivatives needed to be calculated in order to effectively implement a PD controller on yaw. Upon a few iterations of proportional gain tuning, a successful value was determined and the yaw angle could be regulated or tracked.

While this implementation made flights easier to pilot, because it allowed the pilot to keep his or her frame of reference while flying, the other 3 DOFs still needed to be controlled in order to achieve station keeping. The next DOF that was investigated was the heave mode as it was clearly decoupled from the other dynamics. Since this was a simple SISO channel, the first iteration at control of this DOF was a PID control scheme. This method proved to be very effective. After a few iterations of gain tuning a set of 3 gains were selected and implemented, providing regulation and tracking of vehicle altitude.

Lastly, to actually hold a position, the longitudinal and lateral dynamics needed to be controlled as well. While the yaw and heave motions were successfully decoupled and controlled separately, these two needed to be done in the same shift. Without knowledge of the model, the position hold control algorithm was approached in the same manner as altitude and heading control. A PID control scheme on each was attempted. However, the performance would noticeably deteriorate within a couple flights, and the flight characteristics would change with battery charge. As this method failed to yield any successful results, it was determined to develop a power regulator for the vehicle to provide a constant voltage to the electronics from a more powerful battery. This power regulator would be fit into another breakout board which would mold the design of the next version of the microquad.

3.6 5th Iteration

3.6.1 Design

The 5th version of the microquad is the current version at the time of this paper and the subject of the rest of the study of this paper. There are few notable differences from the previous design. This design was intended to be a proof of concept, but due to the success of the platform, it became the focal point of furthering the project. The same GWS™ rotors are incorporated, the ESCs and motors are the same as in the previous two versions. The airframe changed again (seen in Figure 3.15); the symmetry of the 4th version was removed in order to reinstall the helicopter style landing gear of the 3rd version.

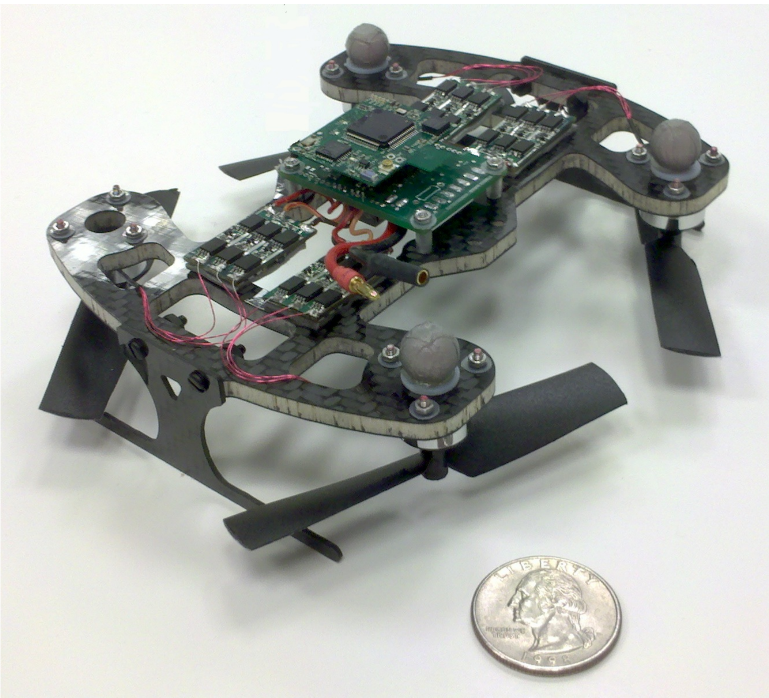


Figure 3.15: Microquad V5

This style of landing gear proved much more robust to landings than that which was installed on the 4th. The other major change is the integration of the new power regulating breakout board seen in Figure 3.16. This board was pretty massive in terms of the rest of the vehicle weighing 12 grams.

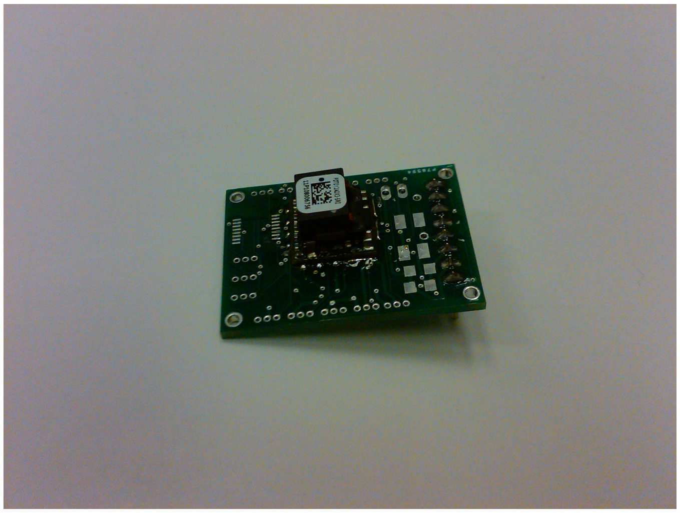


Figure 3.16: Power Regulating Breakout Board

This board allowed the usage of 2 cell (2S) Lithium Polymer (LiPo) batteries. These batteries tend to be a bit heavier than the 1 cell batteries used in all 4 previous versions and put the total vehicle mass at roughly 70 grams. However, with this battery and breakout board configuration, batteries with less capacity could be used. The output voltage also stayed constant at 3.96 V, even as the battery voltage changed from its maximum charge of 8.4 V through its nominal 7.4 V, and to its minimum charge. This constant voltage supply provided more consistent flight behavior. The previous designs all had varying performance characteristics at different battery levels, often requiring different inner loop gains depending on battery charge. Despite the large mass of the board, it did manage to keep the excess weight down, as extraneous wires and connectors were kept to a minimum.

3.6.2 Accomplishments

While this version was intended as a proof of concept design, the breakout board was so successful that the project began moving forward with this platform. All that was accomplished with the previous versions could be done as easily if not easier. As with the 4th iteration of the microquad, the inner loop was closed on the 5th too. A set of inner loop gains providing the vehicle with the most stable flight behavior was determined through iteration across many flights. The 5th version of the microquad was also successfully integrated with Vicon$^{\text{TM}}$ to implement the altitude and yaw control algorithms used on previous versions. The same station keeping control algorithm attempted on the 4th was tested using the same iterative approach used for altitude and heading control. However, as was seen with the 4th version, this method proved to be futile. Only this time the flight behavior was consistent across battery charge, yet the iterative approach was still not effective. Without a solid understanding of the vehicle's properties, obtaining a sufficient gain set to hold the vehicle's position was too difficult of a task.

Therefore, the next step with the project was to determine, as accurately as possible a linear, dynamic model of the microquad. Using the knowledge of the vehicle model, a controller can be constructed. There are many different options for synthesizing gain sets for a linear time invariant (LTI) model. Performance criterion can be set, and parameters can be tweaked to attain the controller that best meets the desired performance levels. Obtaining this model, known as system identification, can be a taxing activity and is the subject of the next chapter of this thesis.

Chapter 4

System Identification

4.1 Overview

As mentioned throughout this thesis, a mathematical model is a very useful tool in the development of an active control scheme for an unstable, agile aircraft such as the microquad. If the system can be accurately modeled in a linear form, such as the common state space formulation seen below, then there is an abundance of options available for control design.

$$\begin{aligned}\dot{\vec{x}} &= \mathbf{A}\vec{x} + \mathbf{B}\vec{u} \\ \vec{y} &= \mathbf{C}\vec{x} + \mathbf{D}\vec{u}\end{aligned} \tag{4.1}$$

Here $\vec{x}$ is the dynamic state vector, $\dot{\vec{x}}$ is the time derivative of the state vector, $\vec{u}$ is a vector of the control inputs, and $\vec{y}$ is a vector of the measured outputs. In the linear time invariant (LTI) realm which is where the analysis present in this thesis occurs, $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are matrices populated with constant parameters. $\mathbf{A}$ is the dynamics matrix, $\mathbf{B}$ is the controls matrix, $\mathbf{C}$ is the observance matrix, and (if present) $\mathbf{D}$ is known as the feedforward matrix [12]. When a system is modeled in such a form there are three basic problems to solve related to dynamics and control, see Figure 4.1 for visual reference.

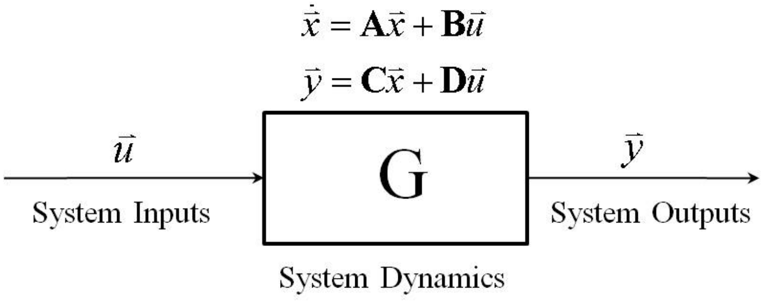

$\dot{\vec{x}} = \mathbf{A}\vec{x} + \mathbf{B}\vec{u}$
$\vec{y} = \mathbf{C}\vec{x} + \mathbf{D}\vec{u}$
$\vec{u}$
System Inputs
G
System Dynamics
$\vec{y}$
System Outputs

Figure 4.1: Dynamical System Block Diagram

The first is simulation, where given the inputs to the system $\vec{u}$ and the dynamics of the system

G, determine the output $\vec{y}$. The next is the control problem, where given the system dynamics G and the desired output $\vec{y}_d$, determine the inputs $\vec{u}$. The last, and what will be the focus of this chapter is know as system identification, where given the inputs $\vec{u}$ and the outputs $\vec{y}$, determine the system G [13].

4.1.1 Basic Process

The general process is to record flight test data of the vehicle and fit the data to a linear equation of parameters estimated through statistical methods. If the control inputs and the kinematics of the vehicle are simultaneously measured, then the data can be analyzed afterwards and an estimate of the system dynamics can be determined [3]. Ideally, this should be done with no feedback control implemented, otherwise the inputs will be affected by the outputs and the system identified will be a system with memory. However, since the microquad is dynamically unstable, there must be some form of active control enabled during flight for a human to pilot the vehicle. In order to obtain the bare airframe model, the control law must be known, as well as the gains, or else the total control signal to the plant must be measureable. Either method requires the feedback control gains be turned down as low as possible to avoid the control distorting the natural dynamic response [13].

The only feedback necessary to make the microquad flyable is attitude rate damping. Attitude proportional feedback control was not necessary, and therefore was left out to remove unnecessary complexity and to remove the possibility of affecting the dynamics. Yaw angle and rate feedback control was absent, as the yaw mode is slow enough to be corrected for by a human pilot. This left only two feedback gains implemented. The roll and pitch axes had a low rate damping term in effect, allowing the vehicle to still exhibit its natural motions, but effectively slowing down the accelerations enough to keep it pilotable.

There are several methods for obtaining the open loop dynamics of a vehicle augmented with an active control scheme. The most obvious method is to directly measure the control signals $\vec{u}$ that are being input to the plant, and to directly measure the full state vector $\vec{x}$. This method will easily identify the $\mathbf{A}$ and $\mathbf{B}$ matrices of the open loop system. For example, a closed loop identification of the roll dynamics of an aircraft with an active controller present can be done by recording the roll states ϕ, and p along with the pilot stick inputs μ_{lat} and determining stability and control derivatives through statistical processes. If the open loop is to be identified instead, the total aileron deflection, δ_a must be recorded including the contributions of the controller, not just the pilot inputs [14]. The total aileron deflections include the inputs from the control system, as well as the pilot inputs. However, direct measurement of the control signals is usually not available. If this is the case, then the form of the control law implemented must be known. If it is known there are methods for backing out the bare airframe dynamics from the closed loop dynamics. A rudimentary and rather inexact method useful for SISO systems is to estimate the root locus of the transfer function. This is easy for single-pole transfer functions, as the effect of the gain is linear. However, it can quickly become very time consuming as many iterations are needed to effectively model more complex transfer functions.

Fortunately, the process for a MIMO system isn't any more complicated than for a SISO. For instance, if the control law is of the common output feedback form $\vec{u} = -\mathbf{K}\vec{y}$. The feedforward matrix is typically taken to be 0, so this simplifies to $\vec{u} = -\mathbf{KC}\vec{x}$. Hence, when the closed loop system is identified it is of the form $\mathbf{A}_{CL} = \mathbf{A} - \mathbf{BKC}$. This means that with the knowledge of the controls matrix $\mathbf{B}$ and the sensor matrix $\mathbf{C}$, which are both to be determined in the system identification process, the open loop dynamics matrix can be calculated with some simple matrix arithmetic. The only drawback is that the feedback gain matrix $\mathbf{K}$ must be known. If it is known, however then the calculation of the open loop matrix is a minuscule task. Since this control of the microquad is custom, the feedback control gains are known exactly. Therefore, the process of identifying the microquad's bare airframe dynamics is to capture closed loop data, then to determine the bare airframe matrix by adding back on the $\mathbf{BKC}$ terms to the closed loop dynamics matrix $\mathbf{A}_{CL}$.

$$\mathbf{A} = \mathbf{A}_{CL} + \mathbf{BKC} \tag{4.2}$$

4.2 Experimental Setup

4.2.1 Vehicle Configuration

The vehicle identified is the 5^{th} version of the microquad discussed in § 3.6.1. A 70 gram quadrotor helicopter in an 'x' configuration, consisting of 4 Walkera brand ESCs, 4 brushless electric motors, 4 GWS counter-rotating 3x2 propellers, a 2 cell LiPo battery, a custom designed airframe of carbon-balsa composite sandwich, a custom designed power regulating breakout board, and lastly, the latest version of the Berkeley GINA Mote. The layout of the hardware, and the body axes used to determine the kinematic motions of the vehicle are depicted in Figure 4.2. The Mote provided the vehicle with estimates of the body axes rotation rates directly from the Invensense$^{\text{TM}}$ ITG-3200 gyroscope, and with estimates of the roll and pitch angles ϕ and θ from a complementary filter explained in § 3.3.2, using measurements from the ITG-3200 gyro and the Kionix$^{\text{TM}}$ accelerometer chips. The motors and ESCs control the RPM of the rotors. The signals sent to each individual ESC is mixed onboard the Mote according to the joystick and compensator commands, and output as a PWM signal of a determined duty cycle. A duty cycle of 0% results in a zero RPM command, while a duty cycle of 100% results in a full throttle command.

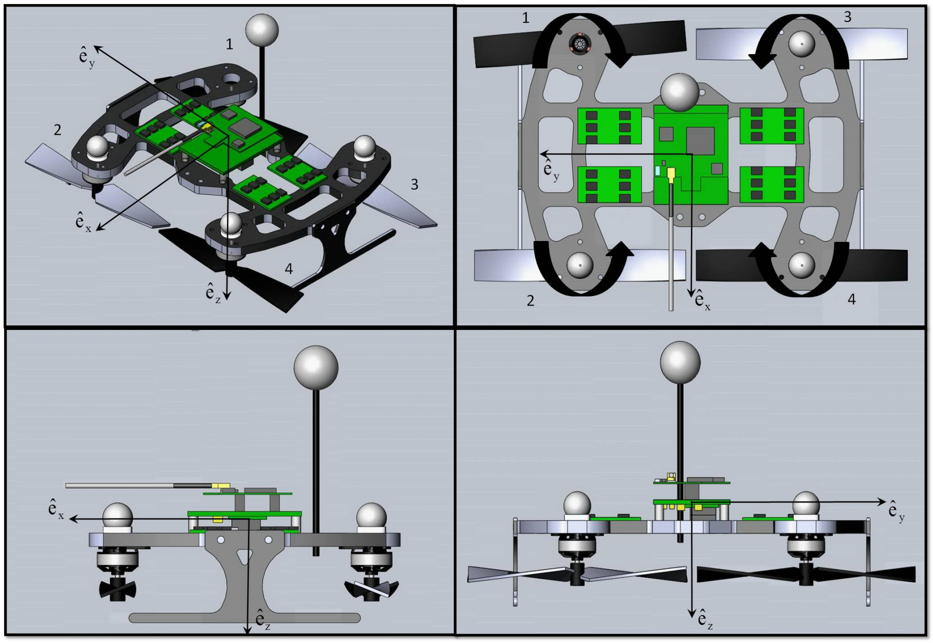


Figure 4.2: Microquad V5 Layout

The vehicle is configured such that there are 4 inputs to the airframe $\vec{u} = [\delta_{lon}, \delta_{lat}, \delta_{yaw}, \delta_{thr}]^T$. These are the total inputs to the system and are composed of two separate signals, the pilot inputs $\mu_{[\cdot]_r}$ and the compensated inputs $\mu_{[\cdot]_c}$, except for the throttle input which is strictly controlled by the pilot input. For example the total lateral input δ_{lat} is the sum of the pilot (reference) input μ_{lat_r} and the controlled input μ_{lat_c}, i.e. $\delta_{lat} = \mu_{lat_r} + \mu_{lat_c}$. They are measured in nondimensional units as they represent the duty cycle of the PWM signal differential across the necessary motor sets, i.e. $\delta_{lon}, \delta_{lat}, \delta_{yaw} \in [-500, 500]$ and $\delta_{thr} \in [0, 1000]$. These parameters are the direct inputs to the airframe, e.g. δ_{lon} is a pitch moment created by increasing the RPM of motors 2 and 4, while subsequently decreasing the RPM of motors 1 and 3 (refer to Figure 4.2). This pitch moment is modeled as a % duty cycle (precise to the 0.1%) increase sent to motors 2 and 4, and the same % duty cycle decrease sent to motors 1 and 3. A roll command δ_{lat} is similar only, now the increased RPM is across motors 1 and 2, while the decreased is across motors 3 and 4. The yaw command δ_{yaw} is the same only it creates a moment using the counter rotating pairs opposing torques. A positive yaw moment is created by increasing RPM on motors 2 and 3, while decreasing RPM by the same magnitude on motors 1 and 4. Throttle command is easily explained. It is just a direct relationship to throttle command on all motors, with 0 being 0% duty cycle and 1000 being 100% duty cycle. Therefore if the vehicle is hovering at 50% throttle, and it is given a pitch command of δ_{lon} of 100 and a roll command of δ_{lat} of 50, then the duty cycle across the motors would be as they are in Table 4.1.

Motor	Command	Duty Cycle
1	350	35%
2	550	55%
3	450	45%
4	650	65%

Table 4.1: Example of Motor Mixing for Command $\vec{u} = [100, 50, 0, 500]^T$

As mentioned in § 4.1.1 feedback is necessary to fly the vehicle. While it is possible to implement proportional attitude feedback, only rate damping is implemented as a proportional feedback gain applied to the gyro outputs, which in terms of attitude is a damping gain. One limitation to only implementing rate feedback and ignoring proportional feedback is that the controller implemented cannot track reference commands as seen in Figure 4.3. A proportional component is needed in the controller or gain matrix K to restore the tracking error ($\vec{e} = \vec{y}_d - \vec{y}$) to 0. Therefore code that typically uses a reference tracking algorithm must be altered to remove the necessity of the proportional gain if the vehicle is to fly without it. With this change the feedback control more closely resembles that of a regulator as seen in Figure 4.4. This limitation offered an inadvertent benefit; it makes the bare airframe dynamics matrix easier to identify. As mentioned in § 4.1.1, the control is of the form $\vec{u} = -\mathbf{K}\vec{y}$. This simplifies the process of backing out the open loop $\mathbf{A}$ matrix from the closed loop matrix identified. If the inner loop control law is of the form $\vec{u} = -\mathbf{K}\vec{e}$, then this significantly complicates the process of determining the open loop dynamics matrix from the identified closed loop matrix.

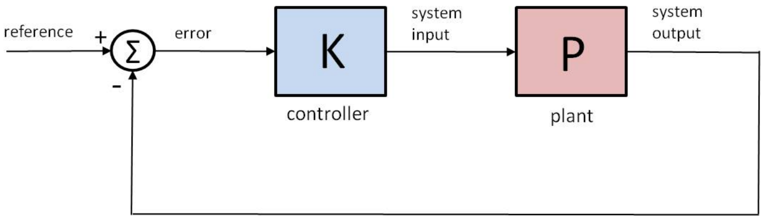


Figure 4.3: Common Feedback Control Tracker Block Diagram

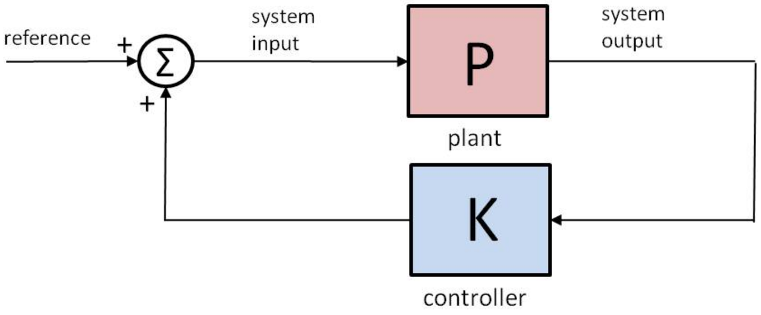


Figure 4.4: Common Feedback Control Regulator Block Diagram

As mentioned the controls of the microquad are $\vec{u} = [\delta_{lon}, \delta_{lat}, \delta_{yaw}, \delta_{thr}]^T$. Yet, the inner loop controls of the microquad disregard the throttle command, because there is no altitude control onboard. Therefore the control vector in the inner loop case is $\vec{u} = [\delta_{lon}, \delta_{lat}, \delta_{yaw}]^T$. The measured outputs are direct scaled measurements of the body rotation rates from the onboard 3-axis gyro.

And estimates of the bank angle ϕ, and the pitch angle θ from the complimentary filter utilizing the accelerometer and the gyro. Therefore the outputs are $\vec{y} = [\hat{\phi}, \hat{\theta}, \hat{p}, \hat{q}, \hat{r}]^T$. As the control is of the form $\vec{u} = -\mathbf{K}\vec{y}$, this resulted in a gain matrix of the form of Equation 4.3.

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \end{bmatrix} \tag{4.3}$$

The commonly implemented gains in this case are those that have a corresponding input and output. For instance the proportional lateral attitude gain would be the feedback gain that provides a lateral response δ_{lat} to a change in the bank angle estimate $\hat{\phi}$. Therefore the proportional roll gain would be k_{21}, while the derivative roll gain would be corresponding to the roll rate estimate $\hat{p}$, so k_{23} would be the derivative roll gain.

The microquad was also outfitted with markers covered in a specially designed retroreflective tape for Vicon$^{\text{TM}}$ tracking as seen in Figure 4.5. Including these markers added a total extra mass of approximately 4 grams. They were placed in a fashion so that the motion tracking system could fit and track a set of body axes to the model as apparent in Figure 4.6. This required non-orthogonal placement of the markers, which will unfortunately change the moments of inertia of the vehicle, however it was still able to fly consistently and controlled without any noticeable difference.

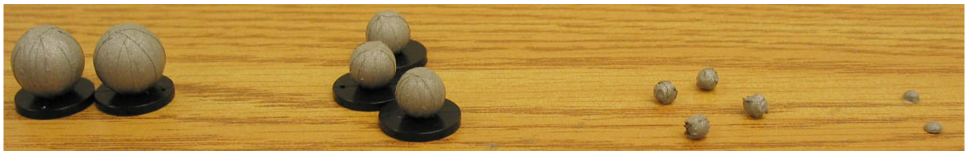


Figure 4.5: Vicon$^{\text{TM}}$ Marker Set

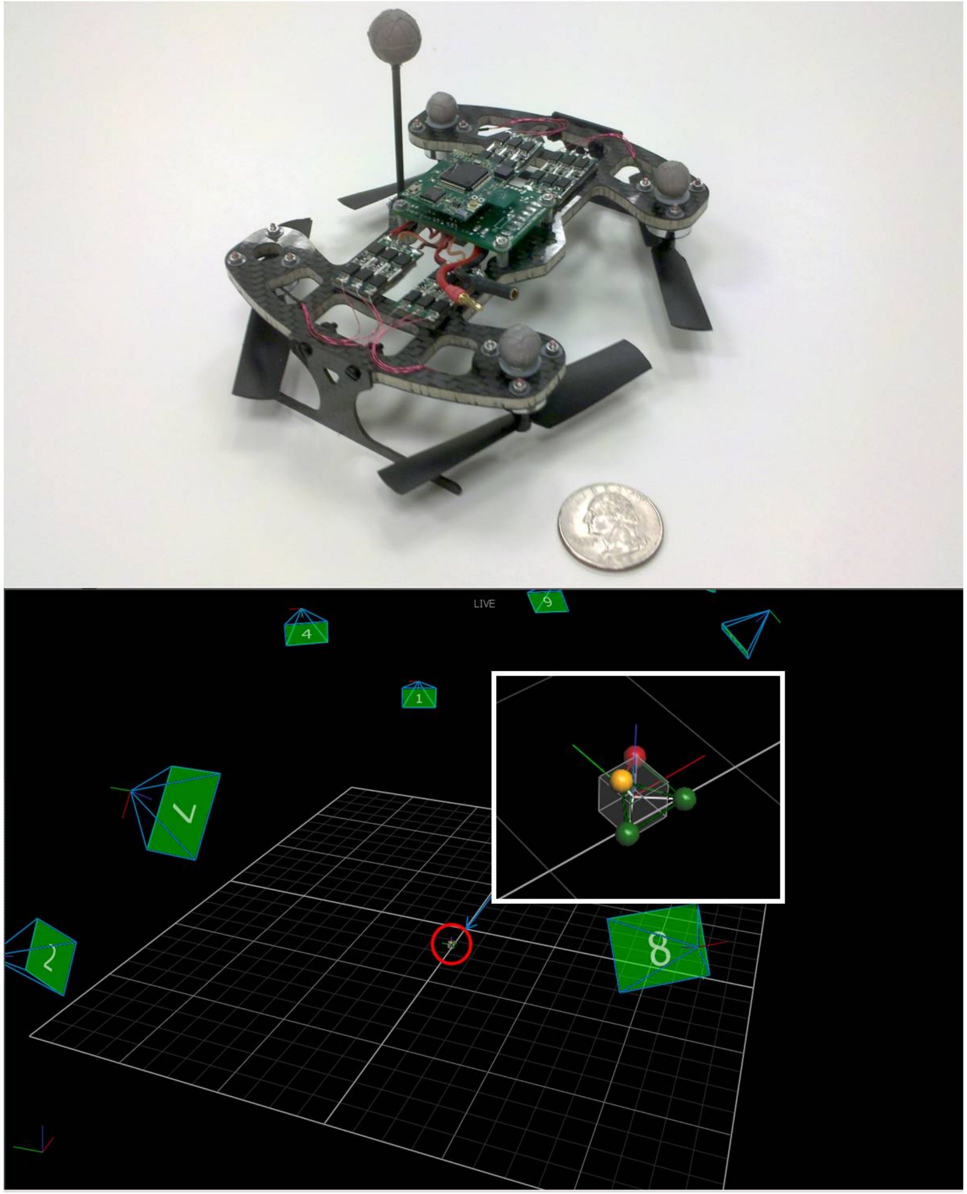


Figure 4.6: Microquad Vicon$^{\text{TM}}$ Model

The last important aspect of the microquad to the system identification process is the Mote. The Mote provides the vehicle with all the sensing, control, and communication bundled in the same package. This is convenient for the sake of system identification. The Mote can easily send back to the Basestation seen in Figure 4.7 the commands, measurements, and any number of other variables.

Figure 4.7: AVR$^{\text{TM}}$ Basestation

The Mote's communication protocol is designed such that it can simultaneously send and receive data, whether it be from another Mote, or a Basestation. It can effectively act as a relay, or a point in a network, or an independent communication node as illustrated in Figure 4.8.

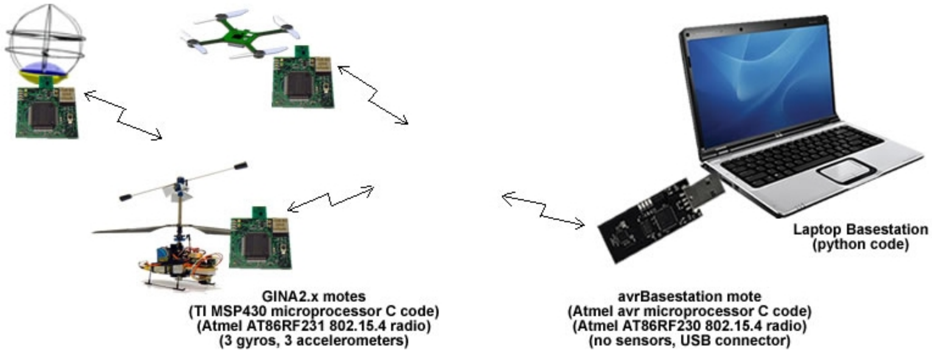

Laptop Basestation
(python code)
GINA2.x motes
(TI MSP430 microprocessor C code)
(Atmel AT86RF231 802.15.4 radio)
(3 gyros, 3 accelerometers)
avrBasestation mote
(Atmel avr microprocessor C code)
(Atmel AT86RF230 802.15.4 radio)
(no sensors, USB connector)

Figure 4.8: Mote Communication Network

This made the onboard data acquisition simple, as all parameters of interest were sent in a single packet of predefined structure. It also offered a level of redundancy that allowed the verification of the open loop dynamic matrix identified from the closed loop data. Since the total control signal sent to the plant can be measured, the open loop system can be identified in another manner other than backing it out from the closed loop matrix. This provides a vital check on the identified parameters.

The variables sent back from the vehicle for analysis were the 4 joystick commands, necessary to do the basic closed loop system identification. They are also necessary if doing the open loop identification as they account for a portion of the entire control signals being sent to the motors, as depicted in Figure 4.4. The compensated commands implemented as a function of the controller present are also sent back. These are necessary for the open loop identification, and therefore helpful for validating the model. Another necessary variable to record is an onboard sensor measurement that is proportional to an offboard measurement. This allows for two data sets sampled at different rates to be resampled at the same rate. For instance, the onboard sampling is done at 333 Hz, while the offboard is done only at 100 Hz. This means that the two data sets will be dissimilar and cannot be used for comparative analysis. In order to correct for this, one set can be linearly interpolated if a sampling time vector is present. However, this process still needs to be verified, and that is where the onboard measurement is used. In this analysis the yaw rate r is measured from the gyro onboard, and it is also measured offboard. At the beginning and end of a data set, a fiducial yaw motion is performed. The idea behind this is that it is a motion that is easily distinguishable from the rest of the data as seen in Figure 4.9 at the beginning and end of the time series, and can therefore verify that two data sets are sampled at the same rate as clearly seen by the two motions lined up after interpolation and resampling. Lastly, a timestamp is included in the packet sent from the vehicle. As was just mentioned, this is necessary to linearly interpolate

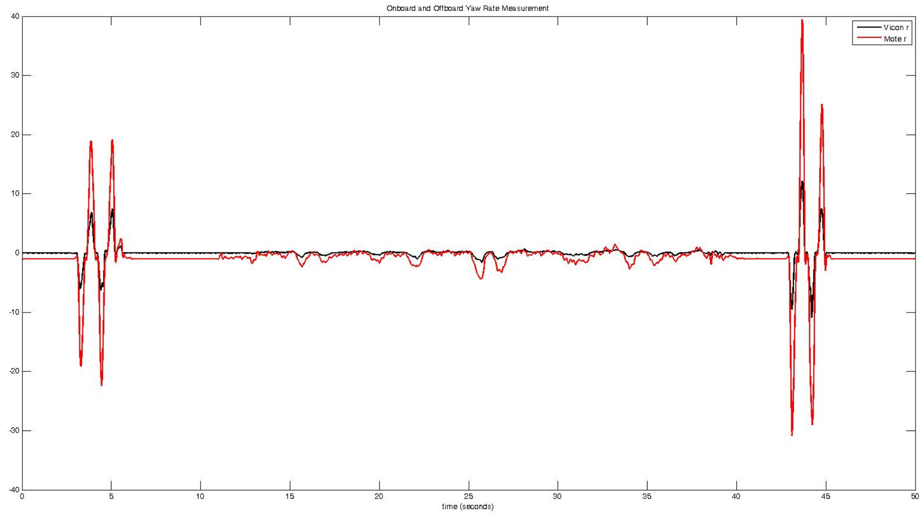


Figure 4.9: Fiducial Motions Present in Measured Yaw Rate

the signal to match a desired sampling rate. With consistent measurements of the variables above, the rest of the system identification should be simple so long as the system responsible for full state measurement can provide clean accurate measurements.

4.2.2 Data Acquisition Setup

The basic idea behind system identification is to estimate a dynamic model by fitting measured system outputs to inputs. Since § 4.2.1 details how the effective system inputs are recorded onboard via the Mote, this leaves the process behind recording the system outputs to be explained. As mentioned at several points previously a Vicon$^{\text{TM}}$ motion tracking system is available at the Autonomous Vehicle Laboratory at the University of Maryland. Using this system, the vehicle position with respect to a preset inertial origin is recorded with sub-millimeter accuracy. The vehicle's Euler Angles ϕ, θ, ψ are also obtained through the vehicle's Directional Cosine Matrix (DCM). With these 6 values, the entire dynamic state vector of the microquad $\vec{x} = [u, v, w, p, q, r, \phi, \theta, \psi, x, y, z]^T$ can be determined at each timestep with outstanding accuracy.

The system is composed of 8 cameras that strobe the test volume with a light close to infrared wavelength. For this test, the cameras are set to strobe at 100 Hz. The cameras are also designed to record only light that is reflected back. This allows the system to track only vehicles that are outfitted with the markers covered in retroreflective tape. The cameras are oriented and focused around a desired flight volume in a manner to maximize the accuracy across the entire volume. Each camera is capable of capturing the reflected light in a 2-D image [3], and the 8 images are compiled at a central processing hub known as an MX Ultranet (see Figure 4.10). After a calibration is performed, the cameras can determine their position relative to each other and can therefore build up a 3-D capture space, like that depicted in Figure 4.6, of all 8 of the 2-D images captured. Using these 8 images, a proprietary algorithm is applied and the 3-D location of each retroreflective marker is calculated. The software known as Tracker$^{\text{TM}}$ then can track all the markers together fit in a known pattern to allow the orientation of the model to be tracked. Therefore the CG inertial position of the model is tracked, and the orientation of the model's body axes are tracked with each strobe of the cameras.

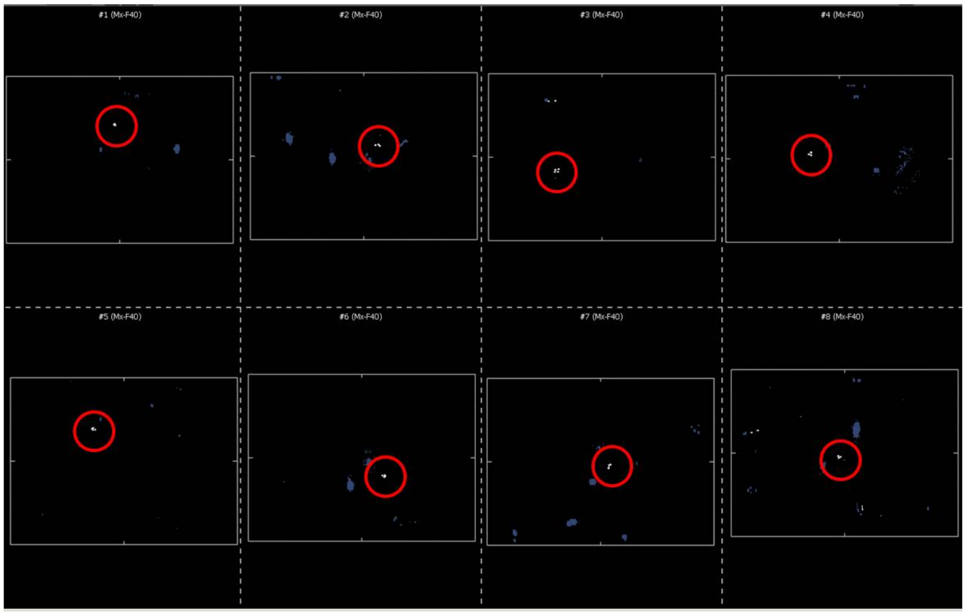


Figure 4.10: Vicon$^{\text{TM}}$ Camera Images (Microquad Markers Circled in Red)

Vicon Tracker$^{\text{TM}}$ is software that is only used to stream the kinematic data. It unfortunately does not record the data, therefore a second program must be used that can interface with the streaming data to record it for post-processing. Fortunately LabVIEW$^{\text{TM}}$ can easily interface with Vicon$^{\text{TM}}$ data while only introducing minimal latency, and offers plenty of data manipulation and recording techniques. Few changes were needed to the existing architecture to record the Vicon$^{\text{TM}}$ kinematic data, the majority of the work needed to be implemented with recording the data off the Mote. Since the LabVIEW$^{\text{TM}}$ program was already capable of communicating with the vehicle, it also was required to handle the recording of the data received. The only real hurdle in this process was to avoid the read buffer on the basestation overflowing. The Mote loop is hardcoded by the internal timer to run every 3 ms, therefore the data is sent over at a rate close to 333 Hz. The LabVIEW$^{\text{TM}}$ program, however, is set to run at 100 Hz, to increase the fidelity of the Vicon$^{\text{TM}}$ measurements. It could be set as high as 350 Hz, but at the cost of the accuracy of the data. Because the Mote code was running over 3 times as fast as the LabVIEW$^{\text{TM}}$ code, this meant that the read buffer on the basestation could easily overflow with data. Therefore before each read, the buffer had to be flushed, and told how many bytes to read from the port. After all necessary data was recorded, it could easily be post-processed into data useful to identifying the dynamics of the system.

4.3 Procedure

There are several nuances involved with the system identification process. There are typically two methods used. The more common method is frequency domain system identification like that done in [15], with the use of the Comprehensive Identification of FRequency Responses (CIFER$^{\text{TM}}$) software developed at the Army Aeroflightdynamics Directorate (AFDD). Frequency domain methods are also implemented in MATLAB's $^{\text{TM}}$ system identification toolbox. This toolbox is effective and very easy to use when working with SISO systems. It offers the option to estimate and validate a system with two sets of data loaded into the session. It also allows the removal of biases, and the option of filtering the working data with a simple pre-processing step [16]. However, because MATLAB's$^{\text{TM}}$ embedded routines tend to be very automated, it allows less freedom over the analysis.

The main idea behind the frequency domain method is that an input of varying frequency is given to the system, and the frequency response is computed by calculating the gain and phase shift of the output from the input. These values are used to piece together a Bode plot, and from this derived Bode plot, a transfer function can be inferred. Coherence measurements are used to determine the accuracy and validity of the models across the frequency range, where a coherence value of less than 0.6 [14] typically means the model is not accurate at that frequency. A definition for the coherence function is provided in [17].

$$\gamma_{uy}^2(\omega) = \frac{|\Phi_{uy}(\omega)|^2}{|\Phi_{uu}(\omega)||\Phi_{yy}(\omega)|} \tag{4.4}$$

Where $\Phi_{uy}(\omega)$ is the cross spectral density between the input u and the output y at the frequency ω. And $\Phi_{uu}(\omega)$ and $\Phi_{yy}(\omega)$ are the auto-spectral densities of the input and output respectively at frequency ω.

The other method, which is less frequently used, is the time domain method. This approach also requires measured inputs and outputs of the system, a requirement common to the system identification process in general, but it also heavily depends on knowledge of the model structure. The functions and algorithms developed by Dr. Morelli as part of the SIDPAC software incorporate both time domain methods and frequency domain methods. This analysis, however will focus on the time domain functions, which analyze the data in the time domain as the nomenclature suggests, and fit estimates of parameters to the structure provided using statistical techniques. The method used to estimate the parameters of the microquad model about hover is known as stepwise regression. This simply means that a pool of candidate regressors are included in the analysis, and each can be added or removed depending on correlation with the structure of the parameters. The regression of each candidate parameter is investigated for a linear relationship among the measured variables used in the statistical analysis [13].

4.3.1 Data Acquisition

The first step in the process of identifying the microquad is to collect the flight data. Using the setup described in § 4.2, all variables necessary to carry out the process could be recorded. However, not all the data necessary to develop a full LTI model of the microquad about hover could be collected in one flight. Due to the decoupled nature of the dynamics, it is necessary to collect data about each input. Therefore, the 4 modes about which to collect data were heave, yaw, lateral, and longitudinal. The reasoning behind this is that if one set of flight data was all that was used to fit the entire model, there would be very weak correlation terms. Across the entire timespan the confidence intervals degrade for each mode. For instance, the heave parameter estimates would be poor because there would be only a small portion of the flight that may have significant heave excitations. Another reason to avoid including all modes, is that the nonlinearities that couple the states would start to have a more significant effect, and the linear regressors would not be able to account for these nonlinear effects. Therefore it is best to break down the dynamics into decoupled modes, under the assumption that the system is decoupled.

It is good practice to collect as much data for each mode as well. With only one data set, the accuracy of the model can only be calculated by its fit across the timespan. If multiple data sets are taken, the statistical average (mean) can be used for each parameter. The standard deviation of estimates can be used to determine the accuracy of the mean as well. If enough sets are taken, the means should converge as the discrepancies in flight data are averaged out across a large enough sample size. Additional to these inherent benefits is the opportunity to validate the model. As more data is collected, one data set can be set aside to use as a validation data set. That is, the model obtained from the rest of the data sets can simulate an output, and that output can be compared to the actual output from the validation data set.

4.3.2 Post-Processing

The raw data collected from onboard the vehicle through the Mote and the kinematic data recorded from Vicon Tracker$^{\text{TM}}$ is not useful for system identification without a considerable amount of post-processing. The first procedure that must be done is to resample the two data sets so the sample rates are equal. The reasoning behind this is that a forced dynamic response must line up with the input that created it, otherwise any analysis resulting from the two off-sampled data sets is invalid. To line the two data sets up, a timestamp is concatenated to both data sets at each sample. This allows both sources to be linearly interpolated to a desired time vector. After aligning the two sensors, they are compared to eachother and verified to have the same sampling rate through the use of a fiducial yaw motion as seen in Figure 4.9. Once the sampling rate of both sources has been validated there are several other important steps necessary to provide data useful for system identification.

The next action is to crop the data. The fiducial motions at the beginning and end should be removed, as they are forced motions by the analyst. Even further after the first motion and further before the end motion must be cropped as well, because the vehicle is not actually flying during these times. The area of interest is when the vehicle is about hover. Therefore, the data to use is when the vehicle is out of ground-effect (OGE). The easy way to determine this is to use the portion of the data where the height is above some chosen altitude, or to select the data where the dynamic excitations are apparent. If the data analyzed is while the vehicle is still in ground-effect (IGE), then the model is not of the microquad about hover, as there are many nonlinearties that affect the aerodynamics and subsequently the dynamics of the vehicle. If these effects are included, an LTI model will not be accurate. An appropriate altitude to assume the microquad is OGE is approximately 0.4 meters above the floor. This is a bit conservative, however it is good to allow a large margin of error to avoid corrupting the data. An example of system identification data in Figure 4.11, shows that the useful flight data is to be taken between the times 18.7 seconds and 46.92 seconds.

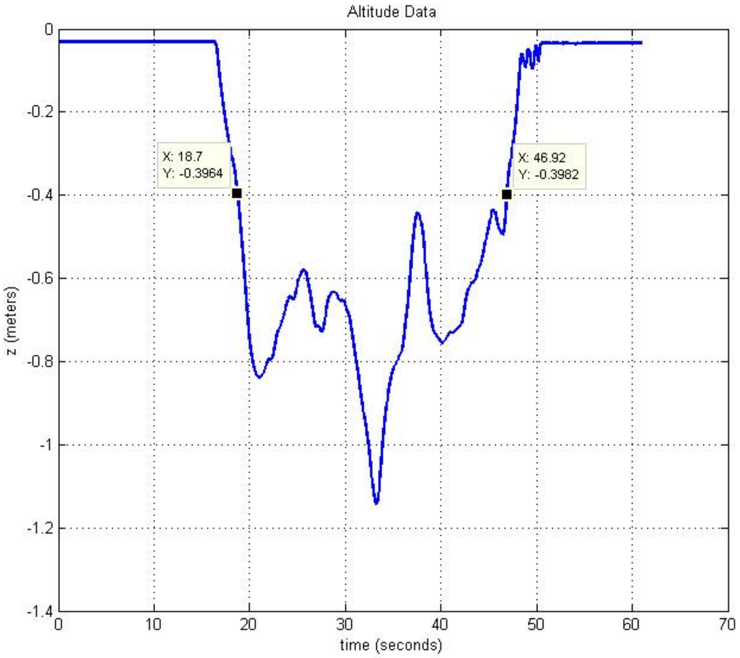


Figure 4.11: Microquad Altitude Data from System Identification Flight

After cropping the flight data to include only the portion that is approximately in the hover regime, it must be smoothed. Even though Vicon Tracker$^{\text{TM}}$ offers very accurate data, it still has a small level of noise present in the measured data. There is also the issue that the signal is very quantized, and therefore any derivatives taken are significantly affected. The smoothing process is done using global Fourier smoothing with an optimal Wiener filter. The function '*smoo.m*' written for use in MATLAB's$^{\mathrm{TM}}$ environment by Dr. Morelli, as part of the SIDPAC 2.0 release is used to perform the global Fourier smoothing of the data. The algorithm used can be found in §11.2.3 of his book [13]. Given a noisy data set, the routine provides the analyst with a plot of the sine series coefficient magnitudes as in Figure 4.12, allowing the user to select the frequency cutoff ω_c for the global smoothing. Because the filters are implemented as low pass, the deterministic signal returned is assumed to be in the frequency range $\omega \in [0, \omega_c]$ [13].

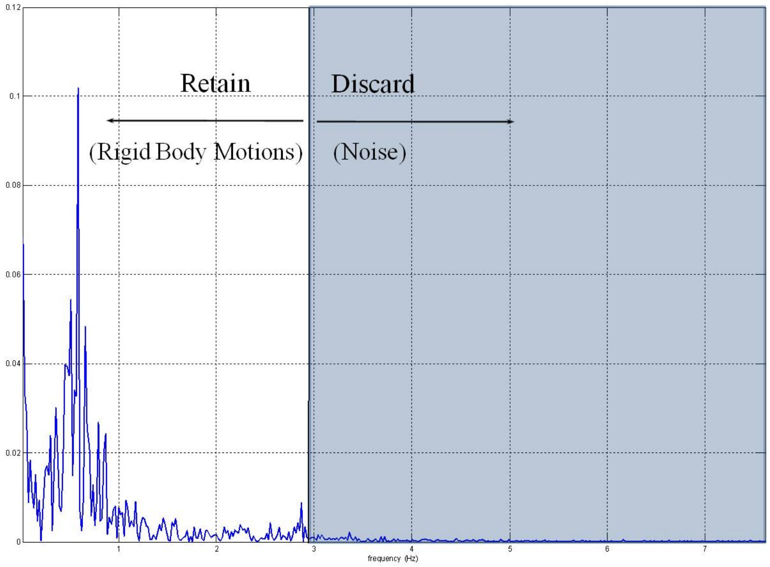


Figure 4.12: θ Sine Series Coefficient Magnitude Plot

When choosing cutoff frequency labeled in Figure 4.12 for the data used to synthesize the plot, the smoothed response can be seen in Figure 4.13. The cutoff chosen was quite conservative given the frequency data. It could be chosen as low as 1.6 Hz. There only seems to be one peak in magnitude that should be included, while the rest of the frequency data from 1.6 to 2.9 Hz doesn't have high coefficient magnitude. If only choosing the highest magnitudes of the coefficients, this frequency might be overlooked. This higher frequency however, could remove some of the vehicle's faster rigid body motions. The microquad is a very fast platform, and pitch motions on the order of 3 Hz are not unlikely.

A more aggressive example of the smoothing is better seen on a sample of data retrieved

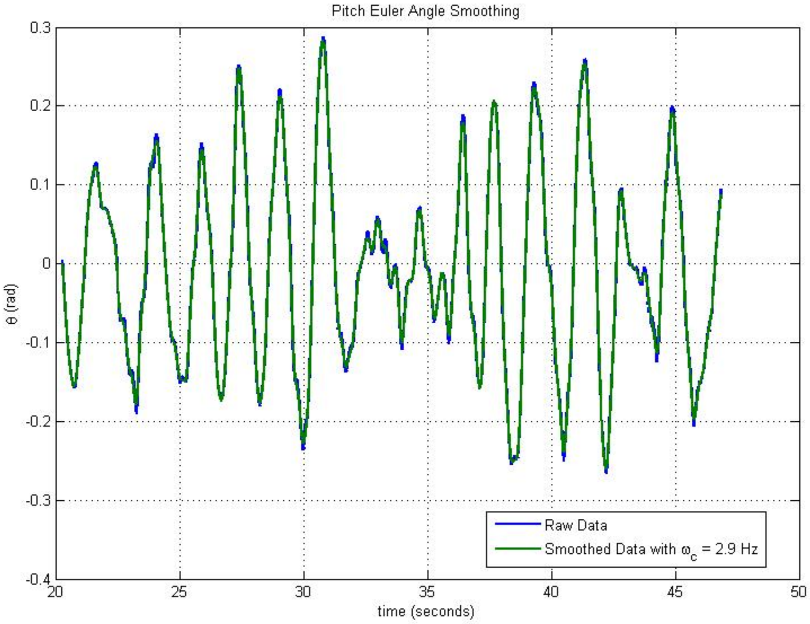


Figure 4.13: Smoothed θ Data Using Global Fourier Smoothing

from the vehicle. The Vicon$^{\text{TM}}$ measurements have much less inherent noise, and therefore very little is actually cut out when using the smoothing function '*smoo.m*' (apparent in Figure 4.13). The onboard sensors have a much more intense noise source in the form of the motor vibrations. This noise is apparent as the larger magnitudes of the sine series coefficients in Figure 4.14 at higher frequencies. Therefore setting the cutoff frequency as displayed will actually have a much more noticeable effect, which can be seen in Figure 4.15.

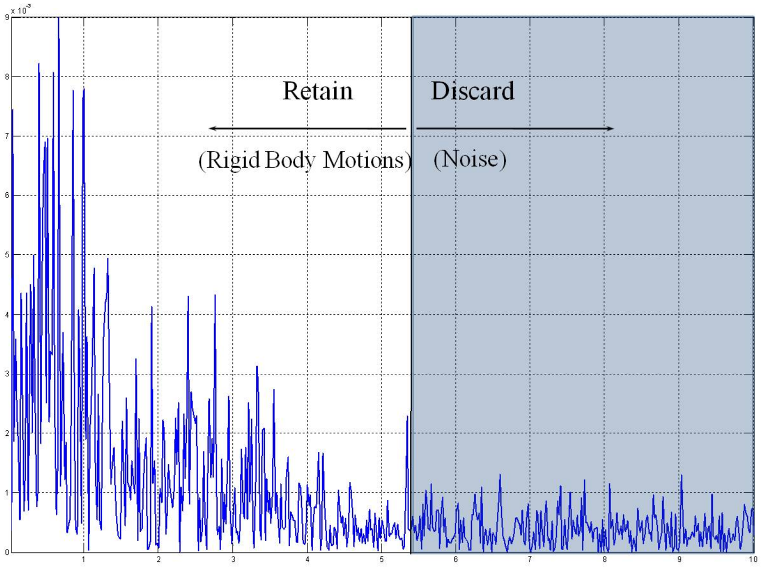


Figure 4.14: δ_{lat} Sine Series Coefficient Magnitude Plot

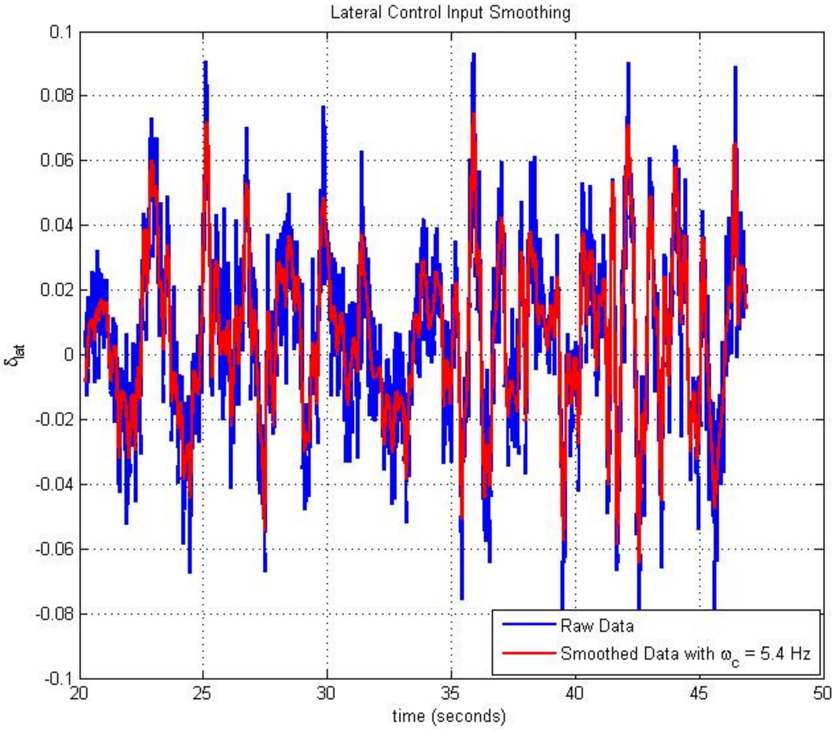


Figure 4.15: Smoothed δ_{lat} Data Using Global Fourier Smoothing, $\omega_c = 5.4$ Hz

After the sensors have been aligned, and the data cropped, and sufficiently smoothed, the data can be manipulated to reach the necessary information to perform a system identification. The post-processed measurements available are the vehicle position $[x, y, z]^T$, the vehicle orientation from the Euler angles $[\phi, \theta, \psi]^T$, the total control inputs $[\delta_{lon}, \delta_{lat}, \delta_{yaw}, \delta_{thr}]^T$, along with the inputs' two separate contributions - pilot $[\mu_{lon_r}, \mu_{lat_r}, \mu_{yaw_r}, \mu_{thr_r}]^T$ and feedback $[\mu_{lon_c}, \mu_{lat_c}, \mu_{yaw_c}]^T$. These measurements must be used to determine all variables present in the 9 equations from the inner loop state space system.

$$\dot{\vec{x}} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{A} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} + \mathbf{B} \begin{bmatrix} \delta_{lon} \\ \delta_{lat} \\ \delta_{yaw} \\ \delta_{thr} \end{bmatrix} \tag{4.5}$$

Therefore the necessary variables to have in order to obtain the stability derivatives that populate the dynamics matrix **A** and the control derivatives that populate the controls matrix **B** are the derivatives of the states $[\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$, along with the states themselves $[u, v, w, p, q, r, \phi, \theta, \psi]^T$, and the control inputs. If the bare airframe model is the system to be identified, then the total control inputs must be measured $[\delta_{lon}, \delta_{lat}, \delta_{yaw}, \delta_{thr}]^T$, but if the closed loop system is the model to be calculated, then the pilot reference commands are the measured inputs $[\mu_{lon_r}, \mu_{lat_r}, \mu_{yaw_r}, \mu_{thr_r}]^T$.

To obtain all necessary recorded variables, from just the measurements available requires a few calculations. First to determine the vehicle's translational rates $[u, v, w]^T$, the navigation equations from § 2.1.1 can be used to determine the body fixed translational rates from the derivatives of the measured inertial positions. To find $[u, v, w]^T$ from the Vicon$^{\text{TM}}$ measurements $[x, y, z]^T$ and $[\phi, \theta, \psi]^T$, we must first use the time derivatives of the inertial position $[\dot{x}, \dot{y}, \dot{z}]^T$. Then we can find $[u, v, w]^T$ from Equation 4.6.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = [T_{BG}] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{4.6}$$

Recall the form of the rotation matrix to get to the body frame from the gravity frame $[T_{BG}]$ from

§ 2.1.1 in Equation 2.2 and the equations to calculate the first three states necessary to identify **A** and **B** are

$$\begin{aligned} u &= (\cos\theta\cos\psi)\dot{x} + (\cos\theta\sin\psi)\dot{y} - (\sin\theta)\dot{z} \\ v &= (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)\dot{x} + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)\dot{v} + (\sin\phi\cos\theta)\dot{z} \\ w &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\dot{x} + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\dot{v} + (\cos\phi\cos\theta)\dot{z} \end{aligned} \tag{4.7}$$

Once the body-fixed translational velocities are calculated, the same must be done for the body-fixed rotation velocities $[p, q, r]^T$. The equations for these are also found in § 2.1.1, however to reiterate the relationships they are (from [6]).

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi}\sin\theta \\ q &= \dot{\theta} - \dot{\psi}\cos\theta\sin\phi \\ r &= \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{aligned} \tag{4.8}$$

With the relationships from Equations 4.7 and 4.8, and the direct measurements of ϕ, θ, and ψ, all 9 states in the inner loop model are available for the system identification. These can all be differentiated easily too. SIDPAC provides a smoothed derivative function '*deriv.m*' that returns an accurate smooth derivative of the input. Using this, the 9 states, and the measured and processed control inputs, all that is needed for the system identification process is available. The structure of the **A** matrix and the populated entries can be chosen freely as candidate regressors in the pool fed to the stepwise regression routine.

4.3.3 Stepwise Regression

The stepwise regression routine provided in the SIDPAC software release is a combination of two other basic categories of computational techniques for a linear regression model [13]. The two basic categories it is composed of are forward selection and backward elimination. Forward selection begins with a model that has no regressors included, only a bias, and then one regressor at a time is added if its correlation is deemed significant through a statistical parameter known as the F-Ratio (a ratio of variances of two independent samples) defined in Equation 4.9.

$$F_{n_1,n_2} = \frac{{}^{y_1}\!/_{n_1}}{{}^{y_2}\!/_{n_2}} = \frac{y_1 n_2}{y_2 n_1} \tag{4.9}$$

Where y_1 is a random variable with n_1 DOF and a χ^2 distribution function, and y_2 also has a χ^2 distribution with n_2 DOF [18].

The process of adding regressors continues, until some termination criterion is satisfied. For the forward selection process a regressor is added if it satisfies the criterion in Equation 4.10.

$$F_0 = \frac{SS_R\left(\hat{\boldsymbol{\theta}}_{p+j}\right) - SS_R\left(\hat{\boldsymbol{\theta}}_p\right)}{s^2} > F_{in} \tag{4.10}$$

F_{in} is a preselected limit to choose how significant a regressor's contribution must be to be included. $SS_R\left(\hat{\boldsymbol{\theta}}_{p+j}\right)$ is the regression sum of squares from adding the j^{th} regressor to the model composed of the p regressors already included. Therefore, this statistic is a ratio of the difference in regression sum of squares made by adding the j^{th} regressor to the fit error variance s^2, which is a metric characterizing the spread present in the parameter estimates error to the actual data.

Backward elimination is basically the opposite, where the model begins with all candidate regressors in the pool included in the model, and the regressor with the smallest F-Ratio is removed from the model; if the F-Ratio is smaller than a preselected limit F_{out}.

$$F_0 = \min_j \frac{SS_R\left(\hat{\boldsymbol{\theta}}_p\right) - SS_R\left(\hat{\boldsymbol{\theta}}_{p-j}\right)}{s^2} < F_{out} \tag{4.11}$$

Similar to Equation 4.10 the F-Ratio must be above a certain value to be included. As the algorithm removes regressors the F-Ratios are updated and weak regressors are removed until a sufficiently accurate model is returned.

The stepwise regression is a combination of both these two techniques. It offers the added benefit, however to reassess the model after including or removing a regressor. The function included in SIDPAC and used on the microquad analysis '*swr.m*' does just as described. It also offers the added bonus of showing F-Ratios of each regressor as one from the pool is added or removed, hinting to the analyst which regressor to add to the model. The data is also plotted to allow the analyst to see the results in a quick method, also including the residuals left behind by the model. An extra parameter displayed is the coefficient of determination, $0 \leq R^2 \leq 1$, which characterizes in a percentage form the model fit to the data. A definition of R^2 is found in [13].

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T} = \frac{\hat{\boldsymbol{\theta}}\mathbf{X}^T z - N\bar{z}^2}{z^T z - N\bar{z}^2} \tag{4.12}$$

Here SS_R is the regression sum of squares, SS_E is the residual sum of squares, and SS_T is the total sum of squares. $\hat{\boldsymbol{\theta}}$ is a vector of the parameter estimates, $\mathbf{X}$ is the matrix of the regressors, z is the vector of N measured outputs, and $\bar{z}$ is the mean value of the repeated measurements. Using the coefficient of determination, the F-Ratio of all regressors included, and the plots of both the model fit and the residuals the analyst can quite quickly step through the candidate pool and arrive with as good of a model as the data will allow, providing the proper pool or candidate regressors is provided.

An extra benefit of the stepwise regression routine is that it is very easy to use it to determine what regressors are best to be included in the model. If the analyst is trying to identify a full state model, all regressors can be included in the candidate pool. The SIDPAC function prints to the screen the F-Ratios of the included regressors. If a regressor is included with a minimal contribution the F-Ratio is low, which is usually indicative that it should not be included. This allows a quick and easy check for decoupled dynamics. For instance, if all regressors are included in a lateral mode identification, and there is only strong correlation with the lateral states, it can be assumed that the other states can be left out of the analysis and that the lateral mode is decoupled from the rest of the system's dynamics.

4.3.4 Model Refinement through Output Error

As is the case with most other difficult problems, one approach is not necessarily a good idea, or a smart one. It is good practice to have another method to either improve the solution or to refute it. As Gauss said in [19]

> It is always profitable to approach the more difficult problems in several ways, and not to despise the good although preferring the better.

Once stepwise regression has been performed on a data set, the next step is to refine the estimates using a technique known as output error. This function is a maximum likelihood parameter estimation method assuming a deterministic model with no process noise, but with measurement noise present. The basic process behind the procedure is to improve upon an initial guess of the parameters populating the state space matrices. Using the initial guess of the model the outputs are simulated and compared to the actual measured outputs. A block diagram representing the flow of the process can be seen in Figure 4.16, courtesy of [13].

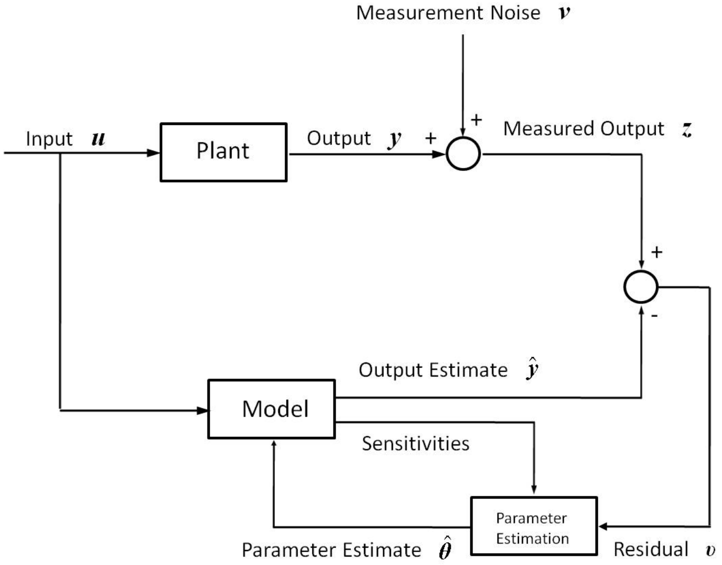

Measurement Noise $\boldsymbol{v}$
Input $\boldsymbol{u}$
Plant
Output $\boldsymbol{y}$
+
+
Measured Output $\boldsymbol{z}$
+
-
Output Estimate $\hat{\boldsymbol{y}}$
Model
Sensitivities
Parameter Estimation
Parameter Estimate $\hat{\boldsymbol{\theta}}$
Residual $\boldsymbol{v}$

Figure 4.16: Output Error Parameter Estimation Flow

The parameter estimates are then altered to minimize a cost function depending on the output error. The cost function is a negative log-likelihood cost function of the form

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{v}(i)\hat{\mathbf{R}}^{-1}\boldsymbol{v}^{T}(i) = \frac{1}{2}\sum_{i=1}^{N}[z(i) - y(i)]\hat{\mathbf{R}}^{-1}[z(i) - y(i)]^{T} \qquad (4.13)$$

Where $\hat{\mathbf{R}}$ is the covariance matrix of the residuals $\boldsymbol{v}(i)$ defined as

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}(i)\boldsymbol{v}^{T}(i) \qquad (4.14)$$

The improved estimates of the initial guess of the parameters $\hat{\boldsymbol{\theta}}_0$ can be defined as the estimates with the maximum likelihood given the statistical information of the model. The expression for the parameter estimator is

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_0 - \mathbf{M}^{-1}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \left[\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{4.15}$$

Where $\mathbf{M}$ is the Fisher information matrix.

$$\mathbf{M} \equiv -E \left[\frac{\partial^2 ln \mathbb{L}(\mathbf{Z}_N; \boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \right] = \sum_{i=1}^{N} \frac{\partial \mathbf{y}^T(i)}{\partial \boldsymbol{\theta}} \hat{\mathbf{R}}^{-1} \frac{\partial \mathbf{y}(i)}{\partial \boldsymbol{\theta}} \tag{4.16}$$

Where $\mathbb{L}(\mathbf{Z}_N; \boldsymbol{\theta})$ is the likelihood function. It is the probability of obtaining the matrix of measured outputs $\mathbf{Z}_N$ given the vector of parameter estimates $\boldsymbol{\theta}$, i.e. p$(\mathbf{Z}_N|\boldsymbol{\theta})$. Since it is a second-order gradient it is used to determine whether an optima is either maximum or minimum. The Cramér-Rao inequality (Equation 4.17) follows, placing a lower bound on the variance of the estimators [13].

$$\text{Cov}(\hat{\boldsymbol{\theta}}) \geq \mathbf{M}^{-1}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{4.17}$$

Using the above relationships and tools, the computational routine, summarized for the output error method found in [13] is as follows:

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_0 + \Delta\hat{\boldsymbol{\theta}} \tag{4.18}$$

$$\Delta\hat{\boldsymbol{\theta}} = -\mathbf{M}^{-1}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \mathbf{g}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{4.19}$$

$$\text{Cov}(\hat{\boldsymbol{\theta}}) \geq \mathbf{M}^{-1}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{4.20}$$

Where

$$\mathbf{M}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \sum_{i=1}^{N} [\mathbf{S}^T(i) \hat{\mathbf{R}}^{-1} \mathbf{S}(i)]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{4.21}$$

$$\mathbf{g}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \sum_{i=1}^{N} [\mathbf{S}^T(i) \hat{\mathbf{R}}^{-1} \boldsymbol{\upsilon}(i)]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{4.22}$$

$$\mathbf{S}(i) = [s_{jk}(i)] = \left[\frac{\partial y_j(i,\boldsymbol{\theta})}{\partial \theta_k}\right] j = 1, 2, ..., n_o;\; k = 1, 2, ..., n_p \tag{4.23}$$

$$\boldsymbol{v}(i) = z(i) - \hat{y}(i,\boldsymbol{\theta}) \tag{4.24}$$

Here, $\mathbf{M}$, the information matrix has dimension $n_p \times n_p$, and the sensitivity matrix $\mathbf{S}(i)$ has dimensions $n_o \times n_p$. The sensitivities of the outputs quantify the change in the outputs due to the change in the parameters. The cost function has a gradient function $\mathbf{g}$, and the covariance matrix $\hat{\mathbf{R}}$ is computed from Equation 4.14. If the nominal value for the parameters $\boldsymbol{\theta}_0$ is not close to the global solution then the routine will likely diverge and never arrive at a maximum likelihood estimate [13].

The output error process continues updating the estimates by minimizing the cost function using the maximum likelihood estimates. The computational algorithm is terminated when convergence criteria is met. A common set of criteria is one or more of the following

1. The magnitude of the updates $\Delta\boldsymbol{\theta}$ are sufficiently small.
2. The cost function $J(\hat{\boldsymbol{\theta}})$ has minimal changes for consecutive iterations.
3. The cost gradient $\mathbf{g}$ has magnitude close to zero (the slope of the cost function is minimal – similar to Number 2).
4. The covariance matrix $\hat{\mathbf{R}}$ changes minimally as well.

Once the convergence criteria is met, the new estimates can be compared with the data, and the two separate models (the stepwise regression parameters, and the output error parameters) can be compared for accuracy as well [13].

4.4 Closed Loop Identification

With a clear procedure outlined in §4.3 the system identification process for the microquad was a simple iterative procedure of collecting data about each of the 4 excitation modes, processing it, and then analyzing it. The structure assumed for the model is similar to that in Equation 2.55. The only difference is the value of the kinematic relationships are not immediately assumed to be 1, nor are the gravitational contributions. The structure of the dynamics matrix $\mathbf{A}$ is then postulated to take the form

$$\begin{bmatrix} X_u & 0 & 0 & 0 & X_q & 0 & 0 & X_\theta & 0 \\ 0 & Y_v & 0 & Y_p & 0 & 0 & Y_\phi & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_v & 0 & L_p & 0 & 0 & L_\phi & 0 & 0 \\ M_u & 0 & 0 & 0 & M_q & 0 & 0 & M_\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi_p & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Theta_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Psi_r & 0 & 0 & 0 \end{bmatrix} \tag{4.25}$$

Where Φ_p, Θ_q, and Ψ_r are all anticipated to be approximately 1, and X_θ is approximately $-g$, while Y_ϕ should be close to g. This is similar to the structure found in [3], however with more coupling in the longitudinal DOFs and in the lateral DOFs. The controls matrix $\mathbf{B}$ is assumed to have the structure

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Z_{thr} \\ 0 & L_{lat} & 0 & 0 \\ M_{lon} & 0 & 0 & 0 \\ 0 & 0 & N_{yaw} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.26}$$

The inputs can be either the pilot inputs or the total control inputs, depending whether the analyst is trying to identify the closed loop system or the bare airframe dynamics respectively. For the remainder of the analysis the nomenclature used will be to use $\delta_{[\cdot]}$ for the inputs used in the analysis. While this was used to denote the total inputs, the approach will be to identify the closed loop system. Therefore the inputs are the reference pilot inputs $\mu_{[\cdot]_r}$. The logic behind this move is that the compensator input contributions $\mu_{[\cdot]_c}$ are very noisy as apparent in Figure 4.14. This makes the system identification analysis difficult. It becomes laborious to differentiate where the noisy signals actually create an effect on the rigid body dynamics. Hence the closed loop system is easier to identify because the pilot inputs are much smoother (obvious in Figure 4.17), and therefore the rigid body responses to these are more obvious. The process will then be to identify the closed loop system, then with the knowledge of the closed loop dynamics matrix, $\mathbf{A}_{CL}$, the controls matrix **B**, the inner loop observation matrix **C**, and the inner loop gain matrix **K**, the bare airframe dynamics matrix **A** can be calculated from Equation 4.2.

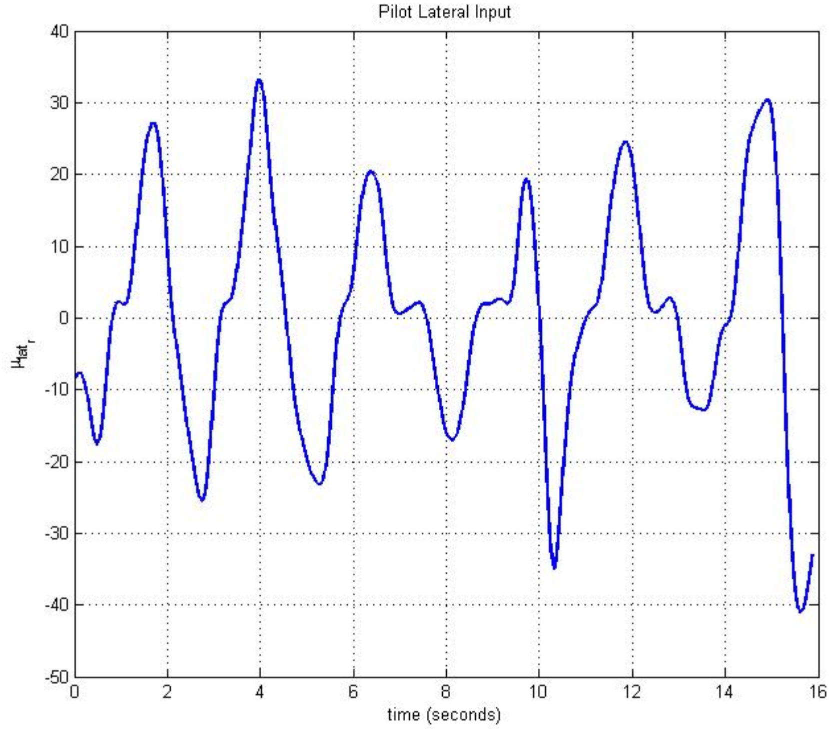


Figure 4.17: Pilot Input Example

As mentioned in §4.3.1, the DOFs are separated by what is affected by each input. This left the analysis categorized into 4 different modes, longitudinal, lateral, yaw, and heave. The longitudinal mode consisted of the states, $[u, q, \theta]^T$, and the control input δ_{lon}. The lateral is similar, in that is consisted of three states also $[v, p, \phi]^T$, and the control input δ_{lat}. The yaw mode contained the two yaw states $[r, \psi]^T$, and of course the yaw input δ_{yaw}. The heave mode is much more simple, consisting of only the heave velocity state w, and the throttle input δ_{thr}. These categories successfully break down the entire system into 4 modes, where each state and control input is accounted for. In doing this, the data collected during the a test flight should be data where the vehicle is excited about the desired identifiable mode, e.g. if trying to identify the heave mode, the aircraft should be given varying throttle commands, and the other DOFs should be isolated with as little excitation from the pilot as possible.

4.4.1 Yaw Mode

The yaw mode is presumed to be a simple two pole system, forced by the yaw input δ_{yaw} only. The form of this mode is selected from Equations 4.25 and 4.26. When isolating only the yaw states and controls as listed in the previous section this results in the following state space formulation.

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} N_r & 0 \\ \Psi_r & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} N_{yaw} \\ 0 \end{bmatrix} \delta_{yaw} \tag{4.27}$$

This structure was validated during analysis, by including all other states in the candidate regressor pool and checking all states' F-Ratios. Only the yaw states and controls present in Equation 4.27 had significant contributions and the decoupled assumption was effectively validated.

5 flights of yaw data were collected. Across all 5 flights, the procedure used to calculate the stability and control derivative estimates was to first determine an initial guess using the stepwise regression function '*swr.m*' and then to refine the estimates using the output error routine '*oe.m*'. In all data sets, the output error estimates provided more approximate results to the actual measured data as seen in Figure 4.18. The change in performance seems small, but it is still better across the yaw maneuver for output error vs. stepwise regression.

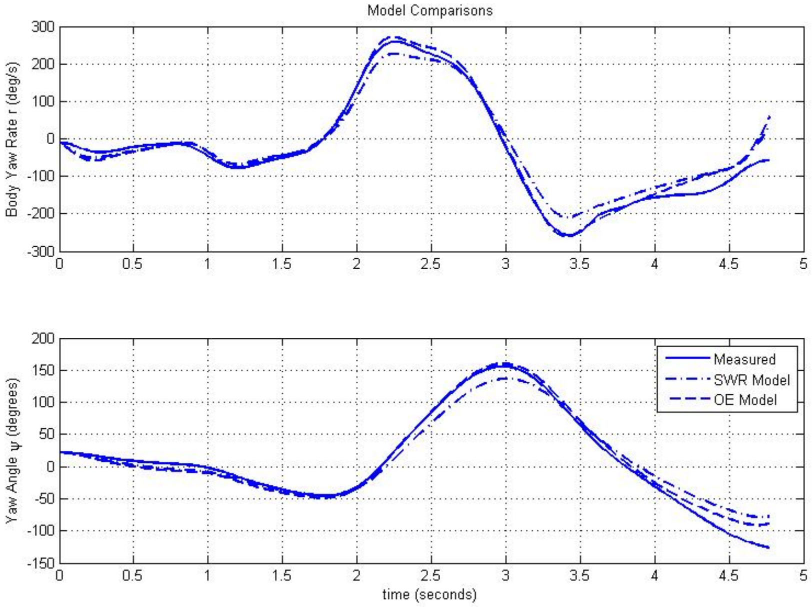


Figure 4.18: Yaw Model Comparison

Using this method, the following derivatives were calculated for the yaw mode. The kinematic assumption that $\dot{\psi} = r$, held as the values for Ψ_r were all approximately equal to 1. Therefore this value is assumed to be 1 in the model. The parameter values along with the standard error for that data set (a measure of the spread in the time series data) are found in Table 4.2.

N_r			N_{yaw}		
Parameter	Standard Error	% Error	Parameter	Standard Error	% Error
-0.55169	± 0.01694	3.07	0.11924	± 9.67e-4	0.81
-1.028	± 0.041412	4.03	0.1184	± 21.04e-4	1.77
-0.59479	± 0.038057	6.40	0.0953	± 19.42e-4	2.04
-0.59954	± 0.014377	2.40	0.1054	± 9.45e-4	0.90
-0.85203	± 0.017684	2.07	0.1038	± 9.04e-4	0.87

Table 4.2: LTI Yaw Stability and Control Derivatives

With the data presented in Table 4.2, an average can be taken to presumably be the best estimate for the parameter. An extra measure of quantification of the data, is to take the standard deviation, and see if it is on the same level as the standard errors seen. They should approximately agree if the sample size is large enough as they are both statistical procedures used to characterize the scatter in the data.

Parameter	Mean	Standard Deviation	Standard Error Mean
N_r	-0.72521	0.206565	± 0.025694
N_{yaw}	0.1084436	0.010248	± 0.0013725

Table 4.3: LTI Yaw Derivatives' Statistics

These values make sense, the damping term is negative, indicating the aerodynamic friction as the vehicle yaws. The control derivative is the right sign, as a yaw command should cause the vehicle to yaw in the positive direction. Yet, unfortunately the standard deviation of the parameter estimates is not on the same scale as the standard error. This, however, is easily explained due to the small sample size, and the existence of a few outliers (particularly runs 2 and 5) greatly increase the standard deviation as they account for 40% of the data. This leads to the standard deviation for both parameters to be an order of magnitude larger than the mean of the standard error.

Another approach to quantify this data, is to only use 4 sets of data to statistically characterize the model, and use the 5$^{\text{th}}$ data set to verify it. The 4 data sets to characterize the average model are the 1$^{\text{st}}$, 2$^{\text{nd}}$, 4$^{\text{th}}$, and 5$^{\text{th}}$; and the 3$^{\text{rd}}$ is used to compare. In doing this, the mean values of the parameters change, and the yaw decoupled model takes the form in Equation 4.28.

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -0.75782 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} 0.111737 \\ 0 \end{bmatrix} \delta_{yaw} \tag{4.28}$$

When using this state space model, the yaw states r and ψ can be simulated and compared to actual flight data. The comparison can be seen in Figure 4.19. The results are satisfactory for a linear model. There is a discrepancy in both at approximately 3 seconds. This could be an artifact of the post-processing, or it could be a nonlinearity not captured by the linear model. Either way, the results are sufficiently close to conclude the yaw structure is as it appears in Equation 4.28. And the final parameter values for the derivatives are found in Table 4.4.

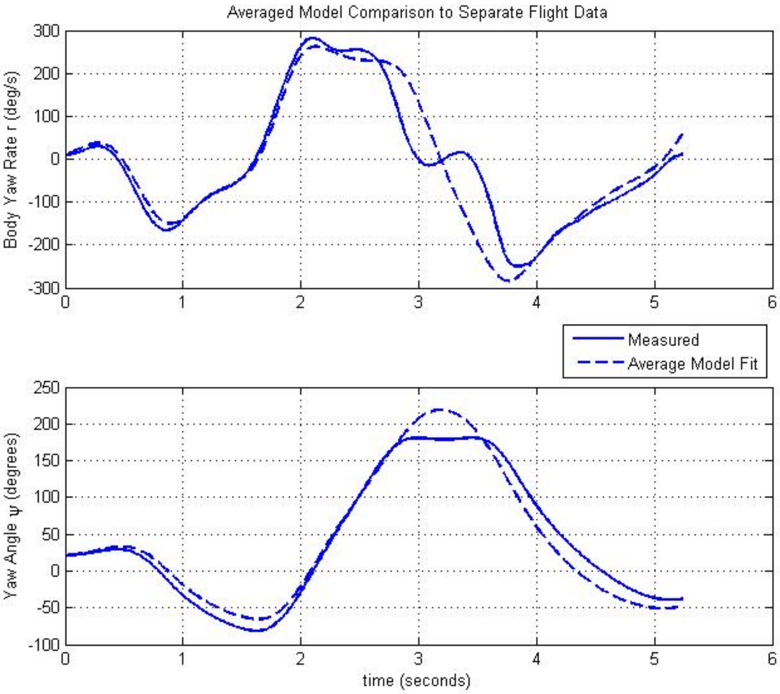


Figure 4.19: Averaged Yaw Model Comparison to Separate Flight Data

Parameter	Value
N_r	-0.75782
N_{yaw}	0.111737
Ψ_r	1

Table 4.4: Final Yaw Derivative Values

4.4.2 Heave Mode

The heave mode is the simplest, most decoupled mode. This was verified through a preliminary analysis of the data using SIDPAC's '*swr.m*' where all states were included in the candidate regressor pool. The only two regressors that had a strong enough correlation were the heave velocity w and the throttle input δ_{thr}. It is a single pole system, that could just as easily be modeled as a transfer function of the form

$$\frac{w(s)}{Z_{thr}(s)} = \frac{Z_{thr}}{s - Z_w} \tag{4.29}$$

This translates to a scalar differential equation. For continuity's sake, it will include brackets to denote it's inclusion in the state space model as a matrix.

$$[\dot{w}] = [Z_w]\,[w] + [Z_{thr}]\,\delta_{thr} \tag{4.30}$$

6 test flights in which heave was the dominant mode excited were completed. The same process was done as in the yaw mode identification. With the post-processed data, the stepwise regression routine '*swr.m*' was run to provide the initial guess to feed through to the output error technique '*oe.m*'. The two separate models were then compared for each test. The output error was better at matching the data across the entire flight regime. It was only slightly better as seen in Figure 4.20, but when trying to fit a linear model to a nonlinear system, every bit of accuracy counts.

The models tend to have more difficulty fitting this data as it is clear there is more nonlinear behavior present. This could be due to the difficulty in piloting the vehicle and exciting only heave in a capture volume with a small z dimension. Regardless, the process to find the parameter

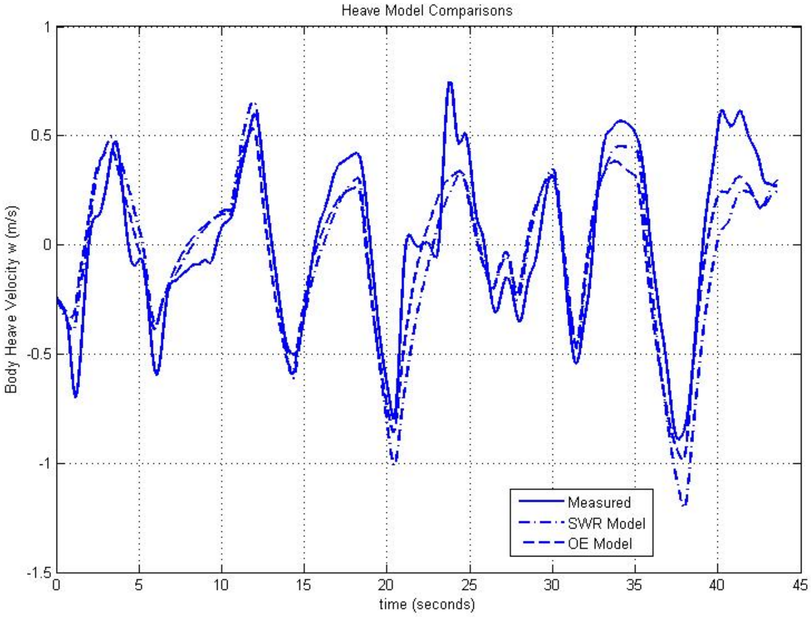


Figure 4.20: Heave Model Comparison

estimates was carried out to provide the results in Table 4.5.

Z_w			Z_{thr}		
Parameter	Standard Error	% Error	Parameter	Standard Error	% Error
-0.40354	± 0.012741	3.16	-0.012691	± 1.89e-4	1.49
-0.35066	± 0.007599	2.16	-0.01555	± 1.3e-4	0.83
-0.60147	± 0.017903	2.98	-0.013145	± 2.18e-4	1.65
-0.50686	± 0.01743	3.44	-0.014112	± 1.54e-4	1.09
-0.58558	± 0.010844	1.85	-0.012945	± 1.4e-4	1.09
-0.48073	± 0.008755	1.82	-0.015367	± 1.2e-4	0.78

Table 4.5: LTI Heave Stability and Control Derivatives

Statistically analyzing this data results in similar findings as with the yaw mode. The parameters are of the right sign. The heave damping term indicates aerodynamic friction resisting the vehicle's motion along the body z-axis. The control derivative Z_{thr} is negative, as thrust provides lift which is along the negative body z-axis. The standard deviation of the data sets provides a value that is an order of magnitude larger than the mean of the standard errors (seen in Table 4.6). This again can be attributed to the small sample size. However, the estimates do seem to be in the same ballpark and therefore a statistical average of them can be safe to use in a linear model.

Parameter	Mean	Standard Deviation	Standard Error Mean
Z_w	-0.48814	0.098794	± 0.012545
Z_{thr}	-0.013968	0.0012518	± 1.59e-4

Table 4.6: LTI Heave Derivatives' Statistics

The same method used in the yaw mode identification is employed on the heave mode. One set of data is reserved for model verification. In this case the 2$^{\text{nd}}$ flight is used as verification data. The 1$^{\text{st}}$, 3$^{\text{rd}}$, 4$^{\text{th}}$, 5$^{\text{th}}$, and 6$^{\text{th}}$ flights are all used to determine the average parameters for the heave model. This changes the average value of the derivatives to those found in Table 4.7. These new values fit the model relatively well, excluding some nonlinear regimes obvious in the flight data shown in Figure 4.21. While this fit is not great, it is the best an LTI decoupled model can do for flight data that obviously includes nonlinear effects, and possibly other faster dynamics that may have manifested themselves as coupled dynamics that may not actually be present.

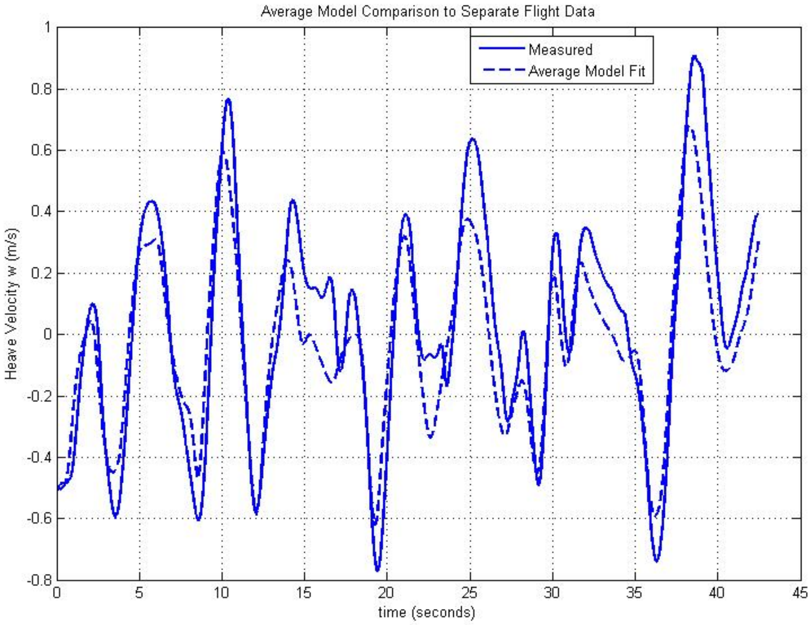


Figure 4.21: Averaged Heave Model Comparison to Separate Flight Data

Parameter	Value
Z_w	-0.515636
Z_{thr}	-0.013652

Table 4.7: Final Heave Derivative Values

4.4.3 Lateral Mode

The rotational DOFs included in the lateral and longitudinal modes are where the analysis begins to become difficult. This posed the need to take more data at the lateral and longitudinal modes than for either the yaw or heave modes. 10 flights were conducted where the DOFs excited were the lateral mode's. The structure of the lateral mode is taken from the full state space model. Because it is assumed to be decoupled, this is possible and the lateral mode is of the form in Equation 4.31. The decoupled assumption was tested in the same way for the yaw and heave cases. The stepwise regression function was called using all the states as regressors. The only F-Ratios that were significant were those that corresponded to the lateral states in Equation 4.31.

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & Y_\phi \\ L_v & L_p & L_\phi \\ 0 & \Phi_p & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ L_{lat} \\ 0 \end{bmatrix} \delta_{lat} \tag{4.31}$$

This model is already more complicated as it has more parameters to identify than the yaw and heave modes combined. A benefit to the SIDPAC software is that the same process used on the other two modes can still be used for this more complex system. One slight difference to note, however, is that the lateral and longitudinal modes are where the feedback control is implemented. The control provided is simply a proportional feedback gain on the onboard estimates of the roll and pitch rates p and q. The stepwise regression function's model was compared with the output error model. As was the case with the yaw mode and the heave mode, the output error provided better estimates of the model behavior for all three lateral states (refer to Figure 4.22).

Using output error to refine the stepwise regression estimates provides similar intuitive findings as those from the heave and yaw cases. However, as a result of the increased number of parameters to estimate, there is a much greater spread because there are more variables to increase the complexity and the scatter. The kinematic parameter Φ_p was approximately 1 for all

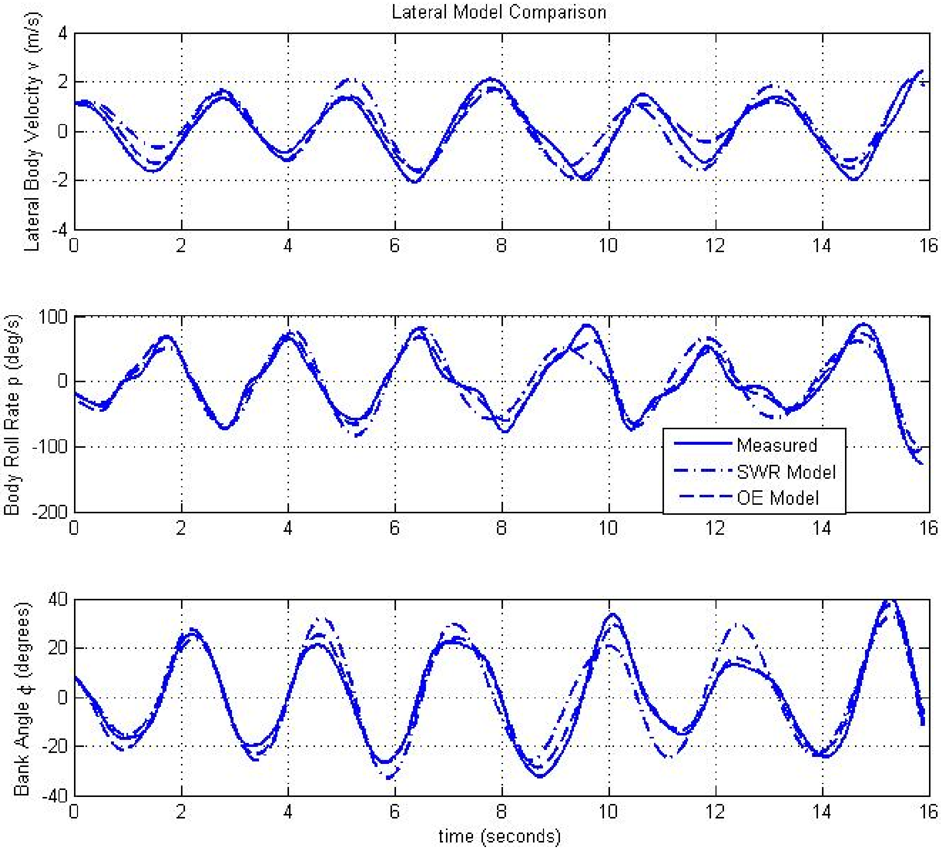


Figure 4.22: Lateral Model Comparison

flights. Therefore it will be assumed as 1 in the model and not analyzed. Table 4.8 provides the characteristics for the lateral translational body velocity's, v, derivatives, and Table 4.9 provides the same characteristics for p, the lateral rotational body velocity. The standard error column is removed for simplicity purposes. It can be determined by multiplying the % error as a decimal by the parameter value.

Y_v		Y_p		Y_ϕ	
Parameter	% Error	Parameter	% Error	Parameter	% Error
-1.0506	4.39	-1.0224	7.73	7.8911	1.95
-0.32082	8.17	0.13124	37.00	7.9371	0.97
-0.30883	19.10	0.34177	26.68	7.6484	1.84
-0.65645	4.84	-0.03536	128.45	9.7584	0.70
-0.91546	3.01	-0.3907	10.57	8.9101	0.99
-0.35533	8.12	0.64598	7.90	6.5594	1.91
-0.7852	2.72	-0.02195	186.67	6.9793	1.44
-2.0308	4.60	1.1583	7.46	8.155	2.73
-1.2183	3.57	-0.69898	9.10	8.2284	1.66
-0.59383	3.15	0.099592	22.63	9.206	0.58

Table 4.8: LTI Lateral Translational Body Velocity, v Stability Derivatives

L_v		L_p		L_ϕ		L_{lat}	
Parameter	% Error	Parameter	% Error	Parameter	% Error	Parameter	% Error
-7.6882	7.90	-18.218	7.06	2.7954	29.28	0.4613	6.26
-3.992	4.91	-9.7022	4.73	-2.2024	10.37	0.25718	4.38
-12.249	25.99	-28.303	25.50	10.014	40.88	0.66036	24.89
-6.8892	0.69	-22.055	6.38	5.3215	15.77	0.59356	6.03
-20.081	13.17	-47.831	12.62	17.513	18.75	1.16978	12.18
-4.0312	3.75	-11.41	2.78	1.7048	22.98	0.36272	2.49
-5.3975	3.22	-15.675	2.32	2.2617	16.21	0.50092	2.09
-4.3963	7.87	-14.613	3.98	2.1844	29.82	0.4699	3.17
-7.5162	7.37	-19.727	6.16	5.017	17.84	0.57684	5.44
-4.8681	2.70	-14.453	2.29	0.81332	32.83	0.426	2.11

Table 4.9: LTI Lateral Rotational Body Velocity, p Stability Derivatives

The added parameters quickly reveal the complexity they add to the problem. Rather than the small error seen in each parameter estimate for both the heave and yaw modes, several parameters have high percentage errors. This trait indicates that the routines had trouble fitting a value to that parameter to make a significantly beneficial change to the model. Despite the rather large variance apparent in this data, it is assumed that taking the mean will provide a sufficiently good model estimate. From a cursory glance, the model does seem to have a Gaussian spread, with only one or two outliers on either side of the mean. For the majority of the data, the parameter estimates are within an acceptable range of the mean estimate. Therefore, the same statistics presented for the heave and yaw modes are available in Table 4.10.

Parameter	Mean	Standard Deviation	Standard Error Mean
Y_v	-0.82356	0.526239	± 0.039675
Y_p	0.020749	0.631299	± 0.057006
Y_ϕ	8.12732	0.974416	± 0.116765
L_v	-7.71087	5.012403	± 0.803691
L_p	-20.1987	11.1001	± 1.922082
L_ϕ	4.538672	5.581698	± 1.182505
L_{lat}	0.547856	0.247603	± 0.0457532

Table 4.10: LTI Lateral Derivatives' Statistics

The next step is to reduce the working data to 9 sets, and reserve one flight validating the statistically averaged model. The flight selected to validate the average model is the 7th flight. Removing this flight changes the mean parameter values to those seen in Table 4.11. When using these parameters to populate the lateral model in Equation 4.31, the simulated data matches up well with the measured flight data from flight 7 (refer to Figure 4.23). Therefore the assumption is to use the parameters in Table 4.11 to populate the full inner loop dynamics matrix.

The model is accurate, even at some of the higher frequency excitations. There are a few areas where the model and test data do not line up very well, mostly in the lateral velocity v model. This could be due to the discrepancy of including the Y_p parameter. It had a large standard error throughout most flights, indicating it may not be a good regressor for the model. However, it was included only because the magnitude of the derivative was relatively small. Another cause for

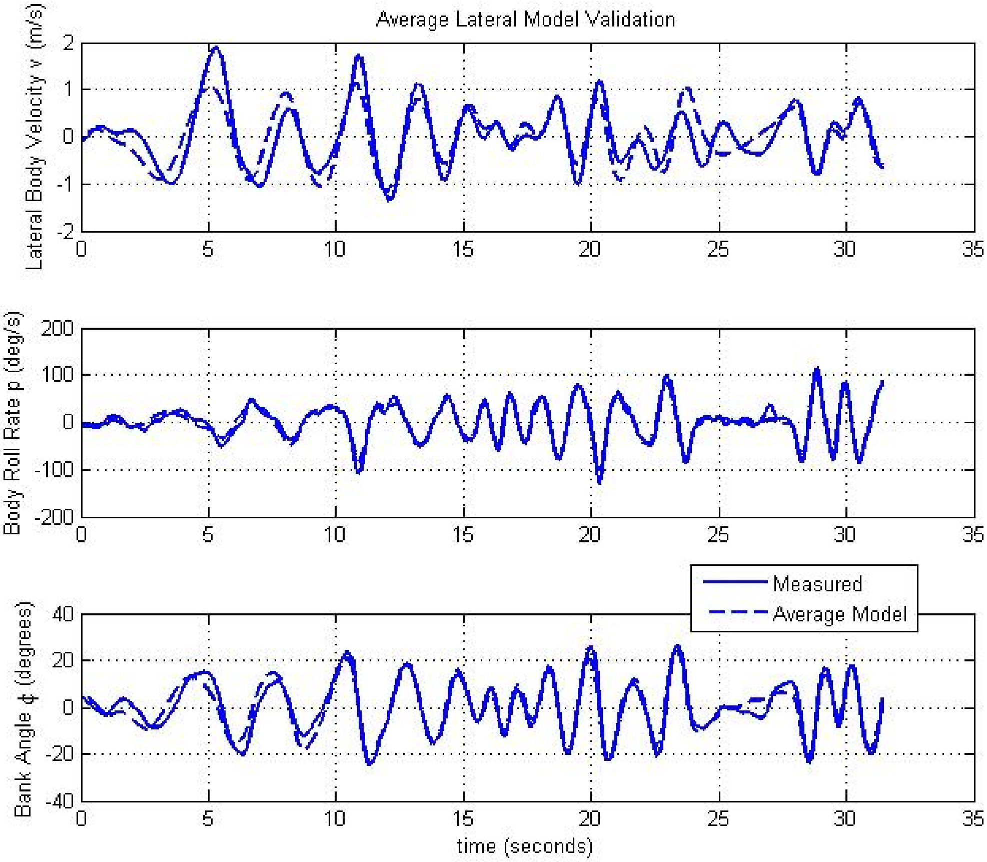


Figure 4.23: Averaged Lateral Model Comparison to Separate Flight Data

Parameter	Value
Y_v	-0.82007
Y_p	0.016868
Y_ϕ	8.022955
L_v	-7.50056
L_p	-19.7875
L_ϕ	4.331675
L_{lat}	0.543589

Table 4.11: Final Lateral Derivative Values

discrepancy in this model could be the value of the Y_ϕ term. Recall from §4.4, this term should be approximately the gravitational constant g, which should be close to 9.81. The natural stiffness term is a red flag too. The standard deviation in the data sets is higher than the actual mean. The standard error is also large for each estimate obtained. This means that the term may not be well-suited or just difficult to identify. However, the eigenvalues of the lateral mode are all stable, therefore it is assumed to take this form to complete the model. To avoid convoluting the model too much by substituting in expected values, the model is kept with the estimates derived. And therefore the parameters in Table 4.11 are those that will be used to determine the bare airframe dynamics.

4.4.4 Longitudinal Mode

The process and results of the the longitudinal mode strongly resemble those of the lateral mode presented in §4.4.3. A small feedback gain was implemented on the longitudinal controls using proportional feedback from the pitch rate gyro. 10 test flights were conducted during which the longitudinal mode was excited. The identification performed is that of the closed loop system, due to the same reasons the lateral mode was identified as the closed loop system. The feedback control was too noisy to discern the rigid body effects, so the cleaner pilot stick inputs were used and the closed loop system was identified with the bare airframe model to be calculated later.

The structure of the longitudinal mode resembles the structure of the lateral mode seen in Equation 4.31. The decoupled nature of the longitudinal mode was tested too using '*swr.m*' and including all states in the regression. The findings were the same as the three previous cases. The longitudinal derivatives were the only significantly contributing regressors. Therefore, the longitudinal mode is decoupled from the other modes, and this separation is possible, and the longitudinal mode can be identified separately from the other 3 modes. The longitudinal dynamics are found in Equation 4.32.

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_q & X_\theta \\ M_u & M_q & M_\theta \\ 0 & \Theta_q & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ M_{lon} \\ 0 \end{bmatrix} \delta_{lon} \tag{4.32}$$

The anticipation is that the kinematic parameter Θ_q will be approximately 1. Also, the stability derivatives X_u and M_q will be negative indicating positive damping. Since the vehicle has feedback control implemented this should definitely be the case. X_θ should be negative and have a magnitude close to the gravitational constant $-g$. From observation of the vehicle behavior, the natural stiffness M_θ should be negative as well. The control derivative, M_{lon} should be positive, since the longitudinal input causes a positive rotation about the body y-axis $\hat{e}_y$. The other two parameters including the translational and rotation coupling effects X_q and M_u are relatively unknown.

As was the case in the previous three mode identifications, the output error method improved upon the stepwise regression estimates. The model has a more difficult time estimating the behavior as seen in Figure 4.24, but nonetheless the output error method improves the accuracy. And therefore the procedure used in the other analyses will be used for the last mode.

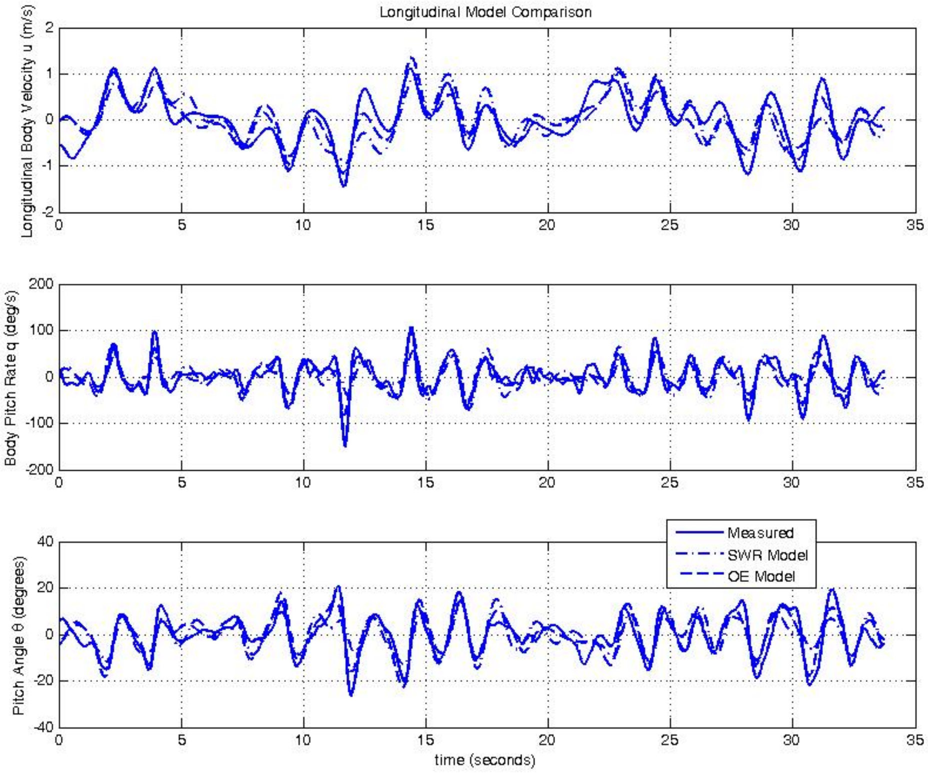


Figure 4.24: Longitudinal Model Comparison

Upon proceeding with this routine the derivatives obtained are presented in Tables 4.12 and 4.13. The spreads present in each parameter is on the same order as that in the lateral. This is due to the fact that the two modes have the same number of unknown parameters. Therefore, the complexity of the identification is of the same degree. Not shown is the kinematic relationship parameter between $\dot{\theta}$ and q, Θ_q. This is because the anticipated value of 1 held across all flights. Hence, the value used in the model is 1. The standard error is also left out again. If desired, it can easily be determined by multiplying the decimal value of the % error by the parameter estimate value.

X_u		X_q		X_θ	
Parameter	% Error	Parameter	% Error	Parameter	% Error
-0.2004	13.28	-0.21191	19.96	-10.287	0.98
-0.42149	7.45	-0.06105	68.31	-10.586	1.13
-0.27055	12.22	- 0.13251	28.21	-10.023	1.02
-0.47435	5.88	0.11572	32.56	-10.305	0.97
-0.64059	7.64	0.15584	34.37	-9.8984	1.50
-0.5735	7.36	0.2164	25.46	-8.1411	0.18
-0.57697	6.09	0.12209	29.13	-7.4173	0.21
-0.34154	8.68	-0.05345	60.65	-8.9373	1.04
-0.61581	5.51	0.17822	21.06	-8.6973	1.34
-0.3762	8.00	-0.07195	35.05	-8.8787	0.99

Table 4.12: LTI Longitudinal Translational Body Velocity, u Stability Derivatives

M_u		M_q		M_θ		M_{lon}	
Parameter	% Error	Parameter	% Error	Parameter	% Error	Parameter	% Error
6.3642	2.38	-6.404	2.43	-15.247	1.77	0.23372	2.31
24.139	7.68	-25.535	7.71	-29.934	5.71	0.79274	7.57
17.351	4.44	-17.098	4.45	-28.008	3.37	0.5393	4.39
19.656	4.09	-18.835	4.15	-29.104	3.47	0.72708	3.99
19.182	10.19	-25.257	10.00	-35.389	6.96	0.77018	9.76
13.327	0.53	-14.377	0.51	-26.767	3.91	0.50244	4.78
18.737	0.25	-18.67	0.21	-37.132	0.23	0.91128	1.86
26.415	3.72	-27.169	3.64	-35.441	3.18	0.97164	3.56
16.716	2.87	-15.037	2.70	-20.736	3.03	0.56722	2.45
26.691	3.66	-23.312	3.55	-23.147	3.02	0.8116	3.45

Table 4.13: LTI Longitudinal Rotational Body Velocity, q Stability Derivatives

The standard errors for the longitudinal case tend to be much smaller than those in the lateral case, with the exception of the X_q derivative. This high error is indicative that perhaps this is not a parameter to include in the model. The small errors across the other estimates is a good indication that the data is fit well with the model. Therefore, to get one value for the parameters from the 10 values, the mean is taken of the data. The standard deviation is also provided with the mean of the standard error for comparison purposes in Table 4.14.

Parameter	Mean	Standard Deviation	Standard Error Mean
X_u	-0.44914	0.152134	± 0.033897
X_q	0.025739	0.148614	± 0.039846
X_θ	-9.31711	1.058079	± 0.089835
M_u	18.85782	6.152287	± 0.809117
M_q	-19.1694	6.380401	± 0.852554
M_θ	-28.0905	6.981827	± 0.998251
M_{lon}	0.68272	0.221625	± 0.031075

Table 4.14: LTI Longitudinal Derivatives' Statistics

The longitudinal data is less spread than the lateral. And because the averaged model did a good job in the lateral identification the same process is employed for the longitudinal mode. The last test flight in this case is the flight removed from the statistical analysis and used to validate the model derived from the other 9 tests. The updated parameters for the longitudinal mode are presented in Table 4.15. A plot comparing the model and measured data can be seen in Figure 4.25.

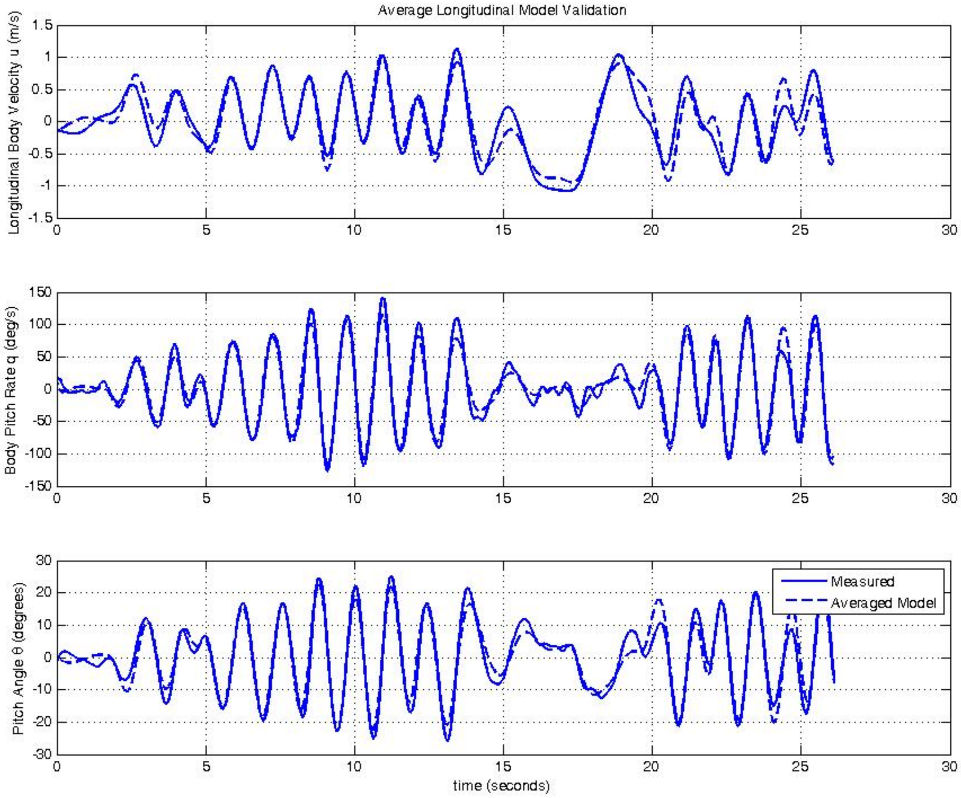


Figure 4.25: Averaged Longitudinal Model Comparison to Separate Flight Data

The averaged model does a good job of matching the measured data. Indicating that the values present in Table 4.15 are appropriate estimates for the parameters in the state space system in Equation 4.32. The signs and magnitudes of the estimates all make intuitive sense. The X_θ term is very close to the anticipated value of the negative gravitational constant $-g$. The damping terms all have negative values. This was to be expected, especially since the vehicle has rate damping

Parameter	Value
X_u	-0.44251
X_q	0.016858
X_θ	-9.277255
M_u	19.56993
M_q	-19.546
M_θ	-27.6411
M_{lon}	0.6944364

Table 4.15: Final Longitudinal Derivative Values

feedback implemented. Similar to the findings in the lateral mode identification, the rotating to translation coupled term X_q is very spread across all tests. Averaging this results in a magnitude that is negligible. However, as was done in the lateral process the estimate was included because of its small magnitude, despite the likelihood that it should not be. With these final parameters, the entire closed loop state space system from Equation 4.25 and Equation 4.26 is completely identified. The last step in obtaining the bare airframe dynamics is to back out the open loop system using the closed loop model. To do this, the feedback contributions to the closed loop system must be available. This requires the knowledge of three matrices, the controls matrix **B**, which was finalized in the process above. The other is the gain matrix **K**, which is known, as it is a controllable variable in the test flights. For the flights conducted to provide the data in this section the gain matrix of the form in Equation 4.3, is enumerated in Equation 4.33.

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0.85 & 0 \\ 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.33}$$

This leaves the last matrix, the inner loop sensor matrix **C**, to be determined. Once this is completed the bare airframe dynamics can be extracted from the entire closed loop system presented in this analysis.

4.4.5 Inner Loop Observation Matrix

The inner loop sensor matrix, the matrix that characterizes the dependence of the inner loop outputs on the states, must be identified in order to proceed with the bare aiframe dynamics extraction. A first cut approach would be to determine the conversion factors present on the gyro and accelerometer chips. However, a number of hard-coded scale factors are present in the firmware code. These can rapidly complicate the process involved with obtaining the conversion factors. Another complication present is with how the attitude angle estimates $\hat{\phi}$, and $\hat{\theta}$ are calculated. Recall the complementary filter uses both the gyro measurements as well as the accel measurements. Because the filter is complementary in nature (as the name suggests) the two portions of the filter have unity gain across all frequencies, and therefore no extra dynamics are assumed. However, there is still a rather complicated mixture of scale factors and conversion factors, along with the sine and cosine functions involved in obtaining an angle estimate from the gravity vector. Fortunately, the inner loop outputs can be transmitted off the vehicle, and recorded. This provided the opportunity to use the system identification software in SIDPAC to determine the linear relationships of the measurements to the states, thus providing the inner loop **C** matrix.

Because this is strictly a linear equation as the outputs $\vec{y}$ are just scalar combinations of the states $\vec{x}$, through the standard output equation

$$\vec{y} = \mathbf{C}\vec{x} \tag{4.34}$$

where

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} & c_{17} & c_{18} & c_{19} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} & c_{27} & c_{28} & c_{29} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} & c_{36} & c_{37} & c_{38} & c_{39} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} & c_{47} & c_{48} & c_{49} \\ c_{51} & c_{52} & c_{53} & c_{54} & c_{55} & c_{56} & c_{57} & c_{58} & c_{59} \end{bmatrix}, \tag{4.35}$$

there are no dynamics involved. Therefore there are no derivatives, and each element of $\vec{y}$ will be a linear combination of the elements of $\vec{x}$. This means that the output error function '*oe.m*' would not be useful, as it assumes a full state space system involved. To avoid complicating the estimation of the **C** matrix, the identified system presented in the previous sections is not included. Because the relationship is a linear equation, the equation estimation technique present in '*swr.m*' would be an appropriate method for determining the parameters.

Determining the regressors for each output was simple. Obviously the only regressors that should be included for the gyro measurements were the Vicon$^{\text{TM}}$ measured angular rates. For the regression analysis done on each gyro measurement, all angular rates were included, to determine if the off-center placement of the gyro chip would cause some inter-axis coupling. The findings were minimal, and the assumption that the gyro measurements were taken at the CG of the vehicle held, and the estimates of the angular rates were just a scaled version of the actual angular rates. Because of the use of the gyro measurements in determining the estimates of ϕ and θ, the angular rates were also included in the regression for these estimates, $\hat{\phi}$, and $\hat{\theta}$. These assumptions alter the structure of the **C** matrix in Equation 4.35, and reduce the number of elements to that seen in Equation 4.36.

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & c_{14} & 0 & 0 & c_{17} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{25} & 0 & 0 & c_{28} & 0 \\ 0 & 0 & 0 & c_{34} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{45} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{56} & 0 & 0 & 0 \end{bmatrix} \tag{4.36}$$

Under the assumption that this is the structure of the inner loop **C** matrix, the resulting calculation of the 7 parameters is straightforward. The vehicle was rotated about each axis, without the motors running to avoid unnecessary vibration noise. Each axis was recorded separately and analyzed after the data was collected and post-processed. The results were good enough, that multiple runs were deemed unnecessary. The coefficient of determination for each case was above 0.9, indicating an accurate fit.

The first onboard measurement to be characterized was the bank angle estimate $\hat{\phi}$. This meant that the entries c_{14} and c_{17} are to be determined. Using a set of data where the vehicle was rotated about its body x-axis $\hat{e}_x$, the values of the **C** matrix were identified using '*swr.m*'. The model fit is seen in Figure 4.26. The values of the entries are provided in Table 4.16. The model does a good job of fitting the measured data, and the coefficient of determination is 0.9234, indicative of a strong correlation.

Entry	Value
c_{14}	-2.5671
c_{17}	70.79

Table 4.16: Bank Angle Linear Combination (R^2 = 0.9234)

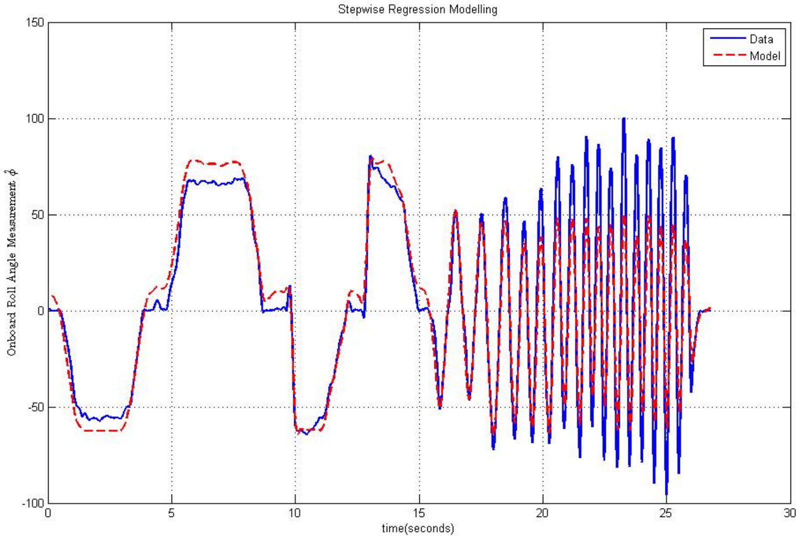


Figure 4.26: Modeling Onboard Bank Angle Estimate $\hat{\phi}$

The next measurement analyzed is the second onboard output, the pitch angle estimate $\hat{\theta}$. The regressors to use as indicated by the entries being identified (c_{25} and c_{28}), are the Vicon$^{\text{TM}}$ measured pitch angle θ, and the pitch rate q. To collect the data necessary to identify these parameters, the vehicle was rotated about its body y-axis $\hat{e}_y$ resulting in varying frequency pitch motions. The values calculated are presented in Table 4.17, and the fit this model provides is seen in Figure 4.27. As was the case with the roll angle estimate, the coefficient of determination is strong, and the model fit reflects that.

The next measurements are all anticipated to have the same magnitude of conversion factors. They are the onboard estimates of the body angular rates $\hat{p}$, $\hat{q}$, and $\hat{r}$. The roll and pitch rate estimates were calculated using the same data that was used to calculate the relationships of the roll and pitch angle estimates. A separate yaw case had to be done where the vehicle was

Entry	Value
c_{25}	-1.4163
c_{28}	65.289

Table 4.17: Pitch Angle Linear Combination ($R^2 = 0.9843$)

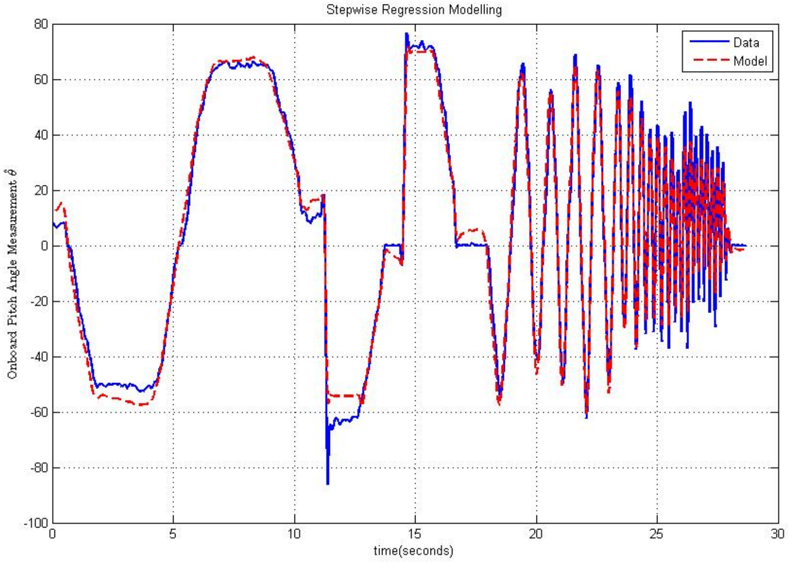


Figure 4.27: Modeling Onboard Pitch Angle Estimate $\hat{\theta}$

rotated about its body z-axis $\hat{e}_z$ at sweeping frequencies. The onboard measurements have only one dependent variable, and therefore the coefficient of determination is dependent only on how well the model is fit with just the one value of the **C** matrix. Table 4.18 presents the conversion factors for each axis rotation, and the coefficient of determination that value provides the model.

Entry	Value	Coefficient of Determination
c_{34}	44.647	0.9744
c_{45}	48.153	0.9144
c_{56}	45.285	0.9556

Table 4.18: Angular Rate Conversion Factors

Figures 4.28, 4.29, and 4.30 all depict how well the estimated models follow the provided data.

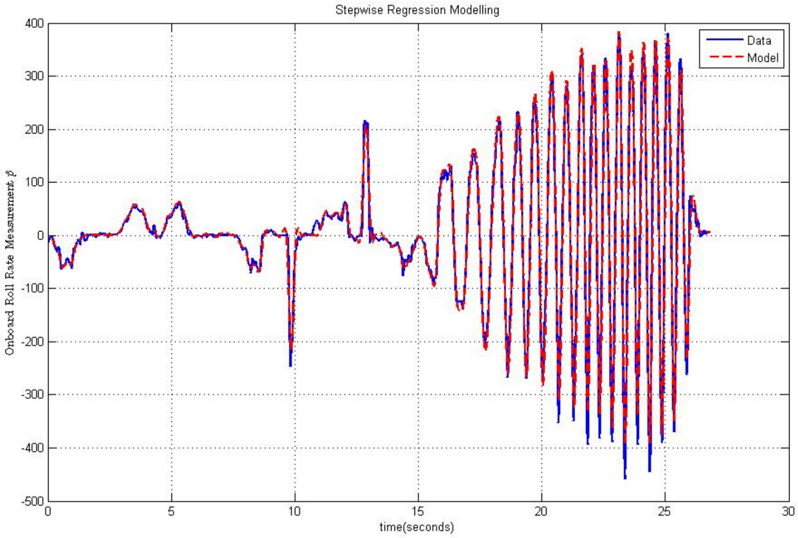


Figure 4.28: Modeling Onboard Roll Rate Estimate $\hat{p}$

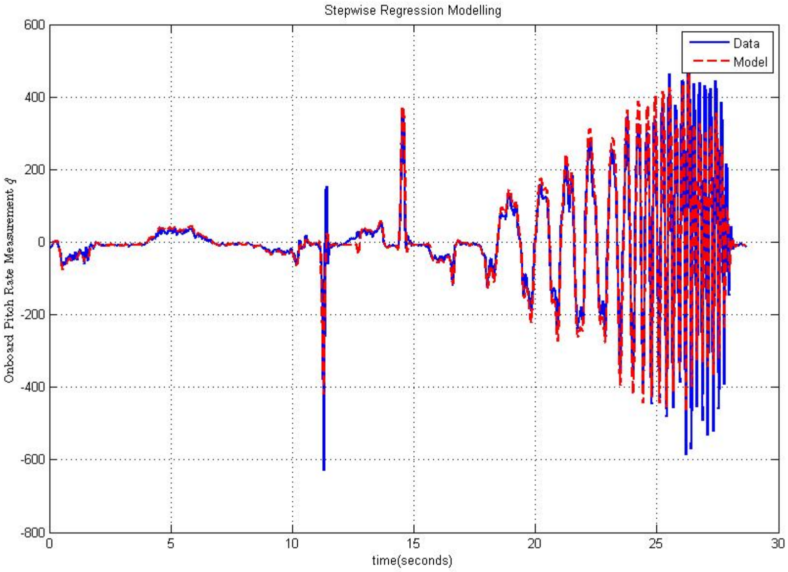


Figure 4.29: Modeling Onboard Pitch Rate Estimate $\hat{q}$

All three models are well suited to predict the measured data. Therefore, these three parameters, along with the four identified values for the roll and pitch angle estimates, are appropriate estimates to use to populate the onboard observation matrix **C**.

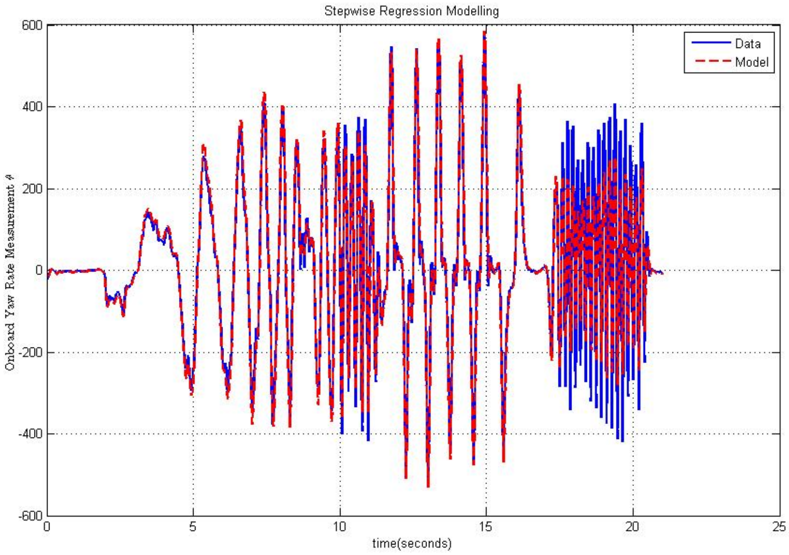


Figure 4.30: Modeling Onboard Yaw Rate Estimate $\hat{r}$

4.5 Bare Airframe Dynamics

Now that all 4 necessary matrices are determined, the open loop dynamics matrix can be calculated through Equation 4.2. To summarize the matrices populated with the parameter estimates obtained in the prior sections, the closed loop dynamics matrix $\mathbf{A}_{CL}$ is presented in Equation 4.37, the controls matrix, $\mathbf{B}$ is provided in Equation 4.38, the inner loop observation matrix, $\mathbf{C}$ is provided in Equation 4.39, and the feedback gain matrix, $\mathbf{K}$ is reiterated in Equation 4.40.

$$\mathbf{A}_{CL} = \begin{bmatrix} -0.44251 & 0 & 0 & 0 & 0.016858 & 0 & 0 & -9.277255 & 0 \\ 0 & -0.82007 & 0 & 0.016868 & 0 & 0 & 8.022955 & 0 & 0 \\ 0 & 0 & -0.51636 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7.71087 & 0 & -20.1987 & 0 & 0 & 4.538672 & 0 & 0 \\ 19.56993 & 0 & 0 & 0 & -19.546 & 0 & 0 & -27.6411 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.75782 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.37}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.013652 \\ 0 & 0.543589 & 0 & 0 \\ 0.6944364 & 0 & 0 & 0 \\ 0 & 0 & 0.111737 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.38}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & -2.5671 & 0 & 0 & 70.79 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.4163 & 0 & 0 & 65.289 & 0 \\ 0 & 0 & 0 & 44.647 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 48.153 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 45.285 & 0 & 0 & 0 \end{bmatrix} \tag{4.39}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0.85 & 0 \\ 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.40}$$

If it is decided that the throttle command δ_{thr} is not to be included, then the last column of the **B** matrix can be removed, along with the bottom row of the **K** matrix. This results in the same computation, because the bottom row of the feedback gain matrix is all zeros. Hence, these are included for completion.

The calculation of the open loop dynamics matrix **A** from Equation 4.2, is as follows.

$$\mathbf{A} = \begin{bmatrix} -0.44251 & 0 & 0 & 0 & 0.016858 & 0 & 0 & -9.277255 & 0 \\ 0 & -0.82007 & 0 & 0.016868 & 0 & 0 & 8.022955 & 0 & 0 \\ 0 & 0 & -0.51636 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7.71087 & 0 & 0.4305 & 0 & 0 & 4.538672 & 0 & 0 \\ 19.56993 & 0 & 0 & 0 & 8.8773 & 0 & 0 & -27.6411 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.75782 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.41}$$

Notice the only two derivatives that change are the rotational damping terms for roll and pitch. This is because the only feedback implemented is the roll and pitch rate damping, and the system is decoupled so these have no effect on any other states. It's important to note that the signs of derivatives have now switched to positive, indicative of negative damping. This may be physically counter-intuitive, because it is anticipated that due to the aerodynamic viscous drag the vehicle should have positive damping, i.e. it should come to a stop if perturbed, not continue to rotationally accelerate. However, it was anticipated that the vehicle be dynamically unstable. While these terms may hint at instability, there are other restorative terms in the lateral and longitudinal modes that may provide other behavior not apparent just by looking at the sign and magnitude of these two derivatives. Hence a look at the eigenvalues or poles of the system's modes is necessary. These are presented in Table 4.19.

Eigenvalue	Damping Ratio	Frequency (rad/s)	DOF
-3.2	1.00	3.2	Longitudinal Translation
$5.82 \pm 5.17\jmath$	-0.747	7.78	Longitudinal Rotation
-4.43	1.00	4.43	Lateral Translation
$2.02 \pm 3.01\jmath$	-0.557	3.62	Lateral Rotation
-0.516	1.00	0.516	Heave
-0.758	1.00	0.758	Yaw
0	-1	0.00	Heading

Table 4.19: Bare Airframe Poles

Upon inspection of the poles of the system, it's apparent that the rotational DOFs for pitch and roll are both unstable. The pitch rotation has a higher natural frequency than its lateral counterpart. This, again is counter-intuitive as it is not explained in the physics of the system. It could be due to a number of reasons though. There could be unmodeled dynamics that manifest instabilities. It could be an effect of a delay in the motor/ESC combination. It could be an artifact of the post-processing of the data, or the feedback control present. Because of these uncertainties, it is important to check this finding.

One method to validate this, is to use the total control inputs, rather than just the pilot stick inputs. This would only affect the calculation of the pitch and roll rotational derivatives as seen from the calculation of the feedback effects in Equation 4.42.

$$\mathbf{BKC} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20.629 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 28.423 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.42}$$

To redo the analysis on the pitch and roll rate stability derivatives, is an easy task. The process actually ends up easier than that involved with the closed loop identification. The reason behind this is that the '*oe.m*' function has trouble converging on account of the unstable nature of the system. This hints at the likelihood that the model identified in Equation 4.41 is actually accurate. Because output error is ineffective for this analysis only '*swr.m*' is used. Instead of using the pilot reference inputs as a regressor, the total inputs δ_{lon} and δ_{lat} are fed to the function. Upon redoing this analysis in the described method, the results are as expected. The open loop stability derivatives L_p and M_q are still positive, suggesting the longitudinal and lateral rotational DOFs are unstable. The values for these (Table 4.20) are calculated using the stepwise regression routine for the same data used to calculate the results presented in Tables 4.8, 4.9, 4.12, 4.13.

While these results agree with the conclusion that the system is actually unstable, it is still important to verify this in another manner. The magnitude of the estimates aren't necessarily on the same scale as those predicted in Equation 4.41. This is likely an effect of not being able to refine these estimates through output error. The other method tested is to do a linear regression on the relationship between the magnitude of the feedback gain and the closed loop stability derivative. Because they are decoupled, and only have rate feedback, this means that the closed loop stability derivatives have a linear relationship with the value of the gain. Therefore, the open loop estimate can be backed out as the intercept of the linear regression (i.e. where the gain value is 0). This method was done for the lateral case only, as it was a very time-intensive procedure, requiring many flights and due to time restrictions, it was not carried out for the longitudinal case. The process is simple, collect data and analyze the rotational stability derivative at a particular gain.

Flight	L_p	M_q
1	0.95875	0.91567
2	1.0797	1.3461
3	1.1312	1.3918
4	0.93124	2.9406
5	0.98762	0.86171
6	1.1287	0.83962
7	1.5042	11.046
8	1.9801	3.4141
9	1.204	1.6679
10	1.4228	1.794
Mean	1.232831	2.62175
Standard Deviation	0.323144	3.084315

Table 4.20: Open Loop Calculation of Rotational Stability Derivatives

Then repeat as the gain is lowered until the vehicle is no longer flyable. The gains used ranged from 1.0 to 0.5; where any gain set lower than 0.5 made the vehicle not pilotable. The gains were usually decremented at an interval of 0.05, and multiple tests were conducted at each gain value. The results of this analysis can be seen in Figure 4.31.

Doing this regression results in an intercept (0-gain) of 1.5485, which is again positive, and yet further indication that the dynamics captured in this system are unstable without feedback. This is also on the same scale as that determined identifying the open loop using stepwise regression. Both of which are slightly larger than the magnitude of the estimate from Equation 4.41. It is apparent when examining Figure 4.31, that the spread is still rather large. Therefore, this method should not be used as the means for obtaining the bare airframe dynamics. This method, is best suited as a sanity check. The stepwise regression modeling of the open loop system is also not as exact because of the inability to refine the estimates through output error. Hence, it is also best suited as a validation to the results obtained in Equation 4.41. The longitudinal results, while not examined, are assumed to take the same form as that in Figure 4.31, It is assumed that

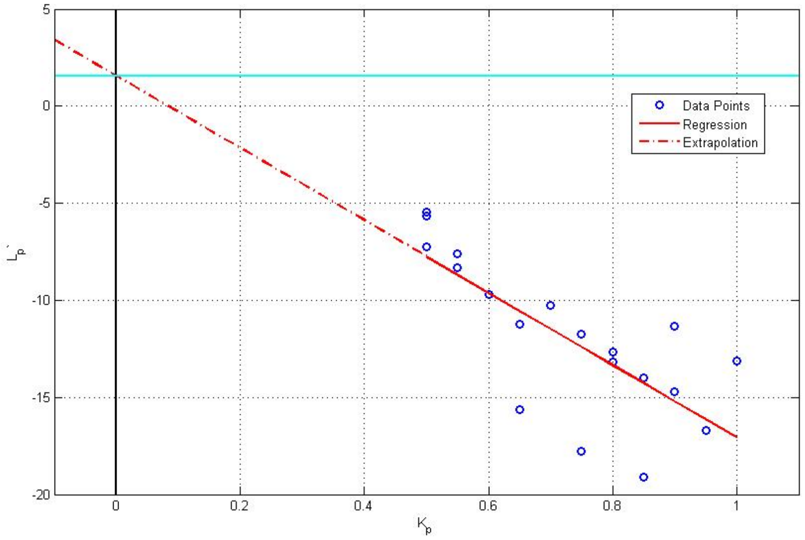


Figure 4.31: Linear Regression of L_p with Respect to Damping Derivative

because both the open loop stepwise regression modeling and the computation of the bare airframe dynamics matrix have the same sign and roughly similar magnitude for the M_q derivative then the same results will follow as they did for the L_p derivative. Therefore the open loop model identified in Equation 4.41 is validated, and the control law design for the model can begin.

Chapter 5

Control Law Design

5.1 System Overview

5.1.1 Inner Loop Control Structure

Now that the microquad has a good LTI state space model about hover, all the well known and commonly used control tools are available. There are numerous options available for any chosen performance criteria. However, the model enumerated in Equations 4.38, 4.39, and 4.41 does not offer much realistic use for the microquad without massaging the model into two different problems. It is useful to separate the full 6 DOF dynamics (including inertial position) into two separate control loops. The inner loop will implement the feedback available from the Mote, and will therefore regulate attitude of the microquad, but not heading. The yaw rate measurement available offers damping for the yaw DOF, but the yaw angle cannot be accurately integrated. A heading lock gyro is not feasible, as the noise present in the measurements integrates off to infinity before the vehicle can take off. Therefore, the inner loop measurements are those mentioned in Chapter 4, and shown in Equation 5.1.

$$\vec{y}_{\mathrm{I}} = \begin{bmatrix} \hat{\phi} \\ \hat{\theta} \\ \hat{p} \\ \hat{q} \\ \hat{r} \end{bmatrix} \tag{5.1}$$

These measurements leave two unobservable modes, (rank of observability matrix $\operatorname{rank} \mathcal{O} = 7$). Because the yaw mode is unobservable, and a pole at the origin, it is technically not detectable, making many of the useful control synthesis techniques are not possible. Therefore it is useful to remove several unused modes that present problems. Particularly, the heading angle ψ is removed from the dynamics. The pole at the origin creates several problems when trying to shape the system dynamics. Many methods ultimately attempt to shape the closed loop dynamics, and a pole that can't be moved from the imaginary axis poses a problem that cannot be solved. In addition to the removal of the yaw angle, the effect of the throttle input is removed. The reasoning behind this, is that the heave mode is decoupled from the other modes, and therefore without any direct measurement of the heave variables, it cannot be controlled. Some control synthesis routines may introduce unnecessary coupling due to the underlying math. Hence the throttle channel is removed from the system as well. Removing these modes results in a new inner loop model outlined below.

$$\mathbf{A}_{\mathrm{I}} = \begin{bmatrix} -0.44251 & 0 & 0 & 0 & 0.016858 & 0 & 0 & -9.277255 \\ 0 & -0.82007 & 0 & 0.016868 & 0 & 0 & 8.022955 & 0 \\ 0 & 0 & -0.51636 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7.71087 & 0 & 0.4305 & 0 & 0 & 4.538672 & 0 \\ 19.56993 & 0 & 0 & 0 & 8.8773 & 0 & 0 & -27.6411 \\ 0 & 0 & 0 & 0 & 0 & -0.75782 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.2}$$

$$\mathbf{B}_{\mathrm{I}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.543589 & 0 \\ 0.6944364 & 0 & 0 \\ 0 & 0 & 0.111737 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.3}$$

$$\mathbf{C}_{\mathrm{I}} = \begin{bmatrix} 0 & 0 & 0 & -2.5671 & 0 & 0 & 70.79 & 0 \\ 0 & 0 & 0 & 0 & -1.4163 & 0 & 0 & 65.289 \\ 0 & 0 & 0 & 44.647 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 48.153 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 45.285 & 0 & 0 \end{bmatrix} \tag{5.4}$$

The inner loop will also have modeled disturbances $\vec{d}_{\mathrm{I}}$, included to offer another level of performance characterization. The inner loop controller will be designed to be robust to gust disturbances. The gust disturbances will be shaped through a Gauss-Markov filter, thus the disturbance inputs to the augmented model will be white noise disturbances. This offers stochastic methods for determining variance in the outputs. Due to the decoupled nature of the dynamics, gust disturbances of varying magnitude will be implemented across all inner loop DOFs $\vec{d}_{\mathrm{I}_G} = [d_u, d_v, d_w, d_r]^T$. In addition to these gust disturbances, the measurement noise will manifest itself as white noise on the longitudinal and lateral input channels due to the feedback control $\vec{d}_{\mathrm{I}_n} = [d_{n_{lon}}, d_{n_{lat}}]^T$. Therefore the total disturbance vector modeled is the combination of the gusts and the input chatter.

$$\vec{d}_{\mathrm{I}} = \begin{bmatrix} \vec{d}_{\mathrm{I}_G} \\ \vec{d}_{\mathrm{I}_n} \end{bmatrix} = \begin{bmatrix} d_u \\ d_v \\ d_w \\ d_r \\ d_{n_{lon}} \\ d_{n_{lat}} \end{bmatrix} \tag{5.5}$$

The gust disturbances are best represented in a state space system of their own. The Gauss-Markov shaping filter is seen in Equation 5.6.

$$\begin{bmatrix} \dot{d}_u \\ \dot{d}_v \\ \dot{d}_w \\ \dot{d}_r \end{bmatrix} = \begin{bmatrix} -1/\tau & 0 & 0 & 0 \\ 0 & -1/\tau & 0 & 0 \\ 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & -1/\tau \end{bmatrix} \begin{bmatrix} d_u \\ d_v \\ d_w \\ d_r \end{bmatrix} \begin{bmatrix} b & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & b \end{bmatrix} \begin{bmatrix} q_u \\ q_v \\ q_w \\ q_r \end{bmatrix} \tag{5.6}$$

Where q_u, q_v, q_w, and q_r are white noise inputs of varying intensity to best match the likely intensity of disturbances in the physical system. The time constant of the filter is chosen to be τ = 3.2 sec, and $b = 1/2$ [20]. The next step is to augment the inner loop dynamics with these states to ensure the only disturbance inputs to the system are white noise. The augmented system in block form is presented in Equation 5.7.

$$\dot{\vec{x}}_a = \begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{d}}_{\mathrm{I}_G} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{D}_{\mathrm{G}} \\ \mathbf{0} & \mathbf{A}_{\mathrm{d}} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{d}_{\mathrm{I}_G} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{u} \end{bmatrix} + \begin{bmatrix} \mathbf{D}_{\mathrm{n}} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{\mathrm{d}} \end{bmatrix} \begin{bmatrix} \vec{d}_{\mathrm{I}_{\mathrm{n}}} \\ \vec{q} \end{bmatrix} \tag{5.7}$$

Where $\mathbf{D}_{\mathrm{G}}$ is the gusts' effects on the dynamics of the matrix, which is typically taken as the negative of the stability derivatives of that respective DOF. $\mathbf{A}_{\mathrm{d}}$ is the dynamics matrix of the Gauss-Markov shaping filter in Equation 5.6, and $\mathbf{B}_{\mathrm{d}}$ is the input matrix, while $\vec{q}$ is the vector of white noise inputs to the Gauss-Markov filter. $\mathbf{D}_{\mathrm{n}}$ is the noise disturbance effect on the system, which is taken as the longitudinal and lateral columns of the controls matrix. Using these augmented states, the steady-state RMS covariance can be computed using simple stochastic techniques.

The inner loop controller will be best suited as a regulator, because the goal of the inner loop controller is to be sufficiently robust to disturbances while stabilizing the attitude of the vehicle. A basic block diagram of the raw inner loop dynamics is apparent in Figure 5.1. The reference input $\vec{u}_0$ will typically be left as 0, seeing as how the controller's performance should be graded

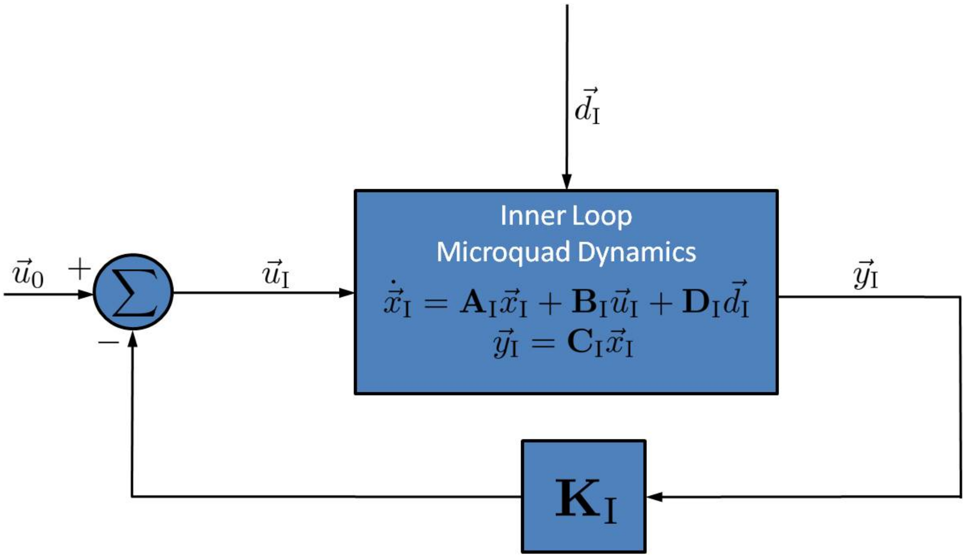

$\vec{d}_\mathrm{I}$
$\vec{u}_0$
+
$-$
$\sum$
$\vec{u}_\mathrm{I}$
Inner Loop
Microquad Dynamics
$\dot{\vec{x}}_\mathrm{I} = \mathbf{A}_\mathrm{I}\vec{x}_\mathrm{I} + \mathbf{B}_\mathrm{I}\vec{u}_\mathrm{I} + \mathbf{D}_\mathrm{I}\vec{d}_\mathrm{I}$
$\vec{y}_\mathrm{I} = \mathbf{C}_\mathrm{I}\vec{x}_\mathrm{I}$
$\vec{y}_\mathrm{I}$
$\mathbf{K}_\mathrm{I}$

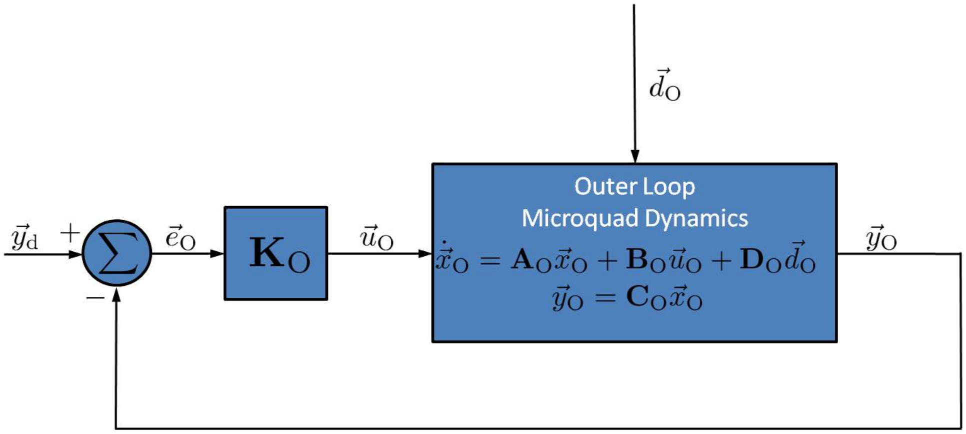
Figure 5.1: Inner Loop Control Structure

on the regulating capabilities. It's also important to note that the outer loop controller will be developed around this inner loop, with active control. Therefore the inner loop dynamics will be the closed loop dynamics, i.e. $\mathbf{A}_{\mathrm{I}_\mathrm{CL}} = \mathbf{A}_\mathrm{I} - \mathbf{B}_\mathrm{I}\mathbf{K}_\mathrm{I}\mathbf{C}_\mathrm{I}$. The outer loop controller will be designed to handle tracking motions.

5.1.2 Outer Loop Control Structure

The outer loop is used to hold the vehicle position, heading, and altitude. This is done by regulating the error signal of the desired position and heading and the measured. Therefore the controller determines the inputs based on the errors, $\vec{u}_\mathrm{O} = \mathbf{K}_\mathrm{O}\vec{e}$. Where $\vec{e} = \vec{y}_\mathrm{O} - \vec{y}_\mathrm{d}$. A basic regulator control loop representing the outer loop controller for the microquad can be seen in Figure 5.2.

In order to develop the outer loop controller, the outputs of the outer loop need to be defined and measurable. The outer loop outputs are defined in Equation 5.8. Fortunately, because of the



$\vec{d}_{\text{O}}$
Outer Loop
Microquad Dynamics
$\dot{\vec{x}}_{\text{O}} = \mathbf{A}_{\text{O}}\vec{x}_{\text{O}} + \mathbf{B}_{\text{O}}\vec{u}_{\text{O}} + \mathbf{D}_{\text{O}}\vec{d}_{\text{O}}$
$\vec{y}_{\text{O}} = \mathbf{C}_{\text{O}}\vec{x}_{\text{O}}$
$\vec{y}_{\text{d}}$
+
−
$\sum$
$\vec{e}_{\text{O}}$
$\mathbf{K}_{\text{O}}$
$\vec{u}_{\text{O}}$
$\vec{y}_{\text{O}}$

Figure 5.2: Outer Loop Control Structure

Vicon$^{\text{TM}}$ motion tracking system these are directly measured with minimal noise and latency.

$$\vec{y}_{\text{O}} = \begin{bmatrix} x \\ y \\ z \\ u \\ v \\ w \\ \psi \end{bmatrix} \tag{5.8}$$

While these are directly measured, they still need to be incorporated into the dynamics. The translational body velocities are already incorporated in the inner loop dynamics, and the yaw angle ψ is also included in the initial identified model, but removed for the inner loop controller design. Adding it back in presents no difficulty. Therefore the only thing to include is the inertial location $[x, y, z]^T$. Because the model is about hover, all the euler angles can be assumed small. This results in three simple relationships.

$$\dot{x} = u \tag{5.9}$$

$$\dot{y} = v \tag{5.10}$$

$$\dot{z} = w \tag{5.11}$$

These inclusions are easy, and offer an added benefit of acting as integrators for the body rates, thus ensuring good tracking [21]. In order to develop the outer loop controller using the full system, these relationships must be added to the dynamics. This is most easily done by concatenating the state vector with the 4 additional states. Doing so results in the following equation.

$$\vec{x}_{\mathrm{O}} = \begin{bmatrix} \vec{x}_{\mathrm{I}} \\ \psi \\ x \\ y \\ z \end{bmatrix} \tag{5.12}$$

Altering the dynamics to include the new states as shown in Equation 5.12, is done easily using the relationships from Equations 5.9 - 5.11. The new outer loop dynamics matrix takes the form seen in Equation 5.13.

$$\mathbf{A}_{\mathrm{O}} = \begin{bmatrix} \mathbf{A}_{\mathrm{I_{CL}}} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} \end{bmatrix} \tag{5.13}$$

Where $\mathbf{A}_{\mathrm{I_{CL}}} = \mathbf{A}_{\mathrm{I}} - \mathbf{B}_{\mathrm{I}}\mathbf{K}_{\mathrm{I}}\mathbf{C}_{\mathrm{I}}$ (as mentioned earlier). And $\mathbf{F}$ is the equivalent of adding the integrators in. It is simply matching the derivatives to the states, as seen in Equation 5.14.

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.14}$$

The rest of the state space system is relatively easy to determine. The outer loop controls matrix reintroduces the throttle channel, and a zero block for the dynamics of the additional outer loop states, as depicted in Equation 5.15.

$$\mathbf{B}_{\mathrm{O}} = \begin{bmatrix} \mathbf{B}_{\mathrm{I}} & \mathbf{B}_{thr} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.15}$$

$\mathbf{B}_{thr}$ is simply the column removed from Equation 4.38, to arrive at Equation 5.3. The observation matrix is simple as well. It is just a matrix of 0's and 1's used to pick out the desired states from the state vector to get the desired structure of the output vector.

$$\mathbf{C}_{\mathrm{O}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.16}$$

Now the only step necessary for developing both inner and outer loop controllers is to define the outer loop disturbance matrix $\mathbf{D}_{\mathrm{O}}$. In order to do so, the disturbances must be identified. A good starting point for doing so is to use the same disturbances used to develop the inner loop controller (see Equation 5.5). The noise from the onboard measurements, although not explicitly modeled in the outer loop system, can be assumed as chatter on the input channels. The same method for augmenting the outer loop with the disturbances in Equation 5.7 can be repeated for the outer loop. Upon doing so, the control development can commence, beginning with the inner loop controller, since the outer loop dynamics depend on the gain set chosen for the inner loop.

5.2 Control Development

5.2.1 Inner Loop Controller

The inner loop controller is obviously the first step in developing an appropriate control scheme for the microquad. The goal of the inner loop controller is to be a stabilizing regulator capable of rejecting external and internal disturbances, such as wind gusts and measurement noise as mentioned previously. Because the control structure is set up as displayed in Figure 5.1, there are a multitude of options for control. Four methods are investigated in this section. The first will be the method used to stabilize the vehicle prior to obtaining the system model, a SISO approach, effectively implementing PD gains on the system outputs. The second will be a very popular method in MIMO systems known as LQG (Linear Quadratic Gaussian) - a combination of LQR (Linear Quadratic Regulator) and the theoretical optimal estimator, the Kalman Filter. The third will be a simpler form of the LGQ controller - output feedback LQR as described in [6]. It is often preferable to LQG because it does not require manipulating the measurements or estimating the states. The last method is a relatively new method proposed in [21], known as $\mathcal{H}_\infty$ output feedback. Where the robustness of $\mathcal{H}_\infty$ controllers is captured in a static set of gains applied to the output feedback. The four of these controllers will be compared and a best choice will be chosen based on the quickness of the response to a disturbance, and its ability to reject random disturbances.

5.2.1.1 Inner Loop SISO PD Controller

Despite the simplistic approach and the gain selection a priori of system knowledge, the SISO PD gains selected over an iterative process provide surprisingly good performance. There is little to be said about the selection process for the gains. As stated the process was iterative, where after a stabilizing set was found, they were continuously refined until the most desirable performance was observed. This procedure yielded a very symmetric set of gains, where the proportional gain applied to both the roll and pitch angles was the same. The same case followed for the roll and pitch body rates. The yaw derivative gain is meant to slow down the yaw rate significantly to prevent fast deviations from the desired heading angle. Even though the gains chosen are SISO gains, they can also be presented in a gain matrix as done so in Equation 5.17.

$$\mathbf{K}_0 = \begin{bmatrix} 0 & 1.25 & 0 & 2.00 & 0 \\ 1.25 & 0 & 2.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.75 \end{bmatrix} \tag{5.17}$$

This gain set provided stable closed loop poles with acceptable damping levels. The closed loop poles can be seen in Table 5.1, and compared with those from the open loop poles in Table 4.19 These could even be improved by increasing the gains further, however, due to the noise in the measurements, a higher gain caused the vehicle to twitch on account of the noise being fed through to the motors via the feedback, most notably through the proportional attitude feedback.

Eigenvalue	Damping Ratio	Frequency (rad/s)	DOF
-55.3	1.00	55.3	Longitudinal Translation
-0.95 ± 1.75$\jmath$	0.478	1.99	Longitudinal Rotation
-45.4	1.00	45.4	Lateral Translation
-0.876 ± 1.18$\jmath$	0.597	1.47	Lateral Rotation
-0.516	1.00	0.516	Heave
-9.61	1.00	9.61	Yaw

Table 5.1: SISO PD Controller Closed Loop Poles

A linear model for the microquad was developed in Simulink$^{\text{TM}}$, similar to the block diagram in Figure 5.3, to simulate the performance of each controller synthesized for the inner loop and outer loop. The model incorporates the gust disturbances shaped from a vector of varying magnitude intensity white noise $\vec{q}$, and measures them, as well as the noise on each of the measured outputs, added in as a vector of white noise $\vec{n}$. Because this is a linear analysis, few nonlinearities are introduced. The only one implemented is a saturation limit placed on the inputs $\vec{u}$. Simulink^{TM} offers the option to include many regularly occurring physical phenomenon into a model, such as a rate limits (e.g., velocity saturation), deadzones, quantizations, and delays. This offers the ability to improve and enhance the simulation already created in the future for better model validation.

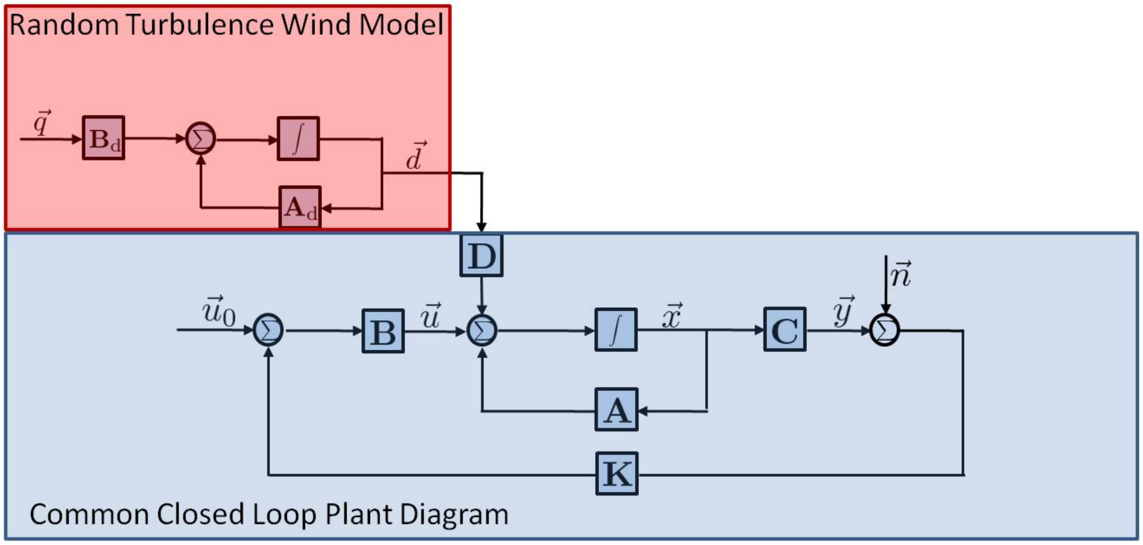


Figure 5.3: Simulation Model Block Diagram

A limitation to the system identification process used, is that it lumps all the physical systems in the microquad into one model. While this may seem good intuitively, it makes modeling the nonlinearities difficult. For instance, the motors and ESC combination has a very apparent slew rate limit (velocity saturation) i.e., the motors can only spin up and slow down so fast. In the model they can do so infinitely, without the rate limit in place. However, implementing such a limit is non-physical. The inputs are the PWM duty cycles being sent to the motors, not the actual motor RPM. Having a separate model for the motors would be more useful for accurately predicting the microquad behavior, and implementing the controllers successfully on the vehicle.

Nonetheless the overall performance of the controllers can be modeled using the system identified, assuming the microquad's overall system behavior does not change. The general trends of each controller's settling times, overshoot, steady state error, and disturbance rejection can be quantified and a best-suited controller can be hypothesized based on the results for each of the controllers.

From the results of the closed loop poles in Table 5.1, the results are as anticipated. There is a fair amount of overshoot present in a corrective maneuver. And without the system knowledge taken into account when deciding the gains, the controller performance is good, but not great. The performance is presented in Figure 5.4, while the disturbances the model is subjected to are shown clearly in Figure 5.5. The vehicle is perturbed by an angle of 15°about its pitch and roll axes, while simultaneously the vehicle is subjected to gust disturbances on the order of 0.5 $^{\mathrm{m}}\!/_{\mathrm{s}}$. The measurement noise is shown in Figure 5.6. The noise is on the scale of what is actually experienced due to the motor vibrations. Unfortunately, these do manifest themselves into fast accelerations in the simulation, where they typically wouldn't in reality. The motors tend to act as a low pass filter, only feeding the lower frequency commands through because of the slew rates present in the RPM.

When investigating the performance in Figure 5.4, the vehicle does well at correcting a perturbation, but does have considerable overshoot. The response due to the gust disturbances introduces a lot of steady state RMS variance in the states of interest. These covariances are presented in Table 5.2. To calculate these values, a basic stochastic principle is used for systems driven by stochastic processes such as white noise. The steady state covariance matrix of the system $\mathbf{P}_{\mathrm{ss}}$ is the solution to the Lyapunov equation below [22].

$$\mathbf{A}_{\mathrm{CL}}\mathbf{P}_{\mathrm{ss}} + \mathbf{P}_{\mathrm{ss}}\mathbf{A}_{\mathrm{CL}}^{T} = -\mathbf{DWD}, \tag{5.18}$$

where $\mathbf{A}_{\mathrm{CL}}$ is the closed loop dynamics, using the specified controller, and $\mathbf{W}$ is the covariance matrix of the white noise inputs. The inputs in this case are uncorrelated white noise, and therefore the matrix is a diagonal matrix of the noise intensities.

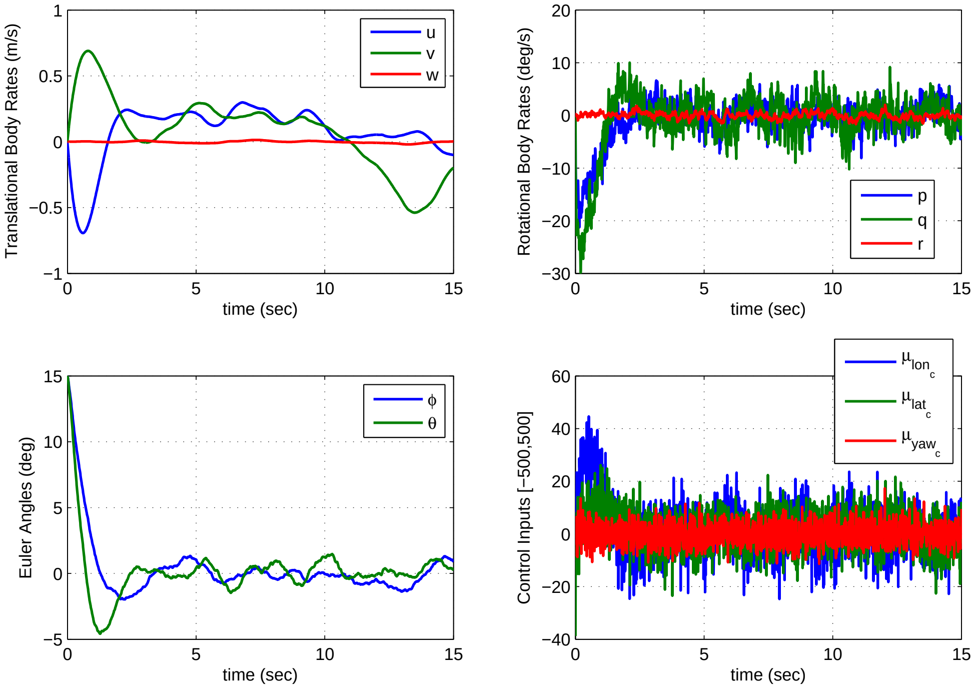


Figure 5.4: SISO PD Gain Performance

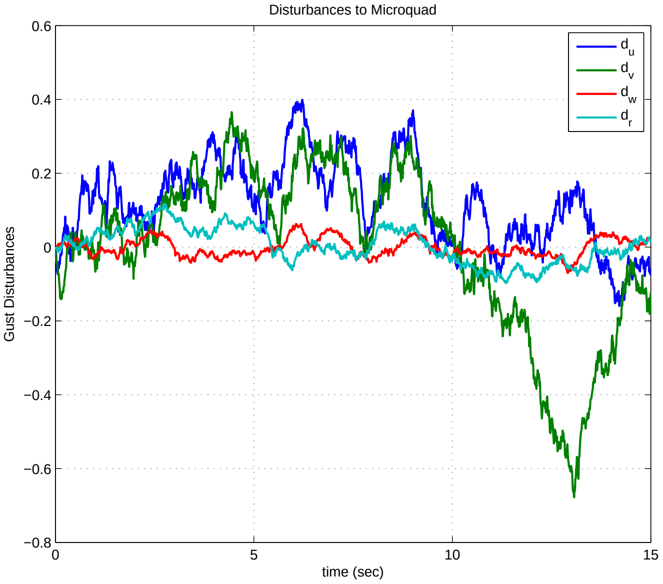


Figure 5.5: Disturbances to Microquad for SISO Simulation

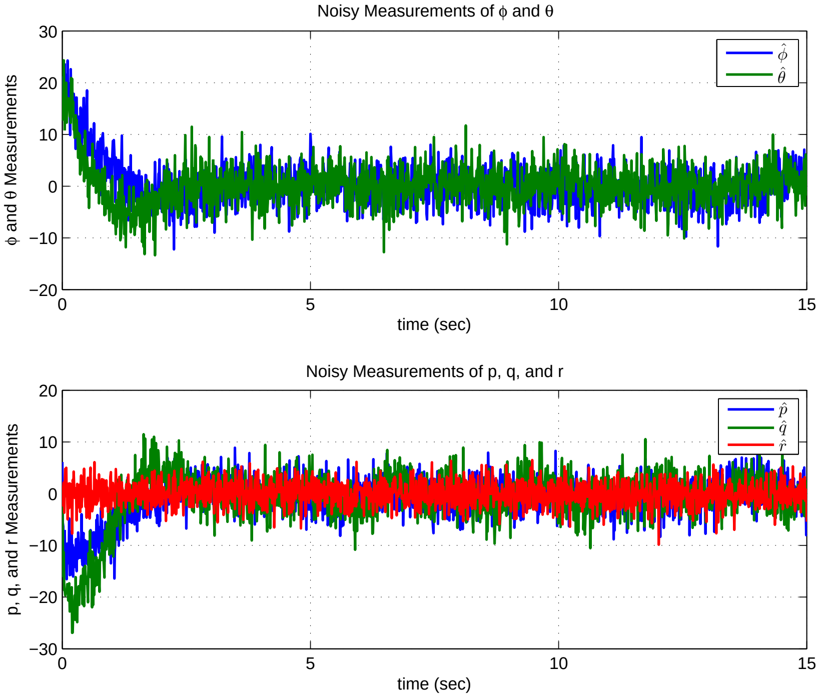


Figure 5.6: Modeled Measurement Noise on Mote Measurements

State	Steady State RMS Covariance
ϕ	0.50417 (degrees)
θ	0.76863 (degrees)
p	1.1929 (deg/s)
q	1.9051 (deg/s)
r	0.28114 (deg/s)

Table 5.2: SISO PD Controller Steady State RMS Covariance

This performance is satisfactory for the vehicle. After all it does allow the vehicle to fly in a stable fashion. However, the performance can clearly be improved, especially given the plethora of available tools when the system model is known. Hence, further options for the inner loop controller must be explored.

5.2.1.2 Inner Loop LQG Controller

The first Model-Based Controller (MBC) tried using the model knowledge is a theoretically optimal controller known as a Linear Quadratic Gaussian controller and estimator combination. This popular method is a well-defined solution inside a particular framework of systems. It combines the optimal regulator of the linear quadratic regulator (LQR) and the optimal estimator or linear quadratic estimator (LQE), more commonly known as the Kalman Filter.

The framework to implement this controller is to assume the plant dynamics are linear and known. To be able to synthesize the appropriate gains the model must also assume a particular form and hold a few more assumptions. The model structure must be as that seen in Equation 5.19 with n states, m inputs, and p outputs. Where the disturbance signals $\vec{d}$ and $\vec{v}$ are stochastic with known statistical properties [6], [23].

$$\begin{aligned} \dot{\vec{x}} &= \mathbf{A}\vec{x} + \mathbf{B}\vec{u} + \mathbf{D}\vec{d} \\ \vec{y} &= \mathbf{C}\vec{x} + \vec{v} \end{aligned} \tag{5.19}$$

The disturbances $\vec{d}$ and the measurement noise $\vec{v}$ are assumed to be zero-mean Gaussian stochastic processes with constant power spectral density, or covariance matrices $\mathbf{W}$ and $\mathbf{N}$ respectively. More specifically,

$$E\left[\vec{d}(t)\vec{d}^T(\tau)\right] = \mathbf{W}\delta(t-\tau) \tag{5.20}$$

$$E\left[\vec{v}(t)\vec{v}^T(\tau)\right] = \mathbf{N}\delta(t-\tau) \tag{5.21}$$

where $E[\cdot]$ is the expectation operator and $\delta(t-\tau)$ is the Dirac delta function [23]. For the microquad model, this is assumed. The disturbance inputs are modeled as white noise shaped through a Gauss-Markov filter. Therefore if the gust states are augmented to the system, then the only disturbances are zero-mean Gaussian noise. The measurement noise is also assumed to be white noise. Hence these requirements are met for the microquad to implement an LQG controller.

In addition to these assumptions, the system must also be stabilizable and detectable. This is a relaxed requirement from the notions of controllability and observability. This means that the controllability matrix have full row rank (rank($\mathcal{C}$) $= n$), and the observability matrix must have full column rank (rank($\mathcal{O}$) $= n$). The controllability matrix for an LTI system is given in Equation 5.22, and the observability matrix is presented in Equation 5.23.

$$\mathcal{C} = \Big[\mathbf{B},\, \mathbf{AB},\, \mathbf{A}^2\mathbf{B}, \cdots, \mathbf{A}^{n-1}\mathbf{B}\Big] \tag{5.22}$$

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{5.23}$$

Instead of requiring all modes to be controllable or observable, stabilizability and detectability only requires the unstable modes be controllable and observable [24]. It is fortunate that this is the requirement because in the inner loop the heave mode is neither observable, nor controllable because the throttle command is removed from the system and no Mote measurements are capable of sensing heave motion. Fortunately the heave mode is a simple single pole, stable system. Hence the microquad model is stabilizable and detectable because the one mode that is neither controllable nor observable in the inner loop is stable.

Because the microquad satisfies the necessary assumptions, and the model is already in the necessary framework, the development of the LQG controller is simple. The steps are easy, and MATLAB$^{\text{TM}}$ has several algorithms built in to handle the majority of the calculations. The first step is to determine the LQR gains. Assuming the system has full state feedback, this can be done using the '*lqr.m*' function. This assumes the control law $\vec{u}(t) = -\mathbf{K}_{LQR}\,\vec{x}(t)$, where $\mathbf{K}_{LQR}$ is the constant gain matrix of the form in Equation 5.24.

$$\mathbf{K}_{LQR} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{X} \tag{5.24}$$

Where $\mathbf{X} = \mathbf{X}^T \geq 0$ is the positive semi-definite unique solution to the algebraic Riccati equation below.

$$\mathbf{A}^T\mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{X} + \mathbf{Q} = 0 \tag{5.25}$$

Here, $\mathbf{Q} \geq 0$ is the positive semi-definite state weighting matrix and $\mathbf{R} > 0$ is the positive definite control input weighting matrix. These matrices are used to define the cost function (Equation 5.26) that the control input $\vec{u}(t) = -\mathbf{K}_{LQR}\,\vec{x}(t)$ minimizes [23].

$$J = \int_0^\infty \left(\vec{x}^T\mathbf{Q}\vec{x} + \vec{u}^T\mathbf{R}\vec{u}\right) \mathrm{d}t \tag{5.26}$$

Using the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$, the MATLAB$^{\text{TM}}$ function '*lqr.m*' only requires the dynamics matrix $\mathbf{A}$ and the controls matrix $\mathbf{B}$ to return the optimal control of the form in Equation 5.24 minimizing the cost function in Equation 5.26.

The next step is to solve the dual problem relating to the optimal estimator and to determine the estimator gains to best estimate the system states. The Kalman Filter does this in a very similar fashion to the steps in the LQR problem. The Kalman Filter has the same structure as that of an ordinary state estimator [23] seen in Equation 5.27.

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + \mathbf{B}\vec{u} + \mathbf{L}\left(\vec{y} - \mathbf{C}\hat{x}\right) \tag{5.27}$$

Where $\hat{x}$ are the estimates of the states $\vec{x}$. The Kalman Filter gains $\mathbf{L}$ are the optimal set of observer gains that minimize the cost function in Equation 5.28.

$$J = E\left\{[\vec{x} - \hat{x}]^T\,[\vec{x} - \hat{x}]\right\} \tag{5.28}$$

The Kalman Filter gains, which are the optimal choice of $\mathbf{L}$ has a similar structure to that in Equation 5.24 as seen in Equation 5.29.

$$\mathbf{L} = \mathbf{Y}\mathbf{C}^T\mathbf{N} \tag{5.29}$$

Where **N** is the covariance matrix of the measurement noise, and as was the case in LQR, the Kalman Filter has a positive semi-definite matrix **Y** which is the solution to the algebraic Riccati equation below.

$$\mathbf{Y}\mathbf{A}^T + \mathbf{A}\mathbf{Y} - \mathbf{Y}\mathbf{C}^T\mathbf{N}^{-1}\mathbf{C}\mathbf{Y} + \mathbf{W} = 0 \tag{5.30}$$

MATLAB$^{\text{TM}}$ also has a built in function to return the Kalman Filter gain matrix **L**. In a very similar fashion to '*lqr.m*', the function '*lqe.m*' requires inputs of the dynamics matrix **A**, the sensor matrix **C**, the disturbance matrix **D**, the disturbance covariance matrix **W**, and the measurement noise covariance matrix **N**. Using these inputs the function will quickly compute and return the Kalman Filter gains **L**.

Now if the systems $\left(\mathbf{A}, \mathbf{B}, \mathbf{Q}^{1/2}\right)$ and $\left(\mathbf{A}, \mathbf{D}\mathbf{W}^{1/2}, \mathbf{C}\right)$ are stabilizable and detectable then the two solutions to the Riccati equations **K** and **L** are guranteed to exist and the closed loop system determined by these gain matrices is internally stable. The introduction of an included estimator in the dynamics complicates the system a bit. Fortunately, the separation principle indicates the closed loop poles of the entire coupled system are simply the combination of the closed loop dynamic poles $(\mathbf{A} - \mathbf{B}\mathbf{K}_{LQR})$, and the estimator poles $(\mathbf{A} - \mathbf{L}\mathbf{C})$[23]. This can be seen from the LQG system structure in Equation 5.31.

$$\begin{bmatrix} \dot{\vec{x}} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{x}} - \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{K}_{LQR} & \mathbf{B}\mathbf{K}_{LQR} \\ \mathbf{0} & \mathbf{A} - \mathbf{L}\mathbf{C} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{D} & -\mathbf{L} \end{bmatrix} \begin{bmatrix} \vec{d} \\ \vec{v} \end{bmatrix} \tag{5.31}$$

A block diagram representation of the LQG environment can be seen in Figure 5.7. When looking at this representation of the LQG controller, it can be quite easily implemented in a Simulink$^{\text{TM}}$ model. The only steps necessary to simulate the LQG performance are to determine the weighting matrices for the LQR gain set, and plug them into MATLAB's$^{\text{TM}}$ built in function and record the LQR gains returned. The next step is to determine the covariance matrices of the noise and disturbances, and then plug them into MATLAB's$^{\text{TM}}$ '*lqe.m*' function and record the Kalman gain set returned. Once these are determined, they can be implemented in the model and the outputs and inputs can be recorded and compared with other controller results.

Before computing these matrices necessary, the system was augmented with the gust states,

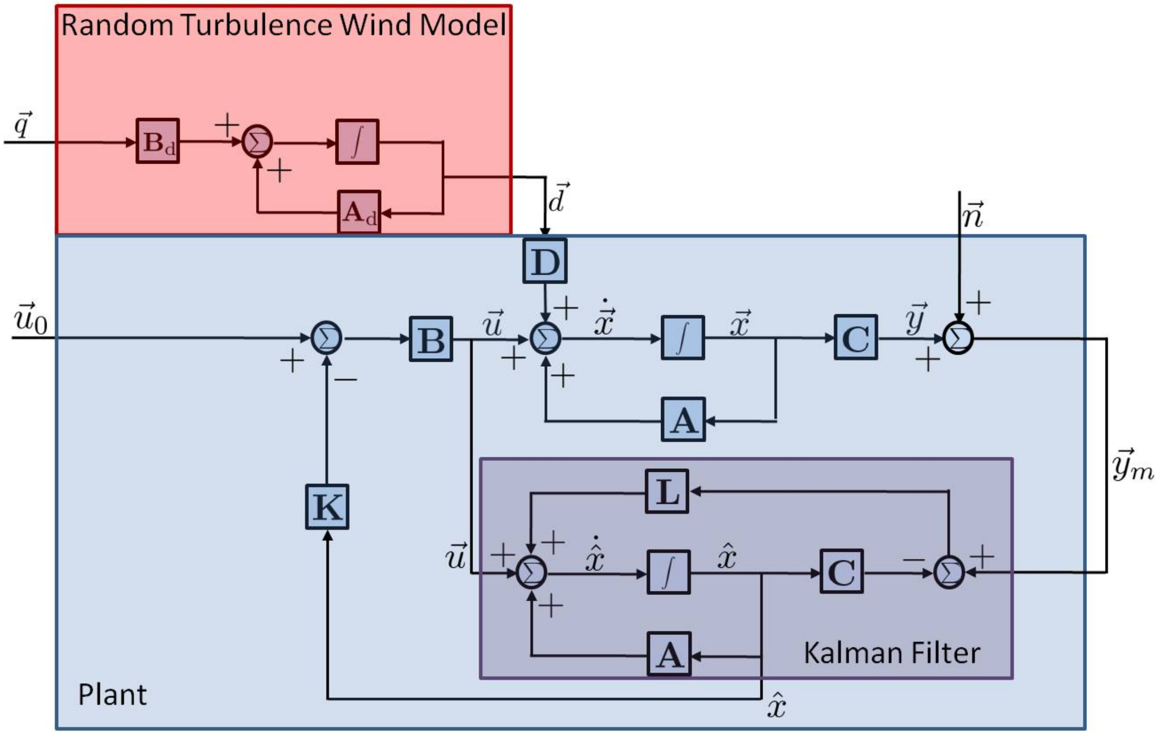


Figure 5.7: LQG Block Diagram

to ensure that the disturbances to the system were zero-mean Gaussian noise. If they were left as the Gauss-Markov shaped gusts, they would not have the desired characteristics and the LQG controller would not be optimal. Since there were 4 gust states, these are added to the dynamics and absorbed into the augmented dynamics matrix.

The weighting matrices used to compute the LQR gains were chosen as follows.

$$\mathbf{Q} = \begin{bmatrix} 0.005 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \tag{5.32}$$

These too, were augmented to account for the gust states included in the model. However the gust states were just weighted with a factor of 0. Which resulted in an LQR gain set as seen in Equation 5.33. Due to the decoupled dynamics many of the entries of the matrix are artifacts of the computation, and are therefore rounded out. Anything with a magnitude less than 10^{-10} is truncated to 0 for simplicity.

$$\mathbf{K}_{LQR} = \begin{bmatrix} 4.402 & 0 & 0 & 0 & 33.517 & 0 & 0 & 92.278 & -7.323 & 0 & 0 & 0 \\ 0 & -5.493 & 0 & 14.872 & 0 & 0 & 53.738 & 0 & 0 & 7.352 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80.086 & 0 & 0 & 0 & 0 & 0 & 6.058 \end{bmatrix} \tag{5.33}$$

The Kalman Filter gains were calculated next. The disturbance covariance matrix **W** is the diagonal matrix of the noise intensity used to shape the random disturbance. The noise covariance matrix **N** is similar, except it is the diagonal matrix of the measurement noise intensities. The two were selected as seen in Equation 5.34. These resulted in measurement noise similar to that seen in actual data taken from the Mote's sensors. And the gusts are on the same order of magnitude as those that the SISO controller was subjected to.

$$\mathbf{W} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}, \mathbf{N} = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0.05 \end{bmatrix} \tag{5.34}$$

Using these power spectral density matrices, and the model knowledge the Kalman Filter gain set **L** can be calculated as seen in Equation 5.35. As was done in Equation 5.33, any entry that had a magnitude less that 10^{-10} is rounded to 0.

$$\mathbf{L} = \begin{bmatrix} 0 & -0.0994 & 0 & 0.1802 & 0 \\ 0.0909 & 0 & -0.092 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 5.571e-4 & 0 & 0.4853 & 0 & 0 \\ 0 & -4.6e-3 & 0 & 0.9388 & 0 \\ 0 & 0 & 0 & 0 & 0.0648 \\ 0.0156 & 0 & 0.0183 & 0 & 0 \\ 0 & 0.0138 & 0 & 0.0136 & 0 \\ 0 & -0.0936 & 0 & -0.4976 & 0 \\ 0.0531 & 0 & 0.5520 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1901 \end{bmatrix} \tag{5.35}$$

Using Equation 5.33 and Equation 5.35 the same maneuver performed in §5.2.1.1 can be simulated. The vehicle is subjected to an offset of $\phi_0 = 15^\circ$and $\theta_0 = 15^\circ$, and the resulting controls and outputs are recorded using a Simulink$^{\mathrm{TM}}$ model developed. The general performance of the microquad can be seen in Figure 5.8. The vehicle settles quickly, and there seem to be less quick oscillations in the body rotation rates $[p, q, r]$. This is because rather than feeding the noisy signals directly through to the control input channel, the Kalman Filter determines an optimal estimate of the states instead, utilizing the known statistical properties of the noise. The disturbances the vehicle is subjected to are seen in Figure 5.9. They are on the same order as those seen in the SISO simulation (Figure 5.5). Lastly, in Figure 5.10 the Kalman Filter's estimates of the inner loop states is seen. Because the gust states are absorbed into the dynamics, they too can be estimated and compared with the original as seen in the plot.

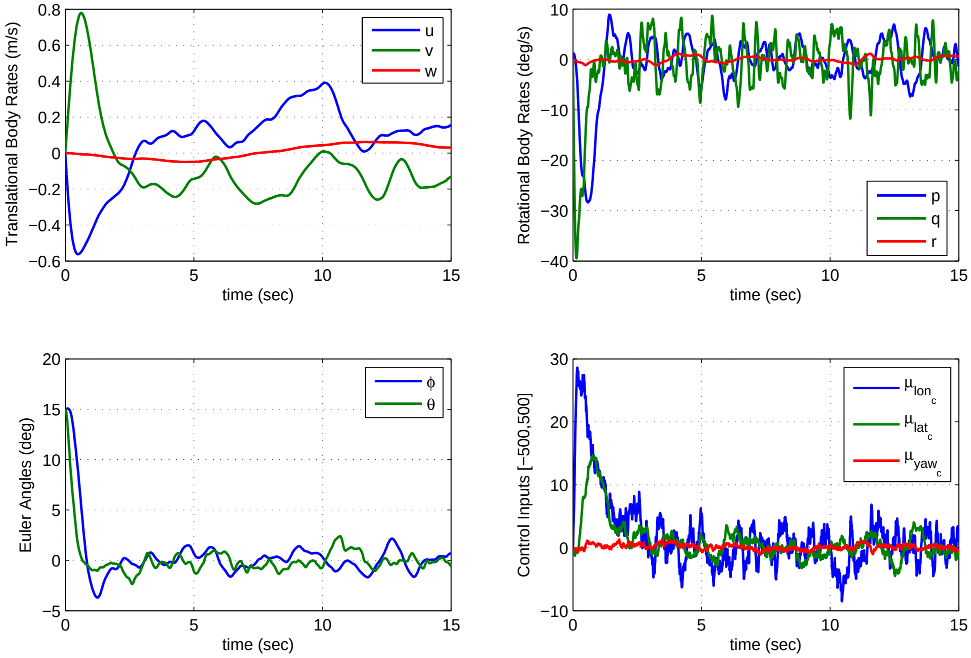


Figure 5.8: LQG Controller Performance

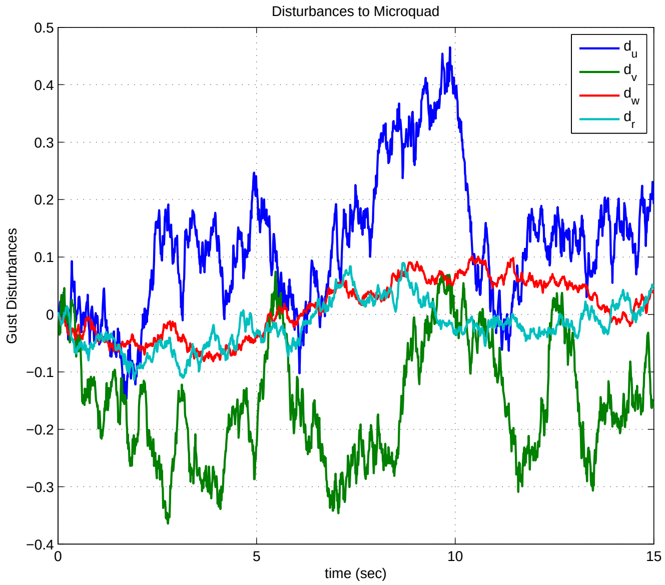


Figure 5.9: Disturbances to Microquad for LQG Simulation

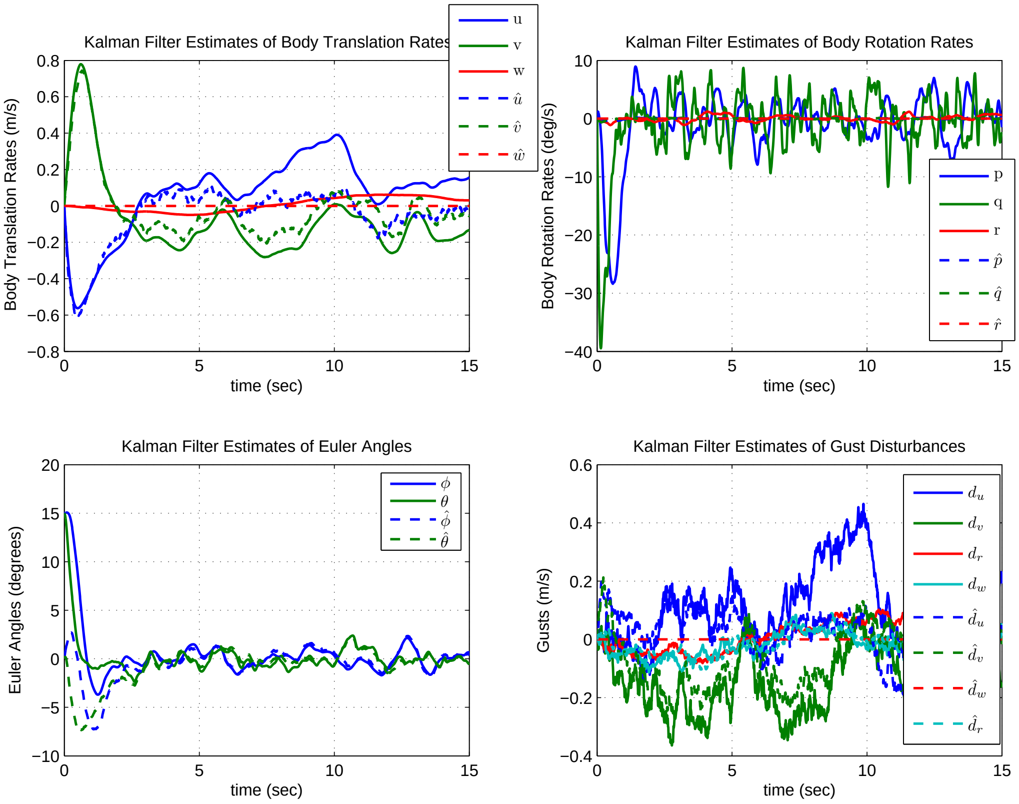


Figure 5.10: Kalman Filter Performance

While these results may be very attractive, the LQG controller is unfortunately too complex for implementation on the microquad. The integration of the Kalman Filter is not possible given the limited computational resources available on the Mote. These results were merely determined for comparison purposes. It's also important to note that the LQG controller is very dependent on the model knowledge. If the model changes even slightly, the results will be very different, and could even be unstable. Because the microquad model can tend to change with different rotors, electronics, or with damage, this controller choice is not very attractive any more.

5.2.1.3 Inner Loop Output LQR Controller

The next choice investigated for the inner loop controller is a regulator similar to LQR, but without full state feedback. It is commonly known as a Linear Quadratic Regulator with Output Feedback, but will be more compactly referred to as Output LQR in this case. This controller assumes a simple LTI plant of the form in Equation 5.36.

$$\begin{aligned} \dot{\vec{x}} &= \mathbf{A}\vec{x} + \mathbf{B}\vec{u} \\ \vec{y} &= \mathbf{C}\vec{x} \end{aligned} \tag{5.36}$$

Rather than using a Kalman Filter to estimate the states and therefore implementing full state feedback, the control law will be a simpler output feedback law [6].

$$\vec{u} = -\mathbf{K}\vec{y} \tag{5.37}$$

Therefore $\mathbf{K}$ is an $m \times p$ matrix of output feedback gains. Because the inner loop is of the form of a regulator, the input will be assumed to have no reference inputs for the analysis.

Output LQ techniques are often popular because of the simplicity. Feeding the measured outputs directly through to the controls is a much less complicated method than estimating the states. The controllers are often of a fixed structure, and thus the LQG estimator/controller combination is difficult to apply. The output LQ gains are calculated via a series of coupled nonlinear matrix equations. To develop these equations a performance index (PI) must be defined. This PI is similar to the cost function of the LQR framework.

$$J = \frac{1}{2} E \left[\int_0^\infty \left(\vec{x}^T \mathbf{Q} \vec{x} + \vec{u}^T \mathbf{R} \vec{u} \right) \mathrm{d}t \right] \tag{5.38}$$

Similar to the LQR requirements, the state weighting matrix $\mathbf{Q} \geq 0$ is a positive semi-definite matrix weighting each state and the control weighting matrix $\mathbf{R} > 0$ is a positive definite matrix weighing each control channel.

From this PI and the fixed structure of the plant, the three optimal gain design equations can be found through a Lagrange multiplier approach, and several linear algebra techniques.

$$0 = \mathbf{A}_c^T \mathbf{P} + \mathbf{P} \mathbf{A}_c + \mathbf{C}^T \mathbf{K}^T \mathbf{R} \mathbf{K} \mathbf{C} + \mathbf{Q} \tag{5.39}$$

$$0 = \mathbf{A}_c \mathbf{S} + \mathbf{S} \mathbf{A}_c^T + \mathbf{X} \tag{5.40}$$

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{S} \mathbf{C}^T \left(\mathbf{C} \mathbf{S} \mathbf{C}^T \right)^{-1} \tag{5.41}$$

Where $\mathbf{A}_c$ is the closed loop dynamics matrix, and $\mathbf{X}$ is the expected value of the squared initial conditions.

$$\mathbf{A}_c = \mathbf{A} - \mathbf{B}\mathbf{K}\mathbf{C}, \quad \mathbf{X} = E\left[\vec{x}_0 \vec{x}_0^T\right]$$

These relationships result in an optimal cost that depends on all three of these coupled equations.

$$J = \frac{1}{2} \operatorname{tr}(\mathbf{P}\mathbf{X}) \tag{5.42}$$

The derivations and proofs of these relationships along with a guideline for the numerical solution algorithm can be found in [6], [25], [26]. Using the formula provided, a numerical iterative routine can be developed in MATLAB$^{\text{TM}}$ to determine the gains given the weighting matrices, the plant structure, and an initial stabilizing gain matrix $\mathbf{K}_0$, the procedure will use as a starting guess. The last piece necessary to be able to solve these equations, is a value of the autocorrelation of the initial states $\mathbf{X}$. If we assume that $\vec{x}_0$ is uniformly distributed on a surface described by $\mathbf{X}$, then we can make an assumption as to what $\mathbf{X}$ is. For this problem, because it is a regulator problem and the goal is to drive nonzero initial states to zero, we can assume the the initial states are uniformly distributed on the unit sphere, and therefore $\mathbf{X} = \mathbb{I}$ [6].

To keep the comparisons pertinent, the same state and control weighting matrices (Equation 5.32) used in the LQG design (§ 5.2.1.2) are used here, and the initial stabilizing gain is the same gain matrix used for the SISO PD case (Equation 5.17). These parameter choices results in a gain matrix seen in Equation 5.43.

$$\mathbf{K} = \begin{bmatrix} -0.002 & 1.538 & -0.001 & 0.784 & 0.005 \\ 0.854 & 0.001 & 0.489 & -0.001 & 0.005 \\ 0.007 & -0.001 & -0.005 & 0 & 1.885 \end{bmatrix} \tag{5.43}$$

Using this gain matrix, the microquad model returns the results seen in the following figures.

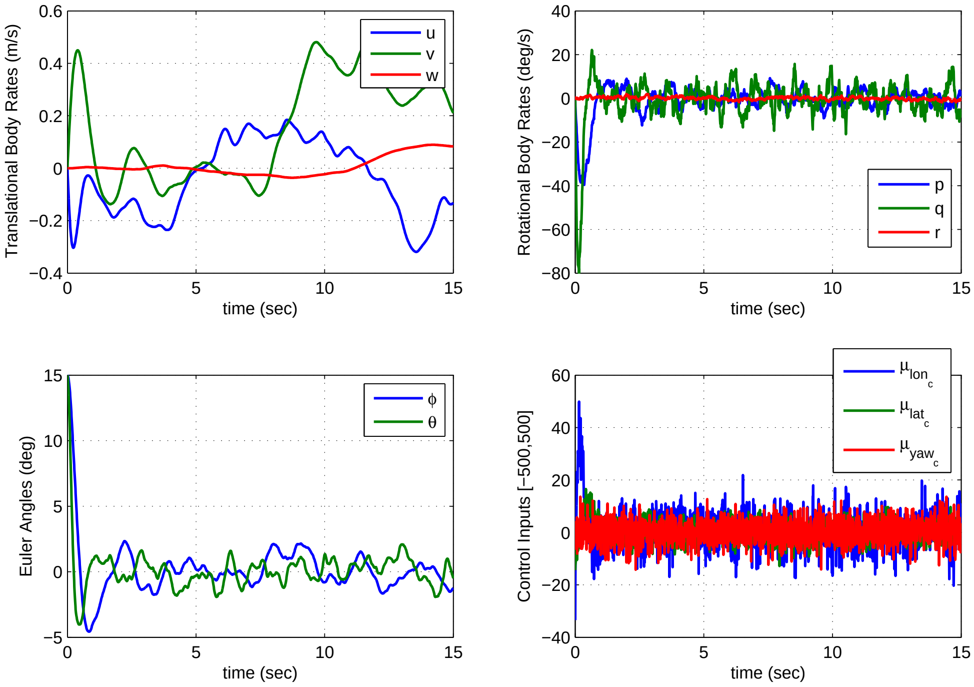


Figure 5.11: Output LQR Gain Performance

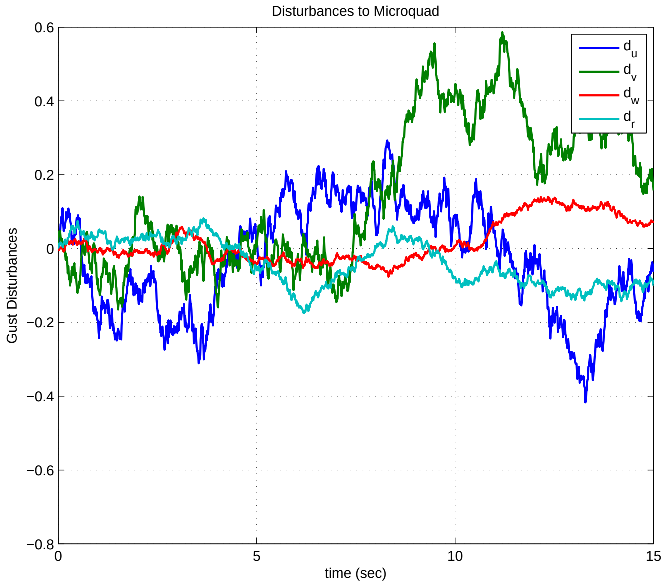


Figure 5.12: Disturbances to Microquad for Output LQR Simulation

In addition to these figures, the steady state RMS covariance of the inner loop states can be seen in Table 5.3. These values are computed using the same formulation in the SISO case (§ 5.2.1.1), from Equation 5.18.

State	Steady State RMS Covariance
ϕ	1.002 (degrees)
θ	0.8528 (degrees)
p	3.4233 (deg/s)
q	3.856 (deg/s)
r	0.2629 (deg/s)

Table 5.3: Output LQR Controller Steady State RMS Covariance

These results are satisfactory, in the sense that they are capable of stabilizing the attitude in the presence of disturbances. The speed of the response is much faster than that of the SISO PD gains, which is a result of the higher proportional gains and the lower damping gains. However, because the noise on the attitude measurements is more intense than that on the rate measurements, these higher proportional gains create a more exaggerated effect of the variance as seen in the lower left hand plot in Figure 5.11. As to be expected from these plots, the RMS covariances computed in Table 5.3 are greater than those in the SISO case (Table 5.2). From these results it would be prudent to choose the SISO gains instead, in order to avoid the unnecessary coupled entries, as well as increasing the covariance and unsteady behavior.

5.2.1.4 Inner Loop Static $\mathcal{H}_\infty$ Controller

The last controller considered is another controller of the same structure as that of the Output Feedback LQR case. However, it incorporates the system's response to disturbances into the synthesis of the gain matrix. $\mathcal{H}_\infty$ control methods are well known for their robustness properties. A detailed description of the environment and relationships for general (mixed sensitivity, loop shaping, etc.) $\mathcal{H}_\infty$ control design can be found in [23]. For brevity's sake and because the Mote would not fair well with implementing a dynamic controller such as those found through these general methods, they will be skipped in this design. While a dynamic controller will always outperform a static controller in terms of performance, due to the fixed structure of the microquad, only the static output feedback scenario is pertinent. Therefore the notions of general $\mathcal{H}_\infty$ control are extended to the design of static feedback systems such as the microquad through the algorithm presented in [20]. This method has several advantages over the Output LQ technique presented in § 5.2.1.3. The solution algorithm only requires solving two coupled matrix equations, rather than three. The algorithm also does not require an initial stabilizing gain matrix either.

Given a typical LTI plant as such shown in Equation 5.44, the Static $\mathcal{H}_\infty$ problem is easy to define.

$$\begin{aligned} \dot{\vec{x}} &= \mathbf{A}\vec{x} + \mathbf{B}\vec{u} + \mathbf{D}\vec{d} \\ \vec{y} &= \mathbf{C}\vec{x} \end{aligned} \tag{5.44}$$

Using a control law of the typical output feedback form $\vec{u} = -\mathbf{K}\vec{y}$, the problem can be placed into the framework for generalized control configuration shown in Figure 5.13. For this problem, $\mathbf{P}$

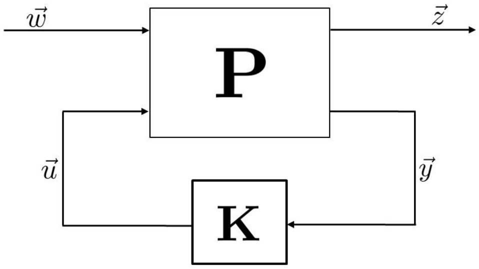


Figure 5.13: Generalized Control Configuration Block Diagram

is the generalized plant, consisting of all dynamics and the formulation of exogenous signals, and errors. The exogenous signals $\vec{w}$ for this problem are the disturbances to the plant $\vec{d}$. The controller is designed to minimize the signal $\vec{z}$ from these exogenous inputs $\vec{w}$. Therefore, to keep a similar environment as used in previous controllers for comparison purposes, the minimization signal $\vec{z}$ is chosen as a quadratic function of the states and controls highlighted below in Equation 5.45.

$$||\vec{z}||^2 = \vec{x}^T\mathbf{Q}\vec{x} + \vec{u}^T\mathbf{R}\vec{u} \tag{5.45}$$

This form is seen in the LQR and output LQR designs. Therefore a familiar LQR type relative weighting scenario coincides with the L_2 gain performance adjustment [27].

The concept is to select a feedback gain matrix $\mathbf{K}$, such that the plant (Equation 5.44) is stabilized, the nominal performance requirements are met, and is robust to disturbances. This problem is a bounded L_2 gain problem, where the system must be output feedback stabilizable. A system is output feedback stabilizable if there exists a matrix $\mathbf{K}$ such that $\mathbf{A} - \mathbf{BKC}$ is stable [20]. The system L_2 gain is said to be bounded or attenuated by γ if

$$\frac{||\vec{z}(t)||_2^2}{||\vec{d}(t)||_2^2} = \frac{\int_0^\infty ||\vec{z}(t)||^2 \mathrm{d}t}{\int_0^\infty ||\vec{d}(t)||^2 \mathrm{d}t} = \frac{\int_0^\infty \left(\vec{x}^T\mathbf{Q}\vec{x} + \vec{u}^T\mathbf{R}\vec{u}\right)\mathrm{d}t}{\int_0^\infty \vec{d}^T\vec{d}\mathrm{d}t} \leq \gamma^2 \tag{5.46}$$

Using the theorem presented in [20], the necessary and sufficient conditions for synthesizing a static $\mathcal{H}_\infty$ gain set are as follows

1. The system is stabilizable and detectable

   - $[\mathbf{A}, \mathbf{B}]$ is stabilizable

- $[\mathbf{A}, \mathbf{C}]$ is detectable

2. There exists matrices $\mathbf{K}$ and $\mathbf{L}$ such that

   - $\mathbf{KC} = \mathbf{R}^{-1}\left(\mathbf{B}^T\mathbf{P} + \mathbf{L}\right)$; where $\mathbf{P} > 0$ is the solution of
   - $\mathbf{A}^T\mathbf{P} + \mathbf{PA}^T + \mathbf{Q} - \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P} + \frac{1}{\gamma^2}\mathbf{PDD}^T\mathbf{P} + \mathbf{L}^T\mathbf{R}^{-1}\mathbf{L} = 0$

If these conditions are met, then [21] presents three different solution algorithms for synthesizing the stabilizing gains $\mathbf{K}$. The first solution algorithm presented will be reiterated here for reference.

Step 1:

Initialize. Set iteration counter $n = 0$, $L_0 = 0$, select weighting matrices $\mathbf{Q}$, $\mathbf{R}$, and the bound γ.

Step 2:

n^{th} iteration step:

- solve for $\mathbf{P}_n$ in $\mathbf{A}^T\mathbf{P}_n + \mathbf{P}_n\mathbf{A}^T + \mathbf{Q} - \mathbf{P}_n\mathbf{BR}^{-1}\mathbf{B}^T\mathbf{P}_n + \frac{1}{\gamma^2}\mathbf{P}_n\mathbf{DD}^T\mathbf{P}_n + \mathbf{L}_n^T\mathbf{R}^{-1}\mathbf{L}_n$
- Evaluate the update directions:

  (a) $\mathbf{K}_{n+1} = \mathbf{R}^{-1}\left(\mathbf{B}^T\mathbf{P}_n + \mathbf{L}_n\right)\mathbf{C}^T\left(\mathbf{RK}_{n+1}\mathbf{C} - \mathbf{B}^T\mathbf{P}_n\right)$

  (b) $\mathbf{L}_{n+1} = \mathbf{RK}_{n+1}\mathbf{C} - \mathbf{B}^T\mathbf{P}_n$
- if $||\mathbf{K}_{n+1} - \mathbf{K}_n||_F$ (Frobenius Norm of $\Delta\mathbf{K}$) is small enough - go to 3, otherwise $n = n+1$ and repeat 2.

Step 3:

Terminate, set $\mathbf{K} = \mathbf{K}_{n+1}$

A MATLAB$^{\text{TM}}$ function was developed to return the gain matrix $\mathbf{K}$, given the system dynamics, the weighting matrices, and an initial bound γ. Typically γ is set high, and then iteratively reduced until the algorithm can no longer solve for $\mathbf{K}$. Because γ is a bound placed on $\vec{z}$ with respect to $\vec{d}$, it has a strong dependence on the intensity of the noise sources, as well as the weighting matrices.

Therefore its magnitude tends to be arbitrary, but generally speaking, the smaller it is the better the disturbance rejection properties. For the weighting matrix selection in Equation 5.32 γ is set to 350. These options result in a quick computation of the gain matrix in Equation 5.47.

$$\mathbf{K} = \begin{bmatrix} 0 & 4.11 & 0 & 2.099 & 0 \\ 1.19 & 0 & 0.57 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.769 \end{bmatrix} \tag{5.47}$$

In comparison to Equation 5.43, these gains are a bit higher than those implemented from the Output LQ synthesis. At a cursory glance, this could be problematic in a hardware implementation. The proportional gains are much higher in comparison to the damping gains (on the order of 2:1 ratio). Recall that the attitude estimates are much noisier and therefore a higher gain will pass this noise through to the actuators, creating a jittery behavior. The Simulink$^{\text{TM}}$ model simulation implemented with these gains results in performance seen in Figure 5.14. The gusts the vehicle is subjected to are plotted in Figure 5.15. In similar fashion the RMS steady state covariance of the inner loop states is seen in Table 5.4.

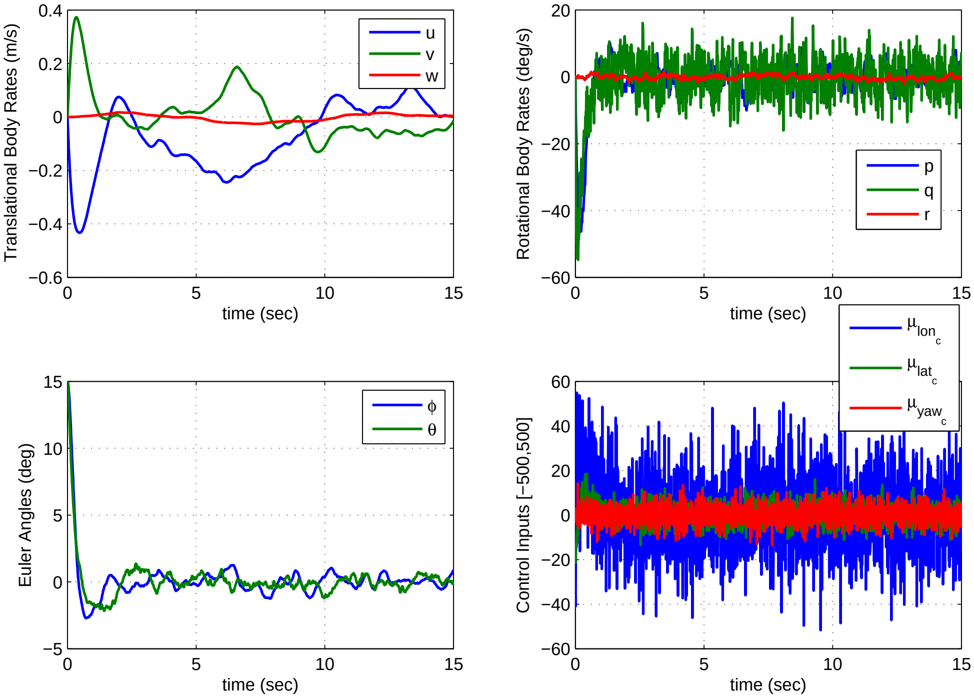


Figure 5.14: Static $\mathcal{H}_\infty$ Gain Performance

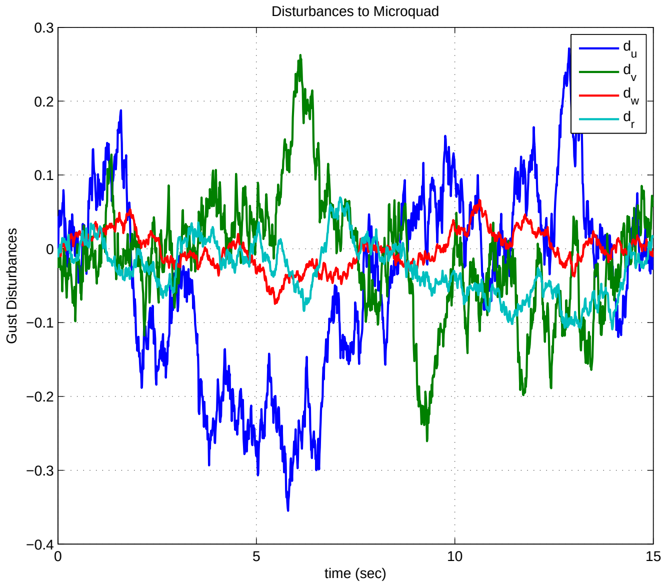


Figure 5.15: Disturbances to Microquad for $\mathcal{H}_\infty$ Simulation

State	Steady State RMS Covariance
ϕ	0.7638(degrees)
θ	0.4402 (degrees)
p	2.8071 (deg/s)
q	1.48146 (deg/s)
r	0.2785 (deg/s)

Table 5.4: Static $\mathcal{H}_\infty$ Controller Steady State RMS Covariance

The performance looks on par with the other controllers. There is more noise in the pitch rate q, as to be expected when looking at the longitudinal control input magnitude. The disturbances are on the same order of magnitude as those the other controllers must reject. However there seems to be less variance in θ after the system has settled, despite this higher longitudinal proportional gain. The steady state RMS covariance of θ also reflects this improved disturbance rejection. In fact all the inner loop states have less RMS covariance for the Static $\mathcal{H}_\infty$ controller than the Output LQ, aside from a slight improvement in r for the Output LQ gains.

5.2.1.5 Inner Loop Controller Gain Selection

To select the appropriate controller for the inner loop of the microquad, several considerations must be taken into account. The performance of the controller in the presence of noise and process disturbances is one of the main points around which to choose the gains. The simplicity of the controller is also a necessary consideration, on account of the fixed structure of the microquad. Robustness to the model uncertainty is also a large consideration because of the inherent uncertainty of the model as a function of the system identification process used. It's important to acknowledge that the MBCs tend to have slightly better performance than the SISO PD controller that has successfully been implemented on the microquad. However, these improvements tend to be only marginal. Therefore, due to the fact that the SISO PD controller is known to work on the vehicle, with slight model changes as well, it can be considered an effective controller with robustness to model uncertainty. Therefore this will be the controller of choice, because it only has slightly reduced performance compared to the MBC options. Hence the inner loop controller is

$$\mathbf{K}_{in} = \begin{bmatrix} 0 & 1.25 & 0 & 2.00 & 0 \\ 1.25 & 0 & 2.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.75 \end{bmatrix} \tag{5.48}$$

Which results in closed inner loop dynamics in Equation 5.49, to be included in the development of the outer loop controller.

$$\mathbf{A}_{\mathrm{I_{CL}}} = \begin{bmatrix} -0.44251 & 0 & 0 & 0 & 0.016858 & 0 & 0 & -9.277255 \\ 0 & -0.82007 & 0 & 0.016868 & 0 & 0 & 8.022955 & 0 \\ 0 & 0 & -0.51636 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7.71087 & 0 & -46.364 & 0 & 0 & -43.562 & 0 \\ 19.56993 & 0 & 0 & 0 & -56.772 & 0 & 0 & -84.315 \\ 0 & 0 & 0 & 0 & 0 & -9.1628 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.49}$$

5.2.2 Outer Loop Controller

The outer loop controller design differs from the inner loop in that it is not simply a regulator. The main goal of the outer loop is to track a desired position, altitude and heading, therefore it is a tracking problem. It is also desired to be robust to gusts as was the case in the inner loop controller. However, in the outer loop, these gusts will actually move the vehicle from its desired position. Therefore the controller must be capable of rejecting these in order to maintain an inertial position. Fortunately, the static gain sets developed from the methods used on the inner loop, can work as tracking controllers too $\vec{u} = \mathbf{K}\vec{e}$. They are simply regulators of the tracking error, regulating it to 0. Formulating the outer loop in the method provided in § 5.1.2, three outer loop controllers are developed and tested.

5.2.2.1 Outer Loop LQG Controller

The first controller to investigate is the LQG controller as discussed in § 5.2.1.2. However, this controller must satisfy tracking requirements. In order to do so, a common procedure known as augmenting the system is implemented with the LQG controller. The idea is to implement a different control, that is known as $\vec{v}(t)$ that is the derivative of the control signal $\vec{u}(t)$.

$$\dot{\vec{u}}(t) = \vec{v}(t) \implies \vec{u}(t) = \int_0^t \vec{v}(\tau)\mathrm{d}\tau \tag{5.50}$$

Introducing this, augments the dynamics to include an integrator of the control signal. This is reflected in a new augmented state space system seen below.

$$\begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{u} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbb{I} \end{bmatrix} \vec{v} \tag{5.51}$$

$$\vec{y} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{u} \end{bmatrix} \tag{5.52}$$

Where the augmented state vector $\vec{x}_a = [\vec{x}, \vec{u}]^T$, and the augmented state space matrices are those seen in Equations 5.51 and 5.52.

Using these augmented matrices, design the LQR gains using Equations 5.24 and 5.25, and the Kalman Filter gains using Equations 5.29 and 5.30. Fortunately, the controller design $G_c(s)$ absorbs the integrators into the MBC component. So the controller takes the structure below in Equations 5.53 and 5.54. The flow of the system is seen below in Figure 5.16 where all the augmented dynamics are integrated in the controller block, simplifying the structure.

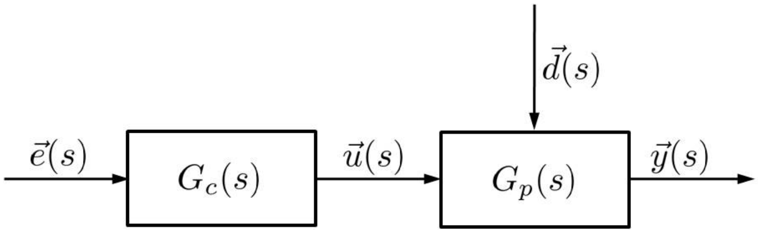


Figure 5.16: Physical System Block Diagram

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\hat{x}_a \\ \vec{u}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_a - \mathbf{B}_a\mathbf{K}_a\mathbf{L}_a\mathbf{C}_a & \mathbf{0} \\ \mathbf{K}_a & \mathbf{0}\end{bmatrix}\begin{bmatrix}\hat{x}_a \\ \vec{u}\end{bmatrix} + \begin{bmatrix}\mathbf{L}_a \\ \mathbf{0}\end{bmatrix}\vec{e} \tag{5.53}$$

$$\vec{u} = \begin{bmatrix}\mathbf{0} & \mathbb{I}\end{bmatrix}\begin{bmatrix}\hat{x}_a \\ \vec{u}\end{bmatrix} \tag{5.54}$$

To determine the LQR gains, the state and control weighing matrices must be determined. The augmented states are not included in the weighting matrices. The full dynamics include the gust dynamic states, augmented in the procedure outlined in Equation 5.7, as well as the zeros to ensure good tracking. However, the only states weighted in the selection of **Q** are the actual dynamical states. The selection of **Q** and **R** are seen in Equation 5.55.

$$\begin{aligned}\mathbf{Q} &= \begin{bmatrix}100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000\end{bmatrix}, \\ \mathbf{R} &= \begin{bmatrix}0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.002\end{bmatrix}\end{aligned} \tag{5.55}$$

Using these matrices, the LQR gain matrix (including the 4 gust dynamic states, and the 4 control integrator states) is computed in Equation 5.56.

$$\mathbf{K}_O = \begin{bmatrix} -102.1 & 0 & 0 & 0 & 8.21 & 0 & 0 & 427.3 & 0 & -200\cdots \\ 0 & 116.4 & 0 & 11.95 & 0 & 0 & 561.2 & 0 & 0 & 0\cdots \\ 0 & 0 & 0 & 0 & 0 & 414.3 & 0 & 0 & 4472 & 0\cdots \\ 0 & 0 & -1065 & 0 & 0 & 0 & 0 & 0 & 0 & 0\cdots \end{bmatrix}$$

$$\begin{bmatrix} \cdots 0 & 0 & -122.9 & 0 & 0 & 0 & 3.38 & 0 & 0 & 0 \\ \cdots 200 & 0 & 0 & 121.8 & 0 & 0 & 0 & 3.6 & 0 & 0 \\ \cdots 0 & 0 & 0 & 0 & 0 & 800.9 & 0 & 0 & 9.62 & 0 \\ \cdots 0 & -707.1 & 0 & 0 & -190.2 & 0 & 0 & 0 & 0 & 5.46 \end{bmatrix} \tag{5.56}$$

For the design of the Kalman filter, all disturbances to the system must be white noise. The four noise signals used to create the Gauss-Markov random gust model, as well as the two random chatter signals on the longitudinal and lateral control signals to model the inner loop measurement noise (i.e. $\delta_{lon} = \delta_{lon} + n$). The outer loop measurement noise is assumed white as well. The covariance matrices of the system noise are represented in Equation 5.57.

$$\mathbf{W} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.75 \end{bmatrix},$$

$$\mathbf{N} = 1e-7 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \tag{5.57}$$

Plugging these matrices, along with the system knowledge, into Matlab's$^{\text{TM}}$ '*lqe.m*' function quickly returns the Kalman Filter gain matric **L** in Equation 5.58 for all states, including gust states and integrator states. It should be noted that the integrator states must have an inclusion in the disturbance matrix, otherwise Matlab$^{\text{TM}}$ cannot solve the algorithm because it is ill-formed. Therefore a small magnitude entry was included in the integrator states' dynamics in the disturbance matrix, on the order of $1e-6$ limiting the effect it would have on the overall gain calculation. Therefore anything within an order of magnitude of $1e-6$ is considered negligible and truncated to remove the effects. Ideally, the last 4 rows (integrator dynamic states) should be zeros, but the solution cannot be solved in such a manner.

$$\mathbf{L} = \begin{bmatrix} 35.414 & 0 & 0 & 8.835 & 0 & 0 & 0 \\ 0 & 34.245 & 0 & 0 & 8.286 & 0 & 0 \\ 0 & 0 & 38.586 & 0 & 0 & 12.03 & 0 \\ 0 & 5.962 & 0 & 0 & 5.153 & 0 & 0 \\ -6.095 & 0 & 0 & -7.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 290.1 \\ 0 & 5.861 & 0 & 0 & 2.112 & 0 & 0 \\ -9.346 & 0 & 0 & -3.916 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 24.086 \\ 6.764 & 0 & 0 & 0.708 & 0 & 0 & 0 \\ 0 & 6.711 & 0 & 0 & 0.685 & 0 & 0 \\ 0 & 0 & 6.884 & 0 & 0 & 0.772 & 0 \\ 89.8 & 0 & 0 & 43.56 & 0 & 0 & 0 \\ 0 & 97.84 & 0 & 0 & 43.62 & 0 & 0 \\ 0 & 0 & 287.1 & 0 & 0 & 181 & 0.00016 \\ 0 & 0 & 0 & 0 & 0 & 0 & 416.1 \\ -0.00206 & 0 & 0 & -0.000957 & 0 & 0 & 0 \\ 0 & 0.00223 & 0 & 0 & 0.00095 & 0 & 0 \\ 0 & 0 & 0.00625 & 0 & 0 & 0.00377 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.00866 \end{bmatrix} \tag{5.58}$$

To characterize the performance of the LQG controller, a step command of a desired inertial position and heading is issued. Assuming the vehicle is hovering at 1 meter above the designated origin ($[x_0, y_0, z_0]^T = [0, 0, -1]^T m$), it is given a desired position of $[x_d, y_d, z_d]^T = [0.2, 0.3, -1.5]^T m$. The heading is assumed offset at an initial angle of 12°, and commanded to return to 0°. The tracking performance of the controller is seen in Figure 5.17, while the disturbances and body translational rates are in Figure 5.18, and the control effort is provided in Figure 5.19.

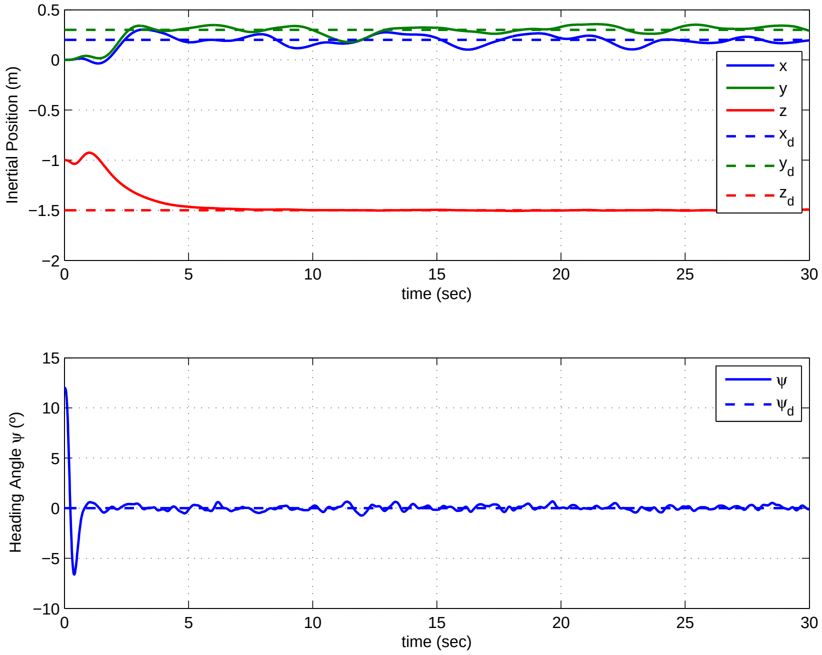


Figure 5.17: Outer Loop LQG Tracking Performance

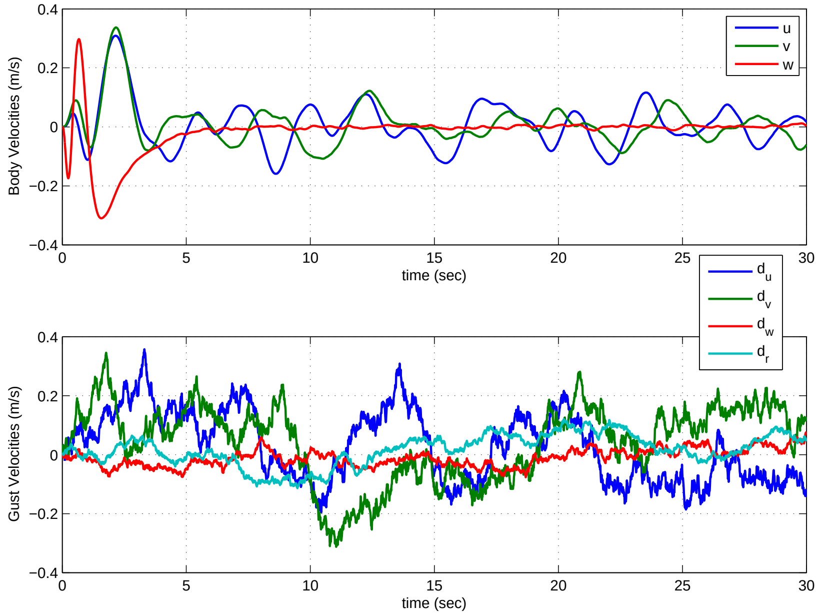


Figure 5.18: Outer Loop LQG Body Rates and Disturbances

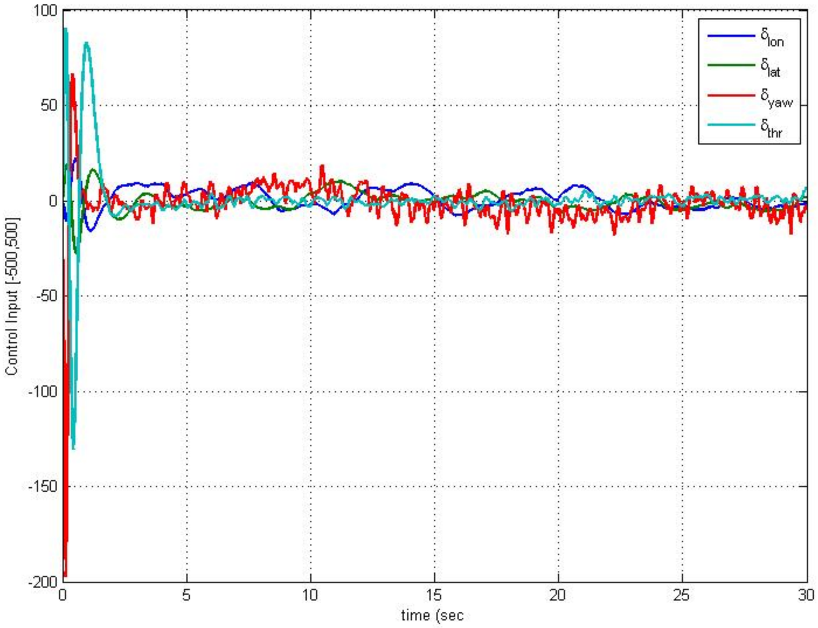


Figure 5.19: Outer Loop LQG Control Effort

The performance is good, which would be expected from a theoretically ideal controller. The tracking of the heading and altitude are very good. There is considerable overshoot in the heading though, perhaps a lower outer loop proportional gain, or a higher inner loop damping gain could remedy this. The altitude also has a noticeable undershoot, indicative of a RHP zero in the closed loop dynamics. There is also a large fluctuation of both the yaw and throttle commands in the beginning of the maneuver. This is an undesirable behavior to see in the simulation, because one difference noticed early in the model prediction versus the real world behavior is that the motors cannot throttle down quick enough to correct the overshoot caused by large control inputs. In general the LQG controller is not a good candidate for an outer loop controller because it is complicated and has a large dependence on the model knowledge. The model in this case is a twice augmented model, with the gust dynamics as well as the integrator states. Hence, the accuracy of a complicated model such as this is questionable. This leads to the desire to find a better, less complicated controller.

5.2.2.2 Outer Loop Output LQR Controller

The second outer loop controller to test is the Output Feedback Linear Quadratic controller discussed in § 5.2.1.3. The design of this controller can be converted from a tracking design to a regulator problem to simplify the solution [6]. Unfortunately, the large weighting factors used for the LQG controller in the section above do not provide a converging solution to the algorithm used to calculate the gains. Therefore a new set of weighting matrices must be selected, and an initial stabilizing gain set must be provided as well. The new selection of the quadratic weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are presented in Equation 5.59.

$$\mathbf{Q} = \begin{bmatrix} 7.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 150 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.075 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.075 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0225 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.75 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 75 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 75 \end{bmatrix},$$

$$\mathbf{R} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.002 \end{bmatrix} \tag{5.59}$$

The initial stabilizing gain is provided below in Equation 5.60. A map of the poles (marked by an 'x') and zeros (marked by an 'o') is shown in Figure 5.20. The poles are all stable, therefore this is a stabilizing gain set and can be used to solve the solution algorithm presented in § 5.2.1.3.

$$\mathbf{K}_0 = \begin{bmatrix} -25 & 0 & 0 & -15 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 80 \\ 0 & 0 & -10 & 0 & 0 & -20 & 0 \end{bmatrix} \tag{5.60}$$

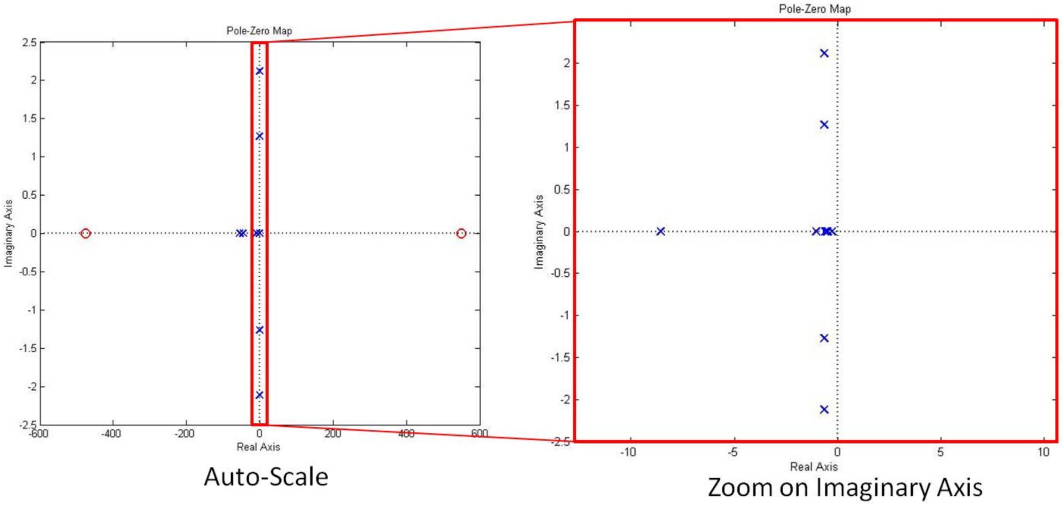


Figure 5.20: Initial Stabilizing Gain Pole-Zero Map

There is a RHP zero in the dynamics, indicative that there may be some undershoot, but nonetheless the system is stable and these parameters (along with the system model) can be plugged into an numerical routine to determine the optimal output feedback gains presented in Equation 5.61.

$$\mathbf{K} = \begin{bmatrix} -42.272 & 0 & 31.912 & -61.978 & 0 & 69.676 & 0 \\ 0 & 41.337 & 0 & 0 & 80.532 & 0 & -55.222 \\ 0 & 3.1865 & 0 & 0 & -361.21 & 0 & 430.36 \\ 7.694 & 0 & -201.58 & 234.74 & 0 & -411.95 & 0 \end{bmatrix} \tag{5.61}$$

The large degree of coupling in the matrix should throw up an immediate red flag. For a mostly decoupled system, the longitudinal control should only depend on longitudinal states, but here it depends on heave states. Therefore the behavior of the vehicle using the Output LQ gains may be unpredictable. When implementing this controller, the results are as expected. There is a large degree of variance in the position hold when looking at the top plot of Figure 5.21. Even the heading hold is very oscillatory (bottom plot of Figure 5.21), because it depends on the other error signals, not just the yaw angle. The disturbances (bottom plot of Figure 5.22) are no greater than those seen in the LQG simulation, but the vehicle has a much more difficult time rejecting them. The control signals seen in Figure 5.23 are very noisy as well. This is because all control inputs depend on more error signals than necessary, therefore the noise present in extra measurements is compounded into each control channel. The steady state RMS covariance of the outer loop outputs is presented in Table 5.5. These will be useful for comparison with the Output Feedback $\mathcal{H}_\infty$ controller in the next section.

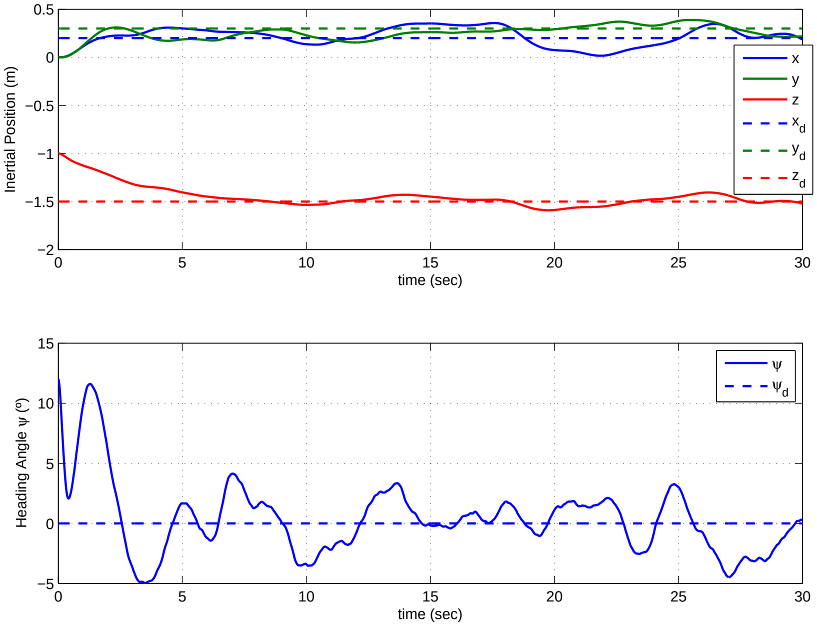


Figure 5.21: Outer Loop Output LQ Tracking Performance

State	Steady State RMS Covariance
x	0.105 (meters)
y	0.071 (meters)
z	0.032 (meters)
u	0.059 (m/s)
v	0.044 (m/s)
w	0.029 (m/s)
ψ	2.134 (degrees)

Table 5.5: Outer Loop Output LQR Controller Steady State RMS Covariance

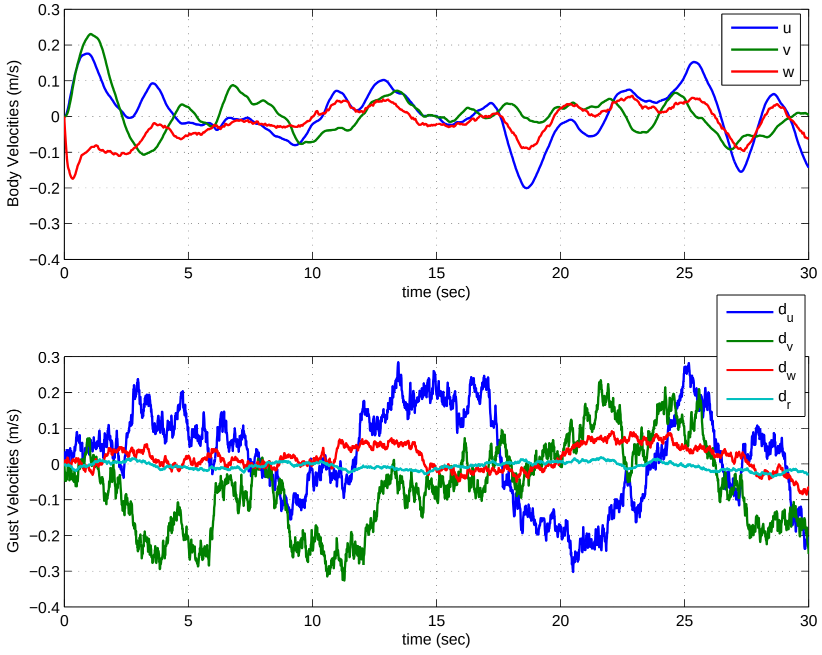


Figure 5.22: Outer Loop Output LQ Body Rates and Disturbances

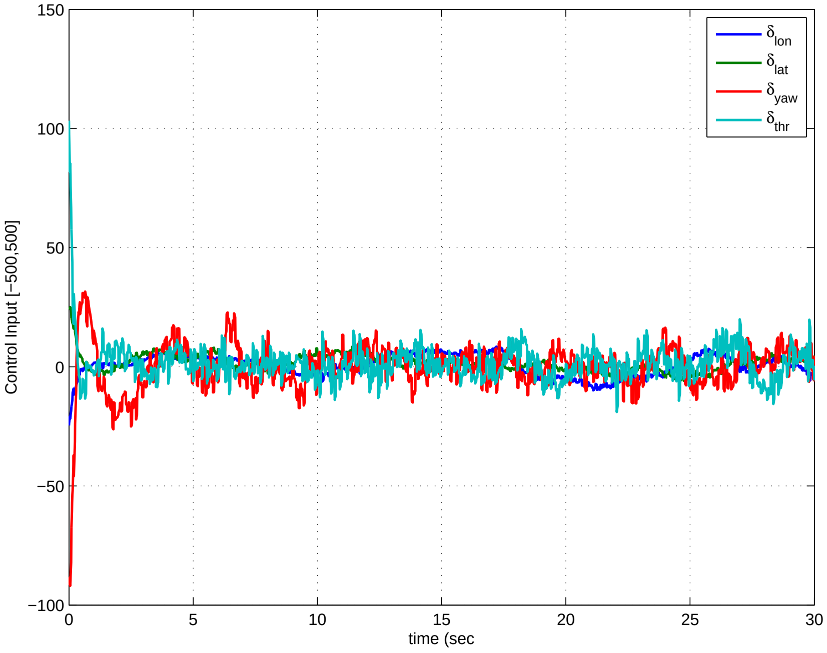


Figure 5.23: Outer Loop Output LQ Control Effort

5.2.2.3 Outer Loop Static $\mathcal{H}_\infty$ Controller

It's expected that the output feedback static $\mathcal{H}_\infty$ will outperform the optimal output feedback gains synthesized in the previous section. The reasoning behind this is the inclusion of the disturbance response in the solution algorithm. This is seen in the inner loop controller design section in § 5.2.1.3 and § 5.2.1.4. The static $\mathcal{H}_\infty$ inner loop gains were better in the presence of gusts. This is expected to be the case for the outer loop controller as well. Especially when looking at the dismal performance of the optimal output feedback gains in the previous section.

To synthesize these gains the same algorithm used in § 5.2.1.4 was implemented with the new weighting matrices in Equation 5.55 to keep the framework for the problem the same as that in the Output LQ case. The disturbance matrix $\mathbf{D}_O$ is the same used in the inner loop problem. The disturbances are still the gusts across all 4 modes, along with the chatter on the longitudinal and lateral input channels. The only item missing to solve the algorithm is the system L_2 gain γ. This was found iteratively, reduced until the algorithm no longer converges. The value selected is 85.7. Recall that the magnitude is relative to the magnitude of the weighting factors in the matrices $\mathbf{Q}$ and $\mathbf{R}$. This selection of parameters and the system model results in the gains in Equation 5.62.

$$\mathbf{K} = \begin{bmatrix} -58.991 & 0 & 0 & -35.013 & 0 & 0 & 0 \\ 0 & 69.477 & 0 & 0 & 52.272 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 274 \\ 0 & 0 & -193.69 & 0 & 0 & -285.72 & 0 \end{bmatrix} \tag{5.62}$$

When these gains are used in the Simulink$^{\text{TM}}$ outer loop model, the performance is better than that of the Output LQ gains, which is the same case as in the inner loop controller design. One obvious difference is there is no coupling in the controller, unlike the Output LQ gains developed in the previous section. The performance of the controller can be seen in the Figures 5.24, 5.25, 5.26, and in Table 5.6.

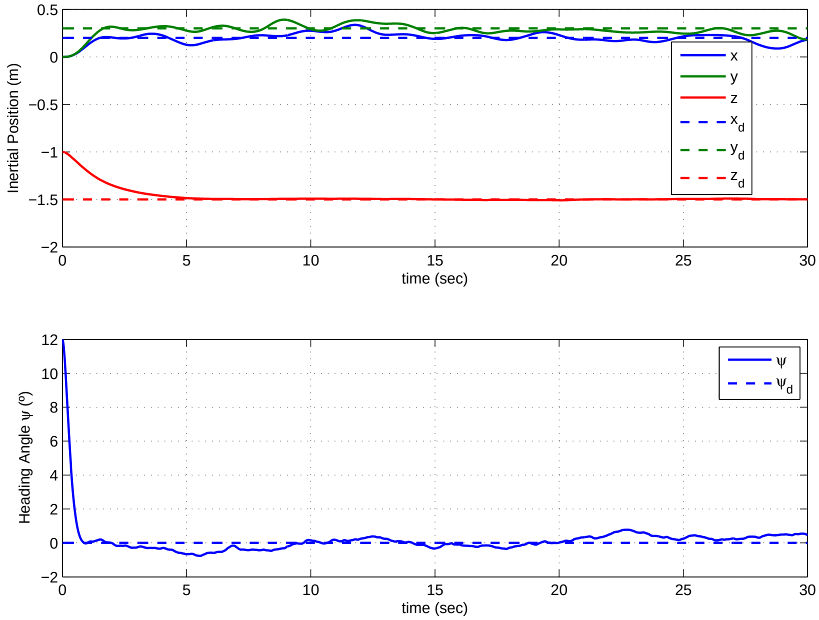


Figure 5.24: Outer Loop Static $\mathcal{H}_\infty$ Tracking Performance

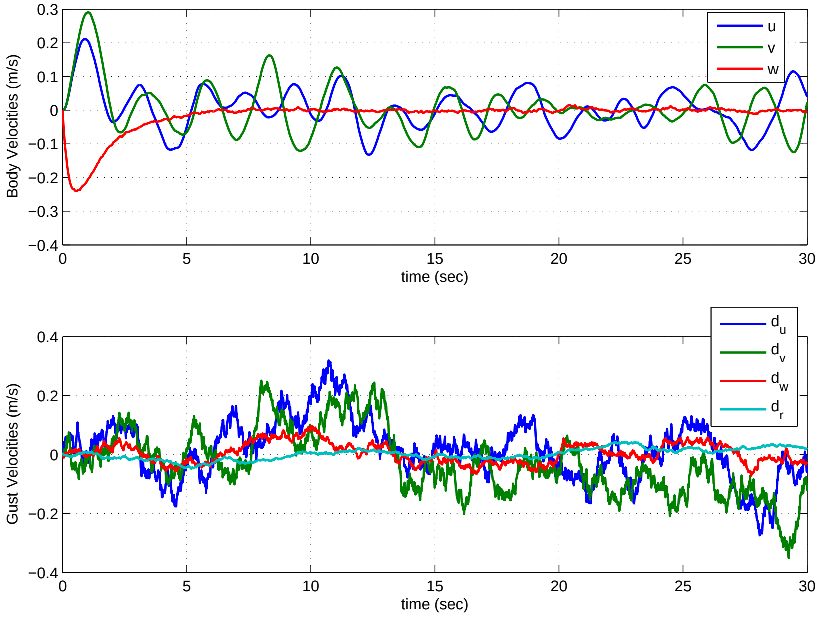


Figure 5.25: Outer Loop Static $\mathcal{H}_\infty$ Body Rates and Disturbances

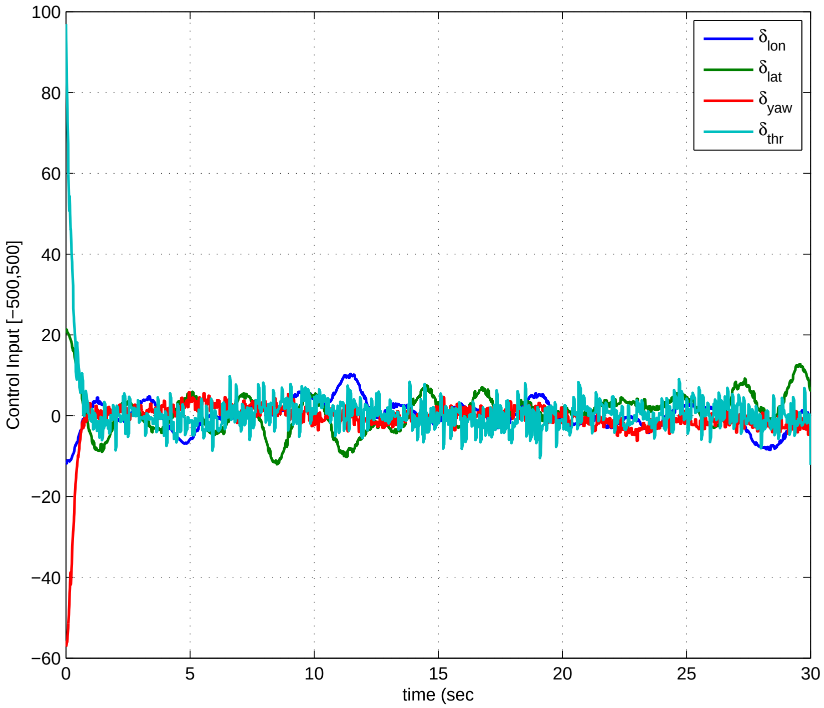


Figure 5.26: Outer Loop Static $\mathcal{H}_\infty$ Control Effort

State	Steady State RMS Covariance
x	0.075 (meters)
y	0.046 (meters)
z	0.007 (meters)
u	0.059 (m/s)
v	0.050 (m/s)
w	0.003 (m/s)
ψ	0.246 (degrees)

Table 5.6: Outer Loop Static $\mathcal{H}_\infty$ Controller Steady State RMS Covariance

The first item to observe from Figure 5.24 is that there is much less variance in the tracking performance because the control inputs are affected only by their respective modal errors. The tracking performance is excellent, with the most significant steady state variance in the position hold. This is to be expected because this is where the majority of the disturbance is acting. The gusts along the $\hat{e}_x$ and $\hat{e}_y$ axes are much stronger than the yaw or heave disturbances when referring to Figure 5.25. The control inputs are all within a reasonable limit as well. Table 5.6 provides the verification of what would intuitively be expected based on the performance in Figure 5.24. The steady state variances in the outer loop states are much less using the gains in Equation 5.62 rather than those in Equation 5.61. From the three outer loop controllers developed, the choice is obvious. The static $\mathcal{H}_\infty$ is the best choice. The design is simplistic and uses the outer loop controller structure. There is no need to implement the Kalman Filter, and therefore only the outputs are necessary. The performance is as good as can be expected from a static controller and therefore Equation 5.62 will be the gain matrix to close the outer loop.

Chapter 6

Conclusions and Future Work

The work in this thesis advances the understanding of the linear dynamics of micro scale quadrotors, and offers further perspective on the stability of such platforms. In spite of their popularity, quadrotors are void of any definitive system identification work. This thesis identifies an unstable state space model of a micro quadrotor using time domain system identification software known as SIDPAC. A state space model offers several advantages to other modeling formats. The stability of a platform can be quickly assessed, as well as the controllability and observability. Using a state space model also presents the ability to develop linear model-based controllers. These controllers are useful because they harness the natural response of the MAV to improve vehicle stability, closed loop damping, or disturbance rejection. Several controller options are also explored after determining the state space model of the microquad. This work helps to understand the requirements to successfully automate MAVs in the future.

6.1 Concluding Remarks

6.1.1 Nonlinear Modeling

A full derivation of the first principles model of the microquad was developed. Included in this derivation are the sources for the moments and forces that manifest the motion of the vehicle. From this point, the entire nonlinear equations of motion would require the calculation and measurement of innumerable parameters such as moments of inertia, aerodynamic force and moment coefficients, reference areas, etc. These measurements could be inaccurate as well as very time consuming. Therefore the assumptions necessary to make to reduce the model into an LTI system are offered as well. Along with these assumptions the assumed structure of the LTI model is presented to assist in the system identification procedure. The structure is decoupled into 4 separate modes: longitudinal, lateral, yaw and heave.

6.1.2 Micro Quadrotor System Identification

Assuming the structure offered by the simplification of the nonlinear equations of motion, a time domain system identification procedure is performed on the microquad. After collecting the time domain data, it must be resampled to ensure proper sampling rates and to ensure the onboard and offboard data are aligned. After resampling, the data must be cropped and smoothed, to remove noise and any flight regimes outside the reference flight condition. Once the data has been post-processed and the parameter estimates are obtained, they are validated by comparing the average parameter value from the data sets with a data set removed heuristically. Because the microquad cannot be flown without feedback control the open loop dynamics are extracted from the stable closed loop model identified. The results are unstable, and because this is essentially counter-intuitive they are checked against two other methods for obtaining estimates of the open loop dynamics. The unstable parameter values are verified and an unstable state space model is assumed for the microquad about hover.

6.1.3 Linear Controller Development

Using the unstable model obtained from the system identification process, several MBCs are investigated for both the inner loop and outer loop of the microquad. 4 choices are explored for a stabilizing inner loop controller designed to quickly reject disturbances and stabilize the attitude of the microquad. The SISO PD gains determined from successful implementation during flight testing yield acceptable results. However, in the presence of disturbances the best performing controllers are the LQG controller, and the constrained output feedback $\mathcal{H}_\infty$ gains were the best performing controllers. However, the LQG controller was foregone because implementing a Kalman Filter on the Mote is not possible, and the static $\mathcal{H}_\infty$ gain set requires too high of a proportional gain. The majority of the sensor noise is present in the estimates of the attitude angles, making a large proportional gain problematic. For these reasons, the SISO PD gain set is chosen to stabilize the inner loop controller because it has only marginally reduced performance in comparison to the MBCs, and because it has been successfully tested to close the inner loop.

The outer loop controller is developed around the closed inner loop to hold a position, altitude, and heading in the presence of wind gusts. In similar fashion to the inner loop, three choices are considered. The LQG is the best performing controller in the presence of disturbances and measurement noises. The output feedback LQR controller introduces unnecessary coupling in the gain matrix, and consequently creates exaggerated disturbance responses. The Static $\mathcal{H}_\infty$ gains have comparable performance to the LQG controller, yet are much simpler to implement because of the fixed structure of the microquad. Therefore, because of the disturbance rejection properties and the simple yet elegant design of the $\mathcal{H}_\infty$ controller, it is chosen as the best option for the outer loop station keeping controller.

6.2 Recommendations for Future Work

To successfully reach an autonomous platform as envisioned, there are still several areas of this research to be extended. The first step is to successfully implement a station keeping control algorithm on the platform. The altitude and heading control is already developed and successfully tested. This just leaves an effective position controller to be created. To accomplish this several tasks should be completed.

The first recommended objective is to improve the model accuracy. In order to do this it would be fruitful to collect more system identification data. The goal of this endeavor is to reduce the standard deviation in the parameter estimates, until the mean of the standard error and the standard deviation converge. This will hopefully better represent the actual stability and control derivatives of the state space model.

The second effort is to improve the Simulink$^{\mathrm{TM}}$ model. There are a number of areas where the model can be improved. As mentioned earlier it would be helpful for the model's prediction accuracy to include several nonlinearities that occur in the physical system. To do this it would be a good idea to develop a model for the actuator (the ESC/DC motor). Where the input to this is the PWM duty cycle, and the output is the motor RPM. This would allow the inclusion of a limit on the RPM rate, and the RPM itself. A big improvement would be to also include delays in the flow of the system and its communication to accurately mimic the communication of the system and to better model phase delay. Other effects to investigate would be coulomb friction in the motor actuator model, and quantization in the PWM commands. Adding these effects should definitely improve the fidelity of the microquad Simulink$^{\text{TM}}$ model, and therefore will better predict the performance of developed controllers.

In addition to improving the model, it would be prudent to develop a controller specifically designed to handle model uncertainty. The idea behind any MAV is to mass develop them for either commercial use or military use. Hence, model parameters will vary with design and tooling constraints, and therefore controller performance will vary and even instability can occur. A commonly used controller designed to handle model uncertainty is known as μ synthesis with DK-Iteration. The algorithm for synthesizing such a controller can be found in [23]. It is also a function embedded in Matlab$^{\text{TM}}$ known as '*dksyn.m*'. To develop and test this controller uncertainty in the model must be characterized as well.

As far as hardware improvements go for the microquad, it would be a wise investment of time to perform a trade study on ESCs in particular. A common feature to many COTS ESCs is known as 'braking'. This works to slow down the RPM of the motors faster than just the inertia and damping of the motor. Better rotors would also be a huge benefit to the microquad. If 3x2 counter rotating pairs were available in carbon fiber this would improve efficiency, reduce mass imbalance (and vehicle mass), increase stiffness and decrease blade flapping. Sensor improvements are also a recommendation. Selecting sensors that are less susceptible to vibration noise would help the development of an inner loop controller. If the noise is small enough to be filtered out, larger proportional gains can be implemented, increasing the response speed.

In addition to improving the sensors already integrated, new sensors are a necessity if the microquad is to implement position control without the use of a Vicon$^{\text{TM}}$ system which is envisioned in the future of the program. A good choice for lightweight sensors to include would be an array of Centeye$^{\mathrm{TM}}$ optic flow sensors. These would provide the microquad with the ability to navigate amongst obstacles. This would bring the microquad even closer to an autonomous MAV, and capturing the full potential of platforms this small. While there is still more to do to obtain a fully autonomous MAV, the development of an efficient controller to effectively stabilize the vehicle is paramount, and this thesis offers an important step to reaching this goal.

Bibliography

[1] S. Boubdallah, *Design and Control of Quadrotors with Application to Autonomous Flying.* Ph.D. Dissertation Ecole Polytech Federale de Laussane, Laussane, Switzerland, 2007.

[2] G. Chowdhary, J. Ottander, E. Salaun, E. Johnson, *Low Cost Guidance, Navigation, and Control Solutions for a Miniature Air Vehicle in GPS Denied Environments.* Georgia Institute of Technology, Atlanta, Georgia, 2009.

[3] G. Gremillion, J.S. Humbert, *System Identification of a Quadrotor Micro Air Vehicle.* AIAA-2010-7644, AIAA Atmospheric Flight Mechanics Conference. Toronto, Ontario, Canada, August, 2010.

[4] J.G. Leishman, *Principles of Helicopter Aerodynamics.* Cambridge University Pres, New York, Second Edition, 2006.

[5] R. Celi, *ENAE635 – Helicopter Stability and Control, Class Notes.* University of Maryland, College Park, 2011.

[6] B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation.* John Wiley and Sons, Inc., Hoboken, New Jersey, Second Edition, 2003.

[7] R.C. Nelson, *Flight Stability and Automatic Control.* McGraw-Hill, New York, 2nd Edition, 1998.

[8] M.De L.C de Oliveira, *Modeling, Identification, and Control of a Quadrotor Aircraft.* Master's Thesis, Czech Technical University in Prague, 2011.

[9] P.-J. Bristeau, P. Martin, E. Salaun, N. Petit, *The Role of Propeller Aerodynamics in the Model of Quadrotor UAV.* Proceedings of the European Control Conference 2009. Budapest, Hungary, August 2009.

[10] E.A. Morelli, *System Identification Programs for Aircraft.* AIAA-2002-4704, AIAA Atmospheric Flight Mechanics Conference. Monterey, California, August 2002.

[11] S. Colton, *The Balance Filter, A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform.* Massachusetts Institute of Technology, Cambridge, Massachusetts, 2007.

[12] C.-T. Chen, *Linear System Theory and Design.* Oxford University Press, Inc., New York, New York, 1999.

[13] V. Klein, E.A. Morelli, *Aircraft System Identification: Theory and Practice.* AIAA Education Series, Reston, Virginia, 2006.

[14] M. Tischler, R. Remple, *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples.* AIAA Educations Series, Reston, Virginia, 2006.

[15] J.K. Conroy, J.S. Humbert, D.J. Pines, *System Identification of a Rotary Wing Micro Air Vehicle.* Journal of the American Helicopter Society, Vol. 56, 2011.

[16] L. Ljung, *The System Identification Toolbox: The Manual.* The Mathworks, Inc. Third Edition, 1997, Natick, Massachusetts, 1997.

[17] E.R. Uhlrich, J.S. Humbert, D.J. Pines, *Pitch and Heave Control of Robotic Samara Micro Air Vehicles.* Journal of Aircraft, Vol. 47, No. 4, 2010.

[18] J.S. Bendat, A.G. Piersol, *Random Data Analysis and Measurement Procedures.* John Wiley and Sons, Inc., Hoboken, New Jersey, Fourth Edition, 2010.

[19] C.F. Gauss, C.H. Davies, *Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Section [microform]: a Translation of Gauss's "Theoria Motus".* Little Brown, Boston, Massachusetts, 1857.

[20] J. Gadewadikar, B.M. Chen, *Structured $\mathcal{H}_\infty$ Command and Control-Loop Design for Unmanned Helicopters.* Journal of Guidance, Control, and Dynamics, Vol. 34, No. 4, 2008.

[21] J. Gadewadikar, *H-Infinity Output-Feedback Control: Application to Unmanned Aerial Vehicle.* PhD Thesis, University of Texas at Arlington, 2007.

[22] R.L. Swaim, D.K. Schmidt, P.A. Roberts, A.J. Hinsdale, *An Analytical Method for Ride Quality of Flexible Airplanes.* AIAA Journal, Vol. 15, No. 1, 1976.

[23] S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design.* John Wiley and Sons, Ltd., Chinchester, West Sussex, UK, 2005.

[24] B.D.O. Anderson, J.B. Moore, *Optimal Control: Linear Quadratic Methods.* Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[25] W.S. Levine, M. Athans, *On the Determination of the Optimal Constant Output Feedback Gains for Linear Multivariable Systems.* IEEE Transactions on Automatic Control, Vol. AC-15, No. 1, 1970.

[26] D.D. Moerder, A.J. Calise, *Convergence of a Numerical Algorithm for Calculating Optimal Output Feedback Gains.* IEEE Transactions on Automatic Control, Vol. AC-30, No. 9, 1985.

[27] F.L. Lewis, V.L. Syrmos *Optimal Control.* John Wiley and Sons, Inc., New York, New York, 1995.