IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2012

Model development, system identification, and control of a quadrotor helicopter

Matthew Rich
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Electrical and Electronics Commons

Recommended Citation

Rich, Matthew, "Model development, system identification, and control of a quadrotor helicopter" (2012). *Graduate Theses and Dissertations*. 12770.
https://lib.dr.iastate.edu/etd/12770

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

المنارة للاستشارات

Model development, system identification, and control of a quadrotor helicopter

by

Matt Rich

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Nicola Elia, Major Professor
Phillip Jones
Namrata Vaswani

Iowa State University
Ames, Iowa
2012

Copyright © Matt Rich, 2012. All rights reserved.

TABLE OF CONTENTS

LIST OF TABLES . . . vi

LIST OF FIGURES . . . vii

ACKNOWLEDGEMENTS . . . ix

ABSTRACT . . . x

CHAPTER 1. Introduction . . . 1

- 1.1 Literature Review . . . 3
- 1.2 Objectives . . . 4
- 1.3 Summary . . . 5
- 1.4 Notation . . . 6

CHAPTER 2. Mathematical/Physical Framework . . . 12

- 2.1 Frames of Reference . . . 12
  - 2.1.1 Inertial (Earth) Frame of Reference . . . 12
  - 2.1.2 Body Frame of Reference . . . 13
  - 2.1.3 Body Orienation . . . 15
  - 2.1.4 Changing Frames . . . 16
- 2.2 Newton-Euler Rigid Body Dynamics . . . 21
  - 2.2.1 Momentum . . . 22
  - 2.2.2 Force . . . 23
  - 2.2.3 Angular Momentum . . . 24
  - 2.2.4 Torque . . . 25
  - 2.2.5 Equation of Motion . . . 27

للاستشارات المنارة

2.2.6 Moment of Inertia Tensor 27

CHAPTER 3. System Architecture 29

3.1 Overall System 29

3.2 Quadrotor 30

3.2.1 Body Frame Axes and Rotor Numbering 30

3.2.2 Physical Architecture 31

3.2.3 Model Architecture 32

3.3 Sensor System 34

3.4 Control System 34

3.5 Communication System 35

CHAPTER 4. Symbolic Modeling 36

4.1 Rigid Body Dynamics 36

4.1.1 Rigid Body Dyanmics State Equation 36

4.2 Powertrain 38

4.2.1 Rotor Force 38

4.2.2 Rotor Torque 43

4.2.3 Rotor System Wrench 46

4.2.4 BLDC Motors 46

4.2.5 Electronic Speed Controls 48

4.2.6 ESC-Motor System 49

4.2.7 Battery 50

4.3 Gravity 51

4.3.1 Body Frame of Reference Gravitational Force 51

4.3.2 Body Frame of Reference Gravitational Torque 51

4.4 Other Aerodynamic Effects 52

4.5 GU-344 53

4.5.1 Gyroscope Feedback 54

4.5.2 Tracking Control 54

للاستشارات المنارة

4.5.3 Signal Mixer . . . 55
4.6 Camera System Model . . . 56
4.7 Communication System Model . . . 56
CHAPTER 5. System Identification . . . 58
5.1 Measurement Equipment and Software . . . 58
5.2 Rigid Body Dynamics Parameters . . . 59
5.2.1 Mass . . . 59
5.2.2 Moment of Inertia Tensor . . . 60
5.3 ESC-Motor Parameters . . . 62
5.3.1 Motor Parameters . . . 63
5.3.2 ESC Parameters . . . 63
5.4 Battery Parameters . . . 64
5.5 Rotor Parameters . . . 65
5.5.1 Thrust Constant . . . 65
5.5.2 Thrust Velocity Adjustment Factor . . . 67
5.5.3 H-Force Constant . . . 72
5.5.4 Drag Constant . . . 72
5.5.5 Equivalent Moment of Inertia . . . 76
5.6 GU-344 Parameters . . . 78
5.6.1 Input/Output . . . 78
5.6.2 Gyroscope Feedback Gains . . . 78
5.6.3 Throttle, Aileron, and Elevator Channel Proportional Gain Elements . . 81
5.6.4 Rudder Channel Gain Elements . . . 84
5.7 Camera System Parameters . . . 86
5.7.1 Sampling Rate . . . 86
5.7.2 Latency . . . 86
5.8 Communication System Identificaiton and Parameters . . . 86
5.8.1 Communication Latency . . . 86
5.8.2 Channel Transfer Characteristics . . . 87

5.9 Collective Results 90

CHAPTER 6. Nested-Loop PID Control 93

6.1 Initial Stabilizing Design 93

6.2 Heading Control 94

6.3 Position Control 96

6.3.1 Orientation Control 96

6.3.2 Lateral and Longitudinal Control 97

6.3.3 Height Control 99

6.4 Pseudo-Nonlinear Extension 101

CHAPTER 7. LQR Control 102

7.1 LQR Weight Selection 102

7.2 Linear Open Loop Model for LQR Design 104

7.2.1 Subsystem Decomposition 107

7.3 Initital LQR Design 108

7.3.1 Common Form LQR Design 108

7.4 Improved LQR Design 110

7.4.1 H_2 Control 110

7.4.2 Design Accounting for Degree of Freedom Conflict 112

7.4.3 Results 117

CHAPTER 8. Conclusion 121

8.1 Summary 121

8.2 Further Work 122

APPENDIX A. Earth as a Frame of Reference 123

APPENDIX B. Pseudo-Inverse Matrix for Overdetermined Systems 124

APPENDIX C. Linearization 126

APPENDIX D. OptiTrack System Setup and Euler Angle Issues 127

BIBLIOGRAPHY 129

LIST OF TABLES

1.1 Recurring Symbols 7

5.1 Equipment 58

5.2 Software 59

5.3 Open Loop Numerical Parameters 90

LIST OF FIGURES

2.1 Frames E, E', and B 17

2.2 Position Radii for Rigid Body 21

3.1 Overall System 29

3.2 Body Axes and Rotor Numbering 31

3.3 Quadrotor Physical Architecture 31

3.4 Quadrotor Model Architecture 32

3.5 Control System Setup 34

3.6 Communication System 35

4.1 Equivalent DC Motor Circuit 47

4.2 Assumed GU-344 Model 53

5.1 Example Battery Discharge Curve 64

5.2 Iterative vs. Quadratic Thrust with Z-Axis Velocity 68

5.3 Iterative vs. Quadratic Thrust with XY-plane Velocity 69

5.4 Iterative vs. Quadratic Torque with Z-Axis Velocity 69

5.5 Iterative vs. Quadratic Torque with XY-plane Velocity 70

5.6 Iterative vs. Quadratic + Adjustment Thrust with Z-Axis Velocity . . 72

5.7 Effects of Thrust Adjustment Factor 73

5.8 Time-Frequency Motor Analysis 78

5.9 Gyroscope Feedback Parameterization Experimental Setup 79

5.10 Pitch/Roll Rate Step Response: Experiment vs. Simulation 85

5.11 Various Communication System Latency Trials 87

5.12 Aileron/Elevator Response through Trainer Port 88
5.13 Aileron/Elevator Response without Trainer Port 89
5.14 Aileron/Elevator Channel Modeled Transfer Function Step Response . 90

6.1 Nested Loop PID Architecture 94
6.2 PID Heading Step Commands 95
6.3 PID Lateral Step Commands 97
6.4 PID Longitudinal Step Commands 98
6.5 PID Height Step Commands 100

7.1 LQR Heading Step Commands 117
7.2 LQR Lateral Step Commands 118
7.3 LQR Longitudinal Step Commands 119
7.4 LQR Height Step Commands 120

ACKNOWLEDGEMENTS

Dr. Nicola Elia *Professor of Electrical Engineering:*

For providing the laboratory, platform, camera system, and other equipment used in this thesis, for useful insight and advice, and for serving as my major professor.

Dr. Phillip Jones *Professor of Computer Engineering:*

For providing the FPGA and necessary VHDL to close the control loop, and for numerous forms of assistance and support throughout this work.

Jeff Wick *Software Engineer / MicroCART senior design team member:*

For writing the software used to implement the control desings of this thesis, and helping with experimental control tuning.

Peter Ha *Electrical Engineer / MicroCART senior design team member:*

For helping with the setup and execution of several parameter identification experiments and data collection, as well as experimental control tuning.

Damek Svec *Electrical Engineer / undergraduate research assistant for Dr. Jones:*

For helping with the setup and execution of parameter identification experiments and data collection.

ABSTRACT

In this thesis a dynamical model is developed for general six degrees of freedom quadrotor vehicle. This is done modularly, and in a layered way. All component models are developed individually with various levels of dynamical complexity parameterized, themselves forming interconnected subsystems that together define the resulting vehicle model. The individual components and subsystems are hence relatively independent of the rest of the model as a whole and can, if desired, be easily extracted with varying levels of complexity selectable through parameters set by the user. Along with the more general vehicle hardware dynamics, existing on board electronics, a network architecture including infrared cameras and operating system based control, and wireless communication systems are modeled. All model parameters are identified with the theoretical background, experimental procedure, and numerical results given for each. Both nested-loop PID and LQR control schemes are developed and implemented, with the resulting performance of each compared to the other as well as the nonlinear simulation predictions. The LQR design is atypical in that it makes advantageous use of a systematic procedure to obtain appropriate cost weights, which capture design specifications while taking direct account of the system structure. The procedure leads to input-state coupling weights consistent with the dynamical limitations of the vehicle, which are key to the successful applicability of the LQR method for the quadrotor. All results are discussed with potential further work, issues and improvements highlighted.

CHAPTER 1. Introduction

Quadrotor helicopters are, as the name suggests, aircraft consisting of four independently controlled rotors which in sum provide all vehicle actuation. The rotation of all four is set up such that there are two oppositely rotating pairs–each member of each pair located directly across from the other–such that they are able to counteract each other to a greater or lesser degree depending on their inputs in order to manipulate the in-plane rotation, or yawing motion, of the vehicle. Depending on how forward/reverse movement is defined for the vehicle, there may either be two opposing pairs of rotors whose balance or imbalance controls rolling and pitching motion, or a single pair controlling each type of motion (see Section 3.2.1). Heave, i.e. up/down motion in the vehicle or *hypothetical* pilot frame of reference, is controlled by the collective thrust of all four rotors. Quadrotor helicopters differ from other common types of rotor aircraft in several ways, depending the type.

Conventional helicopters typically have one main rotor which is used to control heave as well as lateral and longitudinal motions through the use of mechanical manipulation of the collective and cyclic pitch of the rotor blades using a swash plate with accompanying joints and linkages. In such a setup the rotation speed of the main rotor can be regulated to a constant value while the collective pitch, i.e. the effective angle of attack of both rotor blades, can be increased or decreased to increase or decrease the resulting thrust. Lateral and longitudinal cyclic pitch inputs manipulate the effective angle of attack of each rotor blade pair such that it increases and decreases as a function of rotor position reaching a maximum or minimum value at either the right or left, and forward or reverse positions respectively[1]. These inputs essentially cause the effective rotor disk to tip right/left or forward/back directing the thrust vector accordingly to create the desired motion. Yawing control, along with some measure of lateral control, is accomplished through the use of a tail rotor having separate collective pitch actuation. This is the most common helicopter architecture, with examples such as the Bell UH-1 Iroquois (Huey), Sikorsky UH-60 Black Hawk, and Bell 206 model helicopters being probably among the most easily recognizable.

Other architectures typically remove the necessity of a tail rotor in one way or another. They include coaxial designs, in which two opposing rotors are stacked one above the other, tandem rotor designs in which two opposing rotors are offset in the forward/reverse direction, as well as NOTAR designs making use of internal fans and directed air. Less common intermeshing designs where two opposing rotors are set so that their planes of rotation intersect as well as other variations such as tilt rotor aircraft also exist. The Russian Kamov Design Bureau produces many coaxial helicopters. Relatively well known examples of the tandem rotor design would be the Boeing CH-47 Chinook or Boeing Vertol CH-46 Sea Knight.

Quadrotor helicopters require neither a tail rotor or any cyclic pitch controls. Most current quadrotors are smaller electrically powered models which use independently controlled speeds to increase or decrease the thrust and torque generated by each of the four rotors, further removing the need for collective pitch controls. This allows for some important advantages, including the decreased mechanical complexity–hence increased robustness–and the potential for somewhat simplified rotor modeling, e.g. assuming each rotor force and torque vector has an effectively constant orientation with respect to the vehicle fuselage and ignoring flapping dynamics. Also, as with tandem rotor designs, lacking a tail rotor (or other similarly purposed mechanisms) allows the quadrotor to devote all vehicle power to producing lift. This allows for significant payload capacity in relation to vehicle weight. This makes them ideal platforms for applications that might involve attached cameras or other equipment, such as autonomous surveillance or exploration of difficult to reach and/or dangerous areas.

With such a configuration however, the entire vehicle must tip in one direction or another in order to direct the rotor thrusts to actuate lateral and/or longitudinal motion. This could be seen as a potential disadvantage or advantage depending on perspective. It does constrain the dynamics of the vehicle, in that it cannot cause acceleration forward or back or from side to side while maintaining a given orientation. This can be exploited with some control architectures (see Section 6.1), by essentially treating the system as a *orientation system* in series with a *translational system*, but it can also create potential difficulties with control that treats the vehicle as a whole (see Sections 7.1 and 7.3) due to the significant interdependence between angular and linear degrees of freedom.

1.1 Literature Review

Modeling and control of quadrotor helicopters is at present a common area of research and application, with various levels of model complexity and control design described throughout recent literature. The modeling of the vehicle dynamics is typically kept relatively simple. Control methodologies range from linear proportional-integral-derivative (PID)[2, 3, 4, 5, 6, 7], to linear quadratic (LQ) optimal[8, 3, 7, 9, 10, 11, 12], to various adaptive and semi-adaptive schemes[13], to feedback linearization[14, 15], backstepping[14, 16], and dynamic inversion[17, 18], among various others. The objectives and the level of control developed varies, with some sources seeking only to control the orientation of the vehicle, and others seeking to control orientaiton and position.

These and other sources assume various sensor feedback forms such as on board inertial measurement units (IMU) measuring linear accelerations and angular rates, on board infrared and SONAR devices measuring distances, on board cameras, and GPS, and sometimes off board visual or infrared camera systems. Typically when the control is implemented experimentally it is done so using on board hardware, sometimes accompanied by off board high-level or planning control or trajectory generation[1].

---

[1] Unfortunately in some literature control is not always implemented experimentally, i.e. sometimes only mathematical justification or simulation results are provided, and even when experiments are performed sometimes no comparison to model prediction is provided for verification. In this thesis all control designs are implemented experimentally and the results provided and discussed.

1.2 Objectives

The main objectives of this thesis were as follows:

- Develop a detailed, general parametric model of quadrotor vehicle dynamics.
- Develop systematic, repeatable procedures for system parameter identification experiments and data analysis.
- Develop a modular, expandable, parametric nonlinear simulation model from the mathematical model.
- Design and implement effective linear control to stabilize the vehicle, provide simple reference tracking, and model verification.
- Work cooperatively with the MicroCART project.
- Produce a thesis as accessible as possible to readers not already possessing expertise in the topics presented.

In developing a general mathematical model and simulation of quadrotor dynamics, an approach was taken to allow for a maximum amount of modularity and flexibility. Individual components were modeled and parameterized separately, with their interactions within and between subsystems kept generic.

For example the rigid body dynamics system (Figure 3.4) accepts as input a *wrench* vector, i.e. a generic force and torque vector, without any regard for where or how it is produced. Similarly the model each rotor accepts as input a angular speed and acceleration (as scalar values independent of vector frame of reference), and outputs an individual wrench vector defined with respect to a local frame of reference parametrically.

The modeling is intended to be useful for applications and purposes beyond those of this thesis, and all identification experiments are presented with a level of detail intended to be sufficient for easy repeatability and/or adaptation to other similar hardware.

Beyond this, working with the MicroCART senior design project to develop the laboratory network control infrastructure and an initial working vehicle control were intended together to provide an existing platform and environment for educational purposes as well as further research. The parametric and flexible nature of the modeling along with the developed laboratory infrastructure were meant to provide some groundwork for further research into cooperative and networked autonomous systems.

Finally this thesis was written with the intention that it be as accessible and useful as possible to senior level undergraduates, beginning graduate students, and/or those not already possessing a great amount of familiarity with the modeling and control topics presented, while providing a more in depth development of such topics than is typically found in similar work.

Copies of existing data analysis and identification functions, and a custom Simulink library containing all components of the nonlinear model used for simulation purposes in this work are freely available online[19].

1.3 Summary

In what follows, Part I provides a general framework and background. The general notation is presented along with development of the model frames of reference, orientation representation, and change of coordinates matrices. Also the 6DOF Newton-Euler rigid body equations of motion are derived with respect to a general frame of reference within the body and an outline of Taylor Series based linearization of nonlinear differential equation models is given.

Part II deals with the general symbolic model development as well as parameter identification. First an overall system architecture is described, followed by a model architecture which illustrates the component-by-component and nested-subsystem nature of the overall model created. Following this each component is symbolically modeled with emphasis given to thorough derivation of the resulting equations as well as the resulting equations themselves in that they are kept as general purpose and parameterized as possible, allowing for easy extraction, modification, and/or use outside the overall architecture of this work. The modeling presented here is generally speaking more thorough than is usually encountered in similar literature, with some distinct additions made in order to yield increased accuracy without significant increase in complexity. After the symbolic modeling is complete, all system parameters are identified with the theory, procedure, and results provided for each and collected in Table 5.3.

Part III deals with control design and implementation. First a nested-loop PID control architecture is developed, along with a pseudo nonlinear extension allowing the linear control to be effective at a continuum of heading angles, with the results of both simulation and actual implementation provided and discussed. This is followed by a LQR control scheme based on the states not directly observed by the camera system obtained through nonlinear calculation. The LQR is first designed using a fairly typical methodology, and the weaknesses of this approach (also mentioned in [3]) are pointed out. A systematic method for obtaining better results is then presented and the design using it is described and implemented along with the same pseudo-nonlinear extension as in the PID case. Again the results of both nonlinear simulation and implementation are discussed.

1.4 Notation

Effort has been made here to use symbolic notation that is as intuitive, consistent, and clear as possible while still providing adequate representation. To this end, and undoubtedly with some exceptions, the symbolic representation of quantities follows the general structure:

$$ {}^{[\textsc{Frame of Reference}]}\overset{\left[\substack{\textsc{Time Derivative}\\ \textsc{Other Designation}}\right]}{[\text{MAJOR SYMBOL}]}^{[\textsc{Exponent/Transpose}]}_{[\textsc{Designation/Index}]} $$

The frame of reference is the most commonly missing detail in what follows, which is typically only given for those quantities recurring in more than one frame of reference. Many quantities are defined with respect to a particular frame of reference and are only ever used in that local respect. If such a quantity is ever meant to be represented in another frame of reference, it is accompanied immediately by the appropriate change of coordinates matrix which should give some clarification as to which frame of reference the resulting quantity is with respect to.

As a final aid, Table 1.1 provides a reference for *most* important recurring symbols used.

Table 1.1: Recurring Symbols

Symbol	**Units**	**Brief Description**
E	(none)	inertial/Earth frame of reference basis matrix
e_x	(none)	inertial/Earth frame of reference x unit vector
e_y	(none)	inertial/Earth frame of reference y unit vector
e_z	(none)	inertial/Earth frame of reference z unit vector
B	(none)	chassis/body frame of reference basis matrix
b_x	(none)	chassis/body frame of reference x unit vector
b_y	(none)	chassis/body frame of reference y unit vector
b_z	(none)	chassis/body frame of reference z unit vector
m	kg	vehicle flight mass
g	$\frac{m}{s^2}$	scalar acceleration gravity
J	kgm^2	vehicle flight moment of inertia tensor w.r.t. center of mass
J_{xx}	kgm^2	b_x principle moment of inertia
J_{yy}	kgm^2	b_y principle moment of inertia
J_{zz}	kgm^2	b_z principle moment of inertia
${}^{B}v_o$	$\frac{m}{s}$	body frame origin velocity expressed in body frame
u	$\frac{m}{s}$	component of ${}^{B}v_o$ on b_x
v	$\frac{m}{s}$	component of ${}^{B}v_o$ on b_y
w	$\frac{m}{s}$	component of ${}^{B}v_o$ on b_z
${}^{B}\Omega$	$\frac{rad}{s}$	body frame angular velocity expressed in body frame
p	$\frac{rad}{s}$	component of ${}^{B}\Omega$ on b_x
q	$\frac{rad}{s}$	component of ${}^{B}\Omega$ on b_y
r	$\frac{rad}{s}$	component of ${}^{B}\Omega$ on b_z
$[{}^{B}\Omega]$	$\frac{rad}{s}$	skew symmetric matrix equivalent to ${}^{B}\Omega\times$
Θ	rad	vector of Euler angles
ϕ	rad	*roll* Euler angle

Continued on next page...

Table 1.1 –*Continued from previous page.*

Symbol	**Units**	**Brief Description**
θ	rad	*pitch* Euler angle
ψ	rad	*yaw* Euler angle
$L_{\overleftarrow{\mathrm{BE}}}$	(none)	intertial to body change of coordinates matrix
$L_{\overleftarrow{\mathrm{EB}}}$	(none)	body to inertial change of coordinates matrix
$A_{\overleftarrow{\mathrm{BE}}}$	(none)	$\dot{\Theta}$ to ${}^{\mathrm{B}}\Omega$ change of coordinates matrix
$A_{\overleftarrow{\mathrm{EB}}}$	(none)	${}^{\mathrm{B}}\Omega$ to $\dot{\Theta}$ change of coordinates matrix
${}^{\mathrm{E}}r_o$	m	absolute position vector of the origin of B expressed in E
x	m	component of $[{}^{\mathrm{E}}r_o]$ on e_x
y	m	component of $[{}^{\mathrm{E}}r_o]$ on e_y
z	m	component of $[{}^{\mathrm{E}}r_o]$ on e_z
Λ	(mixed)	rigid body dynamics state vector
${}^{\mathrm{B}}r_{oc}$	m	vector from origin of B to body center of mass
${}^{\mathrm{B}}r_{oc}$	m	skew symmetric matrix equivalent to ${}^{\mathrm{B}}r_{oc}\times$
${}^{\mathrm{B}}F$	N	vector of total force acting on vehicle expressed in B
${}^{\mathrm{B}}Q$	Nm	vector of total torque acting on vehicle expressed in B
${}^{\mathrm{B}}W$	(mixed)	total wrench acting on vehicle expressed in B
${}^{\mathrm{B}}F_r$	N	vector sum of rotor forces acting on vehicle expressed in B
${}^{\mathrm{B}}T_i$	N	vector thrust of rotor i expressed in B
ω	$\frac{rad}{s}$	vector of all four scalar rotor speeds
α	$\frac{rad}{s^2}$	vector of all four scalar rotor accelerations
ω_i	$\frac{rad}{s}$	scalar rotational speed of rotor i
K_T	$\frac{kgm}{rad^2}$	rotor thrust constant
δ_T	$\frac{kg}{rad}$	rotor thrust velocity adjustment factor
Γ_{T_i}	(none)	coordinate unit vector giving direction of rotor thrust i w.r.t. B
${}^{\mathrm{B}}H_i$	N	vector in-plane drag of rotor i expressed in B
K_H	$\frac{kg}{rad}$	in-plane rotor drag constant

Continued on next page...

Table 1.1 –*Continued from previous page.*

Symbol	**Units**	**Brief Description**
Γ_H	(none)	matrix selecting xy-plane hub velocity components w.r.t. B
v_{h_i}	$\frac{m}{s}$	velocity of rotor hub i expressed in B
${}^{\mathrm{B}}Q_r$	Nm	vector sum of rotor torques acting on vehicle expressed in B
${}^{\mathrm{B}}Q_{d_i}$	Nm	vector drag and induced torque of rotor i expressed in B
K_d	$\frac{kgm^2}{rad^2}$	rotor drag torque constant
Γ_{Ω_i}	(none)	unit vector giving direction of rotor angular velocity i w.r.t. B
${}^{\mathrm{B}}Q_{L_i}$	Nm	vector rotor angular momenteum torque of rotor i expressed in B
$\tilde{J}_r$	kgm^2	equivalent rotor and motor moment of inertia
${}^{\mathrm{B}}Q_{F_i}$	Nm	vector rotor force lever arm torque of rotor i expressed in B
${}^{\mathrm{B}}W_r$	(mixed)	total rotor wrench acting on vehicle expressed in B
${}^{\mathrm{B}}G$	N	vector force due to gravity acting on vehicle expressed in B
${}^{\mathrm{B}}Q_G$	Nm	vector torques due to gravity acting on vehicle expressed in B
${}^{\mathrm{B}}W_G$	(mixed)	total gravity wrench acting on vehicle expressed in B
${}^{\mathrm{B}}F_D$	N	vector sum of disturbance forces acting on vehicle expressed in B
${}^{\mathrm{B}}Q_D$	Nm	vector sum of disturbance torques acting on vehicle expressed in B
${}^{\mathrm{B}}W_D$	(mixed)	total disturbance wrench acting on vehicle expressed in B
K_V	$\frac{rad}{Vs}$	motor back-emf constant
K_Q	$\frac{Nm}{A}$	motor torque constant
R_m	Ω	motor resistance
i_f	A	motor internal friction current
$\tilde{V}$	V	vector of all four equivalent motor voltage inputs
$\tilde{V}_i$	V	equivalent voltage input for motor i
P	(none)	vector of all four GU-344 to ESC duty cycle percentages
P_i	(none)	GU-344 to ESC i duty cycle percentage
$P_{\perp}$	(none)	ESC minimum turn on duty cycle
$\tilde{P}_{\perp}$	(none)	GU-344 minimum duty cycle output when active

Continued on next page...

Table 1.1 –*Continued from previous page.*

Symbol	**Units**	**Brief Description**
$\tilde{P}_{\top}$	(none)	GU-344 maximum duty cycle output when active
u_{P_i}	(none)	normalized input command to ESC i
V_b	V	battery voltage
V_0	V	nominal constant operating battery voltage
δ_V	$\frac{V}{s}$	approximate constant voltage loss rate
γ_g	$\frac{s}{rad}$	GU-344 gyroscope gain matrix
γ_p	$\frac{s}{rad}$	GU-344 gyroscope p gain
γ_q	$\frac{s}{rad}$	GU-344 gyroscope q gain
γ_r	$\frac{s}{rad}$	GU-344 gyroscope r gain
k_g	(none)	GU-344 gyroscope overall feedback gain
A_κ	(none)	GU-344 rate tracking state space A matrix
B_κ	(none)	GU-344 rate tracking state space B matrix
C_κ	(none)	GU-344 rate tracking state space C matrix
D_κ	(none)	GU-344 rate tracking state space D matrix
κ_T	(none)	GU-344 throttle channel gain
κ_A	(none)	GU-344 aileron channel gain
κ_E	(none)	GU-344 elevator channel gain
κ_{R_P}	(none)	GU-344 rudder channel proportional gain
κ_{R_I}	(none)	GU-344 rudder channel integral gain
ι	(none)	GU-344 rudder channel integrator state
M_G	(none)	GU-344 signal mixing matrix
u_s	(none)	vector of command inputs to GU-344
u_T	(none)	throttle command
u_A	(none)	aileron command
u_E	(none)	elevator command
u_R	(none)	rudder command

Continued on next page...

Table 1.1 –*Continued from previous page.*

Symbol	**Units**	**Brief Description**
Y_c	(mixed)	camera system output vector
T_c	s	camera system frame rate
τ_c	s	camera system latency
τ_T	s	communication system throttle channel latency
τ_A	s	communication system aileron channel latency
τ_E	s	communication system elevator channel latency
τ_R	s	communication system rudder channel latency
G_{T_x}	s	communication system throttle channel transfer function
G_{A_x}	s	communication system aileron channel transfer function
G_{E_x}	s	communication system elevator channel transfer function
G_{R_x}	s	communication system rudder channel transfer function

CHAPTER 2. Mathematical/Physical Framework

In this chapter the mathematical and physical framework for all the modeling to follow is developed. In the first portion, the two main frames of reference are defined along with the quantities used to describe their relative orientation. Next the matrices allowing for change of coordinates between the frames of reference are developed. The second portion provides a thorough derivation of the Netwon-Euler six degree of freedom equations of motion for a rigid body, and a definition of the moment of inertia tensor.

2.1 Frames of Reference

For the modeling done here, two main frames of reference are utilized. These are the inertial (Earth) frame of reference and the local chassis or body frame of reference.

2.1.1 Inertial (Earth) Frame of Reference

The inertial (Earth) frame of reference for this model: $E \in \mathbb{R}^{3\times 3}$, is defined with the orthonormal basis vectors: e_x, e_y and e_z, which are all elements of $\mathbb{R}^3$. In what follows, the symbol E may be used to refer to the frame generally, or to the basis matrix:

$$E = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}$$

The origin of this frame of reference is fixed with respect to Earth. The unit vector e_z is positive along the acceleration of gravity, or what would be in typical everyday experience simply *straight down*. The orientation of the vectors e_x and e_y is generally arbitrary so long as they follow the right hand rule (i.e. $e_x \times e_y = e_z$, $e_y \times e_z = e_x$, and $e_z \times e_x = e_y$) but must be established in order to define the orientation of the body frame unit vectors with respect to the Earth frame unit vectors.

Though the surface of Earth is is not a perfect inertial frame of reference, it is assumed *close enough* for this model. The Earth frame of reference as defined here, fixed with respect to the surface, is in fact accelerating due to the motion of the planet. The effects of this acceleration, e.g. Coriolis effect, are for the purposes of this model negligible (See Appendix A) and the Earth frame of reference can be treated as effectively inertial [20].

2.1.2 Body Frame of Reference

The body frame for this model: $B \in \mathbb{R}^{3\times 3}$, is defined with the orthonormal basis vectors: b_x, b_y and b_z, with each an element of $\mathbb{R}^3$. In the text that follows the symbol B may be used to refer to the body frame generally, or as the basis matrix:

$$B = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}$$

The body frame is completely fixed within the rigid body, i.e. the origin of this frame is fixed at the nominal[1] center of mass, and the unit vectors rotate with any body rotation. The unit vectors orientations within the body are defined using the right hand rule from the perspective of a hypothetical pilot: b_x is positive forward, b_y is positive right and b_z is positive down. Also the body frame axes are assumed along the principle inertia axes of the body.[2] See [21] for a more detailed exposition on various choices of frames of reference relating to a moving body.

The body frame velocites are defined using the unit vectors. The linear velocities: u, v and w are the body frame of reference origin velocity, expressed as components on b_x, b_y and b_z respectively. Together they form the vector ${}^{\mathrm{B}}v_o$ representing the origin velocity expressed in B. The angular velocities: p, q and r are the components of the body angular velocity vector on b_x, b_y and b_z respectively, where a positive component of angular velocity is defined using the right hand rule. Together they form the vector ${}^{\mathrm{B}}\Omega$, the body angular velocity expressed in B.

---

[1] Nominal because the exact position of the center of mass may be uncertain.

[2] Assuming the axes are along the inierta axes of the body greatly simplifies the equations of motion in many ways. This will be mentioned and expanded upon when appropriate throughout what follows. E.g. 2.2.6

As is fairly intuitive[3], the angular velocity of a rigid body expressed in one frame of reference fixed with respect to the body will be identical to that expressed in another frame of reference also fixed with respect to the body if the two frames of reference have axes that are aligned. In the event their axes are not aligned, the angular veloctiy vector in terms of each differs only by a static rotation matrix[22]. Hence ${}^{\mathrm{B}}\Omega$ does not change if the body frame of reference origin is not at the center of mass.

2.1.2.1 Body Frame Time Derivative

It will be necessary to define the time derivative of the body frame B in order to perform many derivations to follow. As detailed more thoroughly in [1] (in the notation used here):

$$\dot{b}_x = (B\,{}^{\mathrm{B}}\Omega) \times b_x = B\left({}^{\mathrm{B}}\Omega \times \begin{bmatrix}1\\0\\0\end{bmatrix}\right)$$

$$\dot{b}_y = (B\,{}^{\mathrm{B}}\Omega) \times b_y = B\left({}^{\mathrm{B}}\Omega \times \begin{bmatrix}0\\1\\0\end{bmatrix}\right)$$

$$\dot{b}_z = (B\,{}^{\mathrm{B}}\Omega) \times b_z = B\left({}^{\mathrm{B}}\Omega \times \begin{bmatrix}0\\0\\1\end{bmatrix}\right)$$

Where the × symbol is used to represent the vector cross prodcut. Putting these together yields the equation:

---

[3] Proofs can be found in [20] and [22]

$$\dot{B} = \begin{bmatrix} \dot{b}_x & \dot{b}_y & \dot{b}_z \end{bmatrix} = \begin{bmatrix} B\left({}^{\mathrm{B}}\Omega \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right) & B\left({}^{\mathrm{B}}\Omega \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right) & B\left({}^{\mathrm{B}}\Omega \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \end{bmatrix}$$

It is convenient to define the matrix $[{}^{\mathrm{B}}\Omega]$ as a skew-symmetric matrix such that left multiplication with some vector a is equivalent to ${}^{\mathrm{B}}\Omega \times a$, i.e.

$$[{}^{\mathrm{B}}\Omega] = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \Rightarrow {}^{\mathrm{B}}\Omega \times a = \begin{bmatrix} qa_3 - ra_2 \\ ra_1 - pa_3 \\ pa_2 - qa_1 \end{bmatrix}$$

Using this matrix the equation for $\dot{B}$ can be expressed as:

$$\dot{B} = B[{}^{\mathrm{B}}\Omega]I = B[{}^{\mathrm{B}}\Omega]$$

So the time derivative of some vector $c = Ba$ would be:

$$\dot{c} = \dot{B}a + B\dot{a} = B[{}^{\mathrm{B}}\Omega]a + B\dot{a} = B(\dot{a} + {}^{\mathrm{B}}\Omega \times a)$$

2.1.3 Body Orienation

Represenation of the orientation of the body, specifically in this case the body frame axes with respect to the earth frame axes, can be accomplished in more than one way. The two more widely used methods are Euler angles and quaternions. The Euler angle method involves the use of three, fairly intuitive angles to describe the orientaton. The quaternion method is conceptually more complicated and involves the use of four parameters to describe the orientation.

The main disadvantage of the Euler angle method is that singularities occur in the change of coordinates matrices. This occurs at certain angles regardless of which sequence of rotations is used[4]. For the one used here, such problems occur when the pitch Euler angle, defined here as $\theta = \pm\pi/2$. The main disadvantage of the quaternion approach is the increased complexity. For the purposes of this model Euler angles are sufficient since the type of flight that would produce the previously mentioned singularity is not intended. For a development of the quaternion approach, see for example [21].

2.1.3.1 Euler Angles

The Euler angles ϕ, θ, and ψ are used to define the orientation of the body frame axes with respect to the inertial frame axes. They are also sometimes called roll, pitch, and yaw respectively. The sequence in which the three angles are taken is generally arbitrary, but must be established in order to change coordinates from one frame of reference to another. The typical [1] sequence for helicopter modeling is yaw, pitch, roll. This is not universal. For example [1], [21], [8], and [2] among others use this sequence, while such well established sources as [23] use another.

2.1.4 Changing Frames

The inertial and body frames of reference as defined here differ in that the origin of the body frame can be displaced from that of the intertial frame, and the axes of the body frame can change orientation with respect to those of the inertial frame. The orientation change is of particular interest and some change of coordinates is necessary to describe a quantity with one set of basis vectors as opposed to another.

In order to explain the derivation of the change of coordinates matrices, it is useful to define some temporary sets of axes. First, a set with an origin that translates with the body frame of reference but does not rotate with it, say $E' = [e'_x \; e'_y \; e'_z]^T$. Second, a set that has been rotated around e'_z by ψ, denoted by $E'' = [e''_x \; e''_y \; e''_z]^T$. Third, a set that has been rotated around e''_y, by θ, denoted by $E''' = [e'''_x \; e'''_y \; e'''_z]^T$. Finally, rotating by ϕ around e'''_x yields B.

---

[4] A related potential disadvantage of the Euler angle approach is that the three consequtive rotations can be performed in various different sequences, hence one always has to be careful to define which sequence is used.

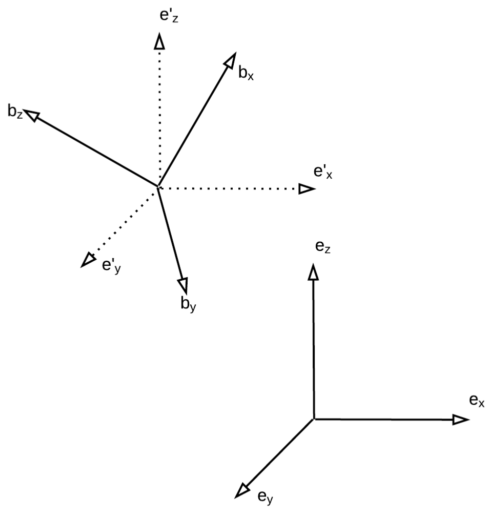


Figure 2.1 Frames E, E', and B

2.1.4.1 Change of Coordinates Matrices

Given some vector element of $\mathbb{R}^3$, represented by coordinates in the body frame of reference (i.e. a weighted sum of the basis vectors in B), there exists an equivalent set of coordinates in the inertial frame basis (i.e. weighted sum of the basis vectors in E) which yield the same vector. In order to obtain the matrix that gives this change of coordinates, it is instructive to look at how one would, using the three rotation process previously described, form each successive set of basis vectors using the previous set.

Starting with E', and first rotating around e'_z by ψ:

$$e''_x = cos(\psi)e'_x + sin(\psi)e'_y + 0e'_z$$

$$e''_y = -sin(\psi)e'_x + cos(\psi)e'_y + 0e'_z$$

$$e''_z = 0e'_x + 0e'_y + 1e'_z$$

Which can be expressed using matrix multiplication as:

$$E'' = E' \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Following the same process for the next two rotations gives:

$$E''' = E'' \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix}$$

$$B = E''' \begin{bmatrix} 1 & 0 & 0) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

Since the orientation of E' is the same as E, this process can be equivalently expressed as:

$$B = E \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

Defining the matrix $L_{\overleftarrow{\mathrm{EB}}}$ as the product the three rotation matrices just used, and multiplying gives:

$$L_{\overleftarrow{\mathrm{EB}}} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

Using this notation, the body frame of reference basis matrix can be obtained from the inertial frame of reference basis matrix using right multiplication by $L_{\overleftarrow{\mathrm{EB}}}$, i.e.

$$B = EL_{\overleftarrow{\mathrm{EB}}}$$

This also means that the change of coordinates from the body to inertial frame coordinates can be accomplished through left multiplication by the matrix $L_{\overleftarrow{\mathrm{EB}}}$. That is to say, for some arbitrary pair a and c such that $Ea = Bc$, the previous development implies:

$$Ea = Bc \Rightarrow a = L_{\overleftarrow{\mathrm{EB}}}c$$

Going from inertial frame coordinates to body coordinates can be accomplished with the matrix inverse of $L_{\overleftarrow{\mathrm{EB}}}$. Since each matrix multiplied to produce $L_{\overleftarrow{\mathrm{EB}}}$ was orthogonal, $L_{\overleftarrow{\mathrm{EB}}}$ is itself orthogonal, and so the inverse is also the transpose[1]:

$$L_{\overleftarrow{\mathrm{BE}}} = L_{\overleftarrow{\mathrm{EB}}}^{-1} = L_{\overleftarrow{\mathrm{EB}}}^{T}$$

$$L_{\overset{\leftarrow}{\text{BE}}} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

And of course:

$$Ea = Bc \Rightarrow L_{\overset{\leftarrow}{\text{BE}}}a = c$$

It is also desirable to change from body frame angular velocities to the equivalent Euler rates. However since the Euler angles are essentially coordinates in multiple reference frames, transitioning between the time derivative of the Euler angles and the body frame angular velocities requires another set of matrices.

The yaw rate represents a coordinate on both $e'_z = e''_z$. Hence in order to translate this coordinate to its equivalent body frame representation, the 2^{nd} and 3^{rd} rotations previously used must be applied. The pitch rate is a coordinate on both $e''_y = e'''_y$, and so only the 3^{rd} rotation matrix should be applied to it. The roll rate exists as a coordinate on both $e'''_x = b_x$, and hence does not require the application of any rotation matrix. Expressing this in mathematical form:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0) \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0) \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

Which can be simplified to give:

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

Where the change of coordinates matrix from Euler rates to body frame angular velocities can be defined as:

$$A_{\overleftarrow{\text{BE}}} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

Going from the body frame angular velocities to the Euler rates can be accomplished with the matrix inverse of $A_{\overleftarrow{\text{BE}}}$.

$$A_{\overleftarrow{\text{EB}}} = A_{\overleftarrow{\text{BE}}}^{-1}$$

$$A_{\overleftarrow{\text{EB}}} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix}$$

2.1.4.2 Change of Coordinates Matrix Time Derivative

In what follows it will also be useful to have an expression for the time derivative of the change of coordinates matrix $L_{\overleftarrow{\text{EB}}}$. Assuming some arbitrary pair a and c such that $Ea = Bc$, i.e. $a = L_{\overleftarrow{\text{EB}}}c$, the equation can be written:

$$EL_{\overleftarrow{\text{EB}}}c = Bc$$

Taking the time derviative of both sides:

$$E(\dot{L}_{\overleftarrow{\text{EB}}}c + L_{\overleftarrow{\text{EB}}}\dot{c}) = B({}^{\text{B}}\Omega \times c + \dot{c})$$

Using the fact that $EL_{\overleftarrow{\text{EB}}} = B$ and cancelling:

$$E\dot{L}_{\overleftarrow{\text{EB}}}c = EL_{\overleftarrow{\text{EB}}}({}^{\text{B}}\Omega \times c)$$

$$\dot{L}_{\overleftarrow{\text{EB}}} = L_{\overleftarrow{\text{EB}}}[{}^{\text{B}}\Omega]$$

2.2 Newton-Euler Rigid Body Dynamics

Newton-Euler equations describe the combined three dimensional linear and angular dynamics of a rigid body. The derivation that follows is general in that the point of reference is not necessarily assumed to be conincident with the body center of mass. The results obtained are then significantly simplified under the assumption that the point of reference (body frame origin) is coincident with the center of mass. The position of the origin of the body fixed frame of reference expressed in terms of the Earth frame unit vectors is:

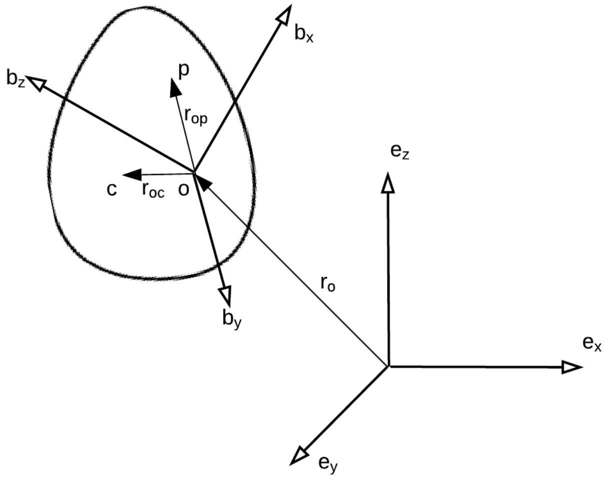


Figure 2.2 Position Radii for Rigid Body

$$r_o = E^{\mathrm{E}}r_o \Rightarrow \dot{r}_o = E^{\mathrm{E}}\dot{r}_o$$

The position of any arbitrary point p within the body relative to the origin of some frame of reference fixed within the body, r_{op}, can be expressed in terms of either the fixed body frame or the Earth frame:

$$r_{op} = E^{\mathrm{E}}r_{op} = B^{\mathrm{B}}r_{op} \Rightarrow {}^{\mathrm{E}}r_{op} = L_{\overset{\leftarrow}{\mathrm{BE}}}{}^{\mathrm{B}}r_{op}$$

The rate of change (velocity) of the body frame origin can be expressed in the body frame of reference by defining the quantity ${}^{\mathrm{B}}v_o$ as follows:

$$\dot{r}_o = E^{\mathrm{E}}\dot{r}_o = B^{\mathrm{B}}v_o$$

Since the Earth frame of reference is being treated as inertial here[5], the basis matrix E can be treated as an identity matrix, and the basis matrix B is hence equal to the change of coordinates matrix $L_{\overleftarrow{\mathrm{EB}}}$. This allows the notation to be simplified as E can be ommitted and B and its time derivative expression replaced by the equivalent forms of $L_{\overleftarrow{\mathrm{EB}}}$ recently derived.

Putting all this together and expressing the derivative of r_{op} using the results of Sections 2.1.2.1 and 2.1.4.2 , the velocity of any arbitrary poiont within the body can be expressed:

$$\dot{r}_p = L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}r_{cp}))$$

2.2.1 Momentum

The momentum of a point mass is simply the product of the mass with the inertial velocity. The momentum of a body can be found by treating the body as a continuum of mass with a density function defined for each point contained within the body volume, and integrating the product of the velocity at each point and the infinitesimal mass at that point over the set defined by the body:

$$P = \int\limits_{\forall r_{cp}} \dot{r}_p \rho(r_{cp}) dV = L_{\overleftarrow{\mathrm{EB}}} \int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}r_{cp}))\rho({}^{\mathrm{B}}r_{cp}) dV$$

Where P is the momentum, the expression $\forall {}^{\mathrm{B}}r_{cp}$ means all radii corresponding to the set of all points defined by the body, and $\rho({}^{\mathrm{B}}r_{cp})dV$ is the mass density as a function of position within the body mulitipled by the differential volume, i.e. the differential mass.

Most of the quantities in the integral for momentum do not depend on the variable ${}^{\mathrm{B}}r_{cp}$ and can be pulled out. Now by definition $\int_{\forall {}^{\mathrm{B}}r_{cp}} \rho({}^{\mathrm{B}}r_{cp})dV = m$. Also, by definition of center of mass, an integral of ${}^{\mathrm{B}}r_{cp}$ over the set of all ${}^{\mathrm{B}}r_{cp}$ must be 0. Therefore an integral of some quantity or quantities not dependent on ${}^{\mathrm{B}}r_{cp}$ and ${}^{\mathrm{B}}r_{cp}$ over this set must also be 0.[6] Hence $\int_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{cp})\rho({}^{\mathrm{B}}r_{cp})dV = 0$, leaving the expression of momentum with respect to the body frame of reference as:

---

[5] See Appendix A

[6] This fact will be used repeatedly throughout the derivation of the rigid body dynamics.

$$^{\mathrm{B}}P = m(^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc})$$

Assuming the reference point is at the center of mass would allow this to be simplified to:

$$^{\mathrm{B}}P = m^{\mathrm{B}}v_o = m^{\mathrm{B}}v_c$$

2.2.2 Force

Force is by definition the time derivative of momentum. Using the previous expression for momentum and the fact that time derivation can be brought inside a volume integral, force can be expressed as:

$$F = \dot{P} = L_{\overleftarrow{\mathrm{EB}}} \int\limits_{\forall^{\mathrm{B}}r_{cp}} (^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\dot{\Omega} \times (^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}r_{cp}) + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times (^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}))\rho(^{\mathrm{B}}r_{cp})dV$$

Again bringing terms not dependent on $^{\mathrm{B}}r_{cp}$ out of the integral and canceling terms dependent on a singular $^{\mathrm{B}}r_{cp}$ and evaluating what remains, the expression for force with respect to a fixed body frame of reference can be simplified to:

$$^{\mathrm{B}}F = m(^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\dot{\Omega} \times {}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}\Omega \times (^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}))$$

Defining the origin acceleration $^{\mathrm{B}}a_o = {}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o$, this can also be expressed as:

$$^{\mathrm{B}}F = m(^{\mathrm{B}}a_o + {}^{\mathrm{B}}\dot{\Omega} \times {}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}\Omega \times (^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}))$$

Which matches the form given in [22, 13]. In terms of the center of mass acceleration:

$$^{\mathrm{B}}a_c = {}^{\mathrm{B}}a_o + {}^{\mathrm{B}}\dot{\Omega} \times {}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}\Omega \times (^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc})$$

The force equation takes the familar $F = ma$ form:

$$^{\mathrm{B}}F = m^{\mathrm{B}}a_c$$

Rearranging the equation for ${}^{\mathrm{B}}a_c$, i.e. solving for ${}^{\mathrm{B}}a_o$ and noting that ${}^{\mathrm{B}}r_{oc} = -{}^{\mathrm{B}}r_{co}$ gives a potentially more intuitive/illustrative form of the relationship:

$${}^{\mathrm{B}}a_o = {}^{\mathrm{B}}a_c + {}^{\mathrm{B}}\dot{\Omega} \times {}^{\mathrm{B}}r_{co} + {}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{co})$$

Which can be read as: *The acceleration of the origin equals the acceleration of the center of mass plus additional tangental acceleration and centripetal acceleration due to the origin offset from center of mass.*

2.2.3 Angular Momentum

As given in [24] and elsewhere, the angular momentum for a body can be expressed similarly to the previous expression for linear momentum as:

$$L = \int\limits_{\forall r_{cp}} (r_p \times \dot{r}_p)\rho(r_{cp})dV$$

Writing these quantities in component form the expression becomes:

$$L = \int\limits_{\forall {}^{\mathrm{B}}r_{cp}} (r_o + L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}r_{cp})) \times L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc} + \Omega_B \times {}^{\mathrm{B}}r_{cp})\rho({}^{\mathrm{B}}r_{cp})dV$$

Multiplying through, separating and bringing terms not dependent on ${}^{\mathrm{B}}r_{cp}$ out of the integral leads to:

$$\begin{aligned}
L = r_o \times L_{\overleftarrow{\mathrm{EB}}} m({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) + L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}r_{oc} \times m{}^{\mathrm{B}}v_o) + \cdots \\
L_{\overleftarrow{\mathrm{EB}}} \int\limits_{\forall r_{cp}} ({}^{\mathrm{B}}r_{oc} \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) + {}^{\mathrm{B}}r_{oc} \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{cp}) + \cdots \\
{}^{\mathrm{B}}r_{cp} \times {}^{\mathrm{B}}v_o + {}^{\mathrm{B}}r_{cp} \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) + {}^{\mathrm{B}}r_{cp} \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{cp}))\rho({}^{\mathrm{B}}r_{cp})dV
\end{aligned}$$

Now as before, canceling terms with only one ${}^{\mathrm{B}}r_{cp}$, the fact that $L_{\overleftarrow{\mathrm{EB}}}m({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) = P$, and further using the identity that for some pair of vectors a and b, $a \times (b \times a) = (a^T a I - a a^T)b$:

$$\begin{aligned} L = r_o \times P + L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}r_{oc} \times m{}^{\mathrm{B}}v_o) + \cdots \\ L_{\overleftarrow{\mathrm{EB}}}\left({}^{\mathrm{B}}r_{oc}^T\,{}^{\mathrm{B}}r_{oc} I - {}^{\mathrm{B}}r_{oc}\,{}^{\mathrm{B}}r_{oc}^T + \int\limits_{\forall r_{cp}} ({}^{\mathrm{B}}r_{oc}^T\,{}^{\mathrm{B}}r_{oc} I - {}^{\mathrm{B}}r_{oc}\,{}^{\mathrm{B}}r_{oc}^T)\rho({}^{\mathrm{B}}r_{cp})dV \right){}^{\mathrm{B}}\Omega \end{aligned}$$

Defining the moment of inertia tensor with respect to center of mass:

$$J = \int\limits_{\forall r_{cp}} ({}^{\mathrm{B}}r_{cp}^T\,{}^{\mathrm{B}}r_{cp} I - {}^{\mathrm{B}}r_{cp}\,{}^{\mathrm{B}}r_{cp}^T)\rho({}^{\mathrm{B}}r_{cp})dV \tag{2.2.3.1}$$

And the moment of inertia tensor with respect to an arbitrary origin:

$$J_o = J + {}^{\mathrm{B}}r_{oc}^T\,{}^{\mathrm{B}}r_{oc} I - {}^{\mathrm{B}}r_{oc}\,{}^{\mathrm{B}}r_{oc}^T \tag{2.2.3.2}$$

Allows for the more compact notation:

$$L = r_o \times P + L_{\overleftarrow{\mathrm{EB}}}(J_o{}^{\mathrm{B}}\Omega + {}^{\mathrm{B}}r_{oc} \times m{}^{\mathrm{B}}v_o)$$

So the angular momentum with respect to the body frame of reference can be expressed:

$${}^{\mathrm{B}}L = J_o{}^{\mathrm{B}}\Omega + {}^{\mathrm{B}}r_{oc} \times m{}^{\mathrm{B}}v_o$$

This can be further simplified using the fact that $m{}^{\mathrm{B}}v_o = {}^{\mathrm{B}}P - {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}$ and simplifying allows an expression for angular momentum using the moment of inertia tensor with respect to center of mass:

$${}^{\mathrm{B}}L = J{}^{\mathrm{B}}\Omega + {}^{\mathrm{B}}r_{oc} \times {}^{\mathrm{B}}P$$

2.2.4 Torque

Torque, or moment of force, is defined as the time derivative of angular momentum:

$$Q = \dot{L} = \int\limits_{\forall r_{cp}} (r_p \times \ddot{r}_p)\rho(r_{cp})dV$$

Which is in expanded form:

$$Q = \int\limits_{\forall {}^{\mathrm{B}}r_{cp}} (r_o + L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}r_{oc} + {}^{\mathrm{B}}r_{cp})) \times L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o + \cdots$$

$${}^{\mathrm{B}}\dot{\Omega} \times {}^{\mathrm{B}}r_{oc} + \dot{\Omega}_B \times {}^{\mathrm{B}}r_{cp} + {}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) + {}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{cp}))\rho({}^{\mathrm{B}}r_{cp})dV$$

As before, multiplying through, separating, collecting and bringing out terms not dependent on position within the body, canceling those depending on a single ${}^{\mathrm{B}}r_{cp}$ term, and using the same simplification as with angular momentum and the moment of inertia tensor (in order to save a large amount of space, the details are not presented this time) gives:

$$Q = mr_o \times L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc})) + \cdots$$

$$m{}^{\mathrm{B}}r_{oc} \times ({}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o) + J_o{}^{\mathrm{B}}\dot{\Omega} + {}^{\mathrm{B}}\Omega \times (J_o{}^{\mathrm{B}}\Omega)$$

Which can be written:

$$Q = r_o \times ma_c + L_{\overleftarrow{\mathrm{EB}}}({}^{\mathrm{B}}r_{oc} \times m({}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o) + J_o{}^{\mathrm{B}}\dot{\Omega} + {}^{\mathrm{B}}\Omega \times (J_o{}^{\mathrm{B}}\Omega))$$

Giving an expression for torque with respect to the body frame of reference as:

$${}^{\mathrm{B}}Q = {}^{\mathrm{B}}r_{oc} \times m({}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o) + J_o{}^{\mathrm{B}}\dot{\Omega} + {}^{\mathrm{B}}\Omega \times (J_o{}^{\mathrm{B}}\Omega)$$

As with angular momentum, if desirable, this expression can be further simplified. Using the fact that $m({}^{\mathrm{B}}\dot{v}_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o) = {}^{\mathrm{B}}F - m{}^{\mathrm{B}}\dot{\Omega} \times {}^{\mathrm{B}}r_{oc} - m{}^{\mathrm{B}}\Omega \times ({}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc})$ and doing some cancelation and simplification leads to a form matching that in [13]:

$${}^{\mathrm{B}}Q = J{}^{\mathrm{B}}\dot{\Omega} + {}^{\mathrm{B}}\Omega \times J{}^{\mathrm{B}}\Omega + {}^{\mathrm{B}}r_{oc} \times {}^{\mathrm{B}}F$$

In any case, if the point of reference is the center of mass:

$${}^{\mathrm{B}}Q = J{}^{\mathrm{B}}\dot{\Omega} + {}^{\mathrm{B}}\Omega \times J{}^{\mathrm{B}}\Omega$$

2.2.5 Equation of Motion

By defining the matrix:

$$[{}^{\mathrm{B}}r_{oc}] = \begin{bmatrix} 0 & -{}^{\mathrm{B}}r_{oc.z} & {}^{\mathrm{B}}r_{oc.y} \\ {}^{\mathrm{B}}r_{oc.z} & 0 & -{}^{\mathrm{B}}r_{oc.x} \\ -{}^{\mathrm{B}}r_{oc.y} & {}^{\mathrm{B}}r_{oc.x} & 0 \end{bmatrix}$$

Meaning that, as with the same form of definition in Section 2.1.2.1, for some vector a

$${}^{\mathrm{B}}r_{oc} \times a = [{}^{\mathrm{B}}r_{oc}]a$$

And bringing the previously derived results for force and torque together, the resulting equation of motion can be expressed in matrix form as:

$$\begin{bmatrix} {}^{\mathrm{B}}F \\ {}^{\mathrm{B}}Q \end{bmatrix} = \begin{bmatrix} mI & -m[{}^{\mathrm{B}}r_{oc}] \\ [0] & J \end{bmatrix} \begin{bmatrix} {}^{\mathrm{B}}\dot{v}_o \\ {}^{\mathrm{B}}\dot{\Omega} \end{bmatrix} + \begin{bmatrix} {}^{\mathrm{B}}\Omega \times m({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) \\ {}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega + {}^{\mathrm{B}}r_{oc} \times {}^{\mathrm{B}}F \end{bmatrix} \tag{2.2.5.1}$$

If the origin of the fixed body frame of reference is taken to coincide with the center of mass, the equation of motion simplifies to:

$$\begin{bmatrix} {}^{\mathrm{B}}F \\ {}^{\mathrm{B}}Q \end{bmatrix} = \begin{bmatrix} mI & [0] \\ [0] & J \end{bmatrix} \begin{bmatrix} {}^{\mathrm{B}}\dot{v}_o \\ {}^{\mathrm{B}}\dot{\Omega} \end{bmatrix} + \begin{bmatrix} {}^{\mathrm{B}}\Omega \times m\,{}^{\mathrm{B}}v_o \\ {}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega \end{bmatrix} \tag{2.2.5.2}$$

2.2.6 Moment of Inertia Tensor

The moment of inertia tensor as previously defined in Equation 2.2.3.1 is:

$$J = \int\limits_{\forall r_{cp}} ({}^{\mathrm{B}}r_{oc}^{T}\,{}^{\mathrm{B}}r_{oc} I - {}^{\mathrm{B}}r_{oc}\,{}^{\mathrm{B}}r_{oc}^{T})\rho({}^{\mathrm{B}}r_{cp})dV$$

Expanding this gives:

$$J = \begin{bmatrix} \int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.y}{}^{2} + {}^{\mathrm{B}}r_{cp.z}{}^{2})\rho({}^{\mathrm{B}}r_{cp})dV & -\int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.x}\,{}^{\mathrm{B}}r_{cp.y})\rho({}^{\mathrm{B}}r_{cp})dV & -\int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.x}\,{}^{\mathrm{B}}r_{cp.z})\rho({}^{\mathrm{B}}r_{cp})dV \\ -\int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.x} r_{cp.y_y})\rho({}^{\mathrm{B}}r_{cp})dV & \int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.x}{}^{2} + r_{cp.z_z}^{2})\rho({}^{\mathrm{B}}r_{cp})dV & -\int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.y} r_{cp.z_z})\rho({}^{\mathrm{B}}r_{cp})dV \\ -\int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.x}\,{}^{\mathrm{B}}r_{cp.z})\rho({}^{\mathrm{B}}r_{cp})dV & -\int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.y}\,{}^{\mathrm{B}}r_{cp.z})\rho({}^{\mathrm{B}}r_{cp})dV & \int\limits_{\forall {}^{\mathrm{B}}r_{cp}} ({}^{\mathrm{B}}r_{cp.x}{}^{2} + {}^{\mathrm{B}}r_{cp.y}{}^{2})\rho({}^{\mathrm{B}}r_{cp})dV \end{bmatrix}$$

Where:

$$^{\mathrm{B}}r_{cp} = \begin{bmatrix} ^{\mathrm{B}}r_{cp.x} \\ ^{\mathrm{B}}r_{cp.y} \\ ^{\mathrm{B}}r_{cp.z} \end{bmatrix}$$

Making the following defintions:

$$J_{xx} = \int_{\forall ^{\mathrm{B}}r_{cp}} ({^{\mathrm{B}}r_{cp.y}}^2 + {^{\mathrm{B}}r_{cp.z}}^2)\rho(^{\mathrm{B}}r_{cp})dV$$

$$J_{yy} = \int_{\forall ^{\mathrm{B}}r_{cp}} ({^{\mathrm{B}}r_{cp.x}}^2 + {^{\mathrm{B}}r_{cp.z}}^2)\rho(^{\mathrm{B}}r_{cp})dV$$

$$J_{zz} = \int_{\forall ^{\mathrm{B}}r_{cp}} ({^{\mathrm{B}}r_{cp.x}}^2 + {^{\mathrm{B}}r_{cp.y}}^2)\rho(^{\mathrm{B}}r_{cp})dV$$

$$J_{xy} = \int_{\forall ^{\mathrm{B}}r_{cp}} (^{\mathrm{B}}r_{cp.x}\, ^{\mathrm{B}}r_{cp.y})\rho(^{\mathrm{B}}r_{cp})dV$$

$$J_{xz} = \int_{\forall ^{\mathrm{B}}r_{cp}} (^{\mathrm{B}}r_{cp.x}\, ^{\mathrm{B}}r_{cp.z})\rho(^{\mathrm{B}}r_{cp})dV$$

$$J_{yz} = \int_{\forall ^{\mathrm{B}}r_{cp}} (^{\mathrm{B}}r_{cp.y}\, ^{\mathrm{B}}r_{cp.z})\rho(^{\mathrm{B}}r_{cp})dV$$

The moment of inertia tensor with respect to the center of mass may be expressed more compactly:

$$J = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \tag{2.2.6.1}$$

As discussed in [23, 24], for any origin fixed within a rigid body there exists an alignment of the frame of reference axes such that the moment of inertia tensor is diagonal. This alignment is when the axes of the body frame of reference are coincident with the body principle axes of inertia. For the purposes of this model, it will be assumed that the nominal body frame of reference axes are aligned with the principle axes. Under this assumption:

$$J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{2.2.6.2}$$

CHAPTER 3. System Architecture

3.1 Overall System

The overall system at a high level consists of the quadrotor vehicle (plant), sensor system in this case composed of infrared cameras and image processing, control system implemented on a PC, and communication system consisting of an FPGA, radio transmitter, and receiver. The communication system compoents (other than the FPGA) are along with the quadrotor commercially avialable off-the-shelf consumer products. The infrared camera system is a commercially available motion tracking system with acconmpanying processing and streaming software. See Table 5.1. Software accomplishing the input/output and control processsing was created specifically for this applicaton. The processing and control agorithms were designed as part of this work, while the bulk of the software background was implemented by members of senior design teams working cooperatively on projects overlapping this work. See Table 5.2.

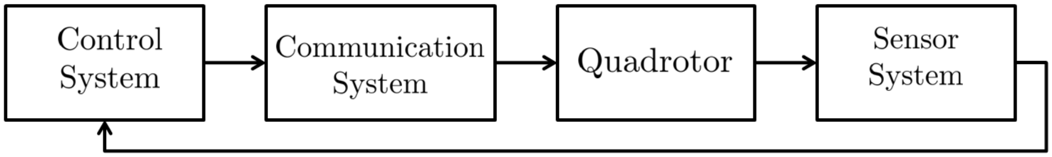


Figure 3.1 Overall System

Each of these systems is described in greater detail in the sections that follow.

3.2 Quadrotor

The specific vehicle used in this work was a GAUI 330X-S model quadrotor helicopter[25] available as a consumer product[1]. The quadrotor system included a lightweight frame, four brushless DC electric motors (GUEC GM-410 brushless DC motors) with 8-inch rotors, four brushless DC motor controllers (GUEC GE-010 10A BLDC ESCs), and an on board input signal mixer and gyroscope unit (GU-344). Each of these components is modeled and parameterized in later sections of this work. The specific convention used in all of what follows for rotor identification and positioning is described below.

3.2.1 Body Frame Axes and Rotor Numbering

The four rotors of the vehicle are numbered 1 through 4. The body frame of reference axes described in Section 2.1.2 are defined such that when viewed against the positive body frame z-axis (b_z), or technically from below the quadrotor, with b_y *pointing up* and b_x *pointing right* in view, rotor 1 is in the 1^{st} (upper-right or +x,+y) quadrant, and rotor 2, rotor 3, and rotor 4 are in the 2^{nd} (-x,+y), 3^{rd} (-x,-y), and 4^{th} (+x,-y) quadrants respectively. Viewed with b_z, rotors 1 and 3 have a negative angular velocity vector, while 2 and 4 have positive angular velocity vectors following the right-hand rule. A *top down*, i.e. b_z *into the page* view of this setup is given in Figure 3.2 with the rotation direction of each rotor indicated.

This choice of axes matches one of two [25] *flying mode* settings available with the vehicle on-board electronics, one of which corredponds to a set of axes as described here, and another which corresponds to those axes rotated by $\frac{\pi}{4}$rad in the xy plane such that b_x and b_y run through the rotors. This setting would primarily affect all hub radii as used in Section 4.2 and the command mixing matrix of Equation 4.5.3.1.

---

[1] At the time of this writing, production of this specific line may be discontinued and hence be hard to find. Similar products from the same manufacturer are likely avialable, with many of the sub-components potentially identical.

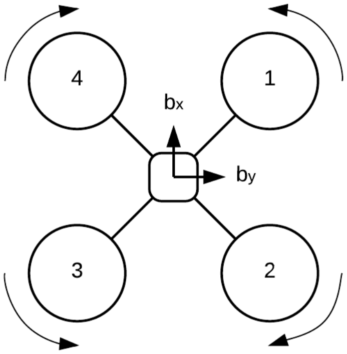


Figure 3.2 Body Axes and Rotor Numbering

3.2.2 Physical Architecture

The physical architecture of the quadrotor used here is shown in Figure 3.3. The receiver shown here as part of the quadrotor system is techincally considered part of the communications system in this model, but is physically present on the vehicle and is shown nonetheless.

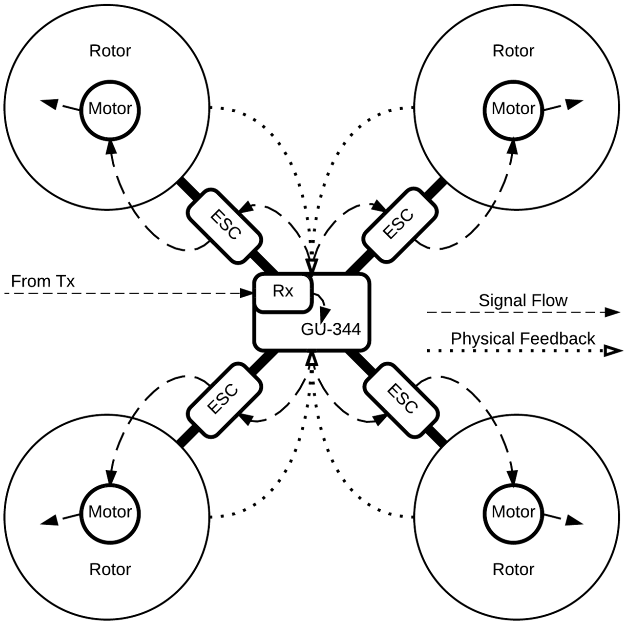

Rotor
Motor
Rotor
Motor
ESC
ESC
From Tx
Rx
GU-344
Signal Flow
Physical Feedback
ESC
ESC
Motor
Rotor
Motor
Rotor

Figure 3.3 Quadrotor Physical Architecture

3.2.3 Model Architecture

The quadrotor model architecture is shown in Figure 3.4. The quadrotor as a whole is broken down successively into smaller and smaller subsystems for modeling purposes. The main three subsystems the quadrotor vehicle is broken down into are the GU-344, Wrench System, and Rigid Body Dynamics System.

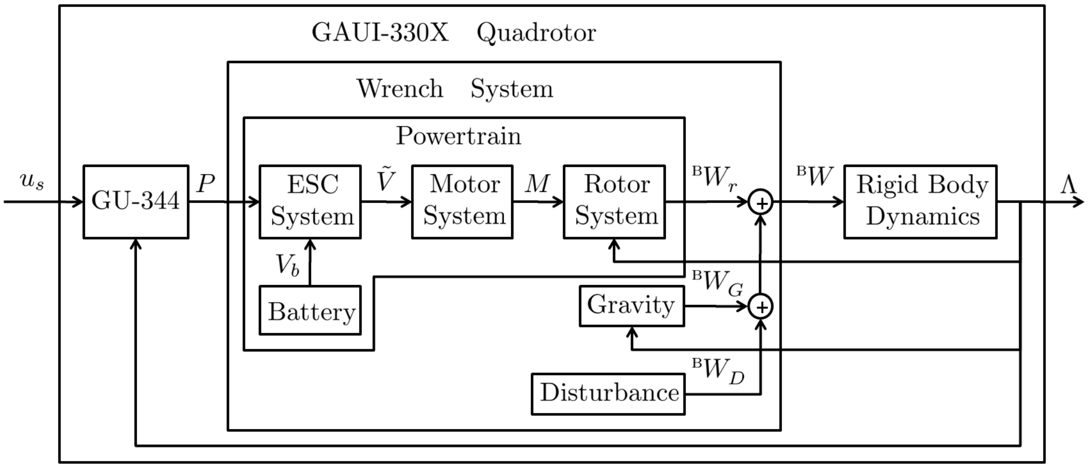

GAUI-330X Quadrotor
Wrench System
Powertrain
u_s
GU-344
P
ESC System
$\tilde{V}$
Motor System
M
Rotor System
${}^{B}W_r$
${}^{B}W$
Rigid Body Dynamics
Λ
V_b
Battery
${}^{B}W_G$
Gravity
${}^{B}W_D$
Disturbance

Figure 3.4 Quadrotor Model Architecture

3.2.3.1 GU-344

The GU-344 subsystem is broken down further into components representing the gyroscope angular velocity feedback, angular velocity tracking, and control mixing functionality of the unit, which are all described in Section 4.5 symbolically and Section 5.6 numerically. The input to the GU-344 is the vector u_s of inputs as obtained from the receiver consisting of the throttle, aileron, elevator, and rudder values. The output P is a vector of four duty cycle percentage values sent to the four motor speed controls.

3.2.3.2 Wrench System

The Wrench System is broken down into the Powertrain subsystem, Gravity, and Disturbance. Wrench here describes a vector of combined force and torque, i.e. for some force F and torque

Q the corresponding wrench is $[F\ Q]^T$. The Wrench System output ${}^{\mathrm{B}}W$ is the sum total of all forces and torques that act on the quadrotor. As modeled these are the Rotor System Wrench, $({}^{\mathrm{B}}W_r)$, Gravity Wrench, $({}^{\mathrm{B}}W_G)$, and Disturbance Wrench, $({}^{\mathrm{B}}W_D)$. The Disturbance Wrench is simple an input to the model to allow for any type of force and torque disturbance. The Gravity Wrench is described in Section 4.3.

3.2.3.3 Powertrain

The Powertrain is further broken down into the Battery, ESC System, Motor System, and Rotor System. The Battery output in this model is the voltage V_b, and the modeling for this component is described in Section 4.2.7 symbolically and Section 5.4 numerically. The ESC and Motor Systems are described symbolically in Sections 4.2.4, 4.2.5, and 4.2.6, and numerically in Sections 5.3 and 5.5.5.

The output of the ESC System and input to the Motor System, $\tilde{V}$, is a vector of the effective voltag for any given energized phase of each of the brushless DC motors. The output of the Motor system M is a vector representing the four angular speeds (represented by the vector ω) and accelerations (represented by the vector α).

3.2.3.4 Rotor System

The Rotor System is where the inputs are actuated. The rotor system is modeled as producing a number of forces and torques. The rotor forces modeled here are the thrust and in-plane drag force or H-Force. These are modeled symoblically in Sections 4.2.1.1 and 4.2.1.2 respectively, and numerically in Sections 5.5.1, 5.5.2, and 5.5.3. The rotor torques modeled here are the in-plane drag and induced torque, change in angular momentum torque, and force lever arm torque. These are modeled symbolically in Sections 4.2.2.1, 4.2.2.2, and 4.2.2.3 respectively, and numerically in Sections 5.5.4 and 5.5.5.

3.3 Sensor System

The sensor system used here consists of 12 infrared cameras (OptiTrack VR:100), two synchronized camera hubs (OptiHub-2) an the Optitrack Tracking Tools processing software. The OptiTrack system provided the 6DOF position and orientation of the quadrotor in real time (see Section 5.7 for frame rate, latency, and accuracy details) using VRPN (see Table 5.2) streaming to the control PC. The orientation data was streamed in terms of quaternions and converted to the correct Euler angles using built in VRPN libraries.

3.4 Control System

The control system used for this work is entirely implemented in software, predominantly written in C++, and run on a PC with Ubuntu 7.10 operating system. The control software not only implements the control and any estimation algorithms but also logs all desired data for analysis purposes. It is assume capable of operating on every sample received from the Camera System with negligable delay and no numerical issues[2]. Its basic architecture is depicted in Figure 3.5.

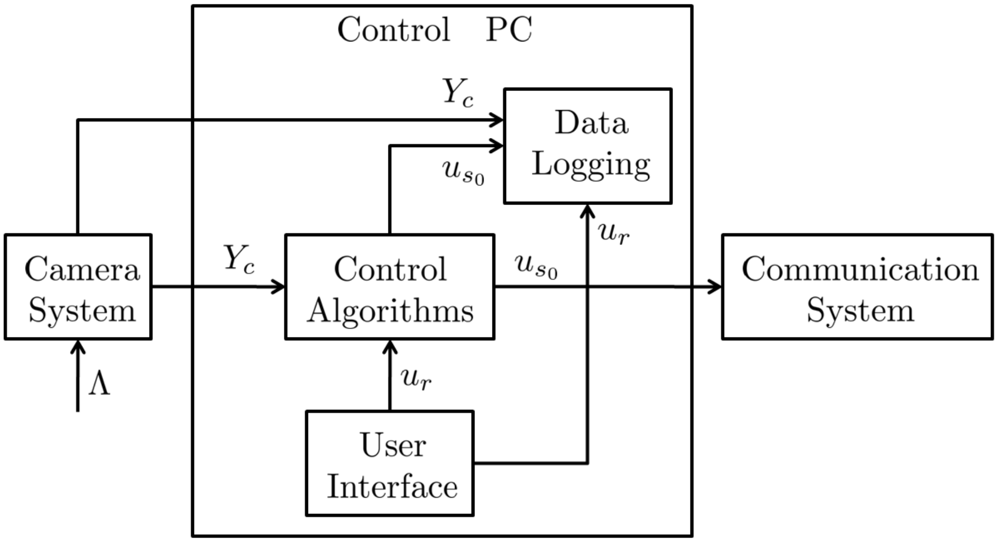


Figure 3.5 Control System Setup

---

[2] E.g. fixed point issues possibly encountered with microprocessors. The PC utilized contained a 64-bit processor.

3.5 Communication System

The communication system as defined in this model consists of an FPGA, radio transmitter, and receiver. The FPGA serves as the interface between the control PC and the radio transmitter, and the receiver as the interface to the quadrotor vehicle GU-344 unit. It is treated as a simple series connection of these components.

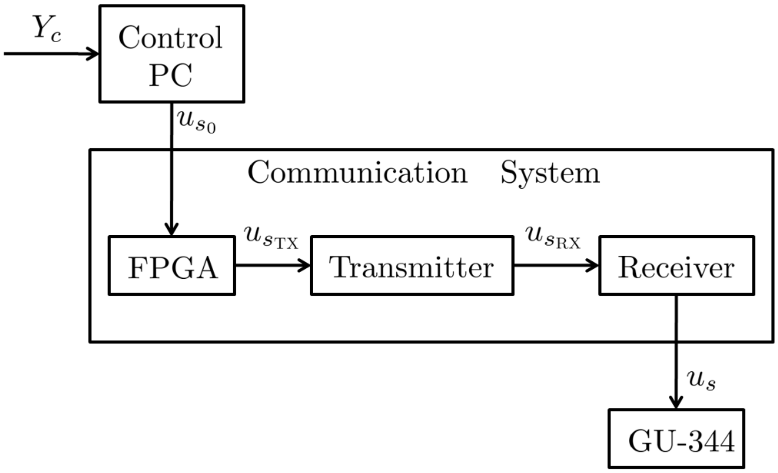

Y_c
Control PC
u_{s_0}
Communication System
FPGA
$u_{s_{\mathrm{TX}}}$
Transmitter
$u_{s_{\mathrm{RX}}}$
Receiver
u_s
GU-344

Figure 3.6 Communication System

CHAPTER 4. Symbolic Modeling

4.1 Rigid Body Dynamics

Equation 2.2.5.1, repeated here:

$$\begin{bmatrix} {}^{\mathrm{B}}F \\ {}^{\mathrm{B}}Q \end{bmatrix} = \begin{bmatrix} mI & -m[{}^{\mathrm{B}}r_{oc}] \\ [0] & J \end{bmatrix} \begin{bmatrix} {}^{\mathrm{B}}\dot{v}_o \\ {}^{\mathrm{B}}\dot{\Omega} \end{bmatrix} + \begin{bmatrix} {}^{\mathrm{B}}\Omega \times m({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) \\ {}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega + {}^{\mathrm{B}}r_{oc} \times {}^{\mathrm{B}}F \end{bmatrix}$$

Was derived in a form that makes it easy to solve for the accelerations. Since the matrix multiplying the accelerations vector for any possible realistic physical scenario must be full rank, it is guaranteed to have an inverse. So the accelerations are given:

$$\begin{bmatrix} {}^{\mathrm{B}}\dot{v}_o \\ {}^{\mathrm{B}}\dot{\Omega} \end{bmatrix} = \begin{bmatrix} mI & -m[{}^{\mathrm{B}}r_{oc}] \\ [0] & J \end{bmatrix}^{-1} \begin{bmatrix} {}^{\mathrm{B}}F - {}^{\mathrm{B}}\Omega \times m({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}r_{oc}) \\ {}^{\mathrm{B}}Q - {}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega - {}^{\mathrm{B}}r_{oc} \times {}^{\mathrm{B}}F \end{bmatrix} \tag{4.1.0.1}$$

4.1.1 Rigid Body Dyanmics State Equation

The rigid body dynamics of the quadrotor are taken as a nonlinear subsystem. The state vector for the system, Λ, is defined to consist of the body frame of reference velocities, the inertial frame of reference position, and orientation, defined as:

$$\Lambda = \begin{bmatrix} {}^{\mathrm{B}}v_o \\ {}^{\mathrm{B}}\Omega \\ {}^{\mathrm{E}}r_o \\ \Theta \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}$$

It is assumed here that the origin of the body frame of reference is coincident with the center of mass and that its axes are aligned with the principle axes of inertia. In this situation, ${}^{\mathrm{B}}r_{oc} = 0$ and the moment of inertia tensor J is diagonal as defined in Equation 2.2.6.2, which allows Equation 4.1.0.1 to be simplified to:

$$\begin{bmatrix} {}^{\mathrm{B}}\dot{v}_o \\ {}^{\mathrm{B}}\dot{\Omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}I & [0] \\ [0] & J^{-1} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{B}}F - {}^{\mathrm{B}}\Omega \times m\,{}^{\mathrm{B}}v_o \\ {}^{\mathrm{B}}Q - {}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\,{}^{\mathrm{B}}F - {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o \\ J^{-1}\,{}^{\mathrm{B}}Q - J^{-1}\,{}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega \end{bmatrix}$$

Now the remaining quantites in the state vector are ${}^{\mathrm{E}}r_o$ and Θ. The portion of the state equation for each may be defined in terms of ${}^{\mathrm{B}}v_o$ and ${}^{\mathrm{B}}\Omega$ respectively as:

$${}^{\mathrm{E}}\dot{r}_o = L_{\overleftarrow{\mathrm{EB}}}\,{}^{\mathrm{B}}v_o$$

$$\dot{\Theta} = A_{\overleftarrow{\mathrm{EB}}}\,{}^{\mathrm{B}}\Omega$$

So defining $f_{\mathrm{RB}}(\Lambda, {}^{\mathrm{B}}W) = \dot{\Lambda}$:

$$f_{\mathrm{RB}}(\Lambda, {}^{\mathrm{B}}W) = \begin{bmatrix} {}^{\mathrm{B}}\dot{v}_o \\ {}^{\mathrm{B}}\dot{\Omega} \\ {}^{\mathrm{E}}\dot{r}_o \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\,{}^{\mathrm{B}}F - {}^{\mathrm{B}}\Omega \times {}^{\mathrm{B}}v_o \\ J^{-1}\,{}^{\mathrm{B}}Q - J^{-1}\,{}^{\mathrm{B}}\Omega \times J\,{}^{\mathrm{B}}\Omega \\ L_{\overleftarrow{\mathrm{EB}}}\,{}^{\mathrm{B}}v_o \\ A_{\overleftarrow{\mathrm{EB}}}\,{}^{\mathrm{B}}\Omega \end{bmatrix} \tag{4.1.1.1}$$

4.2 Powertrain

While conventional helicopters require two cyclic pitch controls to perform longitudinal and lateral movements, quadrotors due to their configuration do not and have rotors without any need of cycylc pitch inputs. Also, conventional helicopters typically use collective pitch adjustment to increase or decrease thrust on either their main or tail rotor, while smaller scale electrically powered aircraft like the quadrotor being modeled here are able to use rotation speed adjustment.[1]

4.2.1 Rotor Force

The forces that apply to the rotors in this model are the thrust T, and in-plane force H. Things like rotor side force developed in [26] are applicable only to rotor systems with cyclic inputs, which the rotors modelled here do not have. It is typical to give both rotor force and torque quantities using coefficient forms. In this case a thrust coefficient and H-force coefficient, defined as:

$$C_T = \frac{|T|}{\rho A(\omega r)^2}$$

$$C_H = \frac{|H|}{\rho A(\omega r)^2}$$

Where ρ is the density of air, A is the rotor disc area, r is the rotor radius, and ω is the rotor angular speed. Nearly identical, and very thorough expressions for the thrust coefficient are given in [1, p. 111] and [26, p. 176], and slightly less detailed in [27, p. 98] and [28, p. 82]. The main difference between the two pairs being the inclusion of terms for rotor blade twist and lateral cyclic pitch. The rotors for this model are taken to have negligable twist effects and of course no cyclics.

---

[1] The ability to control collective pitch, especially to the extent of reversing the direction of thrust, can be used in the performance of aggressive acrobatic maneuvers.

4.2.1.1 Thrust

An iterative scheme for solving a set of equations to define the necessary quantities for calculation of rotor thrust is given in [1] and [29, 30]. Using those from [29, 30] since they are in a simplified form appropriate for this model:

$$C_T = \frac{a_0\sigma}{2}\left(\theta_0\left(\frac{1}{3}+\frac{\mu^2}{2}\right)+\frac{\mu_z-\lambda_0}{2}\right)$$

$$\lambda_0 = \frac{C_T}{2\sqrt{\mu^2+(\lambda_0-\mu_z)^2}}$$

Where a_0 is a constant lift curve slope, σ is the solidity ratio, θ_0 is the collective pitch (in this case a constant), μ is the advance ratio, μ_z is the normal airflow component, and λ_0 is the inflow ratio (the ratio of the induced velocity to the rotor tip speed). The advance ratio and normal airflow componets are defined:

$$\mu = \frac{\sqrt{(u-u_{wind})^2+(v-v_{wind})^2}}{\omega r} \tag{4.2.1.1}$$

$$\mu_z = \frac{w-w_{wind}}{\omega r} \tag{4.2.1.2}$$

Where u,v, and w are the body frame of reference linear velocity components as defined in Section 2.1.2. The iterative process begins by defining a function with a zero at the sought parameter values:

$$g_0 = \lambda_0 - \frac{C_T}{2(\mu^2+(\lambda_0-\mu_z)^2)}$$

Using Newton's method, the estimate following the j^{th} estimate is obtained as:

$$\lambda_{0_{j+1}} = \lambda_{0_j} + f_j h_j(\lambda_{0_j})$$

Where:

$$h_j = -\left(\frac{g_0}{dg_0/d\lambda_0}\right)_{\lambda_0=\lambda_{0_j}}$$

$$\Rightarrow h_j = \frac{\left(2\lambda_{0_j}\sqrt{\mu^2+(\lambda_0-\mu_z)^2}-C_T\right)\left(\mu^2+(\lambda_0-\mu_z)^2\right)}{2\left(\mu^2+(\lambda_0-\mu_z)^2\right)^{3/2}+\frac{a_0\sigma}{4}\left(\mu^2+(\lambda_0-\mu_z)^2\right)-C_T\left(\mu_z-\lambda_{0_j}\right)}$$

Where f_j is a damping factor to stabilize the calculation especially near hover, with both [1, 29] obtaining best results at a value of 0.6.

While the iterative scheme can yeild accurate results over a large range of flight conditions, for the purposes of this model, i.e. relatively low velocity non-acrobatic flight, some simplifying assumptions can be made. Typical rotor speeds for this model will exceed $500\frac{rad}{s}$, so assuming the translational velocities of the craft are kept low enough, the approximation can be made:

$$\mu_z \approx 0 \quad \mu \approx 0$$

The set of equations dealing with rotor thrust simplify:

$$\lambda_0 = \frac{C_T}{2\sqrt{\lambda_0^2}} \implies C_T = 2\lambda_0^2$$

$$C_T = \frac{1}{6}a_0\sigma\theta_0 - \frac{1}{4}a_0\sigma\lambda_0$$

With this simplification the need for an iterative solution is gone and a close form solution for λ_0 can be found. The above equations can be combined to form a quadratic equation in λ_0:

$$2\lambda_0^2 + \frac{1}{4}a_0\sigma\lambda_0 - \frac{1}{6}a_0\sigma\theta_0 = 0$$

Which gives the solution:

$$\lambda_0 = \frac{-a_0\sigma}{16} + \frac{1}{4}\sqrt{\left(\frac{1}{4}a_0\sigma\right)^2 + \frac{4}{3}a_0\sigma\theta_0}$$

Which is for this model, all constants. Also since:

$$C_T = 2\lambda_0^2 = \frac{|T|}{\rho(r\omega)^2\pi r^2}$$

$$|T| = 2\lambda_0^2\rho(r\omega)^2\pi r^2 = [2\lambda_0^2\rho\pi r^4]\omega^2$$

Defining the constant $K_T = 2\lambda_0^2\rho\pi r^4$ the thrust expression becomes:

$$|T| = K_T\omega^2$$

This is the form of the thrust equation found in [8, 2, 13, 3] among many, many others[2]. In order to take some account of the changes in thrust due to velocity while keeping a closed form expression, the adjustment factor[3] $\delta_T = [\delta_{T_x}\ \delta_{T_y}\ \delta_{T_z}]$ is defined such that:

---

[2] Just about every source with a quadrotor model containing an expression for thrust as a function of rotor speed uses this form.

[3] See Section 5.5.2 for detailes on this adjustment factor and explanation for why only the thrust equation has such a parameter.

$$|T| = K_T\omega^2 + \delta_T v_h \omega \tag{4.2.1.3}$$

Where v_h is the velocity of the rotor hub in a local reference frame. For this quadrotor, v_h can be represented using the velocity of the body frame of reference origin and the radius of the hub from that origin r_h as:

$$v_h = {}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_h \tag{4.2.1.4}$$

In order to represent the thrust of each rotor in vector form, the quantity Γ_{T_i} is defined as a unit vector giving the direction of thrust i in the body frame of reference. With each rotor assumed to be attached so that thrust is along $-b_z$ or exactly "straight up" in the body frame of reference:

$$T_i = -(K_T\omega_i^2 + \delta_T v_h \omega_i) b_z \quad \forall i$$

$$\Rightarrow \Gamma_{T_i} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \forall i$$

That is:

$$T_i = B(K_T\omega_i^2 + \delta_T v_{h_i} \omega_i)\Gamma_{T_i}$$

Or as body frame quantities:

$${}^{\mathrm{B}}T_i = (K_T\omega_i^2 + \delta_T v_{h_i} \omega_i)\Gamma_{T_i} \tag{4.2.1.5}$$

4.2.1.2 H-force

The H-force, developed in e.g. [26, p. 177], is the in-plane force due to the drag on the rotors when the vehicle has some in-plane velocity relative to the air. The H-force coefficient similar to the thrust version can be expressed:

$$C_H = \left(\frac{\sigma a_0 \theta_0 \lambda_0 + \sigma C_{d_0}}{4}\right)\mu$$

Where C_{d_0} is defined in Section 4.2.2.1 and all other symbols are as previously described in Section 4.2.1.1. Here the expression needs to be adjusted for each of the four rotors. Adjusting advance ratio μ as defined in Equation 4.2.1.1 and using the expression for hub velocity given in Equation 4.2.1.4, the equivalent advance ratio for each rotor is the same form with the hub velocity quantities subsituted[4]:

$$\mu_{h_i} = \frac{\sqrt{(v_{h_{ix}})^2 + (v_{h_{iy}})^2}}{\omega_i r_i}$$

$$C_{H_i} = \left(\frac{\sigma a_0 \theta_0 \lambda_0 + \sigma C_{d_0}}{4}\right) \mu_{h_i}$$

$$|H_i| = C_{H_i}(\rho A_i (\omega_i r_i)^2) = \left(\frac{\rho A_i r_i^2 \sigma a_0 \theta_0 \lambda_0 + \rho A_i r_i^2 \sigma C_{d_0}}{4}\right) \mu_{h_i} \omega_i^2$$

Using $A_i = \pi r_i^2$ and $\mu_{h_i} = \sqrt{(v_{h_{ix}})^2 + (v_{h_{iy}})^2}/\omega_i r_i$:

$$|H_i| = \left(\frac{\rho \pi r_i^3 \sigma (a_0 \theta_0 \lambda_0 + C_{d_0})}{4}\right) \sqrt{(v_{h_{ix}})^2 + (v_{h_{iy}})^2} \omega_i$$

The direction of this force is always opposite the direction of in plane motion, i.e. it is opposite the vector $[v_{h_{ix}} \; v_{h_{ix}} \; 0]^T$. Defining the unit version of that vector:

$$\frac{1}{\sqrt{(v_{h_{ix}})^2 + (v_{h_{iy}})^2}} \begin{bmatrix} v_{h_{ix}} \\ v_{h_{ix}} \\ 0 \end{bmatrix}$$

Using this and defining the constant $K_H = \left(\frac{\rho \pi r_i^3 \sigma (a_0 \theta_0 \lambda_0 + C_{d_0})}{4}\right)$, the vector for each force can be expressed as a body frame quantity:

$$^{\mathrm{B}}H_i = -K_H \omega_i \begin{bmatrix} v_{h_{ix}} \\ v_{h_{iy}} \\ 0 \end{bmatrix}$$

Defining:

$$\Gamma_H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The H-force of each rotor can be expressed as:

$$H_i = -B(K_H \omega_i \Gamma_H v_{h_i})$$

$$^{\mathrm{B}}H_i = -K_H \omega_i \Gamma_H v_{h_i} \tag{4.2.1.6}$$

---

[4] Wind velocity is not repeated but in circumstances where wind would play a role it would need to be reintroduced

4.2.1.3 Total Rotor System Force

The resulting total force in the body frame of reference due to the rotors can be expressed as:

$$ {}^{\mathrm{B}}F_r = \sum_{i=1}^{4} ({}^{\mathrm{B}}T_i + {}^{\mathrm{B}}H_i) \tag{4.2.1.7} $$

Where ${}^{\mathrm{B}}T_i$ and ${}^{\mathrm{B}}H_i$ are defined in Equation 4.2.1.5 and Equation 4.2.1.6 respectively. Expanding this expression gives:

$$ {}^{\mathrm{B}}F_r = \sum_{i=1}^{4} ((K_T\omega_i^2 + \delta_T({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_{h_i})\omega_i)\Gamma_{T_i} - K_H\omega_i\Gamma_H({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_{h_i})) $$

4.2.2 Rotor Torque

The torques created by the rotors in this model are the in-plane drag and induced torue Q_d, change in angular momentum torque Q_L, and thrust lever arm torque Q_T. In this case only the in-plane torque has a coefficient form, defined as:

$$ C_Q = \frac{|Q_d|}{\rho A(\omega r)^2 r} $$

Where as before ρ is the density of air, A is the rotor disc area, r is the rotor radius, and ω is the rotor angular speed. Again, very similar, and very thorough expressions for this torque coefficient can be found in in [1, p.115], [29, 30], and [26, p. 179-184], and again slightly shorter in [27, p. 102] and [28, p. 84].

4.2.2.1 In-plane Drag and Induced Torque

$$ C_Q = C_T\left(\lambda_0 - \mu_z\right) + \frac{C_{d_0}\sigma}{8}\left(1 + \frac{7}{3}\mu^2\right) $$

Again, following the same assumptions as in Section 4.2.1.1:

$$ C_Q = C_T\lambda_0 + \frac{C_{d_0}\sigma}{8} $$

$$ C_Q = 2\lambda_0^3 + \frac{C_{d_0}\sigma}{8} \Leftarrow C_T = 2\lambda^2 $$

Grouping terms:

$$|Q_{d_i}| = (2\lambda_0^3 + \frac{C_{d_0}\sigma}{8})\rho\pi r^5\omega_i^2$$

Defining the constant $K_d = (2\lambda_0^3 + \frac{C_{d_0}\sigma}{8})\rho\pi r^5$, following the vectors used here:

$$|Q_{d_i}| = \pm K_d\omega_i^2 \tag{4.2.2.1}$$

The quantity Γ_{Ω_i} is defined as a unit vector by right hand rule for each ω_i in the body frame of reference. So depending on the direction of ω_i:

$$\Gamma_{\Omega_i} = \begin{bmatrix} 0 \\ 0 \\ \pm 1 \end{bmatrix} \tag{4.2.2.2}$$

So the drag torque of each rotor can be expressed using B, and $Q_{d_{B_i}}$ defined:

$$Q_{d_i} = B(-K_d\omega_i^2\Gamma_{\Omega_i})$$

$$^{\mathrm{B}}Q_{d_i} = -K_d\omega_i^2\Gamma_{\Omega_i} \tag{4.2.2.3}$$

4.2.2.2 Change in Angular Momentum Torque

The angular velocity vector for each rotor can be defined generally using the scalar ω_i, the unit vector Γ_i which gives the direction of the angular velocity vector in the body frame following the right-hand rule, and the body frame basis B.[5]

The angular momentum of each rotor can be defined:[6]

$$L_i = \tilde{J}_r B(\omega_i\Gamma_{\Omega_i})$$

The time derivative of this angular momentum vector gives the torque applied to each rotor. Hence the torque applied from rotor to body is the negative of the time derivative:

$$Q_{L_i} = -\tilde{J}_r\dot{B}(\omega_i\Gamma_{\Omega_i}) - \tilde{J}_r B(\dot{\omega}_i\Gamma_{\Omega_i})$$

---

[5] To fully describe the detailed dynamics of each rotor (especially if large enough to flex and/or take account of flapping), each would require its own reference frame and a much more complicated development. However, assuming that the plane of rotation and center of rotaiton of each rotor are both absolutely fixed with respect to B, a an expression using just B should be sufficient.

[6] Also, the rotor speed is assumed significantly large compared to any chassis angular velocity to ignore chassis angular velocity in the angular momentum calculations.

$$Q_{L_i} = -\tilde{J}_r B({}^{\mathrm{B}}\Omega \times \omega_i \Gamma_{\Omega_i}) - \tilde{J}_r B \dot{\omega}_i \Gamma_{\Omega_i}$$

Which can be written as body frame components:

$${}^{\mathrm{B}}Q_{L_i} = -\tilde{J}_r({}^{\mathrm{B}}\Omega \times \omega_i \Gamma_{\Omega_i} + \dot{\omega}_i \Gamma_{\Omega_i}) \tag{4.2.2.4}$$

Where the component due to in plane angular acceleration is $\tilde{J}_r(\dot{\omega}_i \Gamma_{\Omega_i})$ and the gyroscopic term is: $\tilde{J}_r({}^{\mathrm{B}}\Omega \times \omega_i \Gamma_{\Omega_i})$.

4.2.2.3 Force Lever Arm Torque

Since each rotor force acts on the body of the quadrotor at a distance from the center, each causes a torque. The contribution of each can be expressed as the cross product of the radius from the body frame origin with the force vector, i.e. the torque due to rotor force i is:

$$Q_{F_i} = (Br_{h_i}) \times F_{r_i}$$

$$Q_{F_i} = B(r_{h_i} \times ((K_T \omega_i^2 + \delta_T({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_{h_i})\omega_i)\Gamma_{T_i} - K_H \omega_i \Gamma_H({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_{h_i})))$$

$${}^{\mathrm{B}}Q_{F_i} = r_{h_i} \times ({}^{\mathrm{B}}T_i + {}^{\mathrm{B}}H_i) \tag{4.2.2.5}$$

4.2.2.4 Total Rotor System Torque

The resulting total torque in the body frame of reference due to the rotors can be expressed as:

$${}^{\mathrm{B}}Q_r = \sum_{i=1}^{4}({}^{\mathrm{B}}Q_{F_i} + {}^{\mathrm{B}}Q_{d_i} + {}^{\mathrm{B}}Q_{L_i}) \tag{4.2.2.6}$$

Where ${}^{\mathrm{B}}Q_{F_i}$ and ${}^{\mathrm{B}}Q_{d_i}$ and ${}^{\mathrm{B}}Q_{L_i}$ are defined in Equation 4.2.2.5 and Equation 4.2.2.3 and Equation 4.2.2.4 respectively. Expanding this expression gives:

$$\begin{aligned}
{}^{\mathrm{B}}Q_r = &\sum_{i=1}^{4}(r_{h_i} \times ((K_T \omega_i^2 + \delta_T({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_{h_i})\omega_i)\Gamma_{T_i}) \cdots \\
&+ r_{h_i} \times (-K_H \omega_i \Gamma_H({}^{\mathrm{B}}v_o + {}^{\mathrm{B}}\Omega \times r_{h_i})) \cdots \\
&- K_d \omega_i^2 \Gamma_{\Omega_i} - \tilde{J}_r({}^{\mathrm{B}}\Omega \times \omega_i \Gamma_{\Omega_i} + \dot{\omega}_i \Gamma_{\Omega_i}))
\end{aligned}$$

4.2.3 Rotor System Wrench

The total rotor system wrench is defined using Equation 4.2.1.7 and Equation 4.2.2.6 as:

$$^{\mathrm{B}}W_r = \begin{bmatrix} ^{\mathrm{B}}F_r \\ ^{\mathrm{B}}Q_r \end{bmatrix}$$

4.2.4 BLDC Motors

The motors used with this quadrotor are three-phase brushless DC motors. Brushless DC motors lack the mechanical commutation of brushed dc motors and hence require the input to each phase to be energized by some external source over the correct timespan during rotation. In order to accomplish this power-electronics many times referred to as ESCs (electronic speed controllers)[7] are utilized to implement the necessary inverter–battery DC to AC for each phase– and commutator functionality. Some models of ESC use Hall effect sensors to obtain feedback of the motor position, while others such as those used in this model known as *sensorless* models use the back EMF in the motor leads to estimate position.

Complete modeling of the physics and several types of control of three-phase BLDC motors can be found in [31]. As is implicit in the equations governing the dynamics of such motors ([31]) and is confirmed in sources such as [8], BLDC motors of the type used here can be modeled similarly to brushed DC motors. In what follows, the notation $\tilde{V}$ is used to describe the effective input voltage, i.e. that which is the result of the power-electronic motor control circuit for BLDCs but is from a modeling equations standpoint equivalent to the same amplitude voltage applied to the terminals of a brushed DC motor. A circuit diagram representing the model to be used here is given in Figure 4.1.

Taking $\tilde{V}$ as the effective input voltage, V_R, V_L, and V_{EMF} as the resistor, inductor, and motor back-emf voltages respectively, and using Kirchhoff's voltage law:

$$\tilde{V} = V_R + V_L + V_{EMF}$$

---

[7] The name may be somewhat deceptive as they many times do not provide any *control* in the sense of feedback based speed reference tracking, e.g. the ESCs used in this model.

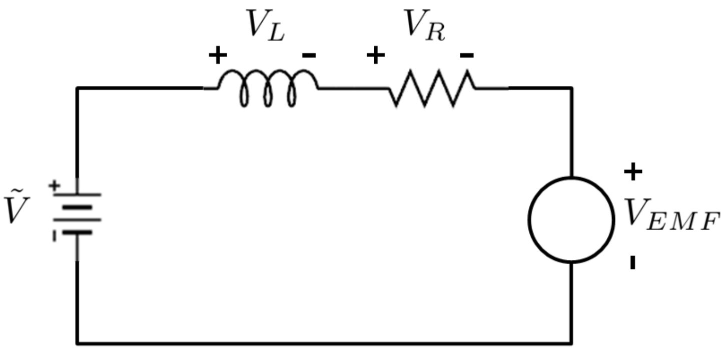


Figure 4.1 Equivalent DC Motor Circuit

The resistor and inductor voltages can be expressed in terms of current using standard circuit laws, and the back-emf in terms of rotor speed [31, 32, 33]:

$$V_R = R_m i \quad V_L = L_m \frac{di}{dt} \quad V_{EMF} = \frac{\omega}{K_V}$$

The effects of inductance can be ignored since for a typical modern DC motor of the type modeled here, the value of L_m is exceptionally small and the transient mechanical response of the motor is much slower than the electrical[32, 33, 2, 34]. Using this, the voltage equation can be simplified to:

$$\tilde{V} = R_m i + \frac{\omega}{K_V}$$

Since the gyroscopic torque effects in Equation 4.2.2.4 have no component along the body frame z-axis b_z, the equation for the angular acceleration of each rotor, and hence each motor[8] can be written in the form:

$$\tilde{J}_r \dot{\omega} = Q_m + Q_f + Q_d$$

Where Q_f represents the internal friction of the motor, expressed here as:

$$Q_f = \frac{-i_f}{K_Q}$$

Where i_f is the what is commonly known as the no-load current, or the current the motor takes to simply rotate wihtout external torque applied. It is modeled as a constant here. A more detailed model of internal losses can be found for example in [33], and in various forms in other sources. Q_m represents the torque supplied by the motor as a function of current, with K_Q as the torque constnat of the motor, i.e. the ratio of current to torque:

---

[8] Assuming no gearbox in this situation.

$$Q_m = \frac{i}{K_Q}$$

$$\Rightarrow Q_m = \left(\left(\tilde{V} - \frac{\omega}{K_V} \right) \frac{1}{R_m} - i_f \right) \frac{1}{K_Q}$$

And finally, Q_d is the torque due to rotor drag as detailed in Section 4.2.2.1, taking the form:

$$Q_r = -K_d \omega^2$$

Substituting these quantities, the equation describing the dynamics of a given motor is expressed:

$$\tilde{J}_r \dot{\omega}_i = \left(\left(\tilde{V}_i - \frac{\omega_i}{K_V} \right) \frac{1}{R_m} - i_f \right) \frac{1}{K_Q} - K_d \omega_i^2$$

4.2.5 Electronic Speed Controls

The input to each ESC is a PWM which drives power-electronics on the board to create the effective voltages for each phase of the motor. Defining the input as a PWM duty cycle percentatge P_i, it can be assumed that the effective voltage is proportional to the duty cycle percentage above the turn-value $P_\perp$ as a fraction of the maximum and the available battery voltage, i.e.

$$\tilde{V}_i = \frac{P_i - P_\perp}{P_\top - P_\perp} V_b$$

Where V_b is the available battery voltage and $P_\top$ is the maximum duty cycle percentage input[9], i.e. $P_\top - P_\perp$ represents the full range of duty cycle input, and is a constant for any given calibration. This can be simplified notationally by defining u_{P_i} as:

$$u_{P_i} = \frac{P_i - P_\perp}{P_\top - P_\perp} \quad , \quad u_{p_i} \in [0, 1] \forall i \qquad (4.2.5.1)$$

---

[9] The ESCs used in this model are flexible in that they can work with various PWM input schemes through a calibration process where the maximum and minimum duty cycle are provided and the effective output is tuned accordingly.

4.2.6 ESC-Motor System

With this, the differential equation defining each motor/rotor velocity is expressed:

$$\tilde{J}_r\dot{\omega}_i = \frac{1}{R_mK_Q}u_{P_i}V_b - \frac{1}{R_mK_QK_V}\omega_i - \frac{1}{K_Q}i_f - K_d\omega_i^2$$

Assuming each motor has the same basic parameters and using the unsubscripted ω to represent the vector of all rotor speeds and unsubscripted u_P as the vector of all normalized inputs, the state equation (with the rotor speeds as the state vector) for the ESC-Motor system can be written:

$$\dot{\omega} = \frac{1}{\tilde{J}_rR_mK_Q}u_PV_b - \frac{1}{\tilde{J}_rR_mK_QK_V}\omega - \frac{1}{\tilde{J}_rK_Q}i_f[1] - \frac{K_d}{\tilde{J}_r}(\omega \circ \omega) \tag{4.2.6.1}$$

Where the $\circ$ used here represents the Hadamard product, i.e.

$$\omega \circ \omega = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

The output equation, defining the ouput from the ESC-Motor System as M, is:

$$M(\omega, u_P, V_b) = \begin{bmatrix} \alpha \\ \omega \end{bmatrix} \tag{4.2.6.2}$$

Where the vector α represents the vector of rotor speed derivatives, i.e. $\alpha = \dot{\omega}$.

4.2.6.1 ESC-Motor System Without Transient

For the purposes of contorl design (not for simulation testing purposes), since the motor dynamics are very fast in comparison to the rigid body dynamics, and in order to keep the state vector used from being unnecessarily large, the motors can be assumed to achieve their commanded speed *effectively* instantaneously, i.e. with no transient. In order to express this, the deritative term in Equation 4.2.4 is set to zero as though the steady state speed value has already been reached, and the resulting quadratic equation is solved for ω:

$$\omega_i = \frac{-1 + \sqrt{1 - 4R_mK_VK_QK_d(K_VR_mi_f - K_Vu_{p_i}V_b)}}{2R_mK_VK_QK_d} \tag{4.2.6.3}$$

4.2.7 Battery

The battery is the sole power supply for the quadrotors motors and electronics. It provides power to each of the motors through the ESCs, which themselves each contain a BEC (battery eliminator circuit) to provide power to the on-board electronics, i.e. the GU-344 unit and Receiver[10]. As is evident from the modeling presented in Section 4.2.6, the battery plays an important role in the rotor speed dynamics and hence the dynamics of the entire vehicle. Circuit runtime battery models and identification experiments can be found in literature, e.g. [35, 36, 37] and online nonlinear estimation found in e.g. [38].

Since the motor currents (and hence power consumption) are not modeled (see Section 4.2.4) and the motor dynamics have been assumed a function of the available battery voltage and input commands only (Equation 4.2.6.1), no thorough battery modeling was done here. Undoubtedly a higher fidelity model could hypothetically be obtained if the motor currents and power consumption were modeled accurately in combination with a good model of the battery runtime characteristics and transient behavior. The increased complexity and difficulty of such a task were seen as outweighing the modest improvements to the model it would make here. For simulation purposes the battery was modeled as simply a decaying voltage (see Section 5.4), and for control design purposes as a static voltage between its maximum and minimum[11] values.

The simulation discharge rate for the battery is represented using the parameter δ_V, while a static operating voltage for contorl design[12] for simulation initial conditions is represented by V_0. With these quantities defined, the battery voltage during any given flight was modeled as following the equation:

$$V_b(t) = V_0 + \delta_V t \tag{4.2.7.1}$$

---

[10] The receiver power is obtained from a connection to the GU-344, which is powered by the battery through the BECs

[11] The vehicle will shut off automatically at a certain voltage level to protect the battery from damage.

[12] In order to avoid assume a time invariant system for desing purposes.

4.3 Gravity

The effects of gravity are modeled here as resulting from a force following Newton's universal law of gravitation. Since all flight conditions being modeled are relatively close to the surface of Earth, the inverse-square law formulation, $F_g = G_0 \frac{m_1 m_2}{r^2}$ is replaced with a standard average acceleration representing the acceleration effects of the inverse-square formulation near the surface of the planet, given by the scalar g.

The inertial/Earth frame of reference for this model was defined such that the acceleration of gravity would be along the positive z-axis. So using m as the mass of the quadrotor in flight, the gravitational force G represented in intertial[13] coordinates can be expressed:

$$G = E\,{}^{\mathrm{E}}G \quad \text{where: } {}^{\mathrm{E}}G = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

4.3.1 Body Frame of Reference Gravitational Force

This model uses forces represented in the local, body frame of reference. The expression for the force of gravity can be given in this frame using the change of coordinates matrices detailed in Section 2.1.4 as:

$$G = E\,{}^{\mathrm{E}}G = B\,{}^{\mathrm{B}}G \quad \text{where: } {}^{\mathrm{B}}G = L_{\overleftarrow{\mathrm{BE}}}\,{}^{\mathrm{E}}G \tag{4.3.1.1}$$

Or expanded as:

$${}^{\mathrm{B}}G = \begin{bmatrix} -mg\sin(\theta) \\ mg\cos(\theta)\sin(\psi) \\ mg\cos(\theta)\cos(\psi) \end{bmatrix}$$

4.3.2 Body Frame of Reference Gravitational Torque

The equation of motion used to describe the dynamics of the system, i.e. Equation 2.2.5.1, is with respect to the body frame of reference origin. The force of gravity always acts at the center of mass. Therefore, if the origin of the body frame of reference is not coincident with the physical center of mass, the force of gravity causes a torque due to the offset with respect to the origin of the reference frame. This is expressed here with respect to the body frame of reference as the torque due to gravity:

---

[13] Again, see Appendix A for an explanation of treating a reference frame fixed with respect to Earth as inertial.

$$^{\mathrm{B}}Q_G = {}^{\mathrm{B}}r_{oc} \times {}^{\mathrm{B}}G \tag{4.3.2.1}$$

4.4 Other Aerodynamic Effects

Aerodynamic effects due to things like fuselage and empennage drag[1, 29, 30] and wake deflection are also not taken into account. The formerly mentioned sources deal mainly with conventional model helicopers with far more significant fuselage cross sectional area in the xy-plane than the quadrotor being modeled here. Also a quadrotor has no tail and hence empennage modeling is completely unnecessary.

In [29, 30] the in-plane rotor forces (Section 4.2.1.2) were lumped into expressions for fuselage drag and wake deflection forces. In essense a sort of opposite approach is taken here, with whatever forces do act directly on the cross sectional area of the fuselage *lumped into* expressions for in-plane rotor drag (4.2.1.6) and/or velocity dependent thrust losses (i.e. the effects of δ_T in 4.2.1.5).

Ground effect was also not taken into account in this model. As given in [1, 28] among others, without significant forward velocity the effects of proximity to the ground or other surface can be calculated as:

$$\frac{|T_{GE}|}{|T_0|} = \frac{1}{1 - (\frac{r}{4h})^2}$$

Where T_{GE} and T_0 represent the thrust produced with ground effect and without, r is the rotor radius, and h is the height above the surface. The thrust increase due to ground effect was therefore assumed to drop off significantly enough at even modest heights that it was not included in the model[14].

---

[14] The thrust increase according to this equation is only approximately 7% when a given rotor is above a

4.5 GU-344

The GU-344 serves several purposes as part of the GAUI 330X system. The three main functions it performs are as follows:

1. **Body Frame Angular Velocity Feedback** - On board 3-axis gyroscope sensor assumed to provide the model equivalent of ${}^{\mathrm{B}}\Omega = [p\ q\ r]^T$.
2. **Angular Velocity Tracking Control** - Feedback control of the body frame angular rates to track reference linearly related to the aileron, elevator, and rudder inputs.
3. **Control Signal Mixing** - Mix the throttle input and the results of the angular rate tracking to produce signals to send to each motor ESC (see Section 4.2.6).

The assumed model of the GU-344 is shown in Figure 4.2 below:

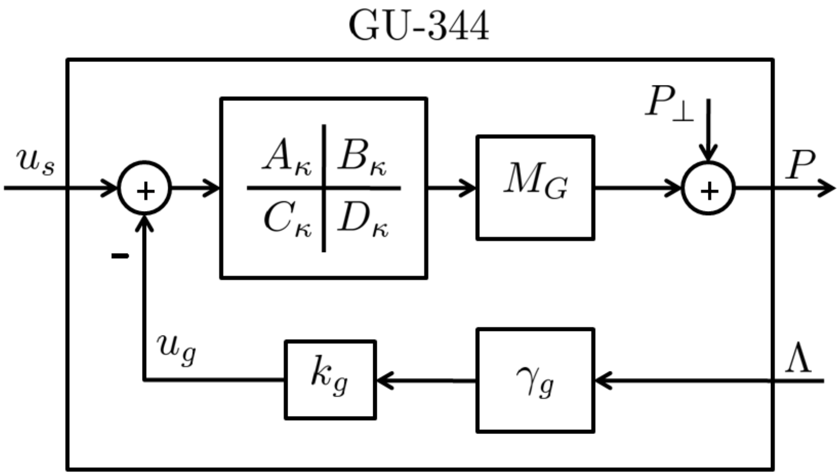


Figure 4.2 Assumed GU-344 Model

---

surface a distance equal to its own radius and the radii of the rotors of the vehicle modeled here are only about 10cm. The vehicle used here as it is set up has rotors that are already beyond this height while landed. By the time the rotors reached a 30cm height (only 1.3 times the vehicle chassis width), the thrust increase would be under 1%. Near ground flight was not part of the intended flight conditions for the control developed here, so ground effect was ignored.

4.5.1 Gyroscope Feedback

The gyroscope feedback capability of the GU-344 is provided by three separate single axis angular rate sensors with specifications given in [39]. The dynamics of the angular rate sensors are fast enough relative to the rest of the system to be assumed effectively instantaneous and the feedback is modeled a static gain matrix:

$$\gamma_g = \begin{bmatrix} [0]_{(4\times 3)} & \begin{bmatrix} 0 & 0 & 0 \\ \gamma_p & 0 & 0 \\ 0 & \gamma_q & 0 \\ 0 & 0 & \gamma_r \end{bmatrix} & [0]_{(4\times 6)} \end{bmatrix}$$

Where γ_p, γ_q, and γ_r correspond to the gain from observed angular velocity to equivalent integer command values. Defining the output of the gyroscop feedback u_g as:

$$u_g = k_g \gamma_g \Lambda = \begin{bmatrix} 0 \\ k_g \gamma_p p \\ k_g \gamma_q q \\ k_g \gamma_r r \end{bmatrix}$$

Where the gyroscope feedback gain k_g is simply a scalar gain affecting the feedback magnitude, which is set by a potentiometer on the hardware.

4.5.2 Tracking Control

The GU-344 tracking control subsystem is modeled as having a single state. This state is defined as ι, and represents the accumulated value in the rudder channel proportional-integral controller. Both the aileron and elevator channels are modeled as having proportional control, while the throttle channel has the equivalent of a proportional gain[15]. The control and gain functionality are considered linear, and the state space matrices for the subsystem are defined:

---

[15] The throttle channel has no feedback within the GU-344.

$$A_\kappa = 0 \qquad B_\kappa = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C_\kappa = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \kappa_{R_I} \end{bmatrix} \qquad D_\kappa = \begin{bmatrix} \kappa_T & 0 & 0 & 0 \\ 0 & \kappa_A & 0 & 0 \\ 0 & 0 & \kappa_E & 0 \\ 0 & 0 & 0 & \kappa_{R_P} \end{bmatrix}$$

Where κ_T is the gain from throttle input to output, κ_A, κ_E and κ_{R_P} are the proportional gains corresponding to body frame angular rate tracking, and κ_{R_I} is the integral gain of the PI controller on the rudder channel. The input to the tracking control subsystem is $u_s - u_g$ or $u_s - k_g \gamma_g \Lambda$, i.e. the integer throttle input and equivalent integer angular rate tracking errors.

4.5.3 Signal Mixer

The signal mixing functionality of the GU-344 is allows the *more-intuitive-for-a-human-pilot* inputs of throttle, aileron, elevator, and rudder to form equivalent commands for each of the four individual motor speed controllers[16]. This mixing is defined by the following matrix:

$$M_G = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \tag{4.5.3.1}$$

Where ±1 designates the either positive or negative effect each input has on each duty cycle percentage command and hence each rotor speed. The elements of this matrix generally depend on the assignment of the body frame axes with respect to the actual quadrotor, and the numerical assignments given to each rotor. The assignments made in this model are explained in Section 3.2.1.

---

[16] This mixing functionality is obviously not necessary under fully autonomous control scenarios where the control could simply output the motor commands directly.

The state equation and output equation, P, for the GU-344 are:

$$\frac{d}{dt}\iota = i = A_\kappa \iota + B_\kappa(u_s - k_g\gamma_g\Lambda) \tag{4.5.3.2}$$

$$P(\iota, u_s, \Lambda) = M_G C_\kappa \iota + M_G D_\kappa(u_s - k_g\gamma_g\Lambda) + P_\perp[1] \tag{4.5.3.3}$$

Or in expanded form:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} \kappa_T u_T - \kappa_A(u_A - k_g\gamma_p p) + \kappa_E(u_E - k_g\gamma_q q) + \kappa_{R_P}(u_R - k_g\gamma_r r) + \kappa_{R_I}\iota + P_\perp \\ \kappa_T u_T - \kappa_A(u_A - k_g\gamma_p p) - \kappa_E(u_E - k_g\gamma_q q) - \kappa_{R_P}(u_R - k_g\gamma_r r) - \kappa_{R_I}\iota + P_\perp \\ \kappa_T u_T + \kappa_A(u_A - k_g\gamma_p p) - \kappa_E(u_E - k_g\gamma_q q) + \kappa_{R_P}(u_R - k_g\gamma_r r) + \kappa_{R_I}\iota + P_\perp \\ \kappa_T u_T + \kappa_A(u_A - k_g\gamma_p p) + \kappa_E(u_E - k_g\gamma_q q) - \kappa_{R_P}(u_R - k_g\gamma_r r) - \kappa_{R_I}\iota + P_\perp \end{bmatrix}$$

4.6 Camera System Model

The camera system was modeled as a discrete system with sampling period T_c, providing measurements of $r_o = [x\ y\ z]^T$ and $\Theta = [\phi\theta\psi]^T$ with some latency τ_c. Defining the camera system output as Y_c:

$$Y_c[n] = \begin{bmatrix} x(n(T_c - \tau_c)) \\ y(n(T_c - \tau_c)) \\ z(n(T_c - \tau_c)) \\ \phi(n(T_c - \tau_c)) \\ \theta(n(T_c - \tau_c)) \\ \psi(n(T_c - \tau_c)) \end{bmatrix}$$

And for simulation purposes the camera system was also modeled as having some small white noise added to all measurements.

4.7 Communication System Model

The communication system described in Section 3.5 is modeled as a latency and transfer function for each channel. The input to the communcation system from the Control PC is u_{s_0} and its output to the GU-344 unit is u_s[17]. Defining the communication system latency for each channel as τ_T, τ_A, τ_E, τ_R and the corresponding throttle, aileron, elevator, and rudder channel transfer functions with no delay as $G_{T_x}(s)$, $G_{A_x}(s)$, $G_{E_x}(s)$, and $G_{R_x}(s)$ respectively:

$$u_s = \begin{bmatrix} e^{-\tau_T s}G_{T_x}(s) & 0 & 0 & 0 \\ 0 & e^{-\tau_A s}G_{A_x}(s) & 0 & 0 \\ 0 & 0 & e^{-\tau_E s}G_{E_x}(s) & 0 \\ 0 & 0 & 0 & e^{-\tau_R s}G_{R_x}(s) \end{bmatrix} u_{s_0} \tag{4.7.0.4}$$

---

[17] It should be pointed out that the s in the notation for these quantities is *not* the Laplace variable that it represents in the transfer functions.

CHAPTER 5. System Identification

5.1 Measurement Equipment and Software

Table 5.1: Equipment

Description	**Manufacturer**	**Model**	**Relevant Specifications**[1]
Scale	Dymo	M10	0-4500g, 2g RES [40]
Photo Tachometer	Extech	461859	5-99,999rpm, 1rpm RES [41]
Photo Tachometer	Kleton	K4010	5-99,999rpm, 1rpm RES [41]
BLDC Motor/Encoder	ECE	220	16,000counts/rev [42]
Oscilloscope	Tektronix	DPO3034	4-Ch., 300MHz BW, DPO [43]
Oscilloscope	HP	54600B	2-Ch., 100MHz BW, DSO [44]
Multimeter	HP	34401A	6.5 digit RES [45]
Waveform Generator	HP	33120A	15MHz BW, 12Bit RES [46]
Power Supply	Kepco	JQE 25-20 M	0-25V/0-20A
Power Supply	Tektronix	CPS250	0-20V/0-0.5A , 0-5V/0-2A [47]
IR Cameras (12)	OptiTrack	V100:R2	100FPS , 10ms latency[48]
IR Camera Hubs[2] (2)	OptiTrack	OptiHub 2	USB2.0[48]
PC Microphone	Realtek		
PC-Transmitter FPGA	Digilent	Nexys 2	
Transmitter/Receiver	Spektrum	DX6i/AR6200	24 GHz DSM2

---

[1] Only specifications relevant to the specific use(s) of the equipment here are given. For full specifications see the equipment manuals and/or contact manufacturer.

[2] Along with all necessary USB cables and Sync Cable

Table 5.2: Software

Description/Application	**Vendor/Author**	**Version**	**Operating System**[3]
MATLAB/Simulink	MathWorks	R2012b [4]	Windows 7
Tracking Tools	OptiTRack	2.3.3	Windows 7
VRPN	many (open source)[5]	7.29	Ubuntu 7.10
Audacity	many (open source)[6]	1.2.6	Windows 7
ECP Executive GUI	ECP	5.1	Windows XP
Command Line Control Interface	Kyle Teske/Roy Lycke	n/a	Ubuntu 7.10
Graphical Control Interface	Jeff Wick	n/a	Ubuntu 7.10
PC-Transmitter FPGA VHDL	Dr. Phillip Jones	n/a	n/a

5.2 Rigid Body Dynamics Parameters

The parameters corresponding to the rigid body dynamics of the quadrotor are the mass m and moment of inertia tensor J. The moment of inertia tensor is itself generally made up of six parameters (see Section 2.2.6). It is assumed in this model however that only three of these paraemters (J_{xx}, J_{yy}, J_{zz}), i.e. those corresponding to the principle axes of inertia, are nominally non-zero.

5.2.1 Mass

The parameter m represents the nominal in-flight mass of the quadrotor in this model, i.e. it represents the mass of the vehicle itself as well as the necessary battery. Represented as symbols this means $m = m_q + m_b$ with m_q the vehicle mass and m_b the battery mass. All batteries used with the quadrotor modeled here were the same mass within ±2grams, which is within the margin of error of hte equipment used for measurement. All masses were measured using a digital scale (Dymo M10 [40]) and are, in kilograms:

---

[3] Operating system specified may only represent that *used* here, not the only one that a given software is capable of being run on.

[4] MATLAB versions R2011a, R2011b, R2012a, R2012b were all used over the course of this work, along with the accompanying versions of Simulink and a number of toolboxes.

[5] At the time of this writing, information on authors and contributors could be found at: http://www.cs.unc.edu/Research/vrpn/

[6] At the time of this writing, information on authors and contributors could be found at: http://audacity.sourceforge.net/about/credits

$$m_q = 0.456 \quad m_b = 0.200 \quad \Rightarrow \; m = 0.656$$

5.2.2 Moment of Inertia Tensor

As detailed in Section 2.2.6, the nominal body frame axes were assumed to be through the principle axes of the quadrotor, making the moment of inertia tensor for the system diagonal. The moment of inertia for each axis was measured using a ECP 220 and accompanying Executive Software by attaching the vehicle to one of the disks and comparing the achieved angular acceleration to that without the it attached.

That is, first the inertia of the disk and rig without quadrotor was established, then with the quadrotor attached to rotate around each of its principle axes (nominally the body frame axes) the inertia measurements were repeated and the difference between the observed inertia with quadrotor and the inertia without quadrotor were taken as the inertia of the quadrotor about that axis using the principle of superposition.

5.2.2.1 Theory

The ECP 220 is documented as producing a constant torque for open loop step trajectory inputs based on input voltage. The calculations here use the constant $0.2\frac{Nm}{V}$ as provided in [42]. Given an approximately uniform torque, an approximately uniform angular acceleration is produced for a specified duration.[7]

Temporarily defining and using the variables α to represent the angular acceleration of the disk+rig or disk+rig+quadrotor as attached for a given experiment, J to represtent the moment of inertia of the system in a given setup, and τ to represent the applied torque, since the rotation in this experimental setup is confined to a single axis (i.e. the cross-product disappears):

---

[7] Here beyond the uniform torque assumption, all friction was assumed to be coulomb in nature and negligable in magnitude. The disk/encoder used in the experiments was found to be very low friction so this assumption was not terribly inaccurate.

$$\tau = J\alpha$$

If the initial position was taken as zero (something that could be guaranteed through the software) and the initial velocity zero, and expression for angular displacement (represented by θ here) is the second antiderivative in time (t) of the constant acceleration:

$$\theta = \frac{1}{2}\alpha t^2$$

Solving this for angular acceleration:

$$\alpha = \frac{2\theta}{t^2}$$

What was set in the software was a constant command voltage amplitude, which was given as corresponding to a constant torque magnitude by the relationship $\tau = \frac{1}{5}V$[42]. Solving for J in the preceeding equation for torque, and substituting:

$$J = \frac{\tau}{\alpha} = \frac{\frac{1}{5}V}{\frac{2\theta}{t^2}} = \frac{Vt^2}{10\theta} \tag{5.2.2.1}$$

5.2.2.2 Procedure

For these measurements, the belt connecting the largest disk at encoder 2 to the intermediate disk and drive was removed, leaving only the belt to the disturbance motor connected. A rig to mount the quadrotor to in appropriate positions for measurement was attached to the disk at encoder 2. The settings in the Executive Software for all measurement runs were:

Trajectory:	**Disturbance:**
Open Loop Step	Step Size (volts): 0.2V
Step Size (volts): 0.00V	Dwell Time (msec): 10000
Dwell Time (msec): 10000	Number of reps: 1
Number of reps: 1	

The steps involved in each measurement set were identical:

1. Ensure rig centered and (if being measured) quadrotor mounted with one of the three principle axes (nominally the body frame axes) aligned with the axis of rotation of the disk at encoder 2.[8]
2. Zero position of encoders.
3. Execute trajectory with disturbance included (the disturbance motor is used to drive the disk in this setup).
4. Export raw data file.
5. Repeat several times.
6. In each data set find the time of the last sample where the input was active, and the angular displacement at that time (taken in counts for these experiments) and if necessary convert to radians.[9]
7. Use this data in Equation 5.2.2.1 to calculate a moment of inertia result.
8. Average the result from all runs in a given configuration to produce a final moment of inertia.

5.2.2.3 Results

The results of the above procecure were, in units of kgm^2, as follows:

$$J_{xx} = 8.1 \times 10^{-3}$$

$$J_{yy} = 7.4 \times 10^{-3}$$

$$J_{zz} = 13.5 \times 10^{-3}$$

5.3 ESC-Motor Parameters

The physical parametrs for the motors (GUEC GM-410 brushless DC motors) used in this model were all obtained from the accompanying documentation. The assumed model for the Electronic Speed Controllers (GUEC GE-010 10A BLDC ESCs) does not have fixed parameters, other than the *turn-on-percentage* $P_\perp$, or minimum PWM duty cycle that will cause the ESC to start turning the motor. The model parameter corresponding to the maximum PWM duty cycle, $P_\top$ is variable depending on a given calibration.

---

[8] If the quadrotor was to be attached, a battery was included as well in flight position.
[9] The converstion for the model used was 16000 counts/revolution

The ESCs and motors used in this model were setup in the *sensorless* configuration, i.e. there were no Hall-effect sensors or any other direct measurements of speed. The ESCs were assumed to instead use back-EMF to estimate which phase to engergize when. As is apparent from the assumed model (Seciton 4.2.6) the ESC-Motor pairs of this model do not explicitly track speed as an input, but instead produce a speed that is a function of the given command, battery state of charge, and any external torque.

The ESCs used here also include a BEC (battery eliminator circuit) in order to provide 5V constant power to the GU-344 without necessitating a separate battery. This voltage was observed as essentially constant regardless of battery fluctutations.

5.3.1 Motor Parameters

The relevant parameters for the motors used here are presented here. Some have been calculated based off of the provided values in order to be in the correct units used here.

Symbol	Nominal Value	Units	Brief Description
R_m	0.19	$Ohms$	Motor Resistance
K_V	110	$rad/s/V$	Back-EMF Constant
K_Q	110	Nm/A	Torqe Constant
i_f	0.39	A	No-Load (friction) Current

5.3.2 ESC Parameters

The turn-on duty cycle for the ESCs used here was determined to be 29%, i.e.

$$P_\perp = 0.29$$

5.4 Battery Parameters

The battires used for flying the quadrotor in this model were Zippy 2100 35C series Lithium Polymer batteries with a maximum open circuit voltage of approximately 12.5V. In order to find a roughly approximate constant discharge rate to use for simulation purposes, a battery was attached to the quadrotor such that the voltage its was supplying at any given time to each of the motor controllers could be observed using a multimeter (in this case HP 34401A[45]) while being given an input to keep it approximately at a hovering condition. The obtained discharge curve could then be fit with a line using least-squares optimization and the slope used as a rough discharge rate. The discharge curve obtained and the linear fit are depicted in Figure 5.1. The equation of the best fit line found for this data is:

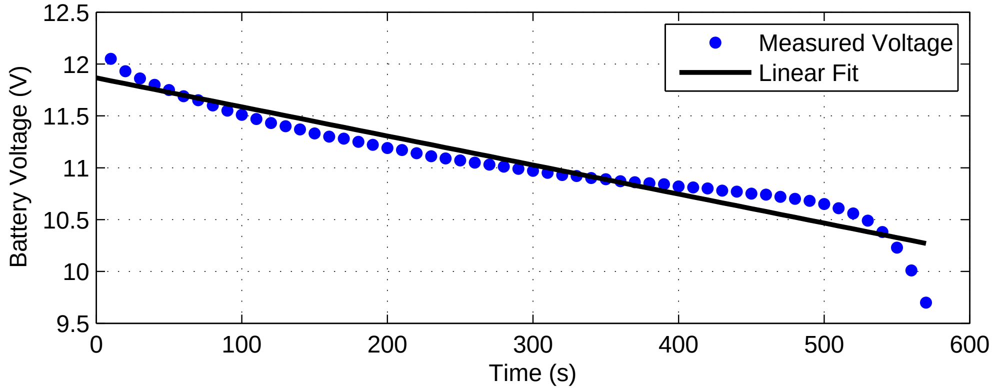


Figure 5.1 Example Battery Discharge Curve

$$V_b = 11.8663 - 0.0028t$$

Assigning the slope obtained as the discharge slope (Section 4.2.7), and assigning the closer-to-mid-range voltage 11.4 as the static operating parameter:

$$\delta_V = -0.0028\frac{V}{s} \quad V_0 = 11.4V$$

5.5 Rotor Parameters

5.5.1 Thrust Constant

The thrust constant was obtained by giving a set of various throttle input commands, and measuring each rotor speed using a optical tachometer and the thrust produced using a digital scale. Since the vehicle was being held stationary during these tests the simple quadratic approximation formula of Section 4.2.1.1 was accurate and the single lift constant K_T could be calculated. The experiments were performed on the quadrotor with all motors active (and gyroscope feedback disabled) and setup inverted with the scale elevated from the floor to minimize any ground effect.

5.5.1.1 Theory

Assuming all rotor speeds are measured along with the corresponding thrust produced, using the results of Section 4.2.1.1 and using the index j for each set:

$$(\omega_{1_j}^2 + \omega_{2_j}^2 + \omega_{3_j}^2 + \omega_{4_j}^2)K_T = |T_j|$$

All these sets of data together form an overdetermined system and can be combined into a matrix equaiton form:

$$AK_T = T$$

Where:

$$A = \begin{bmatrix} \omega_{1_1}^2 + \omega_{2_1}^2 + \omega_{3_1}^2 + \omega_{4_1}^2 \\ \omega_{1_2}^2 + \omega_{2_2}^2 + \omega_{3_2}^2 + \omega_{4_2}^2 \\ \vdots \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} |T_1| \\ |T_2| \\ \vdots \end{bmatrix}$$

With A and T defined in this way, the K_T that minimizes the Euclidian norm of an assumed zero mean measurement error vector (See Appendix B) can be found as:

$$K_T = (A^T A)^{-1} A^T T \tag{5.5.1.1}$$

5.5.1.2 Procedure

The first thing that must be done is to disable the gyroscope feedback in the GU-344[10]. Since the sensors it contains for angular velocity have significant bias and cause the command given to each motor to vary noticeably when the quadrotor is kept stationary, its feedback shold be disabled in order to be able to obtain steady state rotor speed readings that correspond in time to thrust readings. The procedure for disabling the gyroscope feedback is found in [25]. Essentially the *green* wire normally plugged into the *Gear* channel of the receiver is disconnected and the potentiometer on the GU-344 can be used to adjust the gyroscope feedback gain (in this model this parameter is represented by k_g, see Section 4.5) to essentially 0.

With this done, the quadrotor can be placed using some supports onto the scale (Dymo M10[40] here) and provided with various throttle commands. The scale should of course be zeroed out with the quadrotor and all other masses already in place. In this case the control software was used in order to provide these inputs. For each input, all rotor speeds are measured using the optical tachomter (in this case either Extech 461895 or Kleton K4010[41]). To facilitate speed measurement, a thin strip of the IR reflective tape provided with the tachometers should be attached to the side of the rotating motor housing, allowing for measurement without mechanical interference. The data collection process is as follows:

1. Provide some throttle command to the quadrotor.
2. Using the optical tachometer, record the approximate steady state speed reached by each motor/rotor.
3. Record the corresponding digital scale reading.
4. Repeat 1-3 with various inputs and record all data.

The tachometers used here produce readings in RPM, and the scale reads grams. The rotor speeds must be converted to rad/s units using the converstion factor $\frac{\pi}{30}\frac{(rad)(min)}{(rotation)(s)}$ and the grams reading to Newtons thrust by converting grams to kg and multipying by the acceleration of gravity. With this done, Equation 5.5.1.1 can be used to produce a least squares estimate of the parameter K_T. In order to get good coverage the throttle command input should be given at a good number of points over the entire range from its minimum to its maximum value. In this case it was varied over this full range 4 times as the battery drained in order to produce a set of distinict rotor speeds and thrust data.

---

[10] Alternatively nearly identical experiments could be done generating PWM signals for each motor ESC directly.

5.5.1.3 Results

The resulting thrust constant obtained by this process is 7.3956×10^{-6}, i.e.

$$K_T = 7.3956 \times 10^{-6}$$

5.5.2 Thrust Velocity Adjustment Factor

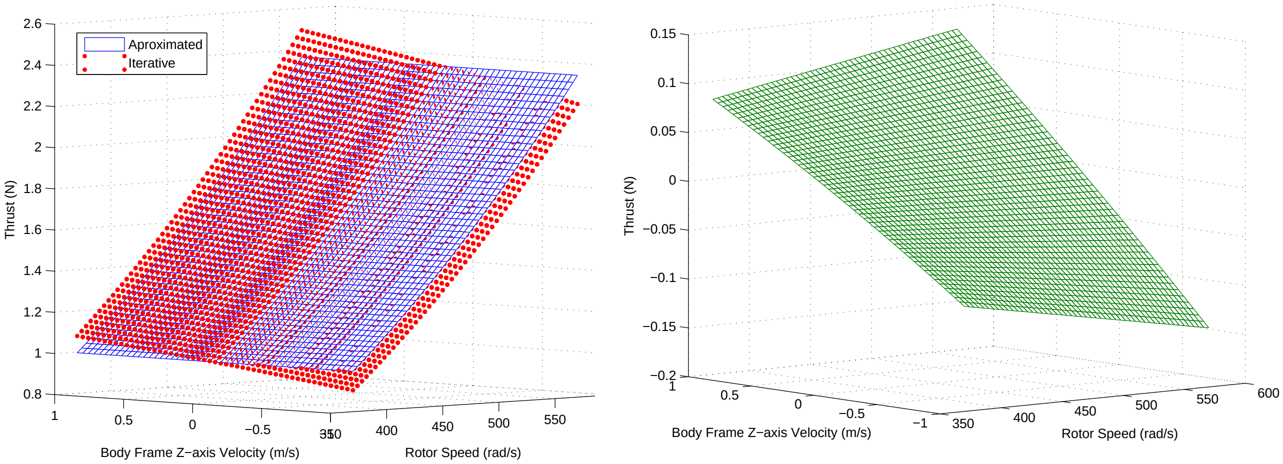
The thrust velocity adjustment factor, δ_T, was a result of comparing the thrust produced using the quadratic approximate thrust (Equation 4.2.1.1) and in-plane drag torque (Equation 4.2.2.1) to the results of iteratively solving a system of more complex equations as described at the start of Section 4.2.1.1. In order to do so various other rotor parameters involved in the system of equations to be iteratively solved, which are otherwise lumped into K_T and K_d, were needed. All could either be obtained as common knowledge, measured, or indirectly calculated from the values of K_T and K_d experimentally obtained in Section 5.5.1 and Section 5.5.4 respectively.

The density of air used in the formulas was $\rho = 1.2041\frac{kg}{m^3}$. The lift curve slope in the iterative equations, a, following various sources putting this general parameter in the range of 5 to 7 (e.g. [4]) for most rotors was taken to be 6. The rotor solidity ratio σ could be approximated using nothing more than a ruler and doing a few calculations based on measurements of the rotors themselves, and the value determined was $\sigma = 0.1147$. The rotor radius was measured as $r = 0.1016m$.

The parameter C_{d_0} used in the iterative in-plane drag equations was lumped into the parameter K_d. Using the determined value of K_d (see Section 5.5.4) the calculated value for C_{d_0} is 0.7659. Using a similar process for the effetive constant collective pitch angle θ_0, lumped into

λ_0 and hence the value of K_T determined in Section 5.5.1, $\theta_0 = 0.2999$ radians (approximatley 17.18^o).

Using these parameters the iterative and quadratic approximate calculations were compared using repeated simulations of the two alternatives under various rotor speed and velocity conditions. In Figure 5.2(a) the resulting thrust under conditions equivalent to the rotor hub having a velocity[11] of $\pm 1 \frac{m}{s}$ normal to its plane of rotation, with rotor speed ranging from a nominal hovering value $\pm 100 \frac{rad}{s}$. Figure 5.2(b) shows the resulting differenct between the two methods of obtaining thrust under the simulated conditions.



(a) Iterative & Quadratic Thrust

(b) Iterative-Quadratic Difference

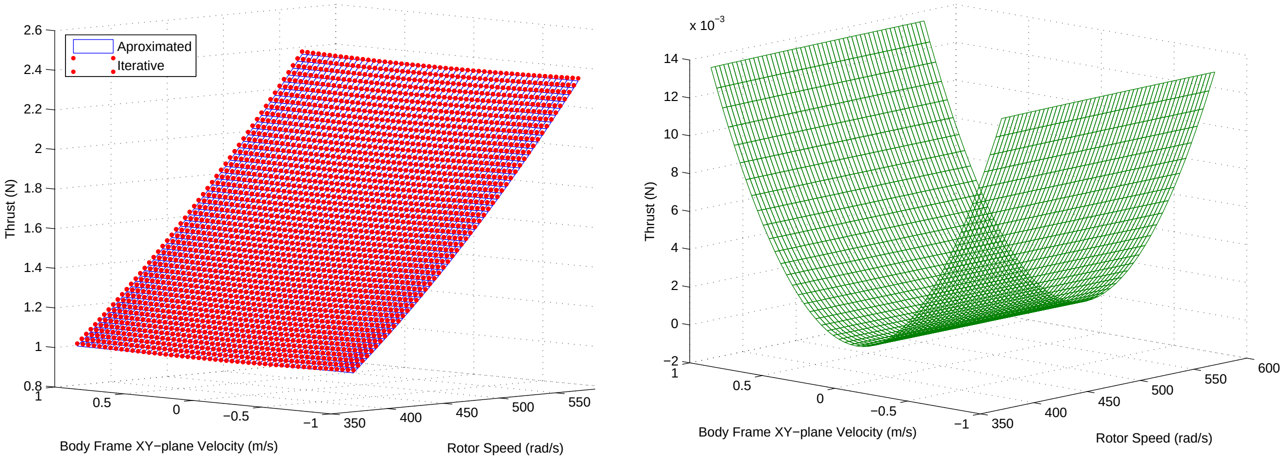
Figure 5.2 Iterative vs. Quadratic Thrust with Z-Axis Velocity

Althought the two methods are exactly equivalent when the relative velocity is zero, there is noticeable deviation when the velocity normal to the rotor plane is either positive or negative[12] that is qualitatively reasonable from a momentum theory perspective. The same sort of simulations were run for relative velocity within the rotor plane, with results shone in Figure 5.3(a) and Figure 5.3(b). As is apparent from these results, with the parameters of this model, the conditions simulated create little to no difference between the iterative and quadratic method of calculating rotor thrust.

---

[11] More specifically, hub relative velocity to the air due to vehicle velocity, wind velocity, or any combination of the two.

[12] Taking account of the fact that in this model a *negative* velocity of this type would correspond to a situation like climbing.

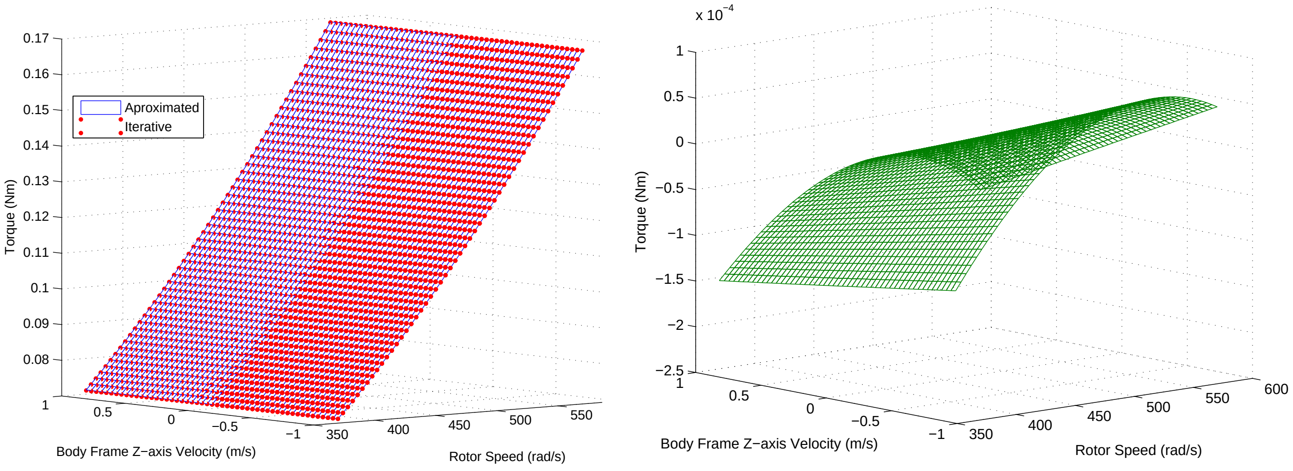


(a) Iterative & Quadratic Thrust

(b) Iterative-Quadratic Difference

Figure 5.3 Iterative vs. Quadratic Thrust with XY-plane Velocity

Next the same process was followed for the in-plane drag torque. The results of simulating the same range rotor speed and relative velocity normal to the rotor plane as with thrust are shown in Figure 5.4(a). Again in this case, looking at Figure 5.4(b), with the model parameters as they are and under simulated conditions there is little to no difference between the iterative and quadratic approximation method of calculating rotor torque. Finally the process was

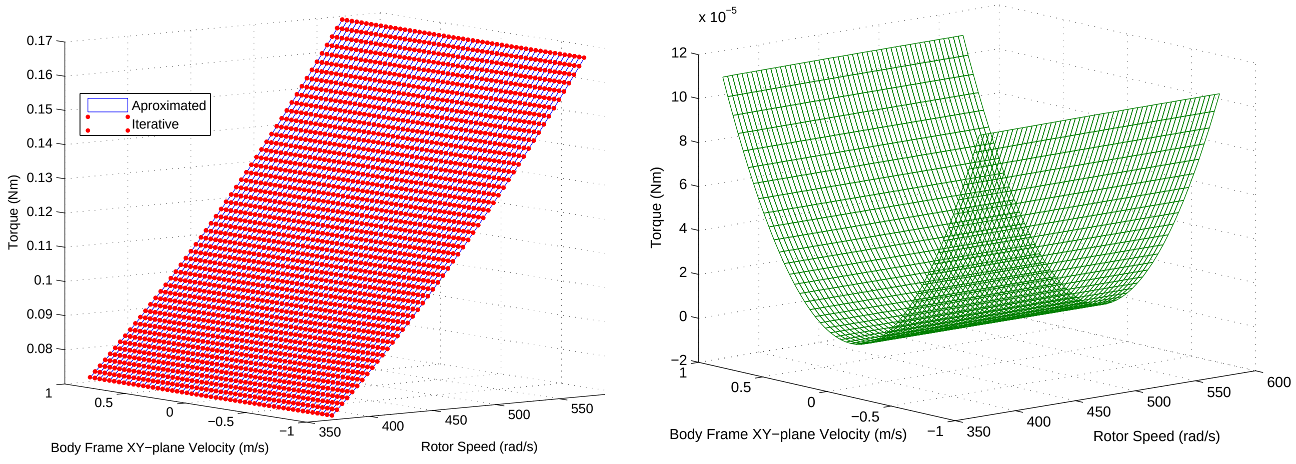


(a) Iterative & Quadratic Thrust

(b) Iterative-Quadratic Difference

Figure 5.4 Iterative vs. Quadratic Torque with Z-Axis Velocity

repeated for in-plane drag torque under the range of rotor speed and in-plane velocity conditions simulated for thrust. The results are shown in Figure 5.5(a) and 5.5(b). Again for the model parameters and conditions simulated no appreciable difference was found between the iterative and closed form approximate methods of calculation. Although for in-plane velocity there was no significant effect on either thrust or torque at various rotor speeds, and no real difference in torque with velocity normal to the rotor plane, Figure 5.2(b) does show a noticeable deviation between the two methods of calculation. It was desireable to take some account of these effects that are normally neglected in the simple quadratic equation for thrust while still keeping a relatively simple closed form solution, i.e. avoiding using a system of equations needing iterative solution. As mentioned in section 4.2.1.1 and defined in Equation 4.2.1.3, this was accomplished by introducing an *adjustment* factor that works off of the product of the *local* z-axis relative air velocity and rotor speed.



(a) Iterative & Quadratic Thrust

(b) Iterative-Quadratic Difference

Figure 5.5 Iterative vs. Quadratic Torque with XY-plane Velocity

As with other parameter fittings, a least-squares process was employed here in order to obtain the *best*[13] adjustment factor value. This was done using the assumed equation (Equation 4.2.1.3) form and defining the result depicted in Figure 5.2(b) for each rotor speed and velocity

---

[13] See Appendix B.

(indexed using j) as:

$$\Delta T_j = \delta_{T_z} \omega_j v_{h_{j_z}}$$

Forming this system of equations into the vectors:

$$[\omega v_z] = \begin{bmatrix} \omega_{j=1} v_{h_{(j=1)_z}} \\ \omega_{j=2} v_{h_{(j=2)_z}} \\ \vdots \end{bmatrix} \qquad [\Delta T] = \begin{bmatrix} \Delta T_{j=1} \\ \Delta T_{j=2} \\ \vdots \end{bmatrix}$$

With these quantities, the optimal least-squares adjustment factor is:

$$\delta_{T_z} = ([\omega v_z]^T [\omega v_z])^{-1} [\omega v_z]^T [\Delta T] \tag{5.5.2.1}$$

The value obtained using this equation and the simulation data and parameters[14] here was:

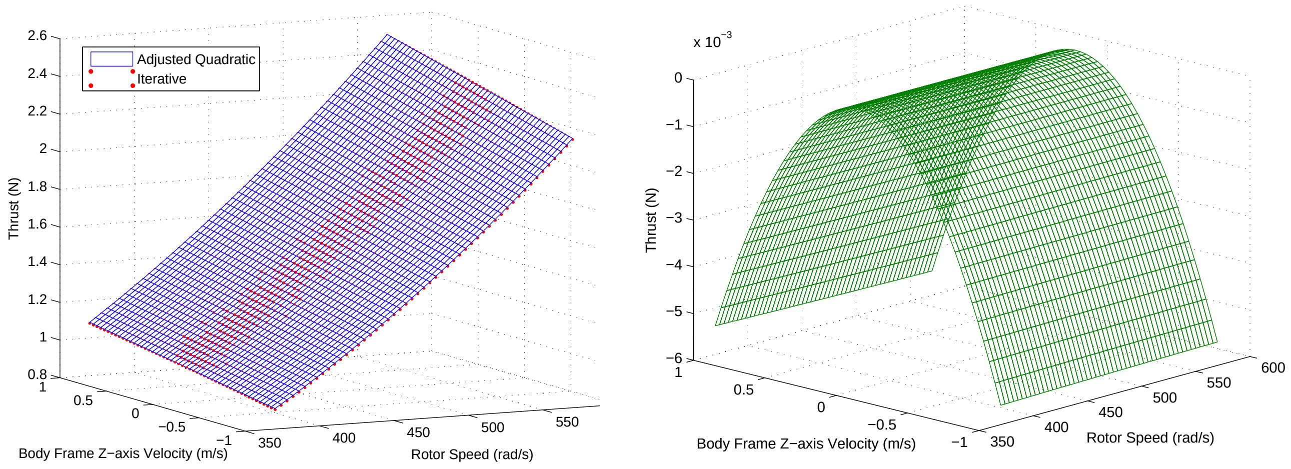
$$\delta_{T_z} = 2.351 \times 10^{-4} \tag{5.5.2.2}$$

Using this result, the *adjusted* closed form equation for rotor thrust (Equation 4.2.1.3) were compared to those obtained by iterative solution, and the reults displayed. As can be seen from the reults in Figure 5.6(a) and Fiture 5.6(b), the adjustment factor found here eliminates the deviation between the iterative and quadratic only calculations observed in Figure 5.2(b). This shows the improvement that can be made to the predictive capabilities of the thrust equation by a thrust adjustment factor. However, since the value in Equation 5.5.2.2 is based off of parameters that are themselves only known approximately, i.e. a_0, θ_0, and σ, the value obtained is itself only guaranteed to be *near* the correct value. Taking this into account, and given some small iterations to the value while comparing the resulting model predictions to experimental data, an updated value was obtaineed:

$$\delta_{T_z} = 1.176 \times 10^{-4}$$

---

[14] It is worth emphasizing that the results here dealing with the magnitude of effects (or lack thereof) created by in-plane velocity and normal velocity at various rotor speeds are all *parameter and model dependant*! That is, for another model with different rotors or other operating conditions, the same type of reults may or may not hold. The operating conditions and specific parameters for a given model have to be taken into account on a case-by-case basis in order to determine what complexity of calculations are sufficient. It can certainly be argued however that generally speaking, some sort of *adjustment* for relative air velocity and rotor speed to the typical constant-times-rotor-speed-squared equation will yield more accurate results. It is the magnitude of such effects that will vary on a case-by-case basis. And of course for some more extreme operating conditions or highly detailed large scale models, simple closed form solution of any type may be completely inadequate.



(a) Iterative & Quadratic+Adjustment Thrust

(b) Iterative-(Quadratic+Adjustment) Difference

Figure 5.6 Iterative vs. Quadratic + Adjustment Thrust with Z-Axis Velocity

The improvement to a simulated height step response given by this δ_T is shown in Figure 5.7.

5.5.3 H-Force Constant

The in-plane rotor drag constant, K_H, defined in Section 4.2.1.2 was determined iteratively by comparing experimental trajectoris resulting from step inputs to the lateral and logitudinal reference channels of the initial PID closed loop (Section 6.1) with the predictions of nonlinear simulation. Best resutls were obtained using a value of 3.4574×10^{-4}, that is:

$$K_H = 3.4574 \times 10^{-4}$$

5.5.4 Drag Constant

Using a DC power supply and function generator to provide input to one of the brushless motor ESCs, and separate power supply as a substitute for the battery, various inputs were given to the subsystem at various equivalent battery voltages. The steady state speed that the motor with rotor reached was measured using an optical tachometer, which combined with the motor

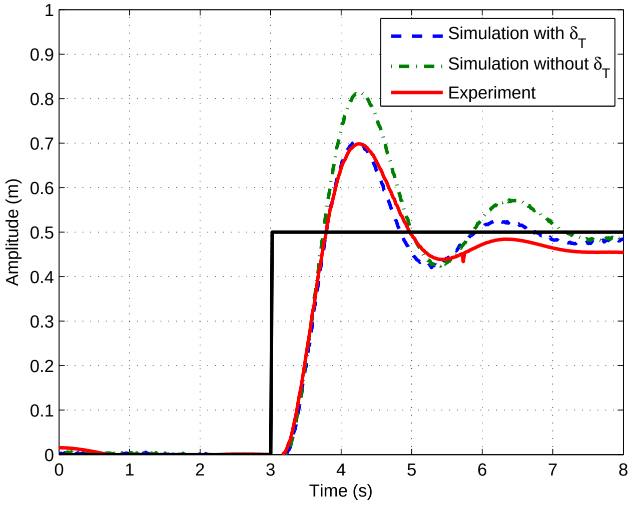


Figure 5.7 Effects of Thrust Adjustment Factor

parameters of 5.3 allowed the in-plane drag and induced torque constant K_d (defined in Section 4.2.2.1) to be calculated.

5.5.4.1 Theory

Assuming a steady state speed has been reached, Equation 4.2.6.1 *for a single rotor* becomes:

$$0 = \frac{1}{R_m K_Q} u_P V_b - \frac{1}{R_m K_Q K_V}\omega - \frac{1}{K_Q} i_f - K_d \omega^2$$

Using the motor parameters R_m, K_Q, K_V, and i_f as provided and the behvaior of the ESC and battery (see Section 5.3 and Section 4.2.6) along with the recorded inputs and measured speeds, this equation consists of known quantities and the unknown quantity K_d. That is if each input PWM and voltage, as well as each output rotor speed are indexed using j, then for the jth instance:

$$(\omega_j^2) K_d = \frac{u_{p_j} V_{b_j}}{R_m K_Q} - \frac{\omega_j}{R_m K_Q K_V} - \frac{i_f}{K_Q}$$

The set of all these equations defines an overdetermined system, and can be combined into a matrix equation with:

$$A = \begin{bmatrix} \omega_{j=1}^2 \\ \omega_{j=2}^2 \\ \vdots \end{bmatrix} \quad , \quad b = \begin{bmatrix} \frac{u_{p_{j=1}} V_{b_{j=1}}}{R_m K_Q} - \frac{\omega_{j=1}}{R_m K_Q K_V} - \frac{i_f}{K_Q} \\ \frac{u_{p_{j=2}} V_{b_{j=2}}}{R_m K_Q} - \frac{\omega_{j=2}}{R_m K_Q K_V} - \frac{i_f}{K_Q} \\ \vdots \end{bmatrix}$$

With these quantities defined, the overdetermined system can be expressed as:

$$AK_d = b$$

So the K_d which minimizes the Euclidian norm of some assumed zero mean measurement error (See Appendix B) can be found using the pseudoinverse of the matrix A, i.e. using the previous definitions of A and b in relation to K_d:

$$K_d = (A^T A)^{-1} A^T b \tag{5.5.4.1}$$

Which in this case can also be expressed:

$$K_d = \frac{\sum_{\forall j} \omega_j^2 \left(\frac{u_{p_j} V_{b_j}}{R_m K_Q} - \frac{\omega_j}{R_m K_Q K_V} - \frac{i_f}{K_Q} \right)}{\sum\limits_{\forall j} \omega_j^4} \tag{5.5.4.2}$$

5.5.4.2 Procedure

The PWM input to the ESC is produced using a DC power supply (in this case Tektronix CPS250[47]) hooked up in series with a waveform generator (in this case HP 33120A [46]). In order to match the output of the GU-344 (ses Section 5.6), the output signal from this series connection must match a PWM within the range 0V to 3.2V, with a total period of 3.6ms and duty cycle range from 29%. To accomplish this the power supply and function generater are set as follows:

Tektronix CPS250:	**HP 33120A:**
Voltage: 1.6V	Amplitude: 1.6V
	Frequency: 280Hz
	Duty Cycle: 28% →

The function generator will allow the duty cycle to be adjusted in 1% intervals, and will cause the motor to turn on at 29%. In order to mimick the battery used for flight, which can have a terminal voltage within the approximate range of 10V to 12V depending on the state of charge, a high current power supply (in this case Kepco JQE 25-20 M) is hooked up to the power input of the ESC.

Before any data can be taken, the ESC should be calibrated! The maximum and minimum PWM duty cycle both define the normalized command output of the ESC defined in Equation 4.2.5.1. The maximum duty cycle is established with a given calibration. The process for calibrating the ESCs used in this model is given in [25, 49]. The following adjusts this procedure to the experimental setup/equipment used in this process:

1. Adjust the high-current power supply to an appropriate voltage such as 11V (since the Kepco JQE 25-20 M has an analog voltage meter, it may be a good idea to have a digital multimeter e.g. in this case HP 34401A [45] in parallel with the supply), and after turning it off connect the supply to the power input of the ESC using appropriate wires and connectors.
2. Connect the low-current power supply and waveform/function generator so as to mimick the PWM duty cycle, and adjust the duty cycle input to what will serve as the *maximum* percentage to allow the ESC to enter calibration mode when powered on. In the caes of the experiements run here 45% was used.
3. Power the ESC by turning on the high-current power supply and ensure that the ESC enters calibration mode by listening for the appropriate tones as indicated in [25, 49].[15] *If calibration mode is not entered something has gone wrong and the process must be restarted, possibly with a higher starting duty cycle percentage.*
4. Wait for the tone indicating *set all to default* and quickly decrease the PWM duty cycle percentage back to the turn off value (28%) to take that setting.

Once this calibration process has been done correctly the data can be collected. In order to do so, the output variable of motor/rotor speed must be measured. This was accomplished here using a optical tachometer (in this case either Extech 461895 or Kleton K4010). A thin strip of the IR reflective tape provided with the tachometers was attached to the side of the rotating motor housing, allowing for the speed to be measured without mechanical interference.

---

[15] Also, by observing that the rotors do not begin spinning at near full speed!

The data collection process is as follows:

1. Set the high-current power supply to a given voltage setting and record the value.
2. Set the PWM duty cycle percentage to some value within the acceptable range and record the value.
3. Using the optical tachometer, record the approximate steady state speed reached by the motor.
4. Repeat 1-3 with various inputs and record all data.

The tachometers used here produce readings in RPM. Convering this data to rad/s units, Equation 5.5.4.1 or equivalently Equation 5.5.4.2 can be used to produce a least squares estimate of the parameter K_d. In order to get good coverage of the in-flight operational range the duty cycle percentage was adjusted several times from the minimum value to the calibrated maximum value in increments of 1% with equivalent battery voltage inputs at 10V, 11V and 12V.

5.5.4.3 Results

The resulting drag constant obtained by this process is 5.194×10^{-7}, i.e.

$$K_d = 5.194 \times 10^{-7}$$

5.5.5 Equivalent Moment of Inertia

The equivalent moment of inertia of the rotating components (in this case the motor rod, upper motor housing, and rotor) affects the acceleration of the rotor/motor, i.e. the transient response to a given speed command. Due to this, the steady state measurements available using the digital tachometers could not provide any useful data to identify this quantity. No direct method of measuring speed as a function of time was available. A similar situation was encountered in [29, 30], where it was dealt with using time-spectral analysis of sound recordings of the response to a speed command step input to give a measure of transient response.

A similar approach is taken here. The main difference being that where [29, 30] used Morlet wavelet transform to perform their time-frequency analysis a short-time Fourier transform approach was used here to obtain equivalent results.

5.5.5.1 Theory

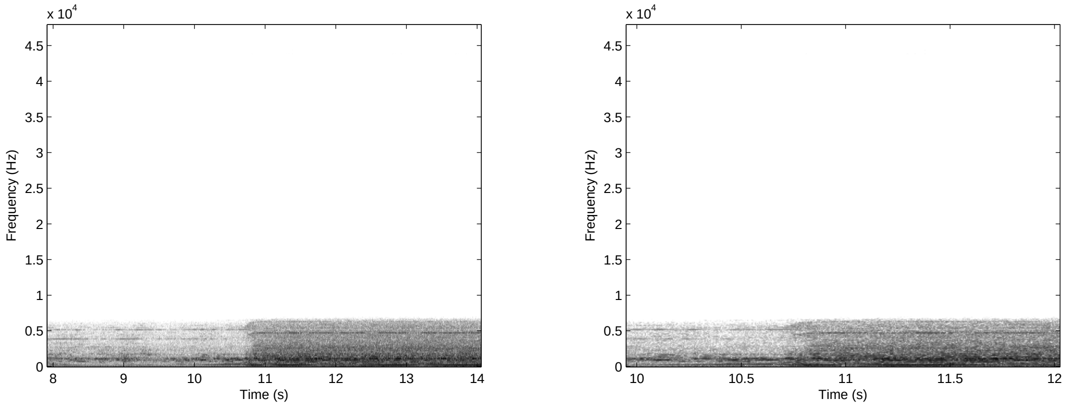
The dynamics of the motor/rotor are described by Equation 4.2.4. Due to the relatively small magnitude of the i_f and K_d constants, the *shape* of the transient response will be dominated by the first two terms on the right hand side of this equation. Using this the approximate response of the system can be treated as linear and a transfer function can be obtained using Laplace transform. Expressing this in terms of the complex variable s:

$$\omega(s) \approx \frac{K_V}{R_m K_V K_Q \tilde{J}_r s + 1} \tag{5.5.5.1}$$

5.5.5.2 Procedure

The procedure for collecting the necessary data to approximate the motor transient is straightforward.

1. Disable the gyroscope feedback as in e.g. Section 5.5.1.
2. Setup the quadrotor so that it cannot lift off or is inverted such that its thrust directs it into a rigid surface without damage.
3. Using the command-line control software issue some small magnitude throttle command keeping all other channels neutral.
4. Setup the recording device (in this case a laptop with microphone and Audacity 1.2.6) to record at a high sampling rate (in this case 96000Hz).
5. Using the command-line control software issue some step increase throttle command.
6. Analyze the data using MATLAB and the FFT algorithm to produce a spectrogram plot showing the approximate transient response of the rotor speed.



(a) Rotor Noise Spectrogram (b) Rotor Noise Spectrogram: Transient Magnified

Figure 5.8 Time-Frequency Motor Analysis

5.5.5.3 Results

It can be seen in Figure 5.8(b) that the settling time of the motor transient response is approximately 0.1s, which indicates a first-order system time constant of approximately 0.0333s. Using this, Equation 5.5.5.1, substituting in all other constants and solving for $\tilde{J}_r$:

$$\tilde{J}_r = 1.376 \times 10^{-5}$$

5.6 GU-344 Parameters

5.6.1 Input/Output

The GU-344 has four PWM outputs, each with a period of 3.6ms. The minimum duty cycle that can be produced on each is 28%, or equivalently 1ms measuring from *brown* to *orange*, with a peak-to-peak amplitude of 3.2V.

5.6.2 Gyroscope Feedback Gains

The GU-344 providing angular velocity feeback meant that the aileron, elevator, and rudder inputs were effectively references for the body frame angular velocity elements. The quadrotor with active gyroscope feedback was given various magnitude inputs on these channels and the steady state velocity was observed for each. This data was then used to calculate the approximate feedback gains for each angular velocity. As with the body moment of inertia elements, the measurements here were performed using the ECP 220 and Executive Software to record the encoder data.

5.6.2.1 Theory

Using the ECP 220 to accurately obtain (as mentioned elsewhere the encoders of the ECP 220 had a resultion of 16000counts/revolution[42]) the angular position of the quadrotor fixed to one of its disks over some timespan, the average speed over that timespan could be calculated as the change in angular position over the total change in time. Temporarily using the variables $\bar{\Omega}$ and $\Delta\theta$ to represent the average angular speed and total angular displacement over the interval Δt:

$$\bar{\Omega} = \frac{\Delta\theta}{\Delta t}$$

A block diagram of the experimental setup is shown in Figure 5.9:

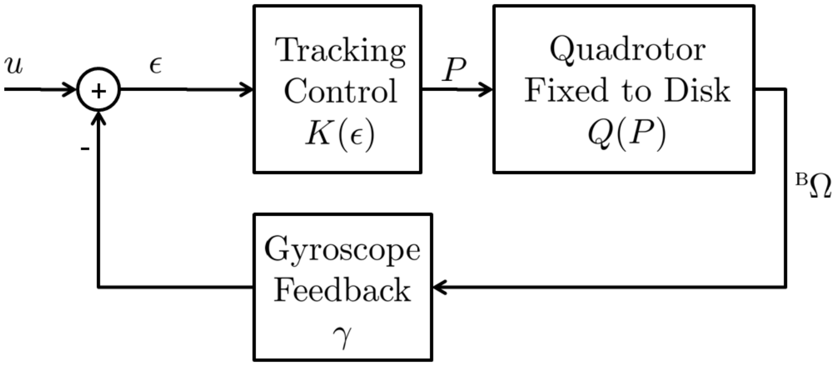

u
+
ϵ
-
Tracking Control $K(\epsilon)$
P
Quadrotor Fixed to Disk $Q(P)$
$^{B}\Omega$
Gyroscope Feedback γ

Figure 5.9 Gyroscope Feedback Parameterization Experimental Setup

Rigidly fixing the quadrotor to one of the disks, i.e. confining it to rotate around single axis, the cross-product term of the rigid body dynamics disappears. Using this the system can be assumed to behave in an effectively linear manner. Treating the systems of Figure 5.9 as effectively continuous and linear, and assuming the Laplace transform of the disk+quadrotor is $Q(s)$, and that of the tracking controller for a given channel is $K(s)$, the speed can be represented as:

$$\bar{\Omega}(s) = \frac{Q(s)K(s)}{1 + Q(s)K(s)\gamma}u(s)$$

Using the final value theorem, i.e. for some time domain function $f(t)$ and its corresponding laplace transform $F(s)$:

$$\lim_{t\to\infty} f(t) = s \lim_{s\to 0} F(s)$$

Giving a step input command of some constant magnitude $\bar{u}$ making $u(s) = \frac{\bar{u}}{s}$, using the fact that $Q(s)$ will involve integral action, and the fact that $K(s)$ will be proportional and integral or just proportional:

$$\lim_{s\to 0} \frac{1}{Q(s)K(s)} = 0$$

$$\Rightarrow \lim_{t\to\infty} \bar{\Omega}(t) = s \lim_{s\to 0} \frac{Q(s)K(s)}{1 + Q(s)K(s)\gamma}u(s) = \frac{\bar{u}}{\frac{1}{Q(s)K(s)} + \gamma} = \frac{1}{\gamma}\bar{u}$$

So allowing enough time for the speed to effectively settle out, and averaging that steady-state value over some time interval, the feedback gain for whichever channel is being activated–corresponding to whichever axis of the quadrotor is setup to rotate on the disk–is calculated as:

$$\gamma = \frac{\bar{u}}{\bar{\Omega}}$$

Repeating the process several times for each axis, and averaging the results, the expression for the feedback gain on a given channel using j to index each of m trails is:

$$\gamma = \frac{1}{m}\sum_{j=1}^{m} \frac{\bar{u}_j}{\bar{\Omega}_j} \tag{5.6.2.1}$$

5.6.2.2 Procedure

The procedure for taking the necessary data to calculate the gyroscope feedback gains is straightforward. No special settings are necessary in the ECP 220 Executive Software other than to set up data aquisition for displacement on whichever disk the quadrotor is to be attached to using the same rig as with the moment of inertia experiments. The ECP 220 and rig can simply be set up the same as in those experiments as well (Section 5.2.2.2). With this setup:

1. Use the command-line version of the control software is used to provide some magnitude step input to the correct channel for the current setup.
2. Give the quadrotor and disk system a moment to settle out to an approximately steady state speed.
3. Activate the Executive Software to begin recording the position for some time interval.
4. Export the data obtained and record the magnitude step input given with it.
5. Repeat several times use Equation 5.6.2.1 to determine an approximate feedback gain for that channel.

5.6.2.3 Results

The results obtained using the the process previously outlined were as follows:

$$\gamma_p = 330.44$$

$$\gamma_q = 330.44$$

$$\gamma_r = 192.68$$

5.6.3 Throttle, Aileron, and Elevator Channel Proportional Gain Elements

5.6.3.1 Theory

For any given set of throttle, aileron, and elevator inputs, u_T, u_A and u_E respectively, keeping the rudder input u_R at its neutral value, a set of four duty cycle percentage outputs will be generated. Indexing each of these using the varible j allows the equations defining the relationships among each set of inputs and each output to be written:

$$
\begin{aligned}
u_{T_j}\kappa_T - u_{A_j}\kappa_A + u_{E_j}\kappa_E + \tilde{P}_\perp &= P_{1_j}\\
u_{T_j}\kappa_T - u_{A_j}\kappa_A - u_{E_j}\kappa_E + \tilde{P}_\perp &= P_{2_j}\\
u_{T_j}\kappa_T + u_{A_j}\kappa_A - u_{E_j}\kappa_E + \tilde{P}_\perp &= P_{3_j}\\
u_{T_j}\kappa_T + u_{A_j}\kappa_A + u_{E_j}\kappa_E + \tilde{P}_\perp &= P_{4_j}
\end{aligned}
$$

Together these sets of data form an overdetermined system of equations. The input and output elements can be formed into vectors for $j = 1, 2, \cdots$ These vectors, are defined:

$$
[u_T] = \begin{bmatrix} u_{T_{j=1}} \\ u_{T_{j=2}} \\ \vdots \end{bmatrix} \quad [u_A] = \begin{bmatrix} u_{A_{j=1}} \\ u_{A_{j=2}} \\ \vdots \end{bmatrix} \quad [u_E] = \begin{bmatrix} u_{E_{j=1}} \\ u_{E_{j=2}} \\ \vdots \end{bmatrix}
$$

$$
[P_1] = \begin{bmatrix} P_{1_{j=1}} \\ P_{1_{j=2}} \\ \vdots \end{bmatrix} \quad [P_2] = \begin{bmatrix} P_{2_{j=1}} \\ P_{2_{j=2}} \\ \vdots \end{bmatrix} \quad [P_3] = \begin{bmatrix} P_{3_{j=1}} \\ P_{3_{j=2}} \\ \vdots \end{bmatrix} \quad [P_4] = \begin{bmatrix} P_{4_{j=1}} \\ P_{4_{j=2}} \\ \vdots \end{bmatrix}
$$

Which can be combined into the matrix equation:

$$
\begin{bmatrix} [u_T] & -[u_A] & [u_E] & [1] \\ [u_T] & -[u_A] & -[u_E] & [1] \\ [u_T] & [u_A] & -[u_E] & [1] \\ [u_T] & [u_A] & [u_E] & [1] \end{bmatrix} \begin{bmatrix} \kappa_T \\ \kappa_A \\ \kappa_E \\ \tilde{P}_\perp \end{bmatrix} = \begin{bmatrix} [P_1] \\ [P_2] \\ [P_3] \\ [P_4] \end{bmatrix}
$$

In order to compact the notation, defining:

$$
A_G = \begin{bmatrix} [u_T] & -[u_A] & [u_E] & [1] \\ [u_T] & -[u_A] & -[u_E] & [1] \\ [u_T] & [u_A] & -[u_E] & [1] \\ [u_T] & [u_A] & [u_E] & [1] \end{bmatrix} \quad x_G = \begin{bmatrix} \kappa_T \\ \kappa_A \\ \kappa_E \\ P_\perp \end{bmatrix} \quad b_G = \begin{bmatrix} [P_1] \\ [P_2] \\ [P_3] \\ [P_4] \end{bmatrix}
$$

The vector of parameters $x_G = \begin{bmatrix} \kappa_T & \kappa_A & \kappa_E & P_\perp \end{bmatrix}^T$ that minimizes the Euclidian norm of an assumed zero mean measurement error (See Appendix B) is found using the pseudoinverse of the matrix A_G and the vector b_G just defined:

$$x_G = (A_G^T A_G)^{-1} A_G^T b_G \tag{5.6.3.1}$$

5.6.3.2 Procedure

As in Section 5.5.1.2, the first thing that must be done is to disable the gyroscope feedback in the GU-344 since the sensors it contains for angular velocity have significant bias and cause the PWM commands given to vary significantly even when the quadrotor is kept stationary. In order to be able to obtain useful readings of how each input affects each duty cycle output this feedback needs to be disabled. The procedure for disabling the gyroscope feedback is found in [25]. Essentially the *green* wire normally plugged into the *Gear* channel of the receiver is disconnected and the potentiometer on the GU-344 can be used to adjust the gyroscope feedback gain (in this model this parameter is represented by k_g, see Section 4.5) to essentially 0.

Next all ESCs are disconnected from the GU344 and using appropriate adaptors along with scope probes, each GU-344 output is connected to an oscilloscope input (in this case the 4-channel Tektronix DPO3034[43]). At this point at least one of the adaptors[16] needs to be connected through to an ESC in order for the GU-344 to receive power (each ESC has a BEC individually capable of providing enough power to the GU-344 unit). The corresponding motor should be disconnected or have the rotor removed.

Either a normal in-flight battery or appropriate power supply can be used to give power to the quadrotor through the normal input, and using the command-line base software (Table 5.2) provided with various integer command sets keeping the rudder channel at neutral. In this case each of the throttle, aileron, and elevator inptus was varied over its full range separately, keeping the throttle input near its middle value while changing the other channel inputs in order to avoid saturation issues. The set of collected data is reformed into the appropriate vectors and matrices as described in Section 5.6.3.1 and Equation 5.6.3.1 can be used to produce least-squares estimates of the throttle, aileron, and elevator channel gains.

---

[16] The *adaptors* mentoned here are made such that one end plugs into an GU-344 output and another end into an ESC input, with exposed sections along the middle of each wire for attaching measurement equipment like oscilloscope probes.

5.6.3.3 Results

The results of collecting and anlyzing the data as previously described were that:

$$\kappa_T = 1.99 \times 10^{-4}$$

$$\kappa_A = 4.3 \times 10^{-5}$$

$$\kappa_E = 4.3 \times 10^{-5}$$

$$\tilde{P}_{\perp} = 0.33$$

These results combined with those of Section 5.6.2 were tested by observing the transient response with the quadrotor attached to the ECP 220 as in Section 5.6.2 and later in Section 5.8.2. In Section 5.8.2 it is shown that the response to a step reference change give through the trainer port of the Dx6i transmitter differs from that given without it. In order to verify the parameters obtaine so far, the response to step inputs of different magnitude given without using the trainer port is compared to that predicted by the nonlinear simulation model, which was made to mimick being fixed to the encoder disk by having all velocity derivatives arbitrarily set to zero.

The resulting responses in Figure 5.10 were normalized for plotting comparison purposes, and a good match between the exerpimental data and simulation prediction was verified. The response for both the simulation and experimental response follow a similar curve and settle out in about 0.3s after the inital reference change.

5.6.4 Rudder Channel Gain Elements

The corresponding gains for the GU-344 rudder channel, κ_{R_P} and κ_{R_I} could not be obtained using the steady state observations that were sufficient for the throttle, aileron, and elevator

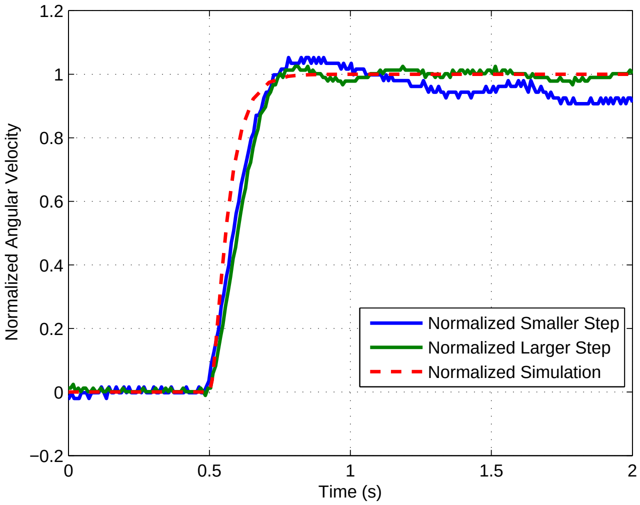


Figure 5.10 Pitch/Roll Rate Step Response: Experiment vs. Simulation

gain elements due to the dynamics of the integrator. With the quadrotor stationary, soon as any non-neutral input was given on the rudder channel, all four outputs of the GU-344 were pushed affected in such a way that they would never settle out, unless given enough time to completely saturate.[17]

To attempt to determine these parameters, while the quadrotor was stationary (meaning any angular velocity command would creat only error at the input to the channel control gains) the *initial value taken* to various magnitude step inputs was observed and taken to correspond to the proportional component of the PI gains, i.e κ_{R_P}. The integrator constant was first approximated roughly by taking the time for the outputs to saturate given varoius input commands and using the slope as an approximate value. The rough estimate was then iteratively refined by comparing experimental *yaw* trajectoris resulting from step inputs to the rudder reference channel of the initial PID closed loop (Section 6.1) with the predictions of nonlinear simulation.

The results obtained by these processes for the rudder channel gains were:

$$\kappa_{R_P} = 4.3 \times 10^{-5} \quad \kappa_{R_I} = 1.6 \times 10^{-4}$$

---

[17] Actually due to the presense of at least some small amount gyroscope drift feedback at all times (even with the potentiometer set to disable this feedback), the rudder channel integrator was probably always having some small unknown effect on all outputs that may have degraded the fit somewhat for the other channel gains.

5.7 Camera System Parameters

Parameters for the camera system, i.e. the cameras and all accompanying hardware are available from multiple sources including the hardware specifications available online and in the user's manual[48]. Those that were relevant to the modeling done here are given below.

5.7.1 Sampling Rate

The VR:100 cameras have a maximum frame rate of 100FPS which was the setting that each was used with for the experiments performed here. In symbolic terms, given as the period:

$$T_c = 0.01s$$

5.7.2 Latency

The latency of the camera system given here is the latency listed in the VR:100 technical specifications of 10ms plus 5ms to serve as an upper bound on the average latency given by Tracking Tools software during operation.

$$\tau_c = 0.015s$$

5.8 Communication System Identificaiton and Parameters

5.8.1 Communication Latency

The communication system latency for the throttle channel, τ_T was approximated by iteratively increaseing the simulated latency until the early response to a step command more closely matched that present in experimental data. This process is illustrated for a height step command in Figure 5.11.

The value that provided adequate similarity to the experimental data was:

$$\tau_T = 0.15s$$

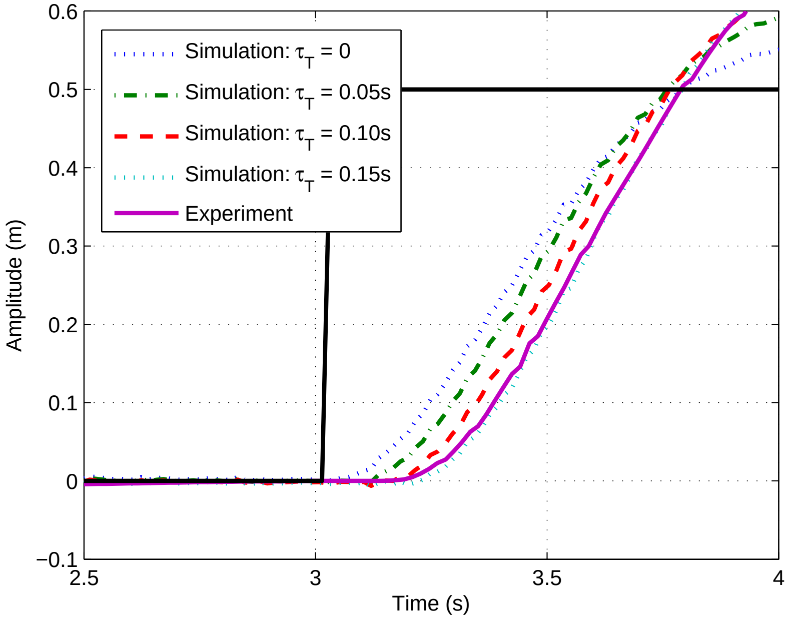


Figure 5.11 Various Communication System Latency Trials

A similar process was used to determine the other channel latencies, and the obtained approximate values are given below:

$$\tau_A = 0.15s \quad \tau_E = 0.15s \quad \tau_R = 0.15s$$

5.8.2 Channel Transfer Characteristics

Ideally the channel transfer elements of Equation 4.7.0.4, $G_{T_x}(s)$, $G_{A_x}(s)$, $G_{E_x}(s)$, and $G_{R_x}(s)$ would all be unity gain. This was approximately[18] the case for the throttle and rudder channels. Unfortunately however, due to an unknown problem in the system, this was not at all the case for the aileron or elevator channels when the trainer port (i.e. the way the FPGA sent signals from the control PC to the transmitter) of the Dx6i transmitter was used.

The abnormal behavior was observed[19] by provding the quadrotor with step input commands using the command-line based control software and recording the transient angular velocity response using the ECP 220 and Executive Software (Table 5.1). Essentially the setup was identical to that described in Section 5.6.2 with the difference in this case being that the transient response is sought rather than steady state amplitude. Figure 5.12 illustrates the observed response using the trainer port:

---

[18] Approximately because there would of course be some frequency response chracteristics e.g. finite bandwidth not modeled here.

[19] The problem was first qualitatively observed during the experiments to obtain the gyroscope feedback gains in Section 5.6.2. Luckily those paramters corresponded to steady state gains and their determination was not prevented by the behavior being described here.

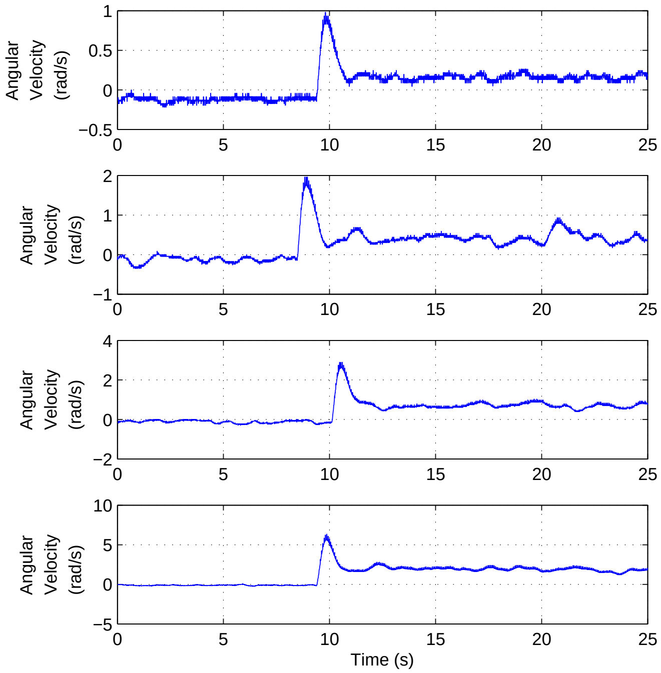


Figure 5.12 Aileron/Elevator Response through Trainer Port

What made this response odd, other than the obvious fact that it has a tremendous amount of overshoot and sluggish settling time, is that if the system was given a similar step input manually using the transmitter without the trainer port plugged in, the behavior disappeard. This is shown in Figure 5.13.

One potential difference between the manual and command-line control step inputs was speed. The command-line software produced its step over the interval of 0.5ms, while the manual input was likely not as fast. To look into the possibility that manually snapping the control stick up or down as quickly as possible was essentially just *not fast enough* i.e. lacking the bandwidth to excite the behavior for whatever reason, another transmitter was hooked up to the trainer port and a step input manually given again except through the trainer port. The

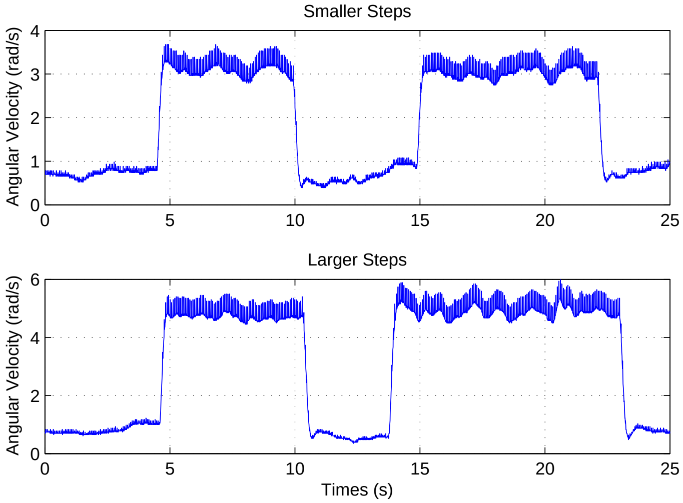


Figure 5.13 Aileron/Elevator Response without Trainer Port

abnormal, extreme overshoot, behavior occurred again with this setup, indicating that it was in fact some unknown issue with the communication through the trainer port of the Dx6i causing the change.

The output of the FPGA to the trainer port was observed using an oscillocope during trials to see if there was some abnormal behavior due to the FPGA (since the observed transient was slow enough to visually observe in the vehicle it would be slow enough to observe in the duty cycle percentage) but nothing abnormal was seen. This coupled with the fact that manual input was able to reproduce the behavior when using the trainer port adds even more indication that it is, however it may be, where the problem is caused (or at least a necessary element in the process of creating it).

The response of these two channels using the trainer port was therefore modeled as a non-unity transfer function. Observing on average approximately 275% overshoot and a peaking time of approximately 0.4 seconds, and starting from a form of third order transfer function similar to some found in [50], the following model for the resulting angular veloctiy system response was found to match the oberved behaviore reasonably well:

$$\frac{18.3s^2 + 159.2s + 102.5}{1s^3 + 14s^2 + 50.25s + 102.5}$$

The pole at $s = -10$ corresponding to a first order response with a time constant of $0.1s$ matches the approximate settling time of the angular rate without the transmitter issue, i.e. about $0.3s$. Hence the aileron and elevator channel transfer function was taken as the unity gain remainder with the first order portion–begin attributed to the vehicle physical dynamics in series–removed:

$$G_{A_x}(s) = G_{E_x}(s) = \frac{1.8304s^2 + 15.9241s + 10.25}{1s^2 + 4s + 10.25}$$

Both the system respose and resulting modeled communication response are shown

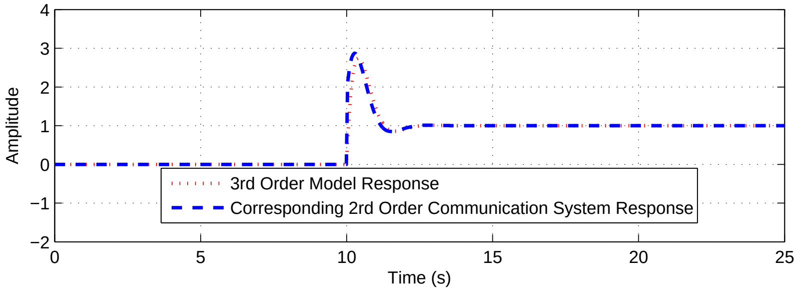


Figure 5.14 Aileron/Elevator Channel Modeled Transfer Function Step Response

The response of the transfer function for the aileron and elevator channels to a unit step input is depicted in Figure 5.14. It can be seen to compare very well to those of Figure 5.12 (the axis scaling matches the third sub-plot of that figure best).

5.9 Collective Results

Table 5.3: Open Loop Numerical Parameters

Symbol	**Nominal Value**	**Units**	**Brief Description**
m_q	0.456	kg	quadrotor mass
m_b	0.200	kg	battery mass
m	0.656	kg	quadrotor+battery mass

Continued on next page...

Table 5.3 –*Continued from previous page.*

Symbol	**Nominal Value**	**Units**	**Brief Description**
g	9.81	m/s^2	acceleration of gravity
J_{xx}	8.1×10^{-3}	kgm^2	quadrotor+battery moment of inertia around b_x
J_{yy}	7.4×10^{-3}	kgm^2	quadrotor+battery moment of inertia around b_y
J_{zz}	13.5×10^{-3}	kgm^2	quadrotor+battery moment of inertia around b_z
$J_{r_{eq}}$	1.376×10^{-5}	kgm^2	rotor+motor m.o.i. around motor axis of rotation
K_T	7.2803×10^{-6}	$\frac{Kgm}{rad^2}$	rotor thrust constant
K_H	3.4574×10^{-4}	$\frac{kg}{rad}$	rotor in-plane drag constant
K_d	5.1994×10^{-7}	$\frac{kgm^2}{rad^2}$	rotor drag constant
δ_{T_z}	1.176×10^{-4}	$\frac{kg}{rad}$	rotor velocity thrust adjustment factor
$\tilde{J}_r$	1.376×10^{-5}	kgm^2	rotor drag constant
$\lvert r_{h_x} \rvert$	0.115	m	x-axis distance from center of mass to a rotor hub
$\lvert r_{h_y} \rvert$	0.115	m	y-axis distance from center of mass to a rotor hub
$\lvert r_{h_z} \rvert$	0.04	m	z-axis distance from center of mass to a rotor hub
R_m	0.19	Ω	motor resistance
K_Q	110	$\frac{A}{Nm}$	motor torque constant
K_V	110	$\frac{rad}{Vs}$	motor back-emf constant
i_f	0.39	A	motor internal friction current
$P_{\perp}$	0.29	(none)	ESC turn-on duty cycle command
$\tilde{P}_{\perp}$	0.3277	(none)	minimum GU-344 output duty cycle command
$P_{\top}$	0.55	(none)	maximum GU-344 output duty cycle command
δ_V	-0.002	V/s	approximate constant battery discharge rate
k_g	1	(none)	gyroscope gain
γ_p	330.44	s	gyroscope rad/s-to-integer feedback gain on p
γ_q	330.44	s	gyroscope rad/s-to-integer feedback gain on q
γ_r	192.68	s	gyroscope rad/s-to-integer feedback gain on r
κ_T	1.99×10^{-2}	(none)	throttle channel proportional gain

Continued on next page...

Table 5.3 –*Continued from previous page.*

Symbol	**Nominal Value**	**Units**	**Brief Description**
κ_A	4.3×10^{-3}	(none)	aileron channel tracking proportional constant
κ_E	4.3×10^{-3}	(none)	elevator channel tracking proportional constant
κ_{R_P}	4.3×10^{-3}	(none)	rudder channel tracking proportional constant
κ_{R_I}	5.33×10^{-4}	(none)	rudder channel rate tracking integral constant
$\tilde{P}_\perp$	0.3277	(none)	minimum active duty cycle command output
T_c	0.01	s	camera system sampling period
τ_c	0.015	s	camera system total latency
τ_T	0.15	s	communication throttle channel latency
τ_A	0.15	s	communication aileron channel latency
τ_E	0.15	s	communication elevator channel latency
τ_R	0.15	s	communication rudder channel latency

CHAPTER 6. Nested-Loop PID Control

6.1 Initial Stabilizing Design

Proportional-integral-derivative (PID) control–or some variant–is almost certainly the most common type of linear control design utilized. The action of each term in a PID is fairly intuitive, and PID control also has the advantage of being able to be effectively tuned for performance in many cases without a detailed (or sometimes any) plant model. Various manual tuning methodologies and heuristics exist such as the ZieglerNichols method, as well as various software implementations allowing for automatic tuning given a linear plant model and some design constraints (e.g. the *pidtool* functionality of MATLAB and related Simulink blocks).

The nested loop PID control architecture used here, depicted in Figure 6.1, was designed while working in cooperation with the MicroCART 2012 senior design team.

At the time of the intial control experiments, the nonlinear model was not complete and all parameters not yet determined. Because of this, the model development was completed in parallel with the PID type control design. An iterative design approach was adopted where initial control was designed based on the existing model–however complete at a given time–and the observations and data obtained used to produce improvements to the model, which could then be used for further control design.

This approach was beneficial in this case. Both the control design and modeling processes were mutually aided at various stages. For example, initial modeling errors in the description of in-plane drag were exposed and fixed using experimental data obtained from early control

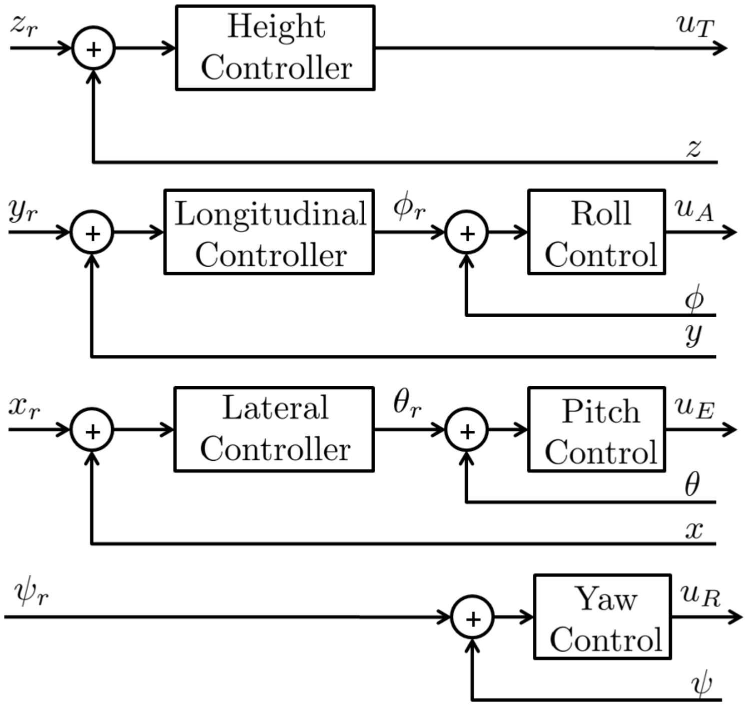

z_r
+
Height Controller
u_T
z
y_r
+
Longitudinal Controller
ϕ_r
+
Roll Control
u_A
ϕ
y
x_r
+
Lateral Controller
θ_r
+
Pitch Control
u_E
θ
x
ψ_r
+
Yaw Control
u_R
ψ

Figure 6.1 Nested Loop PID Architecture

experiments. In the opposite direciton, at on point a significant bug with a portion of the control software was discovered with the aid of the model. Various controller errors were simulated and the resulting predicted degradation in behavior was compared to the actutal behavior, with one of the closest determined *possible sources* being the actual problem.

The control gains for each channel were obtained over time in cooperation with the MicorCART team, using a mixture of experimental trials with manual tuning and simulation based design and redesign. The control software was designed to allow a mixture of manual and autonomous control. One or more input channels could be placed under autonomous control while the others remained under the control of a human pilot.

6.2 Heading Control

Yaw or heading control was the first to be designed in order to maintain a steady–initially always zero–heading. The modeling with respect to the yawing dynamics was essentially complete during by the time of initial PID design. Based on the model linearized at hover, and testing with the nonlinear simulation, the control gains in Equation 6.2.0.1 were obtained and did not require subsequent manual tuning.

$$K_{P_\psi} = 408 \quad K_{D_\psi} = 19.6 \quad \tau_{f_\psi} = 0.468 \tag{6.2.0.1}$$

Where K_{P_ψ} is the proportional gain, K_{D_ψ} is the derivative gain, and τ_{f_ψ} is the equivalent first order low-pass filter time constant applied to the derivative input. The results of this control are shown in Figure 6.2.

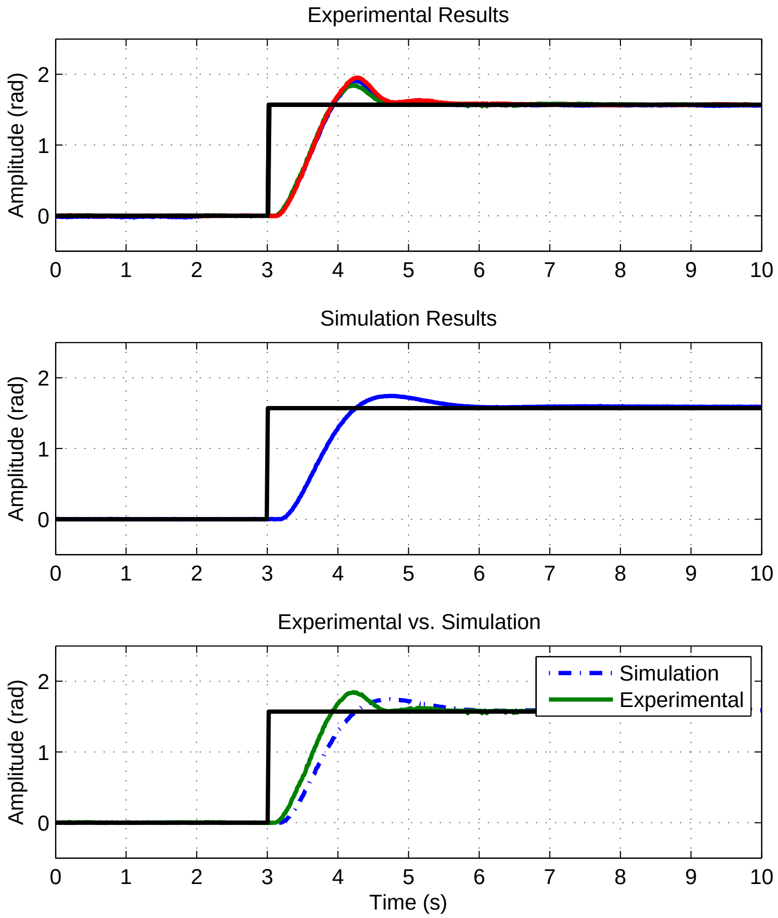


Figure 6.2 PID Heading Step Commands

The experimental response to a yaw/heading reference change follows the simulation prediction well. The simulation predicts a somewhat slower response with slightly less overshoot. One potential source of this discrepancy is the fact that at times, due to some time varrying bias in the GU-344 unit output, the assumed *neutral* rudder input produced a non-zero yawing velocity. The results were however deemed satisfactory.

6.3 Position Control

The architecture takes advantage of the fact that in order to create accelerations in the lateral or longitudinal directions the quadrotor must first change orientation, i.e. it must tip the entire fuselage towards the desired direction of linear motion. This allowed the resulting control loops for the aileron and elevator channels to be designed in two stages. The roll and pitch orientation control was designed first, followed by the lateral and longitudinal control.

6.3.1 Orientation Control

Due to the existing gyroscope feedback of the GU-344, the aileron, elevator, and rudder inputs to the open loop system act effectively as angular rate references. With the addition of a first layer of control for the orientation angles along with the previously designed yaw/heading control, these inputs are given by a angular control layer with angle reference inputs.

For the orientation control an initial estimate proportional value was determined based on the at-the-time-existing model and tuned based on trial and observation. It was found that proportional only control yielded adequate results[1]:

$$K_{P_{roll}} = 251$$

Where K_{P_ϕ} is the roll control proportional gain.

$$K_{P_{pitch}} = 251$$

Where K_{P_θ} is the pitch control proportional gain.

---

[1] This is entirely due to the existing gyroscope angular rate feedback. Without this feedback the system from input to roll and pitch output would act effectively as a double integrator in series with other dynamcs, which is a system that cannot be stabilized by *only* proportional control.

6.3.2 Lateral and Longitudinal Control

Following the nested-loop architecture, the control for the lateral and longitudinal positions was in the form of a closed loop around each of the existing orientation controls. The subsequent design of these controls was predominantly experimental, producing the following proportional-derivative control:

$$K_{P_x} = 0.39 \quad K_{D_x} = 0.05 \quad \tau_{f_x} = 0.01$$

$$K_{P_y} = 0.39 \quad K_{D_y} = 0.05 \quad \tau_{f_y} = 0.01$$

The results of these gains are shown in Figure 6.3 and Figure 6.4.

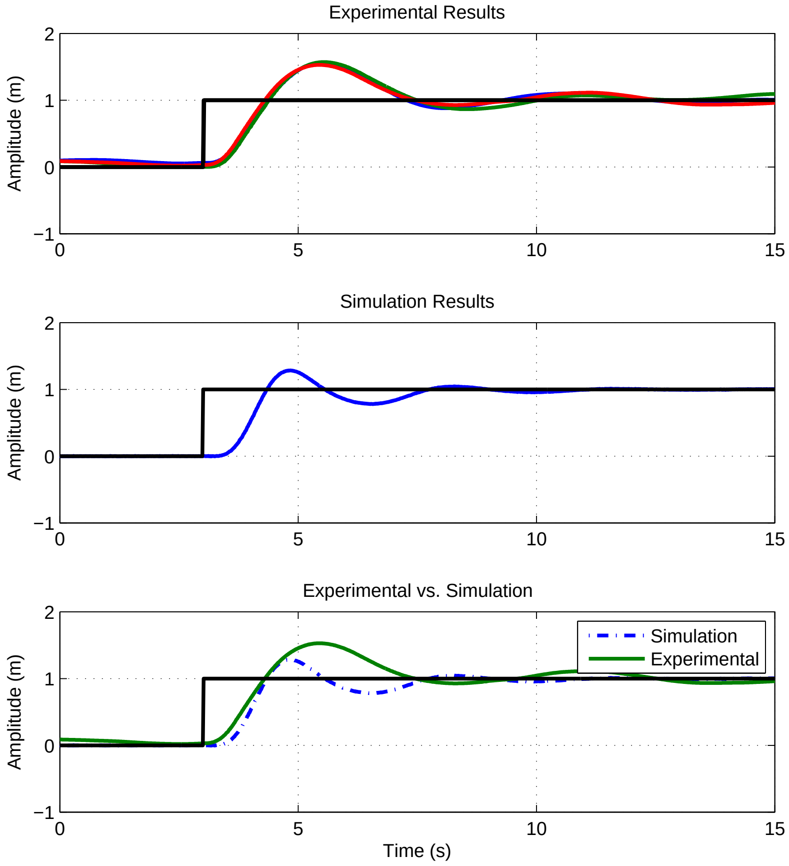


Figure 6.3 PID Lateral Step Commands

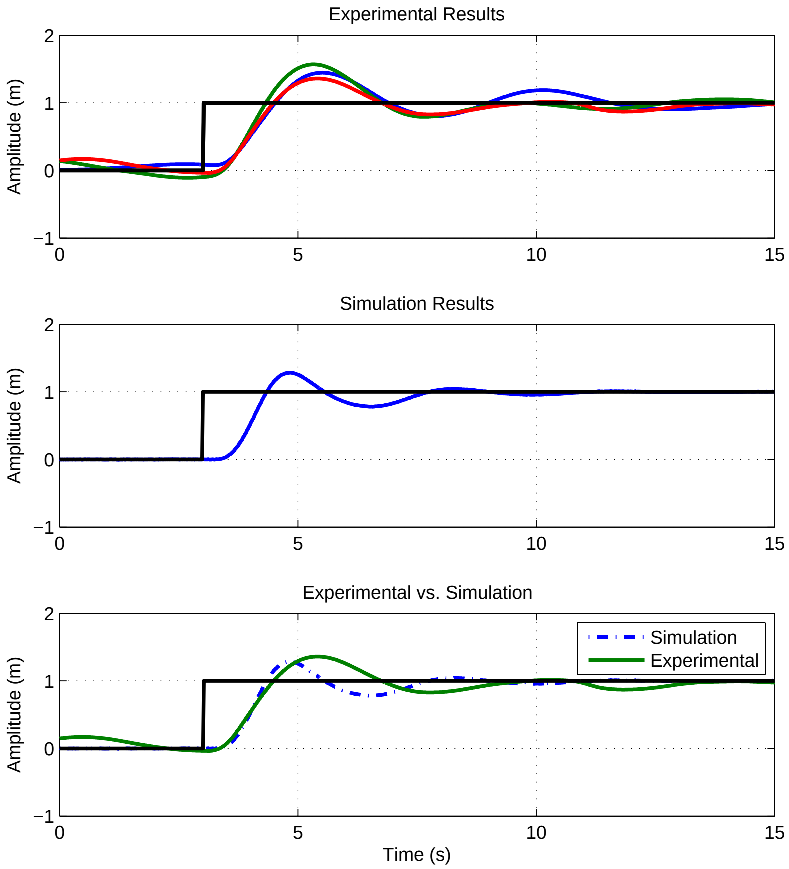


Figure 6.4 PID Longitudinal Step Commands

The response to a lateral and longitudinal reference step inputs was good, though it deviated somewhat more significantly from the simulation predictions than the heading control. The initial response is a good match but the simulation predicts a sharper peak after the initial overshoot in both cases. This is most likely due to the communication system effects described in Section 5.8.2, which were isolated to the aileron and elevator channels, and hence would only affect the lateral and longitudinal response.

Though the observed behavior was given an approximate linear model, the collected data (Figure 5.12) shows some indication of nonlinearity, i.e. varrying amplitude overshoot depending on input magnitude. Also, only isolated angular rate step response tests were used to obtain the approximate model. It is quite possible the model used to describe the communication behavior on these channels was not adequate to capture the behavior during more complicated in-flight maneuvers.

The accuracy with which the simulation model follows the physical response of the system to angular velocity step commands in Figure 5.10, i.e. when the trainer port communications could be avoided, would seem to indicate that whatever unmodeled dynamics cause this mismatch are not part of the physical dynamics of the vehicle, but rather some other portion of the system. Further testing, e.g. substitution of another communication system, would be necessary to firmly establish whether or not the aileron/elevator isssue is the cause of the model/experimental mismatch observed.

6.3.3 Height Control

Height control was designed based primarily on the model available at the time of PID design, which though it was not yet complete still yielded a resulting controller with adequate performance without any subsequent manual tuning.[2] The PID gains used for height control are as follows:

$$K_{P_z} = -44.076 \quad K_{I_z} = 0.7777 \quad K_{D_z} = -139.3324 \quad \tau_{f_z} = 0.2368$$

Where K_{P_z} is the proportional gain, K_{I_z} is the integrator gain, K_{D_z} is the derivative gain, and finally τ_{f_z} is the time constant of a first order low pass filter applied to the error seen by the derivative term. The results of this controller are shown in Figure 6.5.

The results of the height control simulation are nearly identical to the experimental results obtained using the previously described height controller. This gives further confirmation to the accuracy of the nonlinear physical dynamical model since the throttle channel was unaffected by any communication issues other than the determined latency (Figure 5.11). Notice the accuracy of the response depicted in Figure 6.5 is dependent upon not only the accurate determination of the typical constant-times-rotor-speed-squared parameter (K_T, see Sections 4.2.1.1 and 5.5.1),

---

[2] The linearized model used, whether complete or not, is unable to take into account the continuual loss of battery charge and hence manually increasing the integrator gain would help guarantee consistent height control over longer flights.

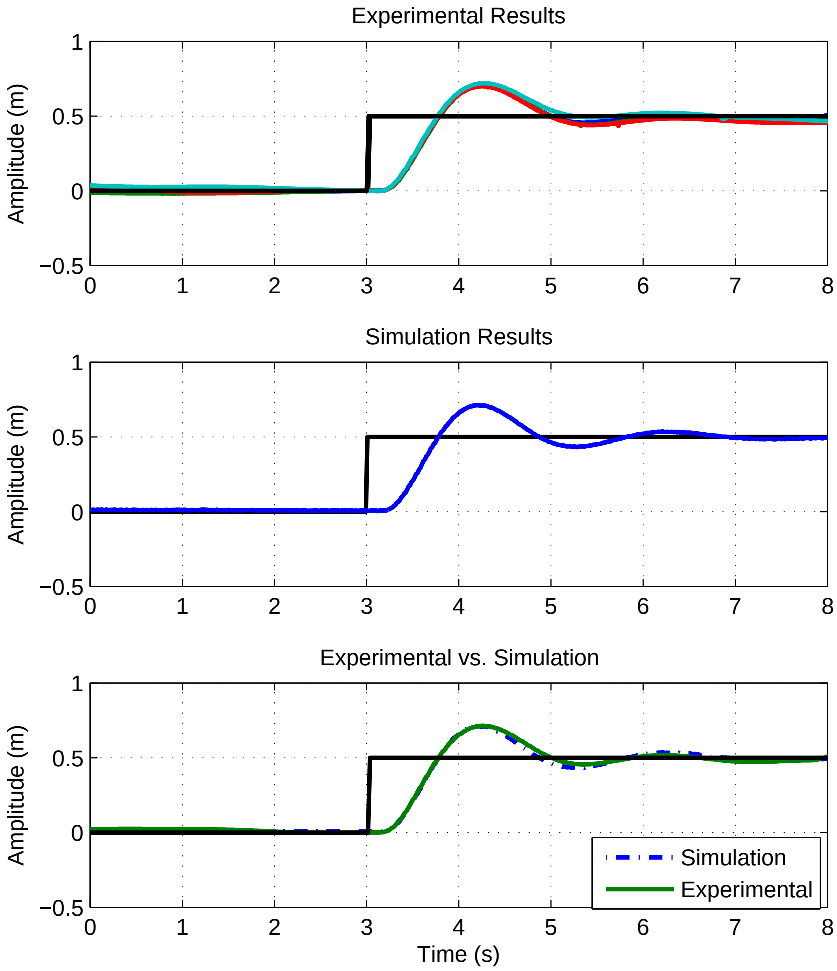


Figure 6.5 PID Height Step Commands

but also the thrust velocity adjustment factor δ_T (Section 5.5.2). The resulting less accurate model prediction without this adjustment is shown in Figure 5.7.

6.4 Pseudo-Nonlinear Extension

In order to allow the system to operate over a continuum of yaw angles, the position errors given to the control system were adjusted so as to account for the current vehicle heading. In order to allow the linear controllers that had been designed to operate at a yaw angle, $\psi = 0$, to function effectively at non-zero angles, the errors in inertial frame of reference position were multiplied by a change of coordinates matrix to project them onto the local body frame axes of the vehicle. That is, the errors given to the controller are:

$$\begin{bmatrix} \tilde{x}_e \\ \tilde{y}_e \\ \tilde{z}_e \end{bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \tag{6.4.0.1}$$

Where x_e, y_e and z_e are the intertial frame of reference errors based on the current vehicle position and user reference commands, and of course ψ is the yaw/heading angle.

In this way, when the yaw angle of the vehicle is changed, the controller effectively works as though the qaudrotor still has zero yaw, and what has happend is the reference position has been rotated by a given angel equal to the non-zero ψ. In a sense this is *fooling* the linear controller to behave in a semi-nonlinear way, hence calling it a pseudo-nonlinear extension.

With such an adjustment the quadrotor was able to hover at any desired heading, as well as execute various trajectories at various non-zero angles. Although the performance when at at angles other than 0 was slightly degraded, any sort of performance at all at large angles away from 0 would have been completely impossible with the static linear system.

CHAPTER 7. LQR Control

Linear quadratic optimal control is a method of control design that seeks to minimize a quadratic cost function of the system states and inputs with the constraint that the system dynamics are governed–at least locally–by a system of linear differential equations, i.e. a linear state space model. Unlike the case of PID control where the controller gains are chosen directly and not necessarily based on a model, LQ design uses a cost function chosen by the designer which is minimized subject to the dynamics of a specific system model.

Optimal control design in general always presents potential difficulties in that the designer must choose some objective or objectives to optimize based on an available model. Because of this the control that results, though *optimal in the sense of the model and chosen criteria*, may suffer in actual performance if the model used or the chosen optimization criteria is not adequate with respect to the real system. Linear Quadratic Regulator (LQR) design–in which a set of matrices is chosen to *weight* the system states and inputs in a cost to be minimized subject to the modeled system dynamics–is no exception.

7.1 LQR Weight Selection

As part of LQR design, the weight matrices must be chosen such that the resulting optimal control gains produce desirable closed loop system performance that is feasable taking into account the real-world system's capabilities and dynamics. In many cases this is accomplished by selecting diagonal matrices to weight both the states and inputs, checking the perfomance of the resulting control (either in simulation or through experimentation if possible), and iteratively adjusting the weight terms and redesigning the controller to improve performance based on the observed results e.g. [3].

One of the advantages of selecting diagonal weight matrices is that it effectively means each state element and each input element is weighted individually. This makes it easier for the designer to have some intuition about how any given value in a weighting matrix will affect the optimizaton process and the resulting control. Heuristics exist for selecting the initial weighting elements such as Bryson's rule in [8, 11] or similar methods focusing on acutator saturation as in [9].

Another slightly more subtle advantage associated with selecting diagonal weighting matrices is that in order to guarantee the cost function is convex, which is necessary for the optimization problem to be well posed, the designer only needs to ensure that each element in the diagonal state weighting matrix is nonnegative and each element in the diagonal input weight is positive. This will guarantee the state weighting matrix is positive semidefinite, and that the input weighting matrix is positive definite, making the resulting cost convex[1].

The main potential disadvantage of the diagonal weights approach is directly related to the first advantage. In forming the cost by taking the sum of positively weighted individual states and inputs, the formation of the weight matrices can be simplified, but this to a large extent destroys the ability of the cost function to take into account coupling or interactions between different states, between different inputs, or between inptus and states.

The quadrotor system of this thesis has multiple states and inputs corresponding to potentially conflicting degrees of freedom. Due to the physics of the system in order to produce lateral or longitudinal acceleration the orientation much be changed first. This means that the rotaional and linear degrees of freedom themselves are interdependent. For example regulation to the origin in the form of correcting errors in the linear states requires manipulation of the angular states away from the origin. These sort of issues were brought up in [3] in relation to nested-loop PID vs. single loop LQR control (designed using the typical diagonal weight structure cost function) architectures.

---

[1] These positive semidefinite/definite conditions are required of deterministic systems, not stochastic systems e.g. [51, 52]

It is desirable to be able to easily form cost matrices with more complicated structure (i.e. not necessarily diagonal) in order to account for interaction among states and among inputs, and also to weight the state/input interactions. For most systems this, especially the input/state weighting matrix, would be very difficult to select on an element-by-element basis as with the digaonal matrix approach and maintain cost convexity. For this reason some systematic procedure is necessery to form a better cost function while maintaining convexity.

In what follows, similar LQR design issues to those described in [3] are encountered, and a systematic procedure for forming the cost function taking into account the specific structure and dynamical constraints of the quadrotor system is presented, and used to produce superior results to those obtained using the more typical approach.

7.2 Linear Open Loop Model for LQR Design

It was desirable to keep the model used for state feedback control design simple while capturing the dynamics enough to facilitate adequate results. In other words, to use the smallest set of states that would yield good resulting control, since the inclusion of states that are very difficult to obtain accurately or affect would only serve to add complexity and uncertainty to the resulting system, thereby reducing robustness. The linear model used for the design of the Linear Quadratic Regulator state feedback was simplified in order to remove states that could not be observed by any available means, could not easily be estimated, and met one or both of the following criteria:

- Could not be regulated or had no need of regulation.
- Had only a transient, and/or *relatively* small effect on the important system dynamics.

The resulting simplified linear model did not have states for the rotor speeds. There was no means available in the system used to measure the motor/rotor speeds during flight, and estimation would have been extremely difficult due to the large operating range and dependence on time varrying unknowns such as battery voltage. The motor dynamics were also relatively fast relative to those of the vehicle motion. For these reasons Equation 4.2.6.3 was used to eliminate the rotor speed states before linearization.

The state representing the accumulation of the integrator on the rudder channel of the GU-344 (Equation 4.5.3.2) also had absolutely no means of measurement available. Though estimation would probably not have been tremendously difficult, the yaw dynamics of the system were relatively well behaved, and furhter it was not desirable or very feasable to regulate the value of this state to a reference value since it would potentially conflict with overall yaw/heading regulation.

The model for the aileron and elevator channel communication system transfer behivor of Sections 4.7 and 5.8.2 involved several states. Again these states could not be measured by any available means, and since the model given to the behavior was a very rough linear approximation, estimation during flight would not have been very feasable. Beyond this there would have been no way to directly affect its state values. Though not an ideal situation, for the purposes of state feedback control *design* the communication system states were ignored.

With these states removed, the linear model used to design the control gains lost some level of accuracty. Due to this, any subsequent control obtained based on the simplified linear model was tested in full-complexity nonlinear simulation (i.e. including the communication behavior, rudder integration and motor dynamics) to help ensure some adequate level of performance before any implementation. This was done based on the previously obtained results using PID control and other model verification experiments that indicated a fairly strong match between simulation predictions and implemented results[2].

The resulting simplified nonlinear open loop system, like the full complexity version, had an equilibrium point at $\Lambda = 0$, corresponding to hovering[3]. It was linearized at this equilibrium to produce the linear state space model:

$$\frac{d}{dt}\Lambda = A\Lambda + B\nu$$

---

[2] With the only somewhat significant exception being the lateral/longitudinal transient behavior affected by the communications issue (Section 6.3.2).

[3] Actually it has a continuum of equilibrium points where all velocity states and the pitch and roll are zero, with the linear position and yaw angle free to take any value.

Where $\nu = u_s - \bar{u}_s$, with $\bar{u}_s$ being the equilibrium input. It is not necessary to define a new state since the equilibrium state for this circumstance is the origin. The A and B matrices with numcerical parameters subsituted are as follows:

$$A = \begin{bmatrix} -0.991 & 0 & 0 & 0 & 0.03964 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 \\ 0 & -0.991 & 0 & -0.03964 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 \\ 0 & 0 & -0.6739 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3.211 & 0 & -20.83 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3.514 & 0 & 0 & 0 & -22.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -5.616 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.03005 & 0 & 0 & 0 \\ 0 & 0.06048 & 0 & 0 \\ 0 & 0 & 0.0662 & 0 \\ 0 & 0 & 0 & 0.02253 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

7.2.1 Subsystem Decomposition

The resulting linear system of Section 7.2 can be formed into four independent subsystems. These four subsystems will be termed the longitudinal, lateral, altitude, and direction subsystems. This necessitates defining the decoupled equivalent state vector, $\tilde{\Lambda}$ as:

$$\begin{bmatrix} \Lambda_{alt} \\ \Lambda_{lat} \\ \Lambda_{lon} \\ \Lambda_{dir} \end{bmatrix}$$

Where each of the components of the equivalent state vector represents the states corresponding to that subsystem:

$$\Lambda_{lon} = \begin{bmatrix} u \\ x \\ q \\ \theta \end{bmatrix} \quad \Lambda_{lat} = \begin{bmatrix} v \\ y \\ p \\ \phi \end{bmatrix} \quad \Lambda_{alt} = \begin{bmatrix} w \\ z \end{bmatrix} \quad \Lambda_{dir} = \begin{bmatrix} r \\ \psi \end{bmatrix}$$

So the equivalent open loop linear system can be written:

$$\frac{d}{dt}\tilde{\Lambda} = \tilde{A}\tilde{\Lambda} + \tilde{B}\nu$$

With the equivalent block diagonal matrices $\tilde{A}$ and $\tilde{B}$ as:

$$\tilde{A} = \begin{bmatrix} A_{alt} & \cdots & \cdots & [0] \\ \vdots & A_{lat} & \ddots & \vdots \\ \vdots & \ddots & A_{lon} & \vdots \\ [0] & \cdots & \cdots & A_{dir} \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} B_{alt} & \cdots & \cdots & [0] \\ \vdots & B_{lat} & \ddots & \vdots \\ \vdots & \ddots & B_{lon} & \vdots \\ [0] & \cdots & \cdots & B_{dir} \end{bmatrix}$$

Substituting the system parameters, the subsystem matrices take the following forms:

$$A_{alt} = \begin{bmatrix} -0.6739 & 0 \\ 1 & 0 \end{bmatrix} \qquad B_{alt} = \begin{bmatrix} -0.03005 \\ 0 \end{bmatrix}$$

$$A_{lat} = \begin{bmatrix} -0.991 & 0 & -0.03964 & 9.81 \\ 1 & 0 & 0 & 0 \\ -3.211 & 0 & -22.67 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B_{lat} = \begin{bmatrix} 0 \\ 0 \\ 0.06048 \\ 0 \end{bmatrix}$$

$$A_{lon} = \begin{bmatrix} -0.991 & 0 & 0.03964 & -9.81 \\ 1 & 0 & 0 & 0 \\ 3.514 & 0 & -22.81 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B_{lon} = \begin{bmatrix} 0 \\ 0 \\ 0.0662 \\ 0 \end{bmatrix}$$

$$A_{dir} = \begin{bmatrix} -5.616 & 0 \\ 1 & 0 \end{bmatrix} \qquad B_{dir} = \begin{bmatrix} 0.02253 \\ 0 \end{bmatrix}$$

7.3 Initital LQR Design

Here the initial attempts to design a LQR controller, following a commonly encoutnered method, are described.

7.3.1 Common Form LQR Design

Linear quadratic regulation is by no means a novel method of control applied to quadrotor helicopters. It has been done in for example [8, 3, 7, 9, 10, 11, 12]. In *all cases the design was performed using the common cost function form*:

$$J = \int_0^\infty x^T Q x + u^T R u \; dt \tag{7.3.1.1}$$

Or the equivalent for the discrete case:

$$J = \sum_{n=0}^{\infty} x^T Q x + u^T R u \tag{7.3.1.2}$$

With Q and R diagonal, and positive semidefinite and definite respectively. This was the form of cost function used in the initial LQR designs used in this thesis. The A and B matrices of Section 7.2 were used along with diagonal weighting matrices chosen using methodology similar to [8, 11].

The structure of the weights used was as follows:

$$Q = \begin{bmatrix} \frac{1}{\hat{u}^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\hat{v}^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\hat{w}^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\hat{p}^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\hat{q}^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{r}^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{x}^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{y}^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{z}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{\phi}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{\theta}^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\hat{\psi}^2} \end{bmatrix} \quad R = \begin{bmatrix} \frac{1}{\hat{u}_T^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\hat{u}_A^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\hat{u}_E^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\hat{u}_R^2} \end{bmatrix}$$

Where each *hat* state and input value corresponds to a maximum desireable amplitude relative to the other states and inputs respectively, to be iteratively adjusted based on acheived results until a satisfactory design is acheived.

Following this process for the quadrotor system here, a large nubmer of weight magnitude combinations were chosen in an attempt to design a satisfactory LQR feedback matrix. One example would be setting all the velocity states maximum amplitude to as well as the pitch and roll angles to unity, the x and y weights to $\frac{1}{\sqrt{0.1}}$ and the weights for z and ψ to $\frac{1}{\sqrt{0.01}}$, the throttle amplitude 50, aileron and elevator to 5 and rudder to 10. These gains produced a stable control loop, but one very lacking performance in x and y tracking, taking 10 or more seconds to track step command on either reference in simulation and faring even worse in implementation due to an inability to reject disturbances.

The state feedback gain matrix resulting from these example weights was:

$$K = \begin{bmatrix} 0 & 0 & -165.75 & 0 & 0 & 0 & 0 & 0 & -487.54 & 0 & 0 & 0 \\ 0 & 9.49 & 0 & 12.61 & 0 & 0 & 0 & 40.95 & 0 & 267.98 & 0 & 0 \\ -9.49 & 0 & 0 & 0 & 11.5 & 0 & -40.95 & 0 & 0 & 0 & 267.06 & 0 \\ 0 & 0 & 0 & 0 & 0 & 17.38 & 0 & 0 & 0 & 0 & 0 & 99.8 \end{bmatrix}$$

The problem essentially boiled down to the fact that the gains on the x and y states (approximately 41 in this case) were simply not large enough to provide adequate tracking and disturbance rejection, and with all the combinations of diagonal state and input weight magnitudes tried (e.g. increasing the weights on x and y while decreasing those on the angles and or rates etc...), any that produced gains for x and y that were high enough[4] to accomplish regulation on those variables produced gains for ϕ and θ in a range high enough (in the 600s) to drive the real system unstable.

The same type of problem had been encountered and mentioned in [3], where a nested-loop PID control architecture similar to that used here was compared to a single loop LQR architecture and the LQR performance was found lacking due to conflict amog the degrees of freedom that it was attempting to simultaneously regulate as opposed to the nested loop architecture where the linear and angular degrees of freedom are in sequence. In [3] an attempt was made to use the nested-loop idea with LQR design but it did not meet with much success. Here a different means of dealing with the issue is proposed, taking the conflicing degrees of freedom and inputs directly into account when forming the cost function.

7.4 Improved LQR Design

7.4.1 H_2 Control

LQR control can be veiwed as a special case of H_2 optimal control[53]. In order to do so, defining the output quantity ζ as:

$$\zeta = Cx + Du$$

---

[4] It is hypothetically possible that the set of diagonal weights that was *just right* to produce desirable control gains was not found even with all the iterations tried, but even in that case the difficult and impractical nature of such a searching process is reason enough to seek out a better method.

A cost function can be formed (given in the discrete case here):

$$J = \sum_{n=0}^{\infty} \zeta^T \zeta$$

Where:

$$\zeta^T \zeta = (Cx + Du)^T (Cx + Du) = x^T C^T Cx + x^T C^T Du + u^T D^T Cx + u^T D^T Du$$

If the output matrices C and D are chosen such that:

$$C = \begin{bmatrix} Q^{\frac{1}{2}} \\ [0] \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} [0] \\ R^{\frac{1}{2}} \end{bmatrix}$$

Where $Q^{\frac{1}{2}}$ and $R^{\frac{1}{2}}$ are the square roots of the positive semidefinite weighting matrices[5] in the typical LQR cost function, then $C^T D = 0$ and $D^T C = 0$, hence:

$$J = \sum_{n=0}^{\infty} x^T Qx + u^T Ru$$

The key here is recognizing that the matrices C and D were chosen to have orthogonal columns arbitrarily. There is no reason why this has to be the case. The C and D matrices can be chosen to represent linear combinations of inputs and states, including those that may have conflit during regulation, in ways that allow the states and inputs to diverge from the origin under appropriate conditions. For example, if one state takes on a non-zero value, and another state and a corresponding input must take a non-zero values in order to allow the system to correct itself, the matrices can be set up so that the cost function reflects the weighted difference of these quantities in a way that they cancel eachother appropriately.

Noting that $x^T C^T Du = u^T D^T Cx \Rightarrow x^T C^T Du + u^T D^T Cx = 2x^T C^T Du$, the more general form of cost function is then:

$$J = \sum_{n=0}^{\infty} x^T C^T Cx + 2x^T C^T Du + u^T D^T Du \qquad (7.4.1.1)$$

Which has elements corresponding to the MATLAB *lqrd* command, where using the syntax of that functions documentation $Q = C^T C$, $N = C^T D$, and $R = D^T D$.

---

[5] Q and R being positive semidefinite guarantees the existence of a square root for each[53]

7.4.2 Design Accounting for Degree of Freedom Conflict

The use of the form of cost function involving non-orthogonal columns in the output matrices is illustrated in what follows for the lateral and logitudinal subsystems described in Section 7.2.1.

7.4.2.1 Lateral Subsystem

The lateral subsystem's state equation matrices and input and state vector:

$$A_{lat} = \begin{bmatrix} -0.991 & 0 & -0.03964 & 9.81 \\ 1 & 0 & 0 & 0 \\ -3.211 & 0 & -22.67 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B_{lat} = \begin{bmatrix} 0 \\ 0 \\ 0.06048 \\ 0 \end{bmatrix} \qquad \nu_A \quad \Lambda_{lat} = \begin{bmatrix} v \\ y \\ p \\ \phi \end{bmatrix}$$

The output function for the lateral subsystem needs to be defined such that it takes account of the conflicting input and degrees of freedom. Using the fact that y is positive *to the right*, and ϕ is negative *banking to the left* and hence the *away from equilibrium* aileron input ν_A would be negative to correct for this lateral error, noting that the same relationships hold for the rates, the output function for this subsystem is defined as:

$$\zeta_{lat} = c_{lat}\Lambda_{lat} + d_A\nu_A$$

$$\zeta_{lat} = \begin{bmatrix} c_v & c_y & c_p & c_\phi \end{bmatrix} \begin{bmatrix} v \\ y \\ p \\ \phi \end{bmatrix} + d_A\nu_A$$

Where the sign on all the terms is kept positive in this case to reflect the fact that it is desirable that if the lateral posision is in error the aileron and roll input be allowed to be negative and still keep the cost minimal, and vice-verca.

The values that would end up producing good results in this case were:

$$c_{lat} = \begin{bmatrix} \frac{1}{2} & \frac{1}{0.6} & \frac{1}{25} & \frac{1}{0.25} \end{bmatrix} \qquad d_A = \frac{1}{60}$$

7.4.2.2 Longitudinal Subsystem

The longitudinal subsystem's state equation matrices and input and state vector:

$$A_{lon} = \begin{bmatrix} -0.991 & 0 & 0.03964 & -9.81 \\ 1 & 0 & 0 & 0 \\ 3.514 & 0 & -22.81 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} B_{lon} = \begin{bmatrix} 0 \\ 0 \\ 0.0662 \\ 0 \end{bmatrix} \quad \nu_E \quad \Lambda_{lon} = \begin{bmatrix} u \\ x \\ q \\ \theta \end{bmatrix}$$

The same reasoning was applied to the longitudinal subsystem as the lateral subsystem. The only difference being that in the case of the longitudinal system, the pitch angle and corresponding elevator input need to be positive to correct for positive longitudinal position error and vice-verca. Hence the output matrices for is system were:

$$c_{lon} = \begin{bmatrix} \frac{1}{2} & \frac{1}{0.6} & \frac{-1}{25} & \frac{-1}{0.25} \end{bmatrix} \quad d_A = \frac{-1}{60}$$

7.4.2.3 Altitude Subsystem

The altitude systems variables were:

$$A_{alt} = \begin{bmatrix} -0.6739 & 0 \\ 1 & 0 \end{bmatrix} B_{alt} = \begin{bmatrix} -0.03005 \\ 0 \end{bmatrix} \quad \nu_T \quad \Lambda_{alt} = \begin{bmatrix} w \\ z \end{bmatrix}$$

The linear altitude subsystem did not have the kind of degree of freedom conflict that the lateral and longitudinal subsystems had, therefore its output matrices could take on the orthogonal form:

$$c_{alt} = \begin{bmatrix} 1 & 0 \\ 0 & 100 \\ 0 & 0 \end{bmatrix}$$

$$d_T = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{50} \end{bmatrix}$$

7.4.2.4 Direction Subsystem

The direction systems variables were:

$$A_{dir} = \begin{bmatrix} -5.616 & 0 \\ 1 & 0 \end{bmatrix} \; B_{dir} = \begin{bmatrix} 0.02253 \\ 0 \end{bmatrix} \quad \nu_R \quad \Lambda_{dir} = \begin{bmatrix} r \\ \psi \end{bmatrix}$$

As with the linear altitude system the linear direction system did not have degree of freedom issues so its output matrices were chosen similarly:

$$c_{dir} = \begin{bmatrix} 1 & 0 \\ 0 & 100 \\ 0 & 0 \end{bmatrix}$$

$$d_R = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{50} \end{bmatrix}$$

Forming each of the output equations into the corresponding cost matrices:

$$Q_{alt} = c_{alt}^T c_{alt} \quad R_{alt} = d_{alt}^T d_{lat} \quad N_{alt} = c_{alt}^T d_{alt}$$

$$Q_{lat} = c_{lat}^T c_{lat} \quad R_{lat} = d_{lat}^T d_{lat} \quad N_{lat} = c_{alt}^T d_{lat}$$

$$Q_{lon} = c_{lon}^T c_{lon} \quad R_{lon} = d_{lon}^T d_{lon} \quad N_{lon} = c_{alt}^T d_{lon}$$

$$Q_{lat} = c_{lat}^T c_{lat} \quad R_{lat} = d_{lat}^T d_{lat} \quad N_{dir} = c_{alt}^T d_{dir}$$

And combining the results into block diagonal matrices:

$$Q = \begin{bmatrix} Q_{alt} & [0] & [0] & [0] \\ [0] & Q_{lat} & [0] & [0] \\ [0] & [0] & Q_{lon} & [0] \\ [0] & [0] & [0] & Q_{dir} \end{bmatrix} \quad R = \begin{bmatrix} R_{alt} & [0] & [0] & [0] \\ [0] & R_{lat} & [0] & [0] \\ [0] & [0] & R_{lon} & [0] \\ [0] & [0] & [0] & R_{dir} \end{bmatrix} \quad N = \begin{bmatrix} N_{alt} & [0] & [0] & [0] \\ [0] & N_{lat} & [0] & [0] \\ [0] & [0] & N_{lon} & [0] \\ [0] & [0] & [0] & N_{dir} \end{bmatrix}$$

Which turn out to be in numerical form for the design here:

$$Q = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.25 & 0.8333 & 0.02 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.8333 & 2.7778 & 0.0667 & 6.6667 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.2 & 0.0667 & 0.0016 & 0.16 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6.6667 & 0.16 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.83 & -0.02 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.8333 & 2.7778 & -0.0667 & -6.667 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.02 & -0.0667 & 0.0016 & 0.16 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -2 & -6.6667 & 0.16 & 16 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10
\end{bmatrix}$$

$$R = \begin{bmatrix}
4 \times 10^{-4} & 0 & 0 & 0 \\
0 & 2.778 \times 10^{-4} & 0 & 0 \\
0 & 0 & 2.778 \times 10^{-4} & 0 \\
0 & 0 & 0 & 4 \times 10^{-4}
\end{bmatrix}$$

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.0083 & 0 & 0 \\ 0 & 0.0278 & 0 & 0 \\ 0 & 0.0007 & 0 & 0 \\ 0 & 0.0667 & 0 & 0 \\ 0 & 0 & -0.0083 & 0 \\ 0 & 0 & -0.0278 & 0 \\ 0 & 0 & 0.0007 & 0 \\ 0 & 0 & 0.0667 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

With the resulting controller, with columns rearranged to work with the original Λ:

$$K = \begin{bmatrix} 0 & 0 & -93.03 & 0 & 0 & 0 & 0 & 0 & -155.9 & 0 & 0 & 0 \\ 0 & 30.28 & 0 & 3.28 & 0 & 0 & 0 & 99.91 & 0 & 241.27 & 0 & 0 \\ -30.27 & 0 & 0 & 0 & 3.26 & 0 & -99.9 & 0 & 0 & 0 & 241.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 31.02 & 0 & 0 & 0 & 0 & 0 & 157.56 \end{bmatrix}$$

It is clear that, in relation to the previous controller given, the magnitude of the terms multiplying the x and y errors have increased by a factor of approximately 150% while keeping those multiplying the orientation angles similar in magnitude, which turned out to be exactly the needed effect.

7.4.2.5 Pseudo-Nonlinear Extension

The same change of coordinates applied to the lateral and longitudinal errors applied to the PID controls in Section 6.4 was applied to the errors receieved by the LQR gain matrix.

7.4.3 Results

The results of implementation of the LQR state feedback matrix previously designed are shown in what follows: The resulting yaw/heading reference step response of the LQR control in

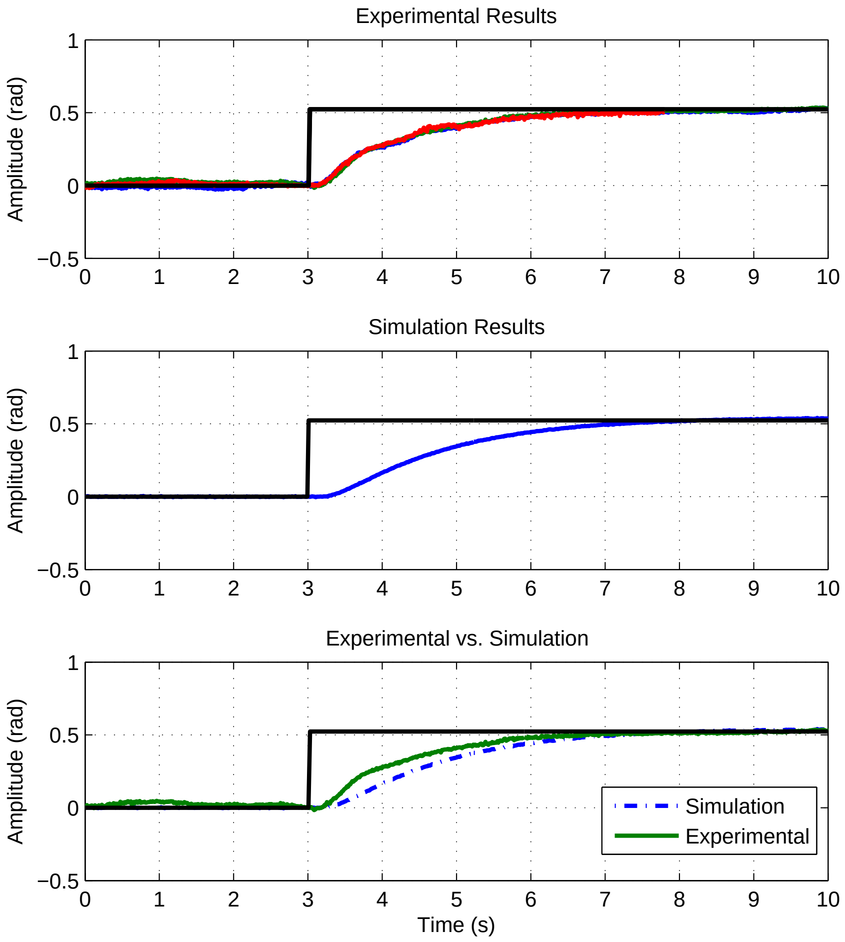


Figure 7.1 LQR Heading Step Commands

Figure 7.1 shows good, though not perfect, matching between the simulation prediciton and experimental data. Though the response is somwhat slower than that of the PID, yaw tracking speed was not really a design goal. The resulting control was seen as adequate.

The response to both lateral and longitudinal reference step changes with the LQR in Figures 7.2 and 7.3, both achieved similar performance to those of the PID controls. The results

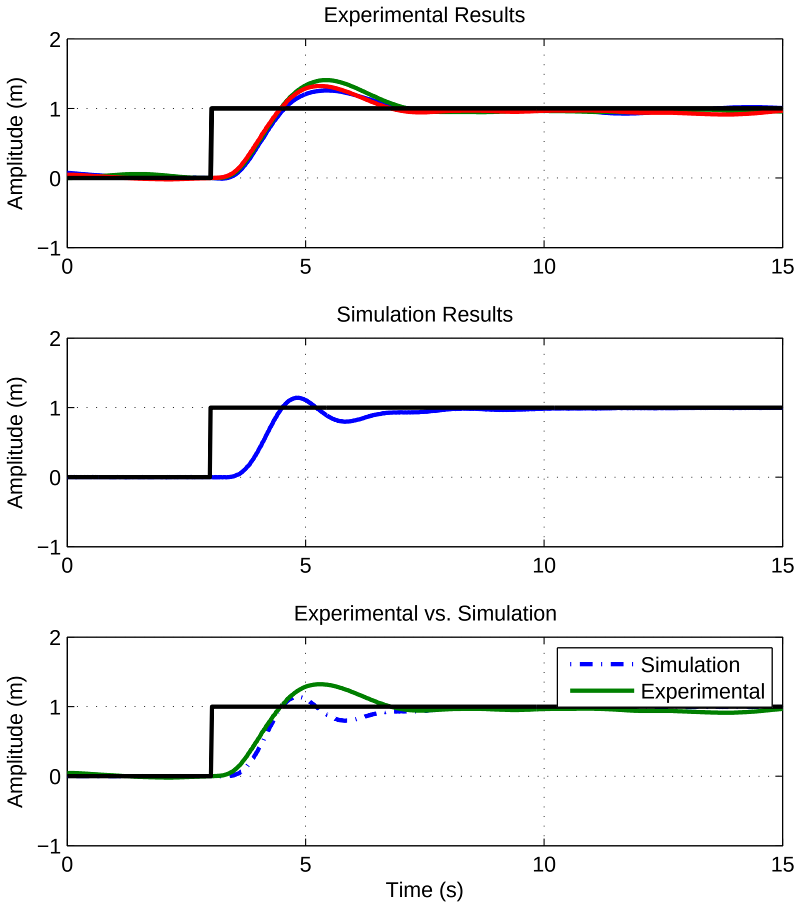


Figure 7.2 LQR Lateral Step Commands

obtained with the LQR are actually slightly better than those of the PID, with approximately 10cm less overshoot on average with an equal or slightly better settling time.

The transient mismatch seen in both responses is very similar to that of the corresponding PID responses, confirming that the source was likely not some controller design or implementation issue as it causes a nearly identical effect with two independent controllers.

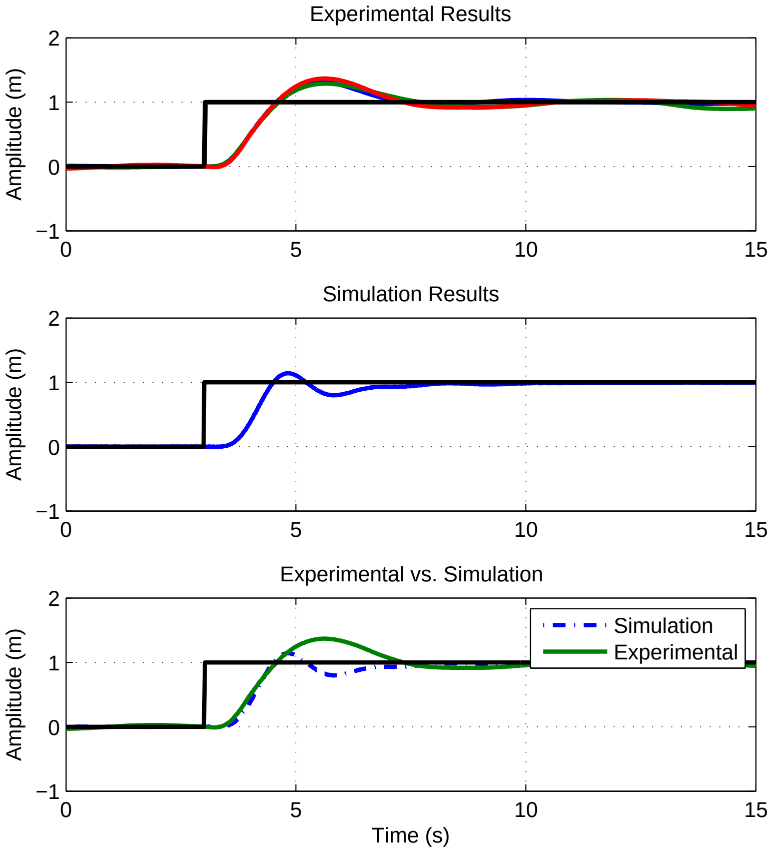


Figure 7.3 LQR Longitudinal Step Commands

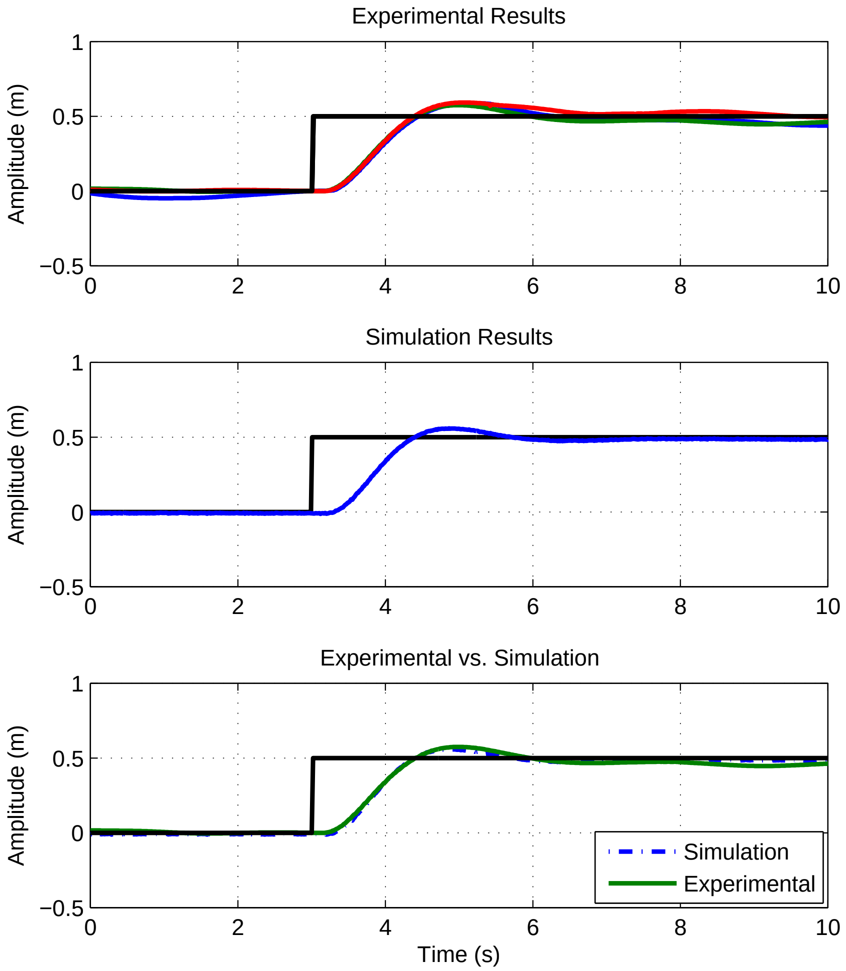


Figure 7.4 LQR Height Step Commands

The response of the LQR control to height reference changes in Figure 7.4 is very good, with approximately 10cm less overshoot on average than the corresponding PID behavior and an approximately equivalent settling interval. As was the case with the PID height tracking, the simulation model and experimental results are nearly identical.

CHAPTER 8. Conclusion

8.1 Summary

In this work, a complete, thorough nonlinear model of a quadrotor helicopter vehicle including on board electronics and an accompanying network control system was developed. The goal to develop complete, accurate and independent symbolic models for each component and each subsystem successively describing the system as a whole, while avoiding unnecessary model complication was accomplished. A general 6DOF rigid body development was provided allowing for analysis from points of reference other than the vehicle center of mass. Symbolic models for each subsystem and component were developed that allow for easy introduction of various parameters, description and analysis of parametric uncertainty, and easy inclusion or exclusion of various levels of dynamics description and accuracy using parameter values.

Model parameters were identified using systematic and as far as possible repeatable procedures and calculations. The resulting numerical nonlinear model and simulation yielded results sufficient to design model based optimal control without any need for subsequent manual tuning. The model has highly accurate predictive capabilities for the physical dynamics of the system such as response to angular rate reference changes, thrust and torque produced by rotors, and closed loop responses to height and heading commands. The model predictions for the lateral and longitudinal closed loop responses suffered due to the presence of unknown dynamics in a portion of the communication system, but were still reasonably accurate and adequate enough to produce the previously mentioned working control.

Finally, a systematic procedure for obtaining improved results with Linear Quadratic optimal control taking into account the physical nature of the system–specifically degree of freedom conflict–was presented and used to successfully produce single-loop state feedback control with equivalent or better performance to a nested-loop PID scheme and superior performance to LQ control designed using more typical methods. Each control was also accompanied by a straight forward, effectively implemented method for extending their performance to arbitrary headings.

8.2 Further Work

The model presented should lend itself very well to various forms of robust control analysis and design involving numerous types of parametric uncertainty, as well as the design and testing of real-time system identification algorithms and general adaptive control schemes. The very rarely presented, and even more rarely developed 6DOF dynamics equations expressed with respect to a point of reference generally different from the rigid body center of mass, along with fairly straight forward application of the multidimensional version of the parallel axis theorem should allow specifically for the representation of a vehicle with unknown mass and inertia characteristics. The symbolic description of such uncertainty, whether inherent in the vehicle model, due to uncertain attached loads, or both, should allow for subsequent production of controllers preserving stability and performance using some type of potentially scheduled H_∞ design, passivity based control, fast adaptive control, or some other method(s) or combination thereof. Also, improvements could undoubtedly be made to the system presented here and to many similar systems by directly taking into account implementation issues such as variable latency and/or sampling rate variance due to either a network/OS setup as was the case here or implementation on local hardware with shared resources.

APPENDIX A. Earth as a Frame of Reference

Assuming some inertial frame of reference, and defining the position of the center of the Earth to be χ, the angular velocity of the Earth around it's axis of rotation as Ω, the radius of the Earth frame of reference origin from the planet's center r, and the vector from the Earth frame of reference origin to any given point p, the position vector of that point in the inertial frame of reference, $X = \chi + r + p$.

Differentiating to obtain velocity:

$$\dot{X} = \dot{\chi} + \Omega \times r + \Omega \times p + \dot{p}$$

And again for acceleration:

$$\ddot{X} = \ddot{\chi} + \dot{\Omega} \times r + \Omega \times (\Omega \times r) + \dot{\Omega} \times p + \Omega \times \dot{p} + \ddot{p} + \Omega \times (\Omega \times p) + \Omega \times \dot{p}$$

Since $\ddot{r} = \dot{r} = 0$, the quantity $R = r + p$ can be defined which allows the above equation to be simplified to:

$$\ddot{X} = \ddot{\chi} + \ddot{R} + \dot{\Omega} \times R + 2\Omega \times \dot{R} + \Omega \times (\Omega \times R)$$

The acceleration of the center of the planet, $\ddot{\chi}$, tangental acceleration $\dot{\Omega} \times R$ due to changes in Earth's angular velocity about it's axis, and centripetal acceleration $\Omega \times (\Omega \times R)$ are all relatively small. The angular velocity of the planet around its own axis very small: $(2\pi/24/60/60 \approx 7.2722 \times 10^{-5} \frac{rad}{s})$. Thus the Coriolis effect $(2\Omega \times \dot{R})$ is important for objects with very high velocity and or few significant external forces, such as high velocity projectiles, rockets, and weather patterns[20]. For the purposes of the model used here, the only significant term is $\ddot{R}$, the acceleration of the craft relative to the Earth frame of reference origin.

APPENDIX B. Pseudo-Inverse Matrix for Overdetermined Systems

Given data that can be expressed–possibly via some manipulation–as multiple linear combinations of some unknown parameters with measured and controlled variables, with the number of combinations greater than the number of parameters, an overdetermined system can be defined. Using the variable j to index each of m equations that can be formed using a set of n unknown parameters x_1 to x_n in linear combination with the measured or controled values a_{1_j} to a_{n_j} equaling the measured or controlled values b_j plus some lumped error ϵ_j (See e.g. Chapter 3 of [54] for more detail), each equation can be expressed:

$$a_{1_j}x_1 + a_{2_j}x_2 + \cdots + a_{n_j}x_n = b_j + \epsilon_j \quad \forall j \in 1, 2, 3, ..., m$$

And collectively formed into:

$$Ax = b + \epsilon$$

With $A \in \mathbb{R}^{m \times n}$, $m > n$. The *best* choice of parameters x is taken to be that which minimizes the Euclidian magnitude squared of the assumed error ϵ, i.e. that which minimizes:

$$\|\epsilon\|_2^2 = \|Ax - b\|_2^2$$

Since $\|\epsilon\|_2^2 = \|Ax - b\|_2^2$ is differentiable in x and can easily be shown to be convex in x, the optimal value can be found by setting its derivative in terms of x to 0. Expressing the function to be minimized as:

$$\|\epsilon\|_2^2 = (Ax - b)^T(Ax - b) = x^T A^T Ax - 2b^T Ax + b^T b$$

And setting the derivative in terms of x to 0 leads to:

$$x^T A^T A = b^T A$$

Which can be solved for x as:

$$x = (A^T A)^{-1} A^T b$$

With the *pseudo-inverse* of the matrix $A \in \mathbb{R}^{m \times n}$ begin the quantity $(A^T A)^{-1} A^T$. For more explanation and background on least-squares estimation and further applications (e.g. recursive estimation) see [54], and for far more in-depth treatment of both least-squares optimization and optimization in general see [55].

APPENDIX C. Linearization

Generally speaking, given the nonlinear system where f is the state equation, y the output equation, x is the state vector, u is the input vector, and t represents time:

$$\dot{x}(t) = f(x(t), u(t))$$

$$y = h(x(t), u(t))$$

Linearization can be accomplished using Taylor Series expansion, and keeping only first order terms:

$$\dot{x}(t) \approx f(x^*, u^*) + \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x^* \\ u=u^*}} (x - x^*) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x^* \\ u=u^*}} (u - u^*)$$

$$y \approx h(x^*, u^*) + \left.\frac{\partial h}{\partial x}\right|_{\substack{x=x^* \\ u=u^*}} (x - x^*) + \left.\frac{\partial h}{\partial u}\right|_{\substack{x=x^* \\ u=u^*}} (u - u^*)$$

Linearization around an equilibrium point or trajectory[1] implies $f(x^*, u^*) = 0$. Defining equilibrium output:

$$y^* = h(x^*, u^*)$$

And each of the Jacobian matrices:

$$A = \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x^* \\ u=u^*}} \quad B = \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x^* \\ u=u^*}} \quad C = \left.\frac{\partial h}{\partial x}\right|_{\substack{x=x^* \\ u=u^*}} \quad D = \left.\frac{\partial h}{\partial u}\right|_{\substack{x=x^* \\ u=u^*}}$$

And finally a change of state/input coordinates:

$$z = x - x^* \quad v = u - u^*$$

The linear system approximating the original around the equilibrium point can be written:

$$\dot{z} = Az + Bv \quad y = y^* + Cz + Dv$$

---

[1] Of course this is under the assumption that the equilibrium point or trajectory is accurately known. Parametric uncertainty, modeling error or lacking model dynamics can lead to non-zero $f(x^*, u^*)$

APPENDIX D. OptiTrack System Setup and Euler Angle Issues

In order to properly use the OptiTrack system, the cameras need to be arranged in some constellation with overlapping fields of view and the whole system calibrated. The calibration process is described on the company website in tutorial videos and user manuals [48]. With a given calibration, a *set ground plane* step is required, which essentially determines the equivalent of the inertial (Earth) frame axes (see Section 2.1) of this model. This is done by using a tool provided by OptiTrack, similar in shape to a carpenter's square with three special reflective markers attached to make a specific shape. This tool and the Tracking Tools software by default[1] set the *ground plane* such that positive y-axis is opposed to the acceleration of gravity, or *up* (assuming a horizontal floor). This setup was not used for the following reasons:

1. The inertial axes of this model as detailed in Section 2.1 are such that the positive z-axis is along the acceleration of gravity, and hence the x and y axes would be in the equivalent *ground plane*. It was desirable to have the axes matching those of the model.
2. More importantly, the sequency of rotations used in this model (as described in Section 2.1.3) to give orientation is Yaw-Pitch-Roll, i.e. a ZYX convention. This is the same convention assumed by the VRPN libraries allowing the quaternion data to be translated into the correct Euler angles. This means that in order to obtain the correct Euler angles through VRPN, **the axes must be set up so that they match those of this model**, i.e. the positive z-axis along the acceleration of gravity[2].

---

[1] By default meaning if the ground plane tool is placed on the floor on its rubber feet as obviously intended.

[2] It may seem as though one could simply use the ground plane in its default position and switch the yaw and pitch angles in order to still obtain the correct magnitude angles with the VRPN functions, and simply negate the resulting yaw to give the correct sign. However, this is *not* the case and such an approach will fail to give the correct angles in general even if it works for some circumstances.

This attempted fix will produce correct magnitude angles only when a single angle is changed. Also, due to the fact that what would be read as *yaw* would be in calculations treated like *pitch* in a ZYX convention, Althougth setting the system up this way does mean that the angles given in the Tracking Tools real time display will be incorrect[3], it is necessary to gurarantee the correct angles are obtained at the control PC. The VRPN functions that use the quaternions provided to calculate yaw, pitch, and roll, assume that the sequence of rotations was yaw first, around the positive z-axis, followed by pitch and roll around equivalent y and x axes as described more accurately in 2.1.3. Luckily the ground plane tool provided is such that it can easily be set on its side to make the axes math those of this model[4].

---

numerical issues witll result at yaw angles of $\pm 90^o$ which is clearly very disadvantageous.

[3] In any case they would not typically match the angles of this model, since the Tracking Tools software (for display purposes) follows a XYZ convention and this model uses a ZYX. *Incorrect* here means incorrect with respect to a *default* ground plane positioning with XYZ convention angle sequence. Of course the software streams the orientation in quaternions which do not have such sequence dependance.

[4] As mentioned in the user's guide, but not entirely obvious at first glance, the *positive* z-axis will be defined in the *opposite* direction of the arrow above the z printed on the ground plane tool.

BIBLIOGRAPHY

[1] G. D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling.* American Insititute of Aeronautics and Astronautics, Inc., 2007.

[2] T. Bresciani, "Modeling, identification and control of a quadrotor helicopter," Master's thesis, Lund University, 2008.

[3] M. D. L. C. de Oliveira, "Modeling, identification and control of a quadrotor aircraft," Master's thesis, Czech Technical University in Prague, 2011.

[4] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2007, pp. 1–20.

[5] R. Goela, S. Shahb, N. Guptac, and N. Ananthkrishnanc, "Modeling, simulation and flight testing of an autonomous quadrotor," *Proceedings of ICEAE*, 2009.

[6] B. Erginer and E. Altug, "Modeling and pd control of a quadrotor vtol vehicle," in *Intelligent Vehicles Symposium, 2007 IEEE.* IEEE, 2007, pp. 894–899.

[7] C. Balas, "Modelling and linear control of a quadrotor," *Cranfield Unicersity, MSc Thesis*, vol. 2007, 2006.

[8] T. Buchholz, D. Gretarsson, and E. Hendricks, "Construction of a four rotor helicopter control system," Master's thesis, Technical University of Denmark, 2009.

[9] I. Cowling, J. Whidborne, and A. Cooke, “Optimal trajectory planning and lqr control for a quadrotor uav,” in *Proceedings of the International Conference Control–2006, Glasgow, Scotland*, vol. 30, 2006.

[10] L. Minh and C. Ha, “Modeling and control of quadrotor mav using vision-based measurement,” in *Strategic Technology (IFOST), 2010 International Forum on.* IEEE, 2010, pp. 70–75.

[11] Y. Al-Younes, M. Al-Jarrah, and A. Jhemi, “Linear vs. nonlinear control techniques for a quadrotor vehicle,” in *Mechatronics and its Applications (ISMA), 2010 7th International Symposium on.* IEEE, 2010, pp. 1–10.

[12] S. Bouabdallah, “design and control of quadrotors with application to autonomous flying,” Ph.D. dissertation, Federal Polytechnic School of Lausanne, 2007.

[13] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, “Design, modeling, estimation and control for aerial grasping and manipulation,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on.* IEEE, 2011, pp. 2668–2673.

[14] E. Altug, J. Ostrowski, and R. Mahony, “Control of a quadrotor helicopter using visual feedback,” in *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 72–77.

[15] A. Mokhtari, N. M’Sirdi, K. Meghriche, and A. Belaidi, “Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle,” *Advanced Robotics*, vol. 20, no. 1, pp. 71–91, 2006.

[16] T. Madani and A. Benallegue, “Control of a quadrotor mini-helicopter via full state backstepping technique,” in *Decision and Control, 2006 45th IEEE Conference on.* IEEE, 2006, pp. 1515–1520.

[17] A. Das, K. Subbarao, and F. Lewis, “Dynamic inversion with zero-dynamics stabilisation for quadrotor control,” *Control Theory & Applications, IET*, vol. 3, no. 3, pp. 303–314, 2009.

[18] M. Hehn and R. DAndrea, “Quadrocopter trajectory generation and control,” in *Proceedings of the IFAC world congress*, 2011.

[19] [Online]. Available: http://home.engineering.iastate.edu/~m87rich/

[20] M. D. Ardema, *Newton-Euler Dynamics.* Springer Science+Business Media Inc., 2005.

[21] J. M. Rolfe and K. J. Staples, *Flight simulation / edited by J.M. Rolfe, K.J. Staples.* Cambridge University Press, Cambridge [Cambridgeshire] ; New York :, 1986.

[22] R. Featherstone, *Rigid Body Dynamics Algorithms.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[23] H. Goldstein, *Classical Mechanics.* Addison-Wesley Publishing Company, Inc., 1959.

[24] J. Wittenburg, *Dynamics of Systems of Rigid Bodies.* Teubner Stuttgart, 1977.

[25] *330X Quad-Flyer Manual*, GAUI.

[26] W. Johnson, *Helicopter Theory.* Princeton, NJ, USA: Princeton University Press, 1980.

[27] A. Bramwell, G. Done, D. Balmford, and E. S. Publishers, *Bramwell's Helicopter Dynamics.* American Insititute of Aeronautics and Astronautics, Inc., 2001.

[28] J. Seddon and S. Newman, *Basic Helicopter Aerodynamics.* American Insititute of Aeronautics and Astronautics, Inc., 2001.

[29] V. Gavrilets, B. Mettler, and E. Feron, “Dynamic model for a miniature aerobatic helicopter,” *Massachusetts Institute of Technology, MIT-LIDS report# LIDS-P-2579*, 2003.

[30] V. Gavrilets, “Autonomous aerobatic maneuvering of miniature helicopters,” Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[31] S. Baldursson, “Bldc motor modelling and control a matlab/simulink implementation,” Master's thesis, Chalmers University of Technology, 2005.

[32] M. Drela, “First-order dc electric motor model,” Massachusetts Institute of Technology, 2007.

[33] ——, “Second-order dc electric motor model,” Massachusetts Institute of Technology, 2006.

[34] K. Ogata, *System dynamics.* Pearson/Prentice Hall, 2004.

[35] M. Chen and G. Rincon-Mora, “Accurate electrical battery model capable of predicting runtime and i-v performance,” *Energy Conversion, IEEE Transactions on*, vol. 21, no. 2, pp. 504 – 511, june 2006.

[36] T. Kim and W. Qiao, “A hybrid battery model capable of capturing dynamic circuit characteristics and nonlinear capacity effects,” *Energy Conversion, IEEE Transactions on*, vol. 26, no. 4, pp. 1172 –1180, dec. 2011.

[37] S. Abu-Sharkh and D. Doerffel, “Rapid test and non-linear model characterisation of solid-state lithium-ion batteries,” *Journal of Power Sources*, vol. 130, no. 1, pp. 266–274, 2004.

[38] M. Partovibakhsh and G. Liu, “Online estimation of model parameters and state-of-charge of lithium-ion battery using unscented kalman filter,” in *American Control Conference (ACC), 2012.* IEEE, 2012, pp. 3962–3967.

[39] *XV-3500CB Datasheet*, EPSON.

[40] *Dymo M10—M25 Manual*, Dymo.

[41] *Model 461895 User’s Guide*, Extech Instruments, 2004.

[42] *Manual for Model 220: Industrial Emulator / Servo Trainer (Instructor’s Edition)*, Educational Control Products, 1995.

[43] *Tektronix DPO3034 Digital Phosphor Oscilloscope Datasheet*, Tektronix, 2008.

[44] *HP 54600-Series Oscilloscopes Datasheet*, Hewlett Packard.

[45] *HP 34401A Multimeter Datasheet*, Hewlett Packard.

[46] *HP 33120A Function/Waveform Generator Datasheet*, Hewlett Packard.

[47] *Tektronix CPS250 Power Supply Manual*, Tektronix.

[48] *Tracking Tools 2.4.0 User’s Guide*, OptiTrack.

[49] *GUEC GE-010 Manual*, GAUI.

[50] R. Gebel and H. Fettis, “Application of third-order transfer functions for edge enhancement of signals from scanned imaging systems,” 1974.

[51] X. Zhou, “Role of uncertainty in stochastic linear quadratic regulators,” in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 2. IEEE, 1997, pp. 1094–1099.

[52] A. Lim and X. Zhou, “Stochastic optimal lqr control with integral quadratic constraints and indefinite control weights,” *Automatic Control, IEEE Transactions on*, vol. 44, no. 7, pp. 1359–1369, 1999.

[53] T. Chen and B. Francis, *Optimal sampled-data control systems.* Springer London, 1995, vol. 124.

[54] P. Wellstead and M. Zarrop, *Self-tuning systems: control and signal processing.* John Wiley & Sons, Inc., 1991.

[55] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge university press, 2004.