

2016

Design, modeling, and control of a two degree of freedom pendulum on an omnidirectional robot

Paul Frederick Uhing
Iowa State University

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

 Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Uhing, Paul Frederick, "Design, modeling, and control of a two degree of freedom pendulum on an omnidirectional robot" (2016).
Graduate Theses and Dissertations. 15825.
<https://lib.dr.iastate.edu/etd/15825>

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.

**Design, modeling, and control of a two degree of freedom pendulum on an
omnidirectional robot**

by

Paul Uhing

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

Umesh Vaidya

Iowa State University

Ames, Iowa

2016

Copyright © Paul Uhing, 2016. All rights reserved.

DEDICATION

To my parents, family and friends who have been with me throughout my education.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
ACKNOWLEDGEMENTS	viii
ABSTRACT	ix
CHAPTER 1. INTRODUCTION	1
1.1 Literature review	2
1.2 Objectives	3
1.3 Summary	3
1.4 Notation	4
CHAPTER 2. SYSTEM DESCRIPTION	11
2.1 Platform	11
2.2 Reference frames	13
CHAPTER 3. SYSTEM DYNAMICS DERIVATION	17
3.1 Position and velocity description	17
3.2 Kinetic energy	18
3.2.1 Pendulum	18
3.2.2 Ground robot	19
3.2.3 Total kinetic energy	20
3.3 Potential energy	20
3.4 Lagrangian	21
3.5 Wheel forces	24

3.5.1	Wheels force	24
3.5.2	Rolling resistance	25
3.6	Quadrotor dynamics	25
3.6.1	Quadrotor motor dynamics	25
3.6.2	Propeller wrench	26
3.6.3	Total propeller force	27
3.7	Input dynamics	27
3.7.1	Saturation and dead zones	27
3.7.2	Mixing	28
CHAPTER 4. PARAMETER IDENTIFICATION		30
4.1	Masses	30
4.2	Lengths	30
4.2.1	Procedure	31
4.2.2	Parameter values	31
4.3	Moments of Inertia	31
4.4	Quad motor parameters	32
4.4.1	BLDC motor parameters	32
4.5	Wheel Parameters	34
4.5.1	Velocity coefficient identification	34
4.5.2	H-bridge parameters	36
4.5.3	Rolling Resistance	37
4.5.4	Parameters	38
4.6	Battery voltage	39
4.6.1	Procedure for propeller motors	39
4.6.2	Procedure for wheel motors	39
4.7	Complete parameter list	39

CHAPTER 5. PID CONTROLLERS	42
5.1 Pitch and roll controllers	42
5.2 Linear velocity control	43
5.3 Heading control	44
5.4 Control mixing	45
5.5 Controller testing	45
5.5.1 Procedure	46
5.5.2 Results	46
CHAPTER 6. LQG CONTROLLER	49
6.1 Model linearization	49
6.2 Controller design	54
6.3 Results	61
6.3.1 Comparison to PID controller	61
CHAPTER 7. SEPARATE LQG CONTROLLER DESIGNS	67
7.1 Subsystem division	67
7.2 LQG controller for the pendulum system	70
7.2.1 Pendulum LQG with open-loop robot results	71
7.3 Ground robot LQG controller design	73
7.4 Results	76
CHAPTER 8. CONCLUSION	78
8.1 Future work	80
BIBLIOGRAPHY	82

LIST OF TABLES

Table 1.1	List of symbols used in this thesis	4
Table 1.2	List of common acronyms	10
Table 2.1	Major system components	16
Table 4.1	Mass parameter values	30
Table 4.2	Length parameters	31
Table 4.3	Moment of inertia parameters parameters	32
Table 4.4	Quadrotor motor parameters	34
Table 4.5	Wheel motor parameters	38
Table 4.6	All measured system parameters	39
Table 5.1	PID controller values for pitch and roll.	43
Table 5.2	PID controller values for x and y velocity.	44
Table 5.3	PID controller values for yaw postion.	45

LIST OF FIGURES

Figure 2.1	RADA system design concept without wheels.	15
Figure 4.1	Time rise data capture.	33
Figure 4.2	Measurement results for identifying K_{V_w} of the wheel motors. . .	36
Figure 4.3	H-bridge parameter identification results.	37
Figure 5.1	PID controller comparison.	47
Figure 5.2	PID controller comparison zoomed in $x-\theta$	48
Figure 5.3	PID controller comparison zoomed in $y-\phi$	48
Figure 6.1	LQG controller comparison.	63
Figure 6.2	LQG controller comparison zoomed in $x-\theta$	64
Figure 6.3	LQG controller comparison zoomed in $y-\phi$	64
Figure 6.4	Comparison of the LQG and PID in $x-\theta$ when accelerating. . .	65
Figure 6.5	Comparison of the LQG and PID in $x-\theta$ when stopping.	65
Figure 6.6	Comparison of the LQG and PID in $y-\phi$ when accelerating. . .	66
Figure 6.7	Comparison of the LQG and PID in $y-\phi$ when stopping.	66
Figure 7.1	LQG control of the pendulum with open-loop cart commands. . .	73
Figure 7.2	Test LQG control of each subsystem independently in $x-\theta$	77
Figure 7.3	Test LQG control of each subsystem independently in $y-\phi$	77
Figure 8.1	Simulation test of a possible wheel force disturbance model. . .	79

ACKNOWLEDGEMENTS

Dr. Nicola Elia *Professor of Electrical Engineering:*

For support of the project both intellectually and financially.

Dr. Phillip Jones *Professor of Computer Engineering:*

For helping support the senior design team that made the prototype possible.

Matt Rich *Electrical Engineer / PhD candidate:*

For his help with model simulation development.

Ian McInernery *Electrical Engineer / RADA 2015 senior design team / Master's student:*

For help running and troubleshooting the system.

Justin Noronha *Electrical Engineer / Master's student:*

For help running and troubleshooting the system .

Dylan Gransee *Computer Engineer / RADA senior design team:*

For rebuilding the software development environment for RADA and solving many software issues.

Robert Larsen *Computer Engineer / RADA senior design team:*

For fixing VHDL issues with the FPGA based motor controller and software development.

Aaron Pederson *Electrical Engineer / RADA senior design team:*

For redesigning the Eris ground robot platform to support the pendulum arm.

Rohit Zambre *Electrical Engineer / RADA senior design team:*

For developing the logging and data processing system used to capture data from testing.

This research has been supported by NSF grant CNS-1239319.

ABSTRACT

In this thesis a dynamical model for a two degree of freedom inverted pendulum on an omnidirectional cart is developed. This system is actuated not only by the wheels on the cart but also by a quadrotor system used to stabilize the inverted pendulum. The dynamic model is designed to be modular allowing for the substitution of different actuators, electrics, and control algorithms without the need to re-derive the model. The parameters of this system are identified with procedures and background where necessary. This thesis presents two controller methods: PID and LQG controllers. The PID controllers were designed using both hand tuning and model based tuning. The LQG controllers were designed using a systematic procedure to initially choose good weights and adjust the weights based off of systems performance in order to get the most out of the system. All sets of controllers are presented along with a comparison of their behavior and plots comparing the real system to that of the non-linear simulation. The results of this design are discussed along with issues that shaped the design decisions, future work and systems improvements are discussed.

CHAPTER 1. INTRODUCTION

The inverted pendulum on a cart system as its name implies is simply a cart with a pendulum that balances above the cart. Standard inverted pendulums on a cart move in a single axis and traverse in a single plane. The pendulum is held in a vertical position by the motion of the cart. This is useful so long as the cart can move along the rotational axis of the pendulum and the moment of inertia of the pendulum is kept relatively small. If these requirements cannot be met either due to the system design or other constraints a new method of stabilizing the pendulum is needed.

The goal of the Robotic Agricultural Data Acquisition (RADA) system is to use a large two degree of freedom (2DOF) inverted pendulum on a cart to traverse a corn field and measure the height of the plants. The pendulum needs to have 2DOF so that wind and other disturbances cannot cause the system to fall over in the field. Clearly the cart will not be able to move in one of the pendulum degrees of freedom and the pendulum will need to be large and heavy in order for sensors to be placed on the pendulum, therefore the moment of inertia will not be small. Because of these design requirements, a quadrotor will be used to stabilize the pendulum instead of the cart's motion.

Quadrotors are traditionally flown like a helicopter using two pairs of counter-rotating propellers to move through space. For this system, the motors will be rotated outwards by 90° and then the quadrotor is attached to the top of the pendulum. This way the rotors will produce torque that can be used to stabilize the system. Then the inverted pendulum can be kept vertical without the need for the cart to move. The propellers will be able to provide additional torque as the system is scaled up to larger pendulums, and the motors will not cause the cart to counter rotate like they would if motors were placed at the base of the pendulum.

This thesis discusses the initial prototype and design of the RADA system. The current system consists of a 2DOF inverted pendulum on an omnidirectional cart. The cart can move in both x and y without having to change its heading as well as control its heading, making it a 3DOF cart. The total system has 5DOF and is designed for use indoors, making testing of the initial system design simpler. This also allows the cart to disturb both degrees of freedom of the inverted pendulum system. While both the pendulum and the cart can be thought of as different entities as in Chapter 5 and Chapter 7, it is most effective to design controllers for the whole system due to the interrelationships between the cart and the pendulum as in Chapter 6. The design of model based control for the pendulum and cart as separate subsystems is investigated in Chapter 7.

1.1 Literature review

The inverted pendulums on a cart is a classic controls problem. Works such as [1], [2], [3], [4], present multi-degree of freedom inverted pendulums. The work in [2] presents a large inverted pendulum system on a ball. The issue with these cases the linear motion of the ball is what stabilizes the pendulum. In some cases, it may be desirable to be able to stabilize an inverted pendulum without having to move the base. Systems such as [4] need to use a large motor to stabilize even a small pendulum. Another option is to use a system of propellers like those used on quadrotors, [5]. A system using propellers could be tall but be contained by a small vehicle and maintain the stability of the system even if the vehicle cannot maneuver to keep the pendulum system stabilized.

One application of this type of system could be crop monitoring. Works such as [6] and [7] present vehicles that can monitor weeds and even participate in weed control. A large inverted pendulum on a cart could be used to monitor weeds and the health of crops such as corn where it could be difficult for other types of systems to navigate into positions to collect the required data. This work will present a two degree of freedom inverted pendulum on an omnidirectional cart using a quadrotor system to stabilize the pendulum system.

1.2 Objectives

The objective of this thesis are as follows:

- Describe design of the system created in conjunction with the RADA senior design team.
- Develop a parametrized non-linear simulation of the designed system.
- Develop additional parameter identification procedures beyond those developed in [8].
- Design and implement linear controllers to effectively control the system.

The RADA system was designed by the RADA senior design team from May of 2015, based off of the input from myself and others within the lab group. The team was given the specific goals for RADA and then they designed the specific implementation of the prototype. The system design is an integral component of this thesis, as other sections depend on the specific system implementation, so the system design is discussed here.

The non-linear model was developed in order to design controllers. It is important that the model is parameterized in such a way that parameters can be changed easily when components of the system are changed. The non-linear model used for the simulation should be designed so that subsystems can be easily changed and substituted as prototypes are improved. Many of the parameter identification procedures were developed in [8] but some parameters are new for this system model and identification techniques needed to be developed.

Finally, the model needs to be used to design linear controllers. This is necessary to verify that the model is good enough at predicting the behavior of the real prototype. Therefore, the system model needs to be accurate enough to design effective controllers.

1.3 Summary

The rest of this thesis is broken down into three parts. The Part I, Chapter 2, breaks down how the RADA prototype was designed. The construction of the system is important for how the model was derived as well as determining what can be expected out of the performance of the system.

Part II, Chapter 3 and Chapter 4, describes how the system is modeled both for controller design and for simulation. There are slight differences between the controller design model and the simulation model that make the simulation more efficient and minimizes the states in the controller. The parameters of model and how they are measured is also discussed. While many of the parameter measurements were previously discussed in [8], some of the parameters measurement techniques have been improved and require further discussion.

Part III, Chapter 5 through Chapter 7, discusses the linear controller design. This includes the model verification through PID controller design and testing. Then discusses how the system is linearized and the design parameters used for controller design. The performance of these controllers are examined as well.

1.4 Notation

The notation used in this thesis is should be fairly consistent. The general structure of the symbols used in this document is as follows:

$${}_{[Frame\ of\ Reference]}^{[Time\ Derivative]} [MAJOR\ SYMBOL]_{[Designation/Index]}^{[Exponent/Transpose]}$$

The designation that is most commonly missing in this document is the reference frame. That is only used if the same variable is used for multiple reference frames. A complete list of important Symbols used in this thesis is shown in Table 1.1. In addition to these symbols there are a few acronyms used throughout this thesis, they are normally defined as they are used and are also defined in Table 1.2.

Table 1.1: List of symbols used in this thesis

Symbol	Units	Description
E	[none]	Earth reference frame marker
P	[none]	Pendulum reference frame marker
R	[none]	Robot reference frame marker
b_x	[none]	Pendulum body x axis

Table 1.1 : *continued*

Symbol	Units	Description
b_y	[none]	Pendulum body y axis
b_z	[none]	Pendulum body z axis
ϕ	rad	Roll Euler angle
θ	rad	Pitch Euler angle
ψ	rad	Yaw Euler angle
Θ	rad	Vector of Euler angles
p	rad/s	Pendulum body angular velocity about b_x
q	rad/s	Pendulum body angular velocity about b_y
r	rad/s	Pendulum body angular velocity about b_z
${}^P\Omega$	rad/s	Pendulum body angular velocity vector
L_{EP}^{\leftarrow}	[none]	Pendulum body reference frame to Earth body reference frame
L_{ER}^{\leftarrow}	[none]	Robot body reference frame to Earth body reference frame
A_{PE}^{\leftarrow}	[none]	Converts $\dot{\Theta}$ to ${}^P\Omega$
${}^E x_p$	m	Pendulum global x position
${}^E y_p$	m	Pendulum global y position
${}^E z_p$	m	Pendulum global z position
${}^E X_p$	m	Pendulum global position vector
${}^P x_{cm}$	m	Pendulum local center of mass x component
${}^P y_{cm}$	m	Pendulum local center of mass y component
l_p	m	Length from point of rotation to pendulum center of mass in ${}^P z$
x	m	Global robot x position
y	m	Global robot y position
z	m	Global robot z position
${}^E X$	m	Vector of global robot position
P_{EP}^{\leftarrow}	[none]	Converts pendulum global linear velocity to global rotational velocity
KE_{pin}	J	Linear kinetic energy of pendulum

Table 1.1 : *continued*

Symbol	Units	Description
$KE_{p_{rot}}$	J	Rotational kinetic energy of pendulum
$KE_{p_{tot}}$	J	Total kinetic energy of pendulum
$KE_{g_{lin}}$	J	Linear kinetic energy of robot
$KE_{g_{rot}}$	J	Rotational kinetic energy of robot
$KE_{g_{tot}}$	J	Total kinetic energy of robot
KE	J	Total kinetic energy
PE	J	Total potential energy
M_P	Kg	Mass of the pendulum body
J_{xx}	$Kg\ m^2$	Moment of inertial of the pendulum about its x axis
J_{yy}	$Kg\ m^2$	Moment of inertial of the pendulum about its y axis
J_{zz}	$Kg\ m^2$	Moment of inertial of the pendulum about its z axis
$^P J$	$Kg\ m^2$	Moment of inertia tensor of the pendulum body
M_C	Kg	Total effective mass of the robot
J_{zzR}	$Kg\ m^2$	Moment of inertia of the robot about its z axis
PE	J	Potential energy of the system
g	m/s^2	Acceleration due to gravity
E_{pen}^z	m	Height of the center of mass of the pendulum
L	J	Lagrangian energy sum
f_x	N	Force produced by propellers in pendulum x
f_y	N	Force produced by propellers in pendulum y
l	m	Total length of the pendulum
l_{mot}	m	Length from the top of the pendulum to the center propeller
R_r	m	Radius of the ground robot
F_{w_i}	N	Force produced by the ith wheel
T_{f_ψ}	Nm	Torque produced from rolling resistance
F_{f_x}	N	Rolling resistance friction on robot x axis

Table 1.1 : *continued*

Symbol	Units	Description
F_{fy}	N	Rolling resistance friction on robot y axis
M	Kg	Mass of the robot
J_m	$Kg\ m^2$	Moment of inertia of the wheel motor rotor
K_g	[none]	Gear ration of wheel motor gearboxes
r_w	m	Radius of the wheels
m_w	Kg	Mass of an individual wheel
η_g	[none]	Efficiency of motor gearbox
η_m	[none]	Efficiency of wheel motors
K_{Q_w}	Nm/A	Wheel motor torque constant
K_{V_w}	$\frac{rad}{V\ s}$	Wheel motor back-emf constant
R_{m_w}	Ω	Resistance of wheel motors
V_{bat}	V	Voltage of wheel battery
a_H	[none]	Slope of H-bridge commands
u_{w_i}	[none]	Command for ith wheel
u_{w_g}	[none]	H-bridge command gain adjustment
$R_{\dot{x}}$	m/s	Robot frame x velocity
$R_{\dot{y}}$	m/s	Robot frame y velocity
μ_{r_x}	[none]	Rolling resistance coefficient for x axis
μ_{r_y}	[none]	Rolling resistance coefficient for y axis
μ_{r_ψ}	[none]	Rolling resistance coefficient for yaw
ω_i	rad/s	Velocity of the ith propeller
\tilde{J}_r	$Kg\ m^2$	Effective moment of inertia of the propeller and motor rotor
K_Q	Nm/A	Propeller motor torque constant
K_V	$\frac{rad}{V\ s}$	Propeller motor back-emf constant
R_m	Ω	Resistance of propeller motors
V_{quad}	V	Voltage of propeller battery

Table 1.1 : *continued*

Symbol	Units	Description
K_d	$\frac{Kg\ m^2}{rad^2}$	Drag of propeller
i_f	A	Propeller motor friction current
u_i	[none]	Command for ith propeller
u_{max}	[none]	Maximum propeller command
K_T	$\frac{Kg\ m}{rad^2}$	Propeller thrust constant
σ_T	Kg/rad	Propeller thrust velocity adjustment
v_i	m/s	Linear velocity of the ith propeller system
$u_{w_{max}}$	[none]	Maximum wheel command
u_{dead}	[none]	Wheel command dead zone offset
$u_{w_{i_{out}}}$	[none]	Effective command of the ith wheel
u_ϕ	[none]	Roll command
u_θ	[none]	Pitch command
u_ψ	[none]	Yaw command
u_x	[none]	x velocity command
u_y	[none]	y velocity command
u_{nom}	[none]	nominal propeller command
\dot{x}_{ref}	m/s	x velocity reference
\dot{y}_{ref}	m/s	y velocity reference
m_{quad}	Kg	Mass of quadrotor system
m_{rod}	Kg	Mass of pendulum rod
l_{quad}	m	Height of quadrotor system
r_{rod}	m	Radius of pendulum rod
$J_{m_{body}}$	$Kg\ m^2$	Moment of inertia of the quad system about its roll and pitch axes
$J_{quad_{zz}}$	$Kg\ m^2$	Moment of inertia of the quad system about its yaw axis
τ	s	Time rise of motor and propeller system
Λ	[mixed]	Vector of states

Table 1.1 : *continued*

Symbol	Units	Description
U	[none]	Vector of input commands
w	[none]	Vector of disturbance inputs
w_ψ	[none]	Yaw disturbance input
w_x	[none]	x axis disturbance input
w_y	[none]	y axis disturbance input
Λ^*	[mixed]	Linearization point of state vector
U^*	[none]	Linearization point of input commands
w^*	[none]	Linearization point of disturbance inputs
G_B	[none]	Gain used on input matrix for controller design
T	s	Sample period of the controller
Q_{xu}	[none]	LQG controller weighting matrix for states and inputs
Q_{wv}	[none]	LQG controller weighting matrix for estimating states and outputs
Q_I	[none]	LQG controller weighting matrix for tracking references

Table 1.2 List of common acronyms

Acronym	Description
RADA	Robotic Agricultural Data Acquisition (the system)
FPGA	Field Programmable Gate Array
CPU	Computational Unit
PID	Proportional Integral Derivative controller
LQG	Linear Quadratic Gaussian controller
LQR	Linear Quadratic Regulator
LQI	Linear Quadratic Regulator with integral tracking
LAN	Local Area Connection
PWM	Pulse-width Modulation
Lipo	Lithium polymer battery
LIDAR	Light detection and ranging
IMU	Inertial measurement unit
GPS	Global Positioning System
VRPN	Virtual Reality Peripheral Network library
SSH	Secure Shell

CHAPTER 2. SYSTEM DESCRIPTION

The RADA system was conceived to collect data about crops while traversing a field. This system primarily is meant for small scale research use as a small vehicle that can fit between two rows of corn. This leads to the use of an inverted pendulum on a cart system. This avoids the issues with having a large sensor mast rigidly attached to a vehicle with a small wheel base. With rough terrain or wind, the vehicle could possibly cause the system to tip over and damage itself or the crops it is trying to measure. A quadrotor or other aerial systems are not ideal because measurements like crop height should be more accurate if taken from a side view.

2.1 Platform

The physical system for this work consists of a 2DOF inverted pendulum attached to an omnidirectional (3DOF) cart. That is, the pendulum is free to both pitch and roll with respect to the cart, and the cart can both translate and rotate on a flat surface. The pendulum is actuated at the top by two orthogonal pairs of propellers, one pair for each degree of freedom, and the cart is actuated by four independent omnidirectional wheels. This setup is depicted in Figure 2.1.

The overall system design is a proof of concept for what could eventually be used for farming or other applications with needs for a long arm with sensors along the length. Especially when the sensor arm cannot be rigidly attached to the system due to terrain or other disturbances. The inverted pendulum is referred to as the pendulum arm, sensor arm and/or sensor mast. The pendulum is a simple hollow aluminum rod with a balsa wood quadrotor frame. The motor mounts for the quadrotor were designed by [9], a senior design team, to point the propellers outwards. After some testing it was determined that the propellers need to be mounted so

that air flows away from the system. The propeller and motors are attached to the motor controller through signal wires run up the pendulum arm and powered by a four cell Lithium polymer (Lipo) battery through a large cable running up the pendulum. The ground vehicle is based off Eris, omnidirectional robot platform developed by a previous senior design team, [10]. It can move linearly in both x and y axes as well control its heading with zero turning radius. The motors powering the wheels are brushed DC motors with gearboxes controlled by H-bridges. These H-bridges are controlled by a forward and reverse pulse-width modulation (PWM) channels for each motor in order to control both the speed and direction of each wheel. The wheel motors are powered by a two cell Lipo battery. The host CPU is an Atom processor running a custom Linux build based off of the Debian distribution. This allows for libraries like the Virtual Reality Peripheral Network library (VRPN) which requires an operating system. A camera localization system determines the 6DOF global position of both the pendulum and the ground vehicle and broadcasts the data over VRPN at 100 Hz. More information can be learned about the camera system from Chapter 3 of [8]. RADA can be controlled through an secure shell (SSH) connection and command prompt. RADA is connected to these other systems through a local area connection (LAN) or through a wireless network connection. Table 2.1 contains a list of the major commercial and open source components.

The current form of RADA clearly could not be taken outside as the ground vehicle cannot handle the rough terrain. The end goal for this initial phase of the project is to determine if the overall system can perform well enough to be useful in the field. This means that the propeller system can stabilize the pendulum arm without needing any assistance from the ground vehicle and can reject disturbances well enough to allow for sensors to produce useful data. If this initial system works well; a new version of the system could be designed to be capable of going out in the field. This first design is useful for two reasons. The first is that it will allow for the sensor arm to be subjected to disturbances in both the pitch, θ , and roll, ϕ , axes, which a non-omnidirectional vehicle could not do while indoors. Secondly it is a more interesting problem for an indoor prototype. Finally, the model derived and verified in this thesis can be simplified to act like new vehicle with fewer degrees of freedom.

For modeling the system, Euler angles were used to represent the orientation of both the pendulum and the ground vehicle. For Euler angles the order in which angles are resolved is important because each angle calculation can be thought of as an intermediate reference frame. This means that the order in which the angles are calculated changes the resulting orientation calculation. For this system the angles will be calculated in the order yaw, pitch, and then roll (ψ, θ, ϕ). The resulting transformation matrices will be discussed further in Chapter 2.2.

Eventually the system will need to be extended to have a sensor on the sensor mast that will include at least one camera, LIDAR, and other sensors that could be used to gather data on the target plants or objects. This will be taken into account by modeling the center of mass location of the inverted pendulum. Data processing of these sensors may happen on board or the data may be stored for processing later.

The overall architecture of RADA has been designed such that with the right ground vehicle the sensor mast will be well controlled in rough terrain and other disturbances. The goal is to model RADA well enough to track velocity commands while quickly rejecting the pendulum's angular displacement from vertical. This way there needs to be less correction of sensor data for orientation of the pendulum and better data from the sensors. This should allow the best scenario for successful data acquisition.

2.2 Reference frames

In order to derive the system's body dynamics, it is important to discuss reference frames. Euler angles are used in this model and the order that the angles are resolved affects the orientation. Because the same setup and camera system is used here as in [8], the same reference frame definitions are used in this derivation. The important transformation matrices are presented in Equation 2.1 through Equation 2.5.

Equation 2.1 converts linear body velocities of the pendulum body to the global reference frame of the camera system, while Equation 2.2 does the same conversion for the robot body. For the way the system is currently designed the yaw, ψ , is the same for both bodies of the system.

$$L_{EP}^{\leftarrow} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (2.1)$$

$$L_{ER}^{\leftarrow} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

It is important to note that L_{ER}^{\leftarrow} inverse is defined by Equation 2.3.

$$L_{ER}^T L_{ER}^{\leftarrow} = I \quad (2.3)$$

Therefore, Equation 2.4 is the reverse transformation.

$$L_{RE}^{\leftarrow} = L_{ER}^T \quad (2.4)$$

Equation 2.5 converts angular velocities from the global camera system reference frame, Euler angles, to rotational velocities in the pendulum's reference frame. This doesn't have an equivalent for the ground robot because since the robot can only rotate in yaw the yaw velocity and local rotational velocity are equivalent. If this model is extended to allow for the ground robot to pitch and roll this will need to be changed and the model would need to be extended for these new axes of rotation.

$$A_{PE}^{\leftarrow} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (2.5)$$

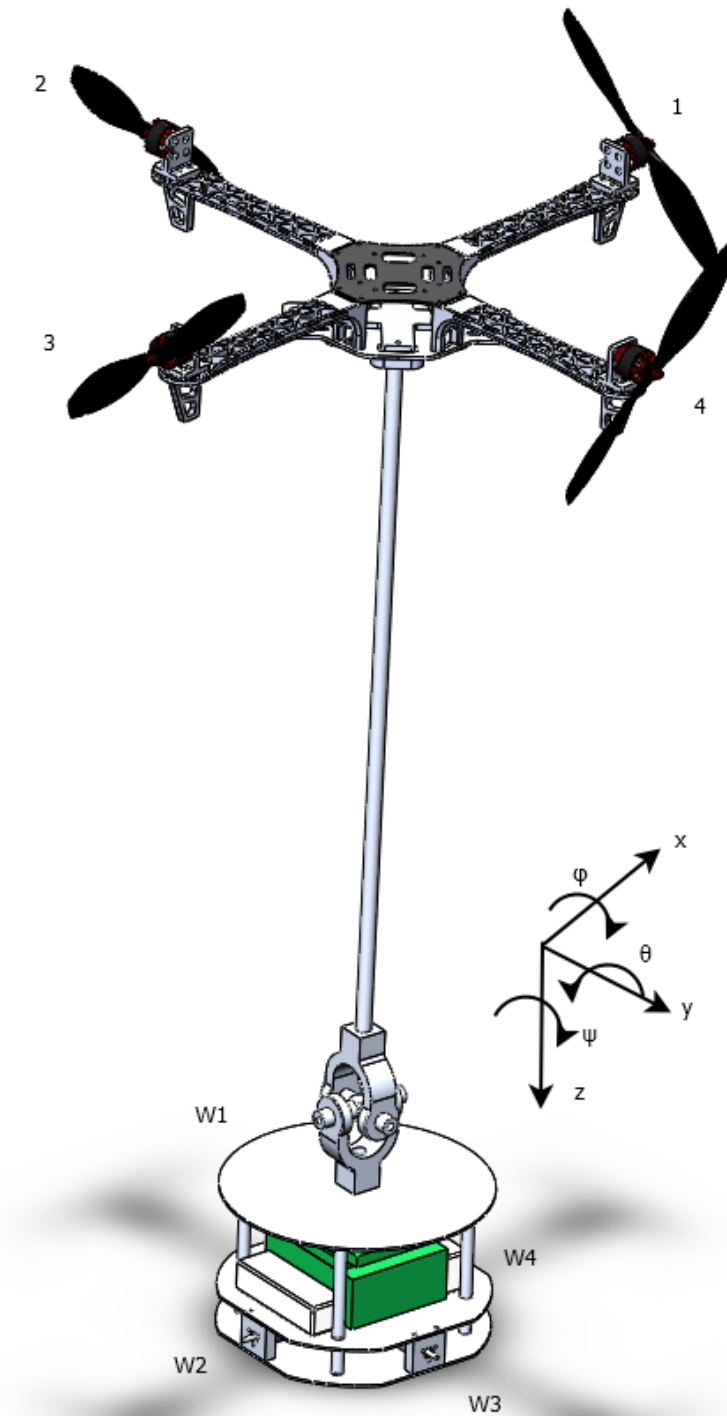


Figure 2.1 RADA system design concept without wheels.

Table 2.1 Major system components

Part	Description	Source
Pluto board	Motherboard with Intel Atom Processor used as the host CPU	[11]
MESA 4i68 FPGA	FPGA used as motor controller for PWM generation	[12]
Pololu Dual MC33926	Motor Driver Carrier (H-bridge) for wheel motors	[13]
Fauhaber 2232U006SR	Brushed DC motors for wheels	[14]
Fauhaber 20/1 14:1	Wheel gearbox paired with wheel motors	[15]
omnidirectional wheels	6 cm omnidirectional wheels	[10]
DJI 2212/920KV	DJI brushless DC motor	[16]
DJI Prop 8045	DJI 8 inch propeller	[16]
OPTO 30A	ESC used with the DJI 2212/920KV motors	[16]
BLHeli	ESC firmware	[17]
Optitrack V100	camera localization	[18]
Tracking Tools	Camera system software	[18]
VRPN	UDP localization broadcasting	[19]

CHAPTER 3. SYSTEM DYNAMICS DERIVATION

The body dynamics of RADA are derived using the Lagrangian method. This requires the calculation of the kinetic and potential energy contained by the system. The Lagrangian function is defined by Equation 3.1, where KE is the total system kinetic energy and PE is the total system potential energy. The dynamics of the actuators will be derived separately. Before the dynamics can be derived the position and orientation of the subsystems, pendulum and ground robot, written in terms of an inertial reference frame. This reference frame is called the earth reference frame throughout this paper.

$$L = KE - PE \quad (3.1)$$

3.1 Position and velocity description

The position and velocity, both linearly and rotationally, must be in the Earth reference frame in order to use the Lagrangian method. The main difficulty is defining the linear position of the pendulum. The linear position of the pendulum is described in Equation 3.2. Where ${}^E X_P$ is the center of mass of the pendulum in the earth reference frame, l_p is the length from the point of rotation of the pendulum to the center of mass of the pendulum, ${}^P x_{cm}$ is x distance in the pendulum reference frame from the center line of the pendulum to the center of mass of the pendulum, ${}^P y_{cm}$ is x distance in the pendulum reference frame from the center line of the pendulum to the center of mass of the pendulum, L_{EP} is the transformation matrix between local positions and global positions.

$${}^E X_P = L_{EP} \begin{bmatrix} {}^P x_{cm} \\ {}^P y_{cm} \\ -l_p \end{bmatrix} + {}^E X \quad (3.2)$$

$${}^E \dot{X}_P = \dot{L}_{EP} \begin{bmatrix} P x_{cm} \\ P y_{cm} \\ -l_p \end{bmatrix} + {}^E \dot{X} \quad (3.3)$$

Taking the derivative of Equation 3.2 results in Equation 3.3. Equation 3.4 will allow Equation 3.3 to be simplified and written in terms of the linear and angular velocity of the system. Where Θ is the vector of Euler angles.

$$\dot{L}_{EP} \begin{bmatrix} P x_{cm} \\ P y_{cm} \\ -l_p \end{bmatrix} = P_{EP} \dot{\Theta} \quad (3.4)$$

The pendulum linear velocity can then be written as in Equation 3.5.

$${}^E \dot{X}_P = P_{EP} \dot{\Theta} + {}^E \dot{X} \quad (3.5)$$

Equation 3.5 can be used for defining the systems kinetic energy.

3.2 Kinetic energy

The kinetic energy of the body can be written as the sum of the linear and rotational kinetic energy of the pendulum and the ground robot.

3.2.1 Pendulum

The linear kinetic energy of the pendulum can be written as Equation 3.6 by using the linear velocity found in Equation 3.5 where M_p is the mass of the pendulum system.

$$KE_{pin} = \frac{1}{2} M_p \left(P_{EP} \dot{\Theta} + {}^E \dot{X} \right)^T \left(P_{EP} \dot{\Theta} + {}^E \dot{X} \right) \quad (3.6)$$

Equation 3.6 can be expanded to Equation 3.7.

$$KE_{pin} = \frac{1}{2} M_p \left(\dot{\Theta}^T P_{EP}^T P_{EP} \dot{\Theta} + 2 {}^E \dot{X}^T P_{EP} \dot{\Theta} + {}^E \dot{X}^T {}^E \dot{X} \right) \quad (3.7)$$

The rotational kinetic energy of the pendulum can be written as Equation 3.8 where A_{PE}^{\leftarrow} converts the Euler angular velocities to local body angular velocities and ${}^P J$ is the moment of inertia tensor of the pendulum about its center of mass.

$$KE_{p_{rot}} = \frac{1}{2} \dot{\Theta}^T A_{PE}^T {}^P J A_{PE}^{\leftarrow} \dot{\Theta} \quad (3.8)$$

The total kinetic energy of the pendulum system is represented by Equation 3.9, is simply the sum of Equations 3.7 and 3.8.

$$KE_{p_{tot}} = \frac{1}{2} \dot{\Theta}^T \left(M_p P_{EP}^T P_{EP} + A_{PE}^T {}^P J A_{PE}^{\leftarrow} \right) \dot{\Theta} + M_p {}^E \dot{X}^T P_{EP}^{\leftarrow} \dot{\Theta} + \frac{1}{2} M_p {}^E \dot{X}^T {}^E \dot{X} \quad (3.9)$$

3.2.2 Ground robot

The ground robot in this model is an omnidirectional vehicle so it can move in any direction on the xy -plane without requiring rotation. The vehicle can yaw and it is assumed that the ground it is traversing is flat and smooth so the model need not deal with the pitch and roll of the ground vehicle. With these assumptions in mind the linear kinetic energy of the ground robot is given by Equation 3.10.

$$KE_{g_{lin}} = \frac{1}{2} M_c {}^E \dot{X}^T L_{RE}^T L_{RE}^{\leftarrow} {}^E \dot{X} \quad (3.10)$$

It can be shown that L_{RE}^T is equal its inverse allowing Equation 3.10 to be simplified to Equation 3.11.

$$KE_{g_{lin}} = \frac{1}{2} M_c {}^E \dot{X}^T {}^E \dot{X} \quad (3.11)$$

The rotational kinetic energy of the ground robot is given by Equation 3.12, where J_{zz_R} is the moment of inertia of the ground robot about its z-axis.

$$KE_{g_{rot}} = \frac{1}{2} \dot{\Theta}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_{zz_R} \end{bmatrix} \dot{\Theta} \quad (3.12)$$

Therefore, the total kinetic energy of the ground robot is given by Equation 3.13

$$KE_{g_{tot}} = \frac{1}{2} M_c {}^E \dot{X}^T {}^E \dot{X} + \frac{1}{2} \dot{\Theta}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_{zz_R} \end{bmatrix} \dot{\Theta} \quad (3.13)$$

3.2.3 Total kinetic energy

The total kinetic energy of the system is given by the sum of the total kinetic energy of the pendulum system and the total kinetic energy of ground robot as given by Equation 3.14.

$$KE = \frac{1}{2} \dot{\Theta}^T \left(M_p P_{EP}^T P_{EP} + A_{PE}^T {}^P J A_{PE} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_{zzR} \end{bmatrix} \right) \dot{\Theta} + M_p {}^E \dot{X}^T P_{EP} \dot{\Theta} + \frac{1}{2} (M_p + M_c) {}^E \dot{X}^T {}^E \dot{X} \quad (3.14)$$

3.3 Potential energy

The potential of the system is defined for this system in general as Equation 3.15 due to the definition of the reference frame where g is the force of gravity.

$$PE = -M_p g {}^E z_{pen} \quad (3.15)$$

Because the ground robot is defined as being on a level, smooth surface that can support the system and that the pendulum actuation system cannot produce enough upward force to lift the system off the ground it is assumed that ${}^E z = 0$. Due to the reference frame definition where the z-axis is in the direction of gravity, the potential energy needs to be inverted. This reduced potential energy is given by Equation 3.16, where ϕ is the Euler angle for roll and θ is the Euler angle for pitch.

$$PE = M_p g \begin{bmatrix} \sin \theta & -\sin \phi \cos \theta & -\cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} {}^P x_{cm} \\ {}^P y_{cm} \\ -l_p \end{bmatrix} \quad (3.16)$$

3.4 Lagrangian

Using Equations 3.14 and 3.16 the Lagrangian function is given by Equation 3.17.

$$\begin{aligned}
 L = \frac{1}{2} \dot{\Theta}^T & \left(M_p P_{EP}^T P_{EP} + A_{PE}^T {}^P J A_{PE} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_{zzR} \end{bmatrix} \right) \dot{\Theta} + M_p {}^E \dot{X}^T P_{EP} \dot{\Theta} \\
 & + \frac{1}{2} (M_p + M_c) {}^E \dot{X}^T {}^E \dot{X} - M_p g \begin{bmatrix} \sin \theta & -\sin \phi \cos \theta & -\cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} {}^P x_{cm} \\ {}^P y_{cm} \\ -l_p \end{bmatrix} \quad (3.17)
 \end{aligned}$$

The body dynamics can be defined by Equation 3.18.

$$\begin{aligned}
 \frac{d}{dt} \frac{\partial L}{\partial \dot{\Theta}} - \frac{\partial L}{\partial \Theta} & = \begin{bmatrix} -f_y (l + l_{mot}) \\ f_x (l + l_{mot}) \\ -R_r \sum_{i=1}^4 F_{w_i} + T_{f_\psi} \end{bmatrix} \\
 \frac{d}{dt} \frac{\partial L}{\partial {}^E \dot{X}} - \frac{\partial L}{\partial {}^E X} & = L_{EP} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} + L_{ER} \begin{bmatrix} -F_{w_1} + F_{w_3} + F_{f_x} \\ F_{w_2} - F_{w_4} + F_{f_y} \\ 0 \end{bmatrix} \quad (3.18)
 \end{aligned}$$

Equation 3.19 through Equation 3.22 define the components of the left hand side of the body dynamics equations from Equation 3.18.

$$\begin{aligned}
 \frac{d}{dt} \frac{\partial L}{\partial \dot{\Theta}} & = \left(M_p \left(\dot{P}_{EP}^T P_{EP} + P_{EP}^T \dot{P}_{EP} \right) + \dot{A}_{PE}^T {}^P J A_{PE} + A_{PE}^T {}^P J \dot{A}_{PE} \right) \dot{\Theta} \\
 & + \left(M_p P_{EP}^T P_{EP} + A_{PE}^T {}^P J A_{PE} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_{zzR} \end{bmatrix} \right) \ddot{\Theta} + M_p P_{EP}^T {}^E \dot{X} + M_p \dot{P}_{EP}^T {}^E \dot{X} \quad (3.19)
 \end{aligned}$$

$$\frac{\partial L}{\partial \Theta} = \frac{1}{2} \begin{bmatrix} \dot{\Theta}^T \left(M_P \left(\frac{\partial P_{EP}^T}{\partial \phi} P_{EP} + P_{EP}^T \frac{\partial P_{EP}}{\partial \phi} \right) + \frac{\partial A_{PE}^T}{\partial \phi} P J A_{PE} + A_{PE}^T P J \frac{\partial A_{PE}}{\partial \phi} \right) \dot{\Theta} \\ \dot{\Theta}^T \left(M_P \left(\frac{\partial P_{EP}^T}{\partial \theta} P_{EP} + P_{EP}^T \frac{\partial P_{EP}}{\partial \theta} \right) + \frac{\partial A_{PE}^T}{\partial \theta} P J A_{PE} + A_{PE}^T P J \frac{\partial A_{PE}}{\partial \theta} \right) \dot{\Theta} \\ \dot{\Theta}^T \left(M_P \left(\frac{\partial P_{EP}^T}{\partial \psi} P_{EP} + P_{EP}^T \frac{\partial P_{EP}}{\partial \psi} \right) + \frac{\partial A_{PE}^T}{\partial \psi} P J A_{PE} + A_{PE}^T P J \frac{\partial A_{PE}}{\partial \psi} \right) \dot{\Theta} \end{bmatrix} \\ + M_p g \begin{bmatrix} y_{cm} \cos \phi \cos \theta + l_p \sin \phi \cos \theta \\ -x_{cm} \cos \theta - y_{cm} \sin \phi \sin \theta + l_p \cos \phi \sin \theta \\ 0 \end{bmatrix} + M_p \begin{bmatrix} E \dot{X}^T \frac{\partial P_{EP}}{\partial \phi} \dot{\Theta} \\ E \dot{X}^T \frac{\partial P_{EP}}{\partial \theta} \dot{\Theta} \\ E \dot{X}^T \frac{\partial P_{EP}}{\partial \psi} \dot{\Theta} \end{bmatrix} \quad (3.20)$$

$$\frac{d}{dt} \frac{\partial L}{\partial E \dot{X}} = (M_p + M_c) E \ddot{X} + M_p \dot{P}_{EP} \dot{\Theta} + M_p P_{EP} \ddot{\Theta} \quad (3.21)$$

$$\frac{d}{dt} \frac{\partial L}{\partial E X} = 0 \quad (3.22)$$

Finally, Equation 3.18 can be solved for the rotational and linear accelerations of the system. Equation 3.23 solves for these accelerations. While this solution solves for six accelerations it was assumed that the floor will not allow the ground robot to accelerate downwards so it is always set to zero.

3.5 Wheel forces

The system is affected by two main forces acting on the system through the wheels. The first is the force produced by each wheel motor. The wheels are assumed to have perfect contact with the ground and are able to rotate without slipping similar to the cart on a rack and pinion gear described in [20]. The second force is rolling resistance, presented in [21], which creates a lumped parameter model including imperfect contact between the wheels and the ground; bearing frictions; and other friction forces.

3.5.1 Wheels force

The wheels being modeled as a perfect system without slip allows the wheel force to be written in terms of the linear and angular velocity of the cart. This also allows for the mass of the cart and the moments of inertia of the wheels to be absorbed into a single parameter as in Equation 3.24; where M_c is the lumped mass of the cart system, M is the mass of the cart, J_m is the inertia of the motor rotor, K_g is the gear box ratio, r_w is the radius of the wheels, and m_w is the mass of the wheels. This equation is the sum of the mass of the cart and the total moment of inertia of the wheels as well as the motor rotor translated into the wheel's reference frame.

$$M_c = M + \frac{4J_m K_g^2}{r_w^2} + 2m_w r_w^2 \quad (3.24)$$

The force produced the wheels can be represented by Equation 3.25, where η_g is the gear box efficiency, η_m is the motor efficiency, K_{Q_w} is the torque constant of the wheel motors, K_{V_w} is the velocity constant of the motors, R_{m_w} is the resistance of the wheel motors, V_{bat} is the voltage of the wheel battery, a_H is an H-bridge constant, u_{w_i} is the command input of the i th motor, and u_{w_g} is again for adjusting for the PWM command generation. Both the wheel force model in Equation 3.24 and Equation 3.25 are described in [20].

$$\begin{aligned} F_{w_1} &= \frac{\eta_g K_g \eta_m}{K_{Q_w} R_{m_w} r_w^2} \left(\frac{V_{bat} r_w a_H u_{w_1}}{u_{w_g}} - \frac{K_g}{K_{V_w}} \left(-R \dot{x} - R_r \dot{\psi} \right) \right) \\ F_{w_2} &= \frac{\eta_g K_g \eta_m}{K_{Q_w} R_{m_w} r_w^2} \left(\frac{V_{bat} r_w a_H u_{w_2}}{u_{w_g}} - \frac{K_g}{K_{V_w}} \left(R \dot{y} - R_r \dot{\psi} \right) \right) \\ F_{w_3} &= \frac{\eta_g K_g \eta_m}{K_{Q_w} R_{m_w} r_w^2} \left(\frac{V_{bat} r_w a_H u_{w_3}}{u_{w_g}} - \frac{K_g}{K_{V_w}} \left(R \dot{x} - R_r \dot{\psi} \right) \right) \\ F_{w_4} &= \frac{\eta_g K_g \eta_m}{K_{Q_w} R_{m_w} r_w^2} \left(\frac{V_{bat} r_w a_H u_{w_4}}{u_{w_g}} - \frac{K_g}{K_{V_w}} \left(-R \dot{y} - R_r \dot{\psi} \right) \right) \end{aligned} \quad (3.25)$$

3.5.2 Rolling resistance

The omnidirectional cart system modeled in Chapter 3.5.1 is assumed to be perfect. This assumption will work with a slight adjustment to correct for the non-idealities in the wheel bearings and in the contact between the wheels and the ground. This can be taken care of by the rolling resistance presented in Chapter 5.3 and Chapter 10.3 of [21]. The corresponding rolling resistance force and torques are represented by Equation 3.26, where μ_{r_x} , μ_{r_y} , and μ_{r_ψ} are the rolling resistance coefficients in the x , y , and yaw axes. While R_r is the radius of the cart.

$$\begin{aligned} F_{f_x} &= -\mu_{r_x} (M_c + M_p) g \operatorname{sign} \left({}^R\dot{x} \right) \\ F_{f_y} &= -\mu_{r_y} (M_c + M_p) g \operatorname{sign} \left({}^R\dot{y} \right) \\ T_{T_\psi} &= -\mu_{r_\psi} R_r \frac{M_c + M_p}{4} g \operatorname{sign} \left(\dot{\psi} \right) \end{aligned} \quad (3.26)$$

Overall this is a good model of the rolling resistance. It does have drawbacks, mainly the sign function is discontinuous at zero. This will be dealt with in two ways. For the purposes of linearization and controller design, $\operatorname{sign} \left({}^R\dot{x} \right)$, $\operatorname{sign} \left({}^R\dot{y} \right)$, and $\operatorname{sign} \left(\dot{\psi} \right)$ will be treated as disturbance inputs. For simulation purposes, Equation 3.27 is used because it is a contiguous function that approximates the sign function. In Equation 3.27 v is any velocity and C_t is a constant that effects the speed the function switches from -1 to 1. This equation comes from [22].

$$\operatorname{sign}(v) = \operatorname{sign}(v) \left(1 - \exp^{-C_t|v|} \right) \quad (3.27)$$

3.6 Quadrotor dynamics

The quadrotor system similar to the system presented in [8] with the motors angled outwards in order to stabilize the sensor mast as opposed to achieving flight.

3.6.1 Quadrotor motor dynamics

The quadrotor motor propeller system used in the RADA system are a set of four brushless DC motors. The dynamics of these motor systems are modeled using the same dynamics presented in [8]. The final results of the dynamical model shown in Equation 3.28, where K_V is the quad motor's velocity coefficient, K_Q is the quad motor's torque coefficient, K_d is the

drag coefficient of the propellers, R_m is the motor resistance, \tilde{J}_r is the total moment of inertia of the motor and propeller system, i_f is the friction current of the motor, V_{quad} is the voltage seen by the quad motors, u_i is the PWM duty cycle command, u_{max} is the maximum PWM command, and ω_i is the motor velocity.

$$\dot{\omega}_i = -\frac{1}{\tilde{J}_r K_Q K_V R_m} \omega_i - \frac{K_d}{\tilde{J}_r} \omega_i^2 - \frac{i_f}{\tilde{J}_r} + \frac{V_{quad}}{u_{max} \tilde{J}_r R_m K_Q} u_i \quad (3.28)$$

The motor propeller system is assumed to be fast in comparison to the rest of the system so for controller design purposes the quadrotor propeller transient dynamics will be reduced out of the system but these dynamics will remain in the model for simulation purposes. Equation 3.28 is solved for zero angular acceleration and it is assumed that the propeller speed must be positive, as is the case with quadrotor motor and propeller systems.

$$\omega_i = \frac{\frac{-1}{R_m K_V K_Q} + \sqrt{\frac{1}{R_m^2 K_V^2 K_Q^2} - 4K_d \left(\frac{i_f}{K_Q} - \frac{V_{quad} u_i}{u_{max} R_m K_V} \right)}}{2K_d} \quad (3.29)$$

3.6.2 Propeller wrench

The propeller system produces both forces and torques. This wrench will be converted into two equivalent force equations that act on the system. The force produced by the propeller system can be shown to be simplified into a few lumped parameter equations as shown by [8]. The total force produced by a single propeller is shown in Equation 3.30, where K_T is the propeller constant, δ_T is a velocity thrust adjustment, and v_i is the velocity of the motor perpendicular to the propeller disk.

$$f_i = K_T \omega_i^2 + \delta_T v_i \omega_i \quad (3.30)$$

The torque produced by the propeller system can be modeled with a simple lumped parameter model related to the drag of the propeller system as presented in [8]. Equation 3.31 represents the magnitude of the torque. This torque is applied about the axis of rotation of the motor in the opposite direction of the motor's rotation.

$$T_i = -K_d \omega_i^2 \quad (3.31)$$

While there are other force and torque parameters presented in [8] as the model was built up and tested it was determined that the model matches the system's behavior well enough that it is unnecessary to add further parameters to this model.

3.6.3 Total propeller force

The total propeller forces are a product of Equation 3.30 and Equation 3.31 and the physical layout of the motors and propellers. The quad motors are positioned as if they were on a quadrotor in the cross configuration, meaning one pair actuates pitch and the other actuates roll. One pair rotates clockwise while the other pair rotates counter-clockwise. The propellers are flipped upside down so that the air flow created by the propellers flows away from the body of the system. This removes any need to attempt to model the aerodynamic effects of pushing a large volume of air into a much smaller space. The total force produced by the propeller system is represented by Equation 3.32 and Equation 3.33.

$$f_x = K_T (\omega_3^2 - \omega_1^2) + \delta_T \left((l + l_{mot}) q - {}^R \dot{x} \right) (\omega_3 - \omega_1) + \frac{K_d}{l + l_{mot}} (\omega_4^2 - \omega_2^2) \quad (3.32)$$

$$f_y = K_T (\omega_4^2 - \omega_2^2) + \delta_T \left((l + l_{mot}) p + {}^R \dot{y} \right) (\omega_4 - \omega_2) - \frac{K_d}{l + l_{mot}} (\omega_3^2 - \omega_1^2) \quad (3.33)$$

3.7 Input dynamics

The input dynamics of RADA add additional actuator dynamics by taking into account the saturation points and dead zone of the H-bridges, electronic speed controllers (ESCs), and PWM generation.

3.7.1 Saturation and dead zones

Saturation is by far the simplest of these dynamics to take into account. The quad motors must have a positive command since the ESCs being used are not setup to reverse the motors, therefore the duty cycle ratio command, the ratio between the command and the maximum command will always be between zero and one. The valid input range for the quad motors is represented by Equation 3.34. This doesn't quite tell the whole story. An additional offset needs to be added to the command to reach the valid range of ESC commands. This matters

in implementation but for the modeling, simulation, and controller design it is an unnecessary addition and then subtraction that won't affect the behavior of the system.

$$0 \leq u_i \leq u_{max}, \quad \forall i \quad (3.34)$$

The wheels however are bi-directional, they can move the system both forwards and backwards. Their valid command range is represented by Equation 3.35.

$$-u_{wmax} \leq u_{w_i} \leq u_{wmax}, \quad \forall i \quad (3.35)$$

This takes care of saturation of the actuators. The H-bridges also have a dead zone where it doesn't activate. The behavior of the output of the command u_{w_i} is shown by Equation 3.36.

$$u_{w_{i,out}} = \begin{cases} u_{w_i} + u_{dead}, & u_{w_i} < -u_{dead} \\ 0, & -u_{dead} \leq u_{w_i} \leq u_{dead} \\ u_{w_i} - u_{dead}, & u_{w_i} > u_{dead} \end{cases} \quad (3.36)$$

3.7.2 Mixing

The system model thus far has eight inputs into the system, one each for the four propellers and one for each of the four wheels. The system only has five degrees of freedom so it would be ideal to reduce the inputs down to one per degree of freedom to reduce computational demands on the system. The propeller commands will be reduced into two commands, one for actuation of the pitch axis and the other for actuation of the roll axis. The wheel commands will be reduced to three inputs, one for x , y , and heading. The mixing Equation is defined by Equation 3.37, this mixer mixes the new inputs: u_ϕ , u_θ , u_ψ , u_x , and u_y ; into the inputs to each of the motors. The mixing matrix also has two pseudo non-linear extensions. The first fixes the wheel commands for a non-zero yaw position. The wheels act on the x and y axes differently as the heading of the vehicle changes. This mixer also allows for the system to change heading and still function correctly. The second is an offset added to u_x and u_y based off of the sign of the commanded velocity \dot{x}_{ref} and \dot{y}_{ref} to deal with the dead zone in the H-bridges and offset portion of the rolling resistance which is acting as a disturbance on the model seen by controllers.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_{w_1} \\ u_{w_2} \\ u_{w_3} \\ u_{w_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -\cos \psi & -\sin \psi \\ 0 & 0 & -1 & -\sin \psi & \cos \psi \\ 0 & 0 & -1 & \cos \psi & \sin \psi \\ 0 & 0 & -1 & \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_x + 6656 \operatorname{sign}(\dot{x}_{ref}) \\ u_y + 6656 \operatorname{sign}(\dot{y}_{ref}) \end{bmatrix} + \begin{bmatrix} u_{nom} \\ u_{nom} \\ u_{nom} \\ u_{nom} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.37)$$

CHAPTER 4. PARAMETER IDENTIFICATION

Most of the parameters that need to be identified can be identified using techniques developed in Chapter 5 of [8]. Many of the parameters are either directly identified using these procedures or indirectly measured and then calculated from these parameters. Any other procedures developed for parameter identification will be outlined in the corresponding section.

4.1 Masses

The masses of the system were just measured simply using the same digital scale in [8]. Table 4.1 contains all the masses needed to define the composite mass M_c , defined in Equation 3.24, and M_p , defined in Equation 4.1.

$$M_p = m_{quad} + m_{rod} \quad (4.1)$$

4.2 Lengths

Lengths are important to parameterize this model for a few reasons. First it allows for the approximation of many of the moments of inertia that would be difficult, unwieldy, or dangerous to measure using the moment of inertia identification technique developed in [8]. It also allows for the approximation of the center of mass of the sensor mast system.

Table 4.1 Mass parameter values

mass parameter	value	unit
M	2.761	Kg
m_w	0.034	Kg
m_{quad}	0.588	Kg
m_{rod}	0.548	Kg
g	9.81	m/sec^2

4.2.1 Procedure

Lengths are simply measured using an Imperial English tape measure then converted into metric units. The center of mass of the sensor mast system is calculated using Equation 4.2.

$$l_p = \frac{m_{rod}l}{2M_p} + \frac{m_{quad} \left(l + \frac{l_{quad}}{2} \right)}{M_p} \quad (4.2)$$

4.2.2 Parameter values

Table 4.2 shows the measured length parameters.

Table 4.2 Length parameters

length parameter	value	unit
l_{quad}	0.0508	m
l	1.8288	m
l_{mot}	0.0413	m
r_w	0.03	m
r_{rod}	0.0095	m
R_r	0.1143	m

4.3 Moments of Inertia

Most of the moments of inertia parameters for this system are calculated because directly measuring them would create safety issues or the estimated moment of inertia produced good enough results that direct measurement was unnecessary. Equation 4.3 through Equation 4.6 show how these moments of inertia are calculated. A few moments of inertia were measured, those values can be found in Table 4.3.

$$J_{zzR} = \frac{1}{2} M_c R_r^2 \quad (4.3)$$

$$J_{xx} = J_{yy} = J_{m_{body}} + m_{quad} \left(l + \frac{l_{quad}}{2} - l_p \right)^2 + \frac{m_{rod}}{12} l^2 + m_{rod} (l - l_p)^2 \quad (4.4)$$

$$J_{zz} = \frac{1}{2} m_{rod} r_{rod}^2 + J_{quad_{zz}} \quad (4.5)$$

$${}^P J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (4.6)$$

Table 4.3 Moment of inertia parameters parameters

moment of inertia parameter	value	unit
J_{mbody}	0.00449	$Kg\ m^2$
J_{quadzz}	0.00878	$Kg\ m^2$

4.4 Quad motor parameters

The quadrotor motor parameters are identified in the same manor presented in [8].

4.4.1 BLDC motor parameters

The only parameter not identified as in [8] for the quadrotor motor parameters is \tilde{J}_r . It was found that the audio method presented for that motor system didn't work well for this set of motors. Instead we obtained a Vernier Dual Range Force Sensor DFS-BTA set to 10N sensitivity as the propeller was not expected to produce more than 10N of force. this was then connected to a LabPro and data acquisition using the LoggerPro software. Data of the motor going from 50% command to 100%command was taken, then using the force measurements the time rise was calculated. From there the time rise of the motor was used to calculate \tilde{J}_r as in [8]. Equation 4.7 shows this calculation.

$$\tilde{J}_r = \frac{\tau}{K_V K_Q R_m} \quad (4.7)$$

4.4.1.1 Procedure for measuring time rise of the motor

1. Connect motor propeller system to Vernier Dual-Range Force Sensor and set the sensitivity to $\pm 10N$ [23]
2. Connect the force sensor to the LabPro interface [24]
3. Connect the CLB 2 to a computer with LoggerPro software installed [25]
4. Open up LoggerPro and configure the session for the force sensor
5. Capture data accelerating the motor from barley on to around 50% command
6. Plot the captured data and manually estimate the time rise
7. calculate \tilde{J}_r

4.4.1.2 Resulting parameters

The results of the time rise measurements for the motor propeller system is shown in Figure 4.1. All of the parameters for the propeller and motor system are shown in Table 4.4.

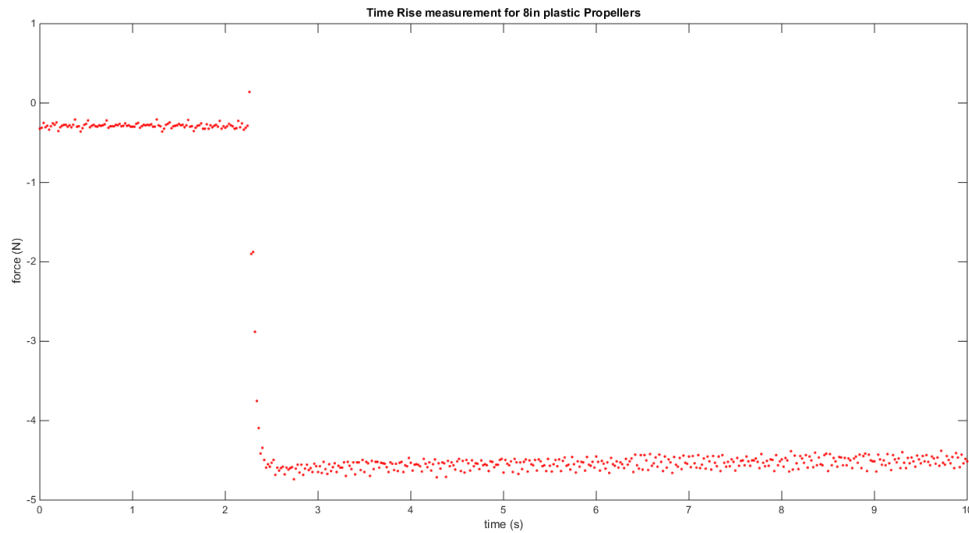


Figure 4.1 Time rise data capture.

Table 4.4 Quadrotor motor parameters

Quadrotor motor parameters	value	unit
τ	0.06	<i>seconds</i>
K_V	96.3422	<i>rad/Vsec</i>
K_Q	96.3422	<i>A/Nm</i>
R_m	0.2340	Ω
i_f	0.422	<i>A</i>
V_{quad}	14.8	<i>V</i>
u_{max}	30000	[<i>none</i>]
K_T	7.9×10^{-6}	<i>mKg</i>
K_d	2.3765×10^{-7}	<i>m²Kg</i>
δ_T	2.8194×10^{-4}	<i>Kg/sec</i>

4.5 Wheel Parameters

The wheel parameters can be obtained in the data sheets for the motor, [14], the H-bridge, [13], and the gear box, [15]. The motor resistance is the sum of the resistance of the motor and the resistance of the H-bridge itself. The parameters K_{V_w} and K_{Q_w} needed to be re-identified because the wheel motors are brushed DC motors that have worn brushes. The H-bridge dead zone and slope also needed to be identified.

4.5.1 Velocity coefficient identification

This was accomplished by supplying a voltage, using a power supply, to each motor; and using an oscilloscope to measure the steady-state velocity of each motor by measuring the encoder period. Then use least-squares to find the slope of the first order line relating the velocity to voltage input. This slope is the new K_{V_w} and K_{Q_w} .

4.5.1.1 Procedure

1. Connect one motor to power supply
2. Start the voltage at zero and increase by 0.5V
3. Measure the period of the encoder signal using an oscilloscope, [26].
4. Repeat step 3 four times to get an average period.

5. Convert the period of the encoders to motor speed.
6. Repeat steps 2-5 until the voltage supplied reaches the maximum voltage expected
7. Repeat steps 1-6 for the other wheel motors
8. Convert speed data to RPMs
9. Use least square to determine the line $y = ax + b$ to relate voltage command to RPMs.
10. The value a is K_{V_w} in RPM/V for use in the model this is converted to $rad/(V s)$

4.5.1.2 Results

The results of measuring K_{V_w} are shown in Figure 4.2. This data identified K_{V_w} as 1218.8 RPM/V . While is only about 2% larger than the datasheet value of 1190 RPM/V but because this is a slope of how fast the wheel will spin per volt applied it makes a difference in the model.

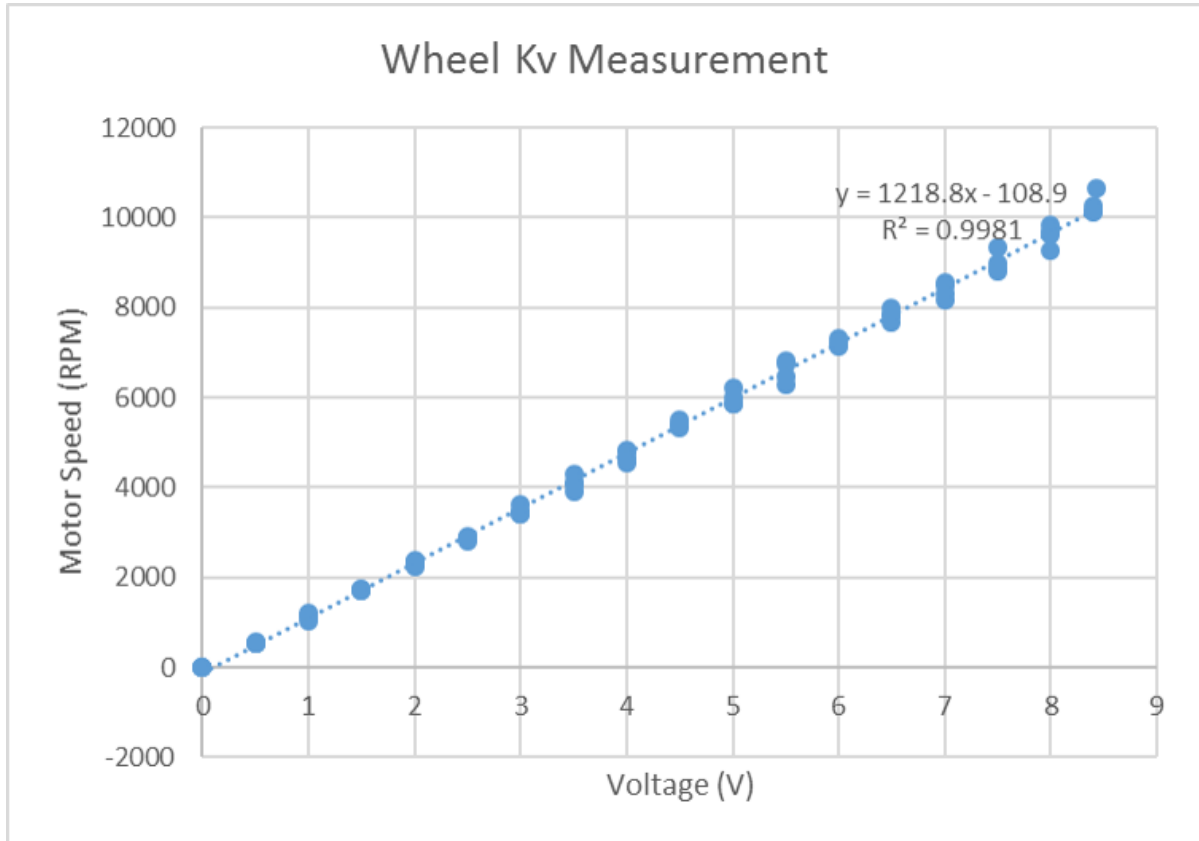


Figure 4.2 Measurement results for identifying K_{V_w} of the wheel motors.

4.5.2 H-bridge parameters

The H-bridge parameters were identified by giving open loop commands to the H-bridge and using the HP Oscilloscope from [8] to measure the output wave to determine the effective voltage seen by the motor then find the relationship between input PWM command and the output voltage seen by the motor. This produces a line with a linear slope and a non-zero y-intercept. The y-intercept constant is used to figure out where to set the dead zone range.

4.5.2.1 Procedure

1. Setup code to run PWM generation in open loop and disconnect the motor
2. Connect a 2 cell Lipo battery to the H-Bridge battery input and oscilloscope, [26], to the H-bridge output

3. Increase the command to the H-bridge and measure the voltage and duty cycle of the H-bridge output
4. Repeat step 3 until the maximum command is reached
5. Use least squares to calculate the slope of the H-bridge and the dead zone of the H-bridge

4.5.2.2 Results

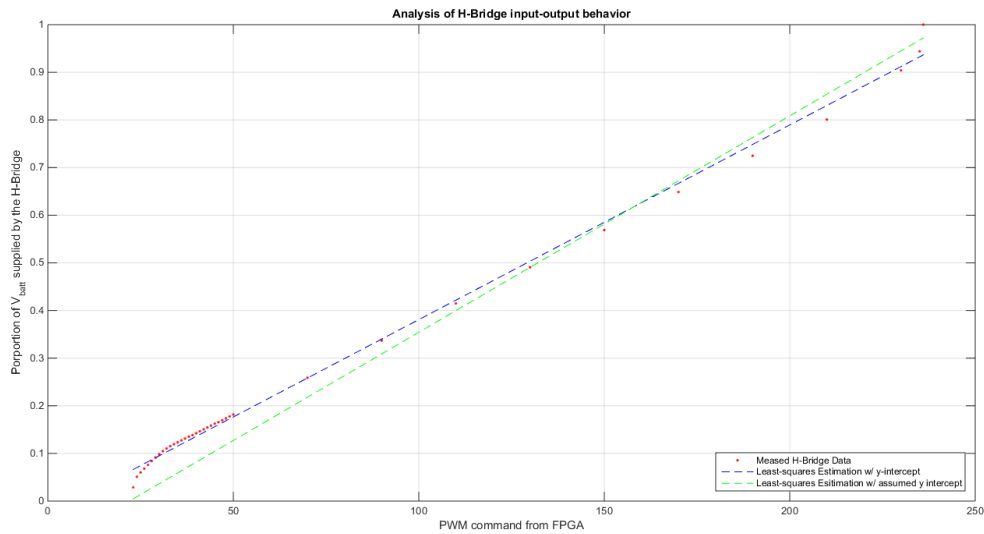


Figure 4.3 H-bridge parameter identification results.

4.5.3 Rolling Resistance

The rolling resistance coefficient is determined by giving two of the wheels, so the cart only moves in one direction, an open loop PWM command while the sensor mast isn't attached to the cart, and measure the steady state speed of cart. For this test the cart was placed on a table covered with a section of the floor material the cart normally drives over. This allowed steady state wheel velocity to be measured more easily using the same tachometer. Then calculate what the excess force the wheels are producing at steady state. Then μ_r can be found through Equation 4.8. This is repeated several times and then average to find the best coefficient. It turns out that there was enough of a difference between the two axes to have a separate

coefficient for both the x and y direction and the use the average of the two coefficients for yaw. This difference wouldn't normally be the case but one of the motors has a slightly bent axle it tougher to move in that direction.

$$\mu_r = \frac{F_{excess}}{2M_c g} \quad (4.8)$$

4.5.3.1 Procedure

1. Setup the robot to run in open loop with a constant command on a flat surface
2. Measure the battery voltage supplied to the H-bridge
3. Calculate the steady-state speed from the test
4. Calculate the force being produced by the wheels at the steady-state velocity
5. Use the force to calculate the rolling resistance coefficient

4.5.4 Parameters

The full set of wheel parameters is listed in Table 4.5.

Table 4.5 Wheel motor parameters

Wheel motor parameters	value	unit
K_{V_w}	127.6347	rad/Vsec
K_{Q_w}	127.6347	A/Nm
R_{m_w}	1.4075	Ω
V_{bat}	7.4	V
u_{max}	65535	[none]
μ_{r_x}	0.1014	[none]
μ_{r_y}	0.1014	[none]
μ_{r_ψ}	0.1014	[none]
K_g	14	[none]
η_m	0.87	[none]
η_g	0.80	[none]
a_H	0.0040	[none]
u_{w_g}	256	[none]
u_{dead}	5632	[none]
J_m	4.8×10^{-7}	Kg m ²

4.6 Battery voltage

The system uses two batteries, a four cell lipo for the propeller system and a two cell lipo for the wheels. For linearization and controller design the nominal battery voltage is used but for simulation and comparison of real data the actual voltage needs to be measured for more accurate simulation results.

4.6.1 Procedure for propeller motors

1. Setup code to run all four propellers at 50% throttle
2. Power off propellers
3. Connect multimeter to measure the voltage across one of the ESCs
4. Power on the propellers
5. Run the system at 50% throttle and measure the steady-state voltage during the test

4.6.2 Procedure for wheel motors

1. After test disconnect wheel battery from the system
2. Measure voltage of battery using multimeter

4.7 Complete parameter list

Table 4.6: All measured system parameters

Parameters	value	unit
M	2.761	Kg
m_w	0.034	Kg
m_{quad}	0.588	Kg
m_{rod}	0.548	Kg
g	9.81	m/sec^2

Table 4.6 : *continued*

Parameters	value	unit
l_{quad}	0.0508	m
l	1.8288	m
l_{mot}	0.0413	m
r_w	0.03	m
r_{rod}	0.0095	m
R_r	0.1143	m
J_{mbody}	0.00449	$Kg\ m^2$
J_{quadzz}	0.00878	$Kg\ m^2$
τ	0.06	<i>seconds</i>
K_V	96.3422	$rad/Vsec$
K_Q	96.3422	A/Nm
R_m	0.2340	Ω
i_f	0.422	A
V_{quad}	14.8	V
u_{max}	30000	[<i>none</i>]
K_T	7.9×10^{-6}	mKg
K_d	2.3765×10^{-7}	m^2Kg
δ_T	2.8194×10^{-4}	Kg/sec
K_{V_w}	124.6165	$rad/Vsec$
K_{Q_w}	124.6165	A/Nm
R_{m_w}	1.4075	Ω
V_{bat}	7.4	V
u_{max}	65535	[<i>none</i>]
μ_{r_x}	0.1562	[<i>none</i>]
μ_{r_y}	0.1765	[<i>none</i>]
μ_{r_ψ}	0.1663	[<i>none</i>]

Table 4.6 : *continued*

Parameters	value	unit
K_g	14	[none]
η_m	0.87	[none]
η_g	0.80	[none]
a_H	0.0040	[none]
u_{w_g}	256	[none]
u_{dead}	5632	[none]
J_m	4.8×10^{-7}	$Kg\ m^2$

CHAPTER 5. PID CONTROLLERS

For the purpose of verifying the model developed in Chapter 3 and Chapter 4 it was best to start by designing simple PID's. Because PID controllers can be hand tuned and refined without the need for the explicit use of the model, the controllers can be developed independent of the model and then tested against it. This allows for the model to be independently verified. A total of five PID's were designed, one to control each degree of freedom of the system. There is one controller for the pitch and roll position of the pendulum arm. One controls the yaw position of the entire system. The last two control the local body linear velocity of the cart. The goal of the system of controllers are to track the cart's linear velocity in x and y and to track yaw position commands. Because the pendulum arm will eventually support sensors for analyzing the surroundings of the vehicle it is desirable try to reject disturbances that stop the pendulum from balancing vertically. Normally controller result plots would be placed along with the controller design sections but because the performance of the system is dependent on the other controllers the controller behavior plots will be presented at the end.

5.1 Pitch and roll controllers

The pitch and roll controllers need to track zero degrees in angular position. In order to track angular position according to the model a PD controller would achieve this goal but the model does not capture all aspects of the physical system. The model makes an assumption that the center of mass is perfectly aligned in x and y but in reality it is slightly off center so a PD controller won't actually track zero angular position. Therefore, it is necessary to design PID controllers to allow the system to track zero angular position. This is an important observation for designing future controllers. This mismatch between the system and the model

won't be an issue once sensors are added to the pendulum arm as it will be easy to figure out where the center of mass of the pendulum system is, but without the sensors the center of mass will vary over time due to small and changing asymmetries of the physical system such as moving the location of ESCs and wires. This makes it difficult to characterize the center of mass of the current pendulum system. Using a PID controller will allow the model and the physical system behavior to match with only slight deviation in terms of the controller commands. The two axes are more or less symmetric so the same PID values are used for both axes.

The controllers were initially designed with the pendulum on a stationary base confined to one axis in order to simplify the controller design process. Once a stabilizing controller was found manually it was then compared against the model. Once the model had been sufficiently refined the model was then used to further adjust the PID controller design. Table 5.1 provides the constants for the final PID controller.

Table 5.1 PID controller values for pitch and roll.

Controller parameter	value	units
$P_{\phi/\theta}$	48000	[none]
$I_{\phi/\theta}$	18000	[none]
$D_{\phi/\theta}$	21000	[none]
$N_{\phi/\theta}$	16	[none]
f_s	100	Hz

5.2 Linear velocity control

The goal for controlling the linear velocity of the vehicle is to track the given velocity command quickly without causing the vehicle to slip. The linear velocity controllers were designed before the heading controller. Because of slight motor differences, the ground vehicle cannot hold a heading without control. Therefore, x and y velocity controllers were designed to control the body reference frame velocity. Initially, it was thought that a full PID controller would be required. Due to the sensors available to the controller, the cameras can only measure x and y position and the system needs to calculate velocity. Taking an additional derivative will start to really amplify the measurement noise to the point where there is no way to

compensate for the noise. The integral term is necessary to get the system to track a velocity command due to the rolling resistance disturbance. The initial PID designs were done using hand tuning methods in order to verify the model. After it became clear that the derivative term wouldn't help the controller performance it was decided to switch to a PI controller. It was initially thought that the proportional term of the controller should be much larger than the integral term because for most systems having a large integral term cause the system to respond slower. After discussion it was realized that designing a PI controller for velocity is equivalent to designing a PD controller for position control with a ramp input. This makes the weighting of the controller make much more sense than it originally did. The final controller is shown in Table 5.2.

Table 5.2 PID controller values for x and y velocity.

Controller parameter	value	units
$P_{R\dot{x}/R\dot{y}}$	100000	[none]
$I_{R\dot{x}/R\dot{y}}$	380000	[none]
f_s	100	Hz

5.3 Heading control

The heading (yaw) controller needs to be able to keep the robot pointed in the desired direction. In order to accomplish this goal a PID controller will be required. The main reason the integration term is needed in this case is the rolling resistance, which not only effects the x and y motion but it also effects the heading of vehicle. RADA needs the integration term to help correct for small errors in yaw over time. The second reason that the integral term is useful for this controller due to the slight discrepancy in motor parameters that could cause the vehicle to yaw when it starts to accelerate. This is something that the model presented in Chapter 3 wouldn't account for. This influenced how the controller was tuned. The model was used to design an initial controller, but from there hand tuning needed to be done in order to make sure the controller was strong enough to prevent the vehicle from yawing too much at the beginning of linear motion. It ended up that the original weighting ratios were acceptable but the overall gain needed to be increased. The cost of this gain increase is that tracking steps

become more difficult for the controller but they can still track a ramp heading command. The final controller weights are shown in Table 5.3.

Table 5.3 PID controller values for yaw position.

Controller parameter	value	units
P_ψ	40000	[none]
I_ψ	21800	[none]
D_ψ	14500	[none]
N_ψ	16	[none]
f_s	100	Hz

5.4 Control mixing

Due to the fact that the linear velocity PID controllers are designed to work on the body linear velocities of the system the mixing matrix from Equation 3.37 is not quite the correct mixing matrix for the PIDs. Instead the mixing is simplified by letting ψ always equal zero. This simplification is represented by Equation 5.1.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_{w1} \\ u_{w2} \\ u_{w3} \\ u_{w4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_x + 6656 \operatorname{sign}(\dot{x}_{ref}) \\ u_y + 6656 \operatorname{sign}(\dot{y}_{ref}) \end{bmatrix} + \begin{bmatrix} u_{nom} \\ u_{nom} \\ u_{nom} \\ u_{nom} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.1)$$

5.5 Controller testing

Controller testing was done systematically with a set procedure. What made this possible was a data analysis tool developed by the RADA Senior Design team from 2014-2015, [9].

5.5.1 Procedure

1. Setup the system and run the test.
2. Record the battery voltage seen by the quad ESCs and H-bridges.
3. Process data using MATLAB.
4. Setup the Simulink simulation of the non-linear system using the measured battery voltage and the reference commands recorded from the test data.
5. Run the simulation for the same length of time as the test.
6. Store the data from the simulation for latter use before plotting.
7. Plot the system behavior and the simulation behavior over each other and compare results.

5.5.2 Results

The results of testing the controllers in tandem in order to see if the controllers are able to achieve the total goals of all the controller together. Figure 5.1 show the total system behavior, and Figure 5.2 and Figure 5.3 show zoomed in versions of the plots.

The x and y velocity controllers settle out from a step in velocity to a non-zero velocity in about 0.9 seconds while stopping only takes 0.13 seconds. Both the pitch and roll axes have a disturbance peak amplitude 1.6° for accelerating from zero to 35 cm/s and 1.8° when stopping. The disturbance rejection time is measured as the time when the system is within 1% of the maximum of the disturbance. The pitch and roll axes take about 5.8 seconds from the start of the disturbance.

Overall the system and model behavior are similar and match very well in general. There are some small differences between the two systems mostly some additional oscillations and vibrations from actual pendulum. Overall the model behaves well enough to design model based controllers. More advanced controllers are necessary because the disturbance seen by the pitch and roll axes are rejected slowly and it is desirable to reduce time to reject the disturbances.

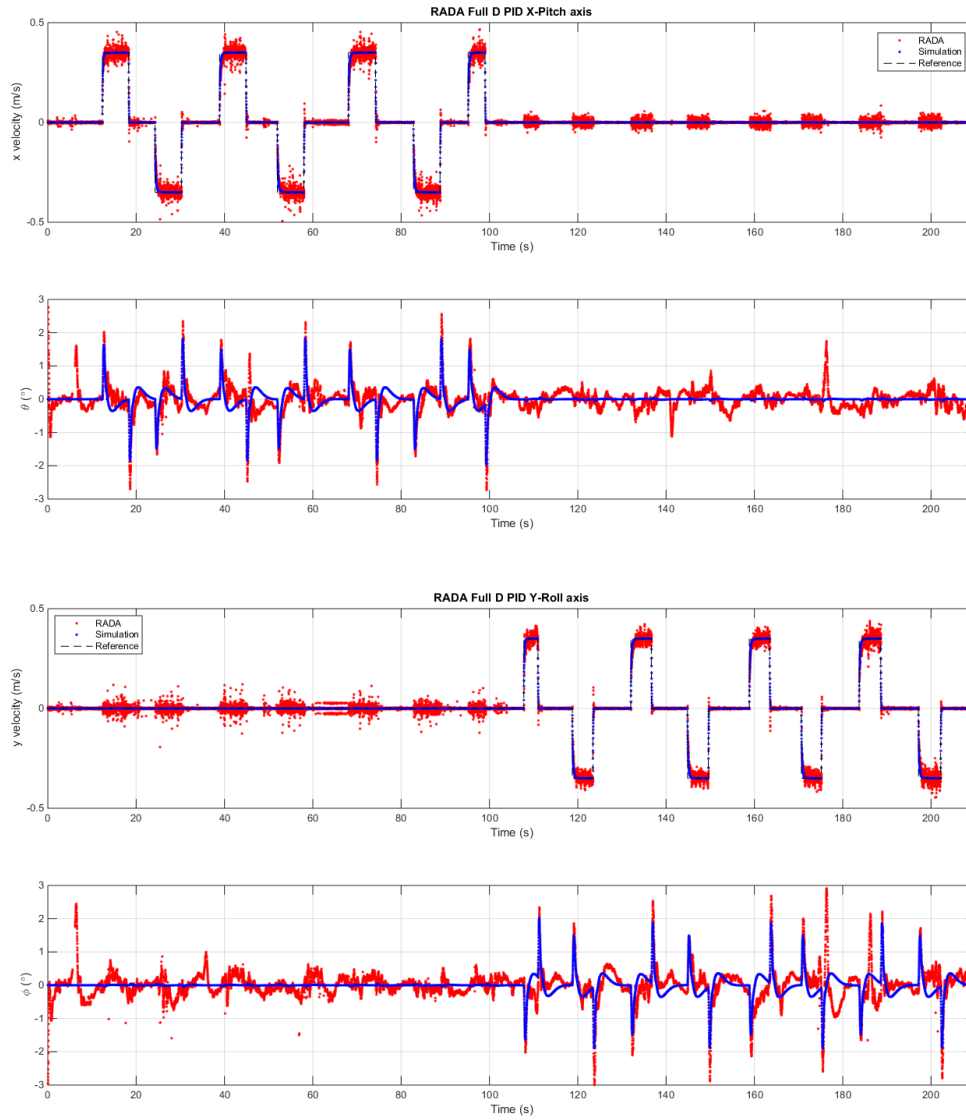


Figure 5.1 PID controller comparison.

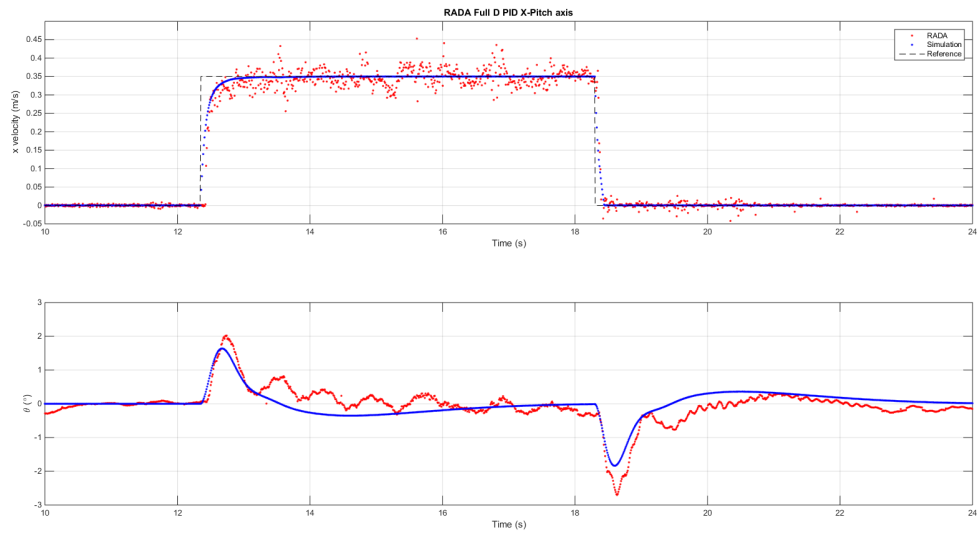


Figure 5.2 PID controller comparison zoomed in $x-\theta$.

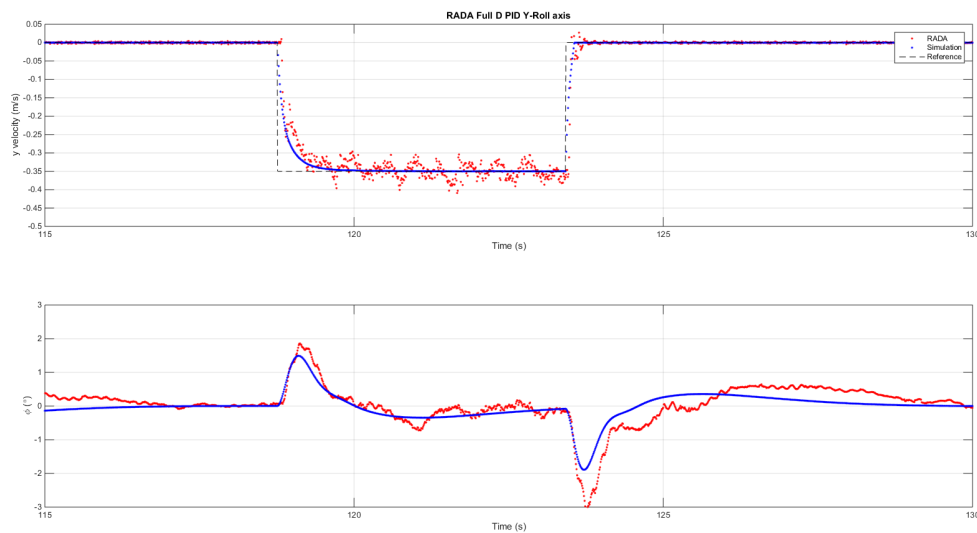


Figure 5.3 PID controller comparison zoomed in $y-\phi$.

CHAPTER 6. LQG CONTROLLER

It was decided to design a linear quadratic gaussian (LQG) controller, state estimator and regulator system, because the PID controller could be improved by increasing the speed of disturbance rejection by the pendulum and decrease amplitudes of these disturbances. Initially a linear quadratic regulator (LQR) was designed but this controller could not track commands for the same reason PD controllers would not have been sufficient. Then a linear quadratic integral (LQI) regulator was designed next. This would allow the regulator to track command inputs but this didn't work either because simply using a digital derivative to calculate the angular velocity states were not good enough to improve the system performance over the PIDs. Finally, it was determined that an estimator would need to be designed, so an LQG controller was the best option because it can simultaneously design an optimal linear regulator and an optimal linear estimator simultaneously for the best performance. The LQG controller can also be modified to track reference inputs. This will allow an LQG controller to fulfill all of the performance goals for the system.

6.1 Model linearization

The non-linear model described in Chapter 3 was linearized using the normal partial derivative method. For this system, it was determined to split the inputs into two sets. The first is the user defined inputs that can be used to control the system, U , defined by Equation 6.2, and the set of disturbance inputs, w , described in Equation 6.3. Where the disturbance inputs are defined by Equation 6.4. The linearized system has states Λ , defined by Equation 6.1.

Remember that for the linearization, controller design, and estimation; the quad motor system is assumed to reach the steady-state commanded velocity infinitely quickly and so the motor states are reduced from the system.

$$\Lambda = \begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (6.1)$$

$$U = \begin{bmatrix} u_{\phi} \\ u_{\theta} \\ u_{\psi} \\ u_x \\ u_y \end{bmatrix} \quad (6.2)$$

$$w = \begin{bmatrix} w_{\psi} \\ w_x \\ w_y \end{bmatrix} \quad (6.3)$$

$$\begin{aligned} w_{\psi} &= \text{sign}(\dot{\psi}) \\ w_x &= \text{sign}(R\dot{x}) \\ w_y &= \text{sign}(R\dot{y}) \end{aligned} \quad (6.4)$$

The main purpose of including the disturbance vector is that the sign function is discontinuous for an input of zero. This means that the system could not be linearized around zero velocity and for at least $\dot{\psi}$ it is necessary to use a linearization point of zero. It is also desirable to linearize the system at zero velocity in both the x and y axes because otherwise a piecewise linearization would be required for the system to be able to effectively move forward, backwards,

and stop. This would also require the design of three controllers and estimator systems. This would not be ideal as this type of linearization would require three times the memory for the controller plus additional computational overhead to handle the context switching between the three modes. By using the sign functions as disturbances the discontinuity issue is solved for linearization purposes. As discussed in Chapter 3.26 a continuous evaluation of the sign function is used for simulation. This is not ideal for linearization due to the large slope of this function at zero and it still does not have a nice evaluation at zero. With this out of the way the system can now be linearized around the equilibrium point (Λ^*, U^*, w^*) , describe by Equation 6.5.

$$\Lambda^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, U^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, w^* = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.5)$$

The resulting linearized Equation will look like the system of equations in Equation 6.6.

$$\begin{aligned} \dot{\Lambda} &= A\Lambda + BU + Ww \\ Y &= C\Lambda + D_U U + D_w w \end{aligned} \quad (6.6)$$

Where A is represented by Equation 6.7, B by Equation 6.8, W by Equation 6.9, C by Equation 6.10, D_U by Equation 6.11, and D_w by Equation 6.12.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2.8489 & 0 & -3.9536 & 0 & 0 & 0 & 0 \\ -2.8489 & 0 & 0 & 0 & -3.9536 & 0 & 0 & 0 \\ 7.7252 & 0 & 0 & 0 & 2.4127 & 0 & 0 & 0 \\ 0 & 7.7252 & 0 & -2.4127 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -11.6817 \end{bmatrix} \quad (6.7)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -3.5071 \times 10^{-7} & 2.1804 \times 10^{-5} & 0 & 0.0001 & 0 \\ -2.1804 \times 10^{-5} & -3.5071 \times 10^{-7} & 0 & 0 & 0.0001 \\ 0.0004 & 6.1693 \times 10^{-6} & 0 & 0 & -7.5686 \times 10^{-5} \\ -6.1693 \times 10^{-6} & 0.0004 & 0 & 7.5686 \times 10^{-5} & 0 \\ 0 & 0 & 0.0032 & 0 & 0 \end{bmatrix} \quad (6.8)$$

$$W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1.2838 & 0 \\ 0 & 0 & -1.2838 \\ 0 & 0 & 0.7834 \\ 0 & -0.7834 & 0 \\ -4.1482 & 0 & 0 \end{bmatrix} \quad (6.9)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6.10)$$

$$D_U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.11)$$

$$D_w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6.12)$$

It is straight forward enough to show that the controllability and observability matrices are both full rank making the system both observable and controllable. The linearized system can now be used to design linear regulators and estimators. The outputs of the system are assumed to be the angular position of the pendulum, yaw of the vehicle, and the global linear velocities of the vehicle. The camera system being used to measure the position of both the pendulum and the vehicle, but it cannot directly measure velocities. It was determined that the calculations of \dot{x} and \dot{y} are good enough not to require filtering either velocity calculation. However, the angular velocity calculations are not good enough, therefore the angular velocity calculation need to be filtered. For linear controller design this means that an estimator is needed. This leads to the need to design a LQG controller as opposed either a LQR or LQI.

6.2 Controller design

Early linear regulator designs had issues with being able to drive the eigenvalues of the system by changing the gains of the state weighting matrix Q . This shouldn't happen in a controllable system. It turns out that MATLAB 2014b has an issue with calculating regulators when the A and B matrices have large magnitude differences. Therefore the B matrix needs to be gained, in this case, so that the regulator calculation will work appropriately. The output regulator will also need to be gained by the same value to get the regulator for the actual system. In Equation 6.13 the max and min functions refer to the MATLAB max and min functions. In MATLAB the max function when operating on a 2D matrix returns the largest value in each column. So effectively this equation taking the minimum of the maximum is finding the smallest value of the set of the largest column values.

$$G_B = \frac{1}{\min(\max|B|)} = 8063 \quad (6.13)$$

The system still needs to be digitized before a controller and estimator can be designed. The system is sampled at 100 Hz using the standard equations to digitize the system demonstrated by Equation 6.14 and Equation 6.15, where $T = 0.01$. Equation 6.15 is also used to digitize the disturbance input matrix W except that G_B is set one.

$$A_d = e^{AT} \quad (6.14)$$

$$B_d = G_B \int_0^T e^{A(T-\tau)} B d\tau \quad (6.15)$$

For now, the estimator will not use the disturbance inputs in the design. If the estimator doesn't produce good enough estimations of the states of the system it would be good to try and add the disturbance inputs into the estimation system.

As discussed previously it was decided to design an LQG controller capable of tracking the outputs of this system. Using the MATLAB `lqg` function to design an LQG controller with tracking given the digital state-space system, a state and input weighting matrix for regulation, Q_{xu} , a input and state weighting matrix for the estimator, Q_{wv} , and an output weighting matrix for tracking, Q_I . Selecting the weights for these matrices should be fairly straight forward. Similar to Chapter 7.3.1 of [8], it was believed that the weights could be chose to match the

inverse square of the maximum desired deviation. This could not be used because some of the limitations of the system, like the noise floor of the cameras. This would cause an issue on a real system so the weights had to be found more through tuning than using the desired behavior of the system to determine the weights. Through much simulation and experimentation the weights in Equation 6.16 were found to produce a controller and estimator system that had adequate performance to meet the desired controller goals of RADA. The matrices Q_{xu} and Q_{wv} are further subdivided into two block diagonal matrices, while this doesn't necessarily need to be the case there was found to be no need to cross weight the actuator inputs with the states of the system. The same can be said of cross weighting the outputs of the linear model and the estimated state outputs that are feed into the regulator. These sub-matrices are defined in Equation 6.17 through Equation 6.20.

$$\begin{aligned}
 Q_{xu} &= \begin{bmatrix} Q_x & \mathbf{0} \\ \mathbf{0} & Q_u \end{bmatrix} \\
 Q_{wv} &= \begin{bmatrix} Q_w & \mathbf{0} \\ \mathbf{0} & Q_v \end{bmatrix} \\
 Q_I &= \begin{bmatrix} 800 & 0 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 60000 & 0 \\ 0 & 0 & 0 & 0 & 60000 \end{bmatrix} \tag{6.16}
 \end{aligned}$$

$$Q_x = \begin{bmatrix} 120 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 120 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1800}{\pi^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.17}$$

$$Q_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.18)$$

$$Q_w = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 25 \end{bmatrix} \quad (6.19)$$

$$Q_v = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (6.20)$$

The resulting system described by Equation 6.21, where Λ_{est} is the estimated state vector including tracking states, A_{lqg} represents the internal estimator based off of the state estimation, B_{lqg} is matrix relating to the input of the estimation system, Y_{ref} is the vector of reference inputs, Y is the set of measurements from the camera system, U is the set of inputs to the mixing matrix for the actuators, C_{lqg} is the quadratic regulator gained by Equation 6.13 in order to undo the gained B matrix in the system the LQG regulator was designed for, D_{lqg} is regulation for the system relating the current inputs to outputs, k is the current sample number, and T is the sample period of the controller.

$$\begin{aligned}\Lambda_{est} [kT + 1] &= A_{lqq}\Lambda_{est} [kT] + B_{lqq} \begin{bmatrix} Y_{ref} [kT] \\ Y [kT] \end{bmatrix} \\ U [kT] &= C_{lqq}\Lambda_{est} [kT] + D_{lqq} \begin{bmatrix} Y_{ref} [kT] \\ Y [kT] \end{bmatrix}\end{aligned}\quad (6.21)$$

The matrix A_{lqq} is defined by Equation 6.22, due to the size of the matrix it couldn't fit on one page and needed to be split into two sub-matrices, A_{lqq1} and A_{lqq2} . The matrix A_{lqq1} is represented in Equation 6.23 and the matrix A_{lqq2} is represented by Equation 6.24.

$$A_{lqq} = \begin{bmatrix} A_{lqq1} & A_{lqq2} \end{bmatrix}\quad (6.22)$$

$$A_{lqq1} = \begin{bmatrix} 0.7762 & 0 & 0 & 0 & 0.0030 & 0.0091 & 0 \\ 0 & 0.7762 & 0 & -0.0030 & 0 & 0 & 0.0091 \\ 0 & 0 & 0.9267 & 0 & 0 & 0 & 0 \\ 0 & -0.0475 & 0 & 0.6865 & 0 & 0 & -0.0057 \\ 0.0475 & 0 & 0 & 0 & 0.6865 & 0.0057 & 0 \\ -2.1157 & 0 & 0 & 0 & 0.1645 & 0.8202 & 0 \\ 0 & -2.1157 & 0 & -0.1645 & 0 & 0 & 0.8202 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\quad (6.23)$$

$$A_{lqg_2} = \begin{bmatrix} 0 & 0.0039 & 0 & 0 & 0 & -0.0074 \\ 0 & 0 & 0.0039 & 0 & 0.0074 & 0 \\ 0.0086 & 0 & 0 & 0.02532 & 0 & 0 \\ 0 & 0 & 0.02864 & 0 & 2.0985 & 0 \\ 0 & -0.02864 & 0 & 0 & 0 & 2.0985 \\ 0 & 0.7827 & 0 & 0 & 0 & -1.4658 \\ 0 & 0 & 0.7827 & 0 & 1.4658 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.24)$$

The matrix B_{lqg} is defined by Equation 6.25, due to the size of the matrix it couldn't fit on one page and needed to be split into two sub-matrices, $B_{lqg_{ref}}$ and $B_{lqg_{meas}}$. The matrix $B_{lqg_{ref}}$ shown in Equation 6.26 is the portion of the estimator input that relates to the reference inputs. The matrix $B_{lqg_{meas}}$ shown in Equation 6.27 is the portion of the estimator input related to the state measurements.

$$B_{lqg} = \begin{bmatrix} B_{lqg_{ref}} & B_{lqg_{meas}} \end{bmatrix} \quad (6.25)$$

$$B_{lqg_{ref}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.0100 & 0 & 0 & 0 & 0 \\ 0 & 0.0100 & 0 & 0 & 0 \\ 0 & 0 & 0.0100 & 0 & 0 \\ 0 & 0 & 0 & 0.0100 & 0 \\ 0 & 0 & 0 & 0 & 0.0100 \end{bmatrix} \quad (6.26)$$

$$B_{lqg_{meas}} = \begin{bmatrix} 0.2207 & 0 & 0 & 0 & -0.0022 \\ 0 & 0.2207 & 0 & 0.0022 & 0 \\ 0 & 0 & 0.0557 & 0 & 0 \\ 0 & 0.0264 & 0 & 0.0205 & 0 \\ -0.0264 & 0 & 0 & 0 & 0.0205 \\ 1.4947 & 0 & 0 & 0 & -0.0050 \\ 0 & 1.4947 & 0 & 0.0050 & 0 \\ 0 & 0 & 0.0524 & 0 & 0 \\ -0.0100 & 0 & 0 & 0 & 0 \\ 0 & -0.0100 & 0 & 0 & 0 \\ 0 & 0 & -0.0100 & 0 & 0 \\ 0 & 0 & 0 & -0.0100 & 0 \\ 0 & 0 & 0 & 0 & -0.0100 \end{bmatrix} \quad (6.27)$$

The matrix $C_{l_{qg}}$, Equation 6.28, relates the estimate state matrix Λ_{est} to the actuator commands U . Like both $A_{l_{qg}}$ and $B_{l_{qg}}$ this matrix had to be split up into three sub-matrices: $C_{l_{qg1}}$, Equation 6.29; $C_{l_{qg2}}$, Equation 6.30; and $C_{l_{qg3}}$, Equation 6.31.

$$C_{l_{qg}} = \begin{bmatrix} C_{l_{qg1}} & C_{l_{qg2}} & C_{l_{qg3}} \end{bmatrix} \quad (6.28)$$

$$C_{l_{qg1}} = \begin{bmatrix} -180351.7108 & 2900.8682 & 0 & -84.1008 & -5228.6856 \\ -2900.8682 & -180351.7108 & 0 & 5228.6856 & -84.1008 \\ 0 & 0 & -114211.0493 & 0 & 0 \\ 0 & -8590.5627 & 0 & -210052.2678 & 0 \\ 8590.5627 & 0 & 0 & 0 & -210052.2678 \end{bmatrix} \quad (6.29)$$

$$C_{l_{qg2}} = \begin{bmatrix} -47689.2319 & 767.0577 & 0 & 206607.2323 \\ -767.0577 & -47689.2319 & 0 & 3323.1752 \\ 0 & 0 & -5587.5173 & 0 \\ 0 & 3614.7004 & 0 & 0 \\ -3614.7004 & 0 & 0 & 12808.0790 \end{bmatrix} \quad (6.30)$$

$$C_{l_{qg3}} = \begin{bmatrix} -3323.1753 & 0 & -803.5457 & -49957.7480 \\ 206607.2323 & 0 & 49957.7480 & -803.5457 \\ 0 & 164133.1219 & 0 & 0 \\ -12808.0790 & 0 & 1716866.3198 & 0 \\ 0 & 0 & 0 & 1716866.3198 \end{bmatrix} \quad (6.31)$$

Finally, $D_{l_{qg}}$ relates the current reference inputs and system measurements to the actuator commands, show in Equation 6.32.

$$D_{l_{qg}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.32)$$

The controller was implemented in C++ code and run on the host processor on the ground vehicle in order to test and compare the system behavior and non-linear model simulation. The same procedure described in Chapter 5.5.1.

6.3 Results

After testing the controller on the current system and comparing it with the model, the data was processed. Figure 6.1 was produced; it compares the actual system behavior to that of the non-linear system. The blue dotted line represents the simulation and the red dotted line represents the actual system behavior. Figure 6.2 shows a zoomed in view of the x and pitch, θ , axis. While Figure 6.3 shows a zoomed in view of the y and roll, ϕ , axis.

Overall the system takes about 1.5 seconds to linearly accelerate in x or y from zero to ± 35 cm/s, while it only takes about 0.5 seconds to linearly decelerate from ± 35 cm/s in either x or y . The pitch and roll axes have a maximum disturbance amplitude of 1° when accelerating from zero to ± 35 cm/s. When stopping the pitch and roll disturbances max out at about 2° and it only takes about 2.5 seconds to reject this disturbance. Most of the difference between the simulation and implementation of the controller are some oscillations due to vibrations and a few other un-modeled dynamics. Overall the system behavior is acceptable and much faster. The model could probably be refined more but this controller demonstrates that the model is good enough to design model based controller and the simulations behave reasonable well.

6.3.1 Comparison to PID controller

The LQG controller performs better in controlling the pendulum because it only takes about 2.5 seconds to reject the disturbance caused by accelerating the cart where the PIDs took 5.8 seconds to reject the disturbance, an increase in speed by about 57%. The magnitude of this disturbance when accelerating from zero to ± 35 cm/s is reduced by about 0.6° or 62.5% of the PID disturbance; but increases by 0.2° or 111.11% of the PID disturbance, when the cart comes to a stop. While the cart now takes 1.5 seconds to accelerate from zero to ± 35 cm/s compared to 0.9 seconds that it took the PID controllers to accelerate the system or 166.67% of the acceleration time of the PID. The cart takes 0.5 seconds to come to a stop under the LQG

controller compared to the 0.13 seconds it took the cart to stop using the PID controller. This is a 384.6% change in the stopping time. The increase in the carts acceleration and stopping time isn't too worrying because the cart is meant to travel long distances between stops so this time difference isn't a big deal. The improved acceleration response of the pendulum is important because the system will be able to reject pendulum disturbance faster and keep the pendulum very near vertical most of the time.

Figure 6.4 through Figure 6.7 show a comparison of the system behavior in simulation and implementation of both the PID and LQG controllers where data has been shifted so that the simulation and implementation results align for both systems. In these figures the red dots represent the actual implementation of the LQG, the green dots represent the data from the actual PID implementation, the blue dots represent the LQG simulation data, the magenta dots represent the PID simulation data, and the black dashed line represents the reference velocity. There is one plot each for comparing the acceleration and stopping data for both axes as this was the easiest way to make both responses line up. Finally, the testing each controller one right after the other and swapping wheel and propeller batteries. This way both tests would have reasonably similar voltages so the comparison is valid.

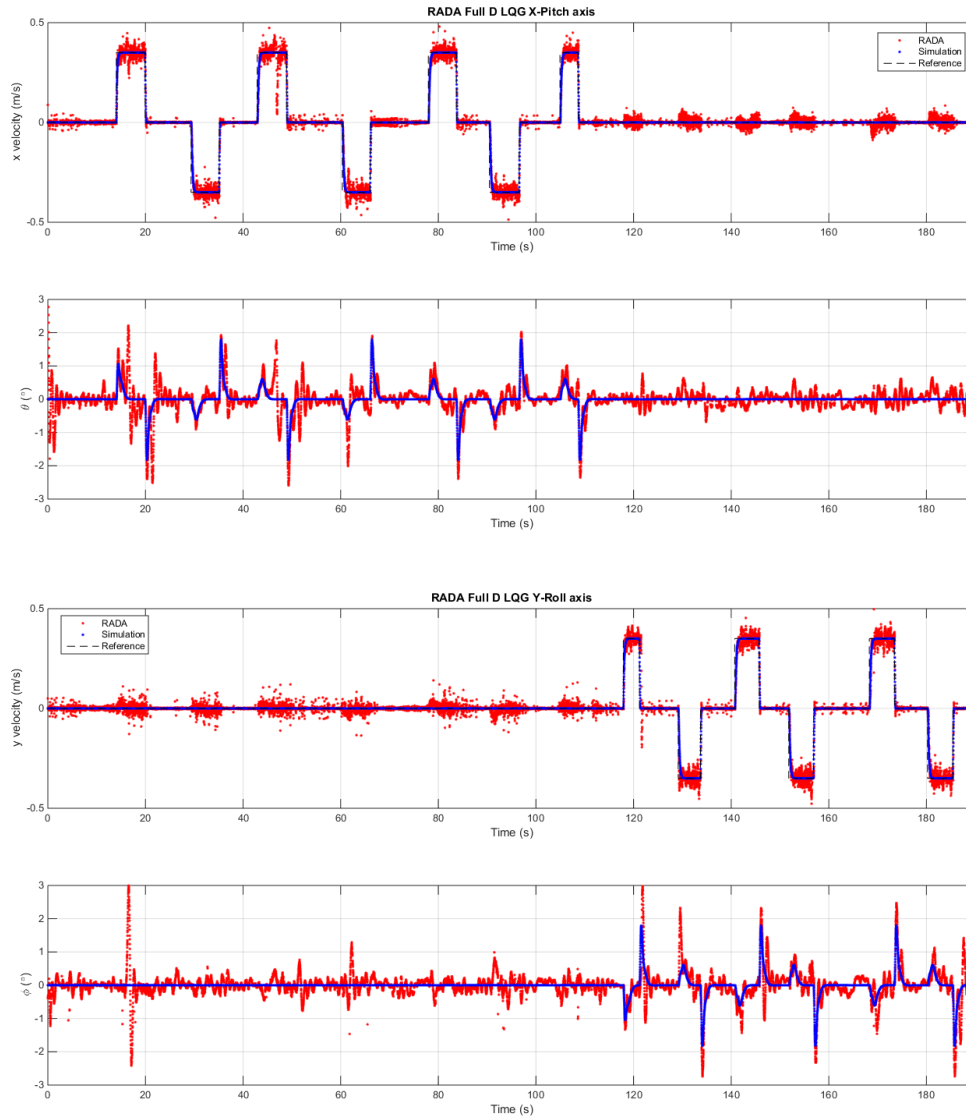


Figure 6.1 LQG controller comparison.

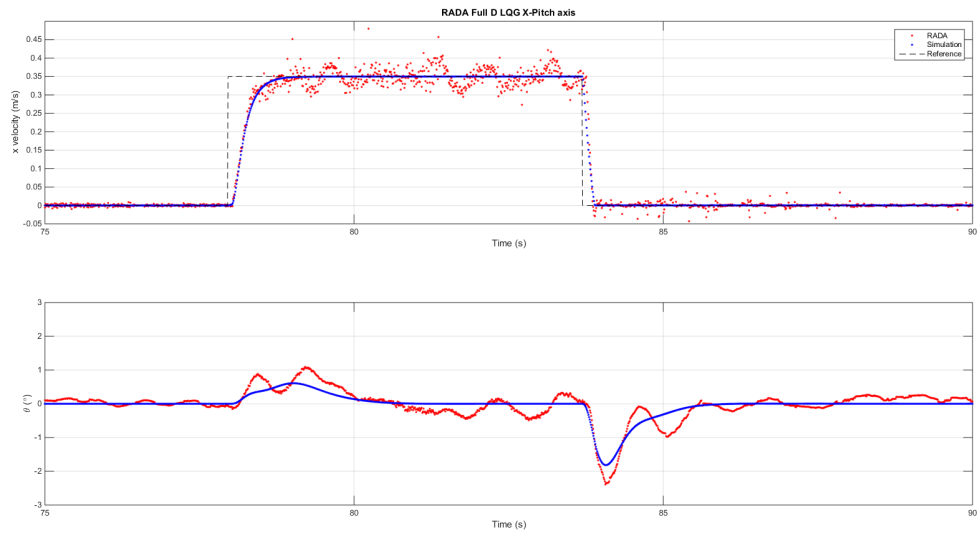


Figure 6.2 LQG controller comparison zoomed in $x-\theta$.

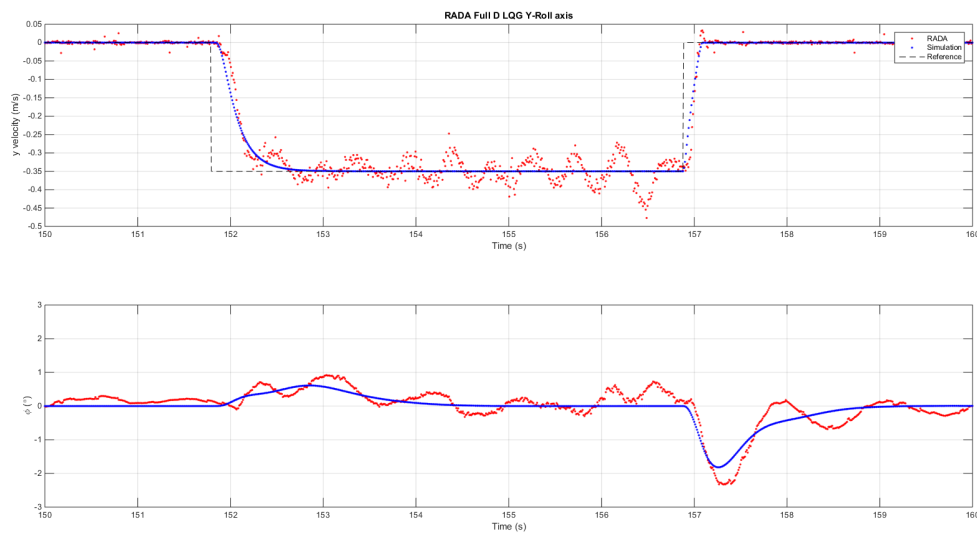


Figure 6.3 LQG controller comparison zoomed in $y-\phi$.

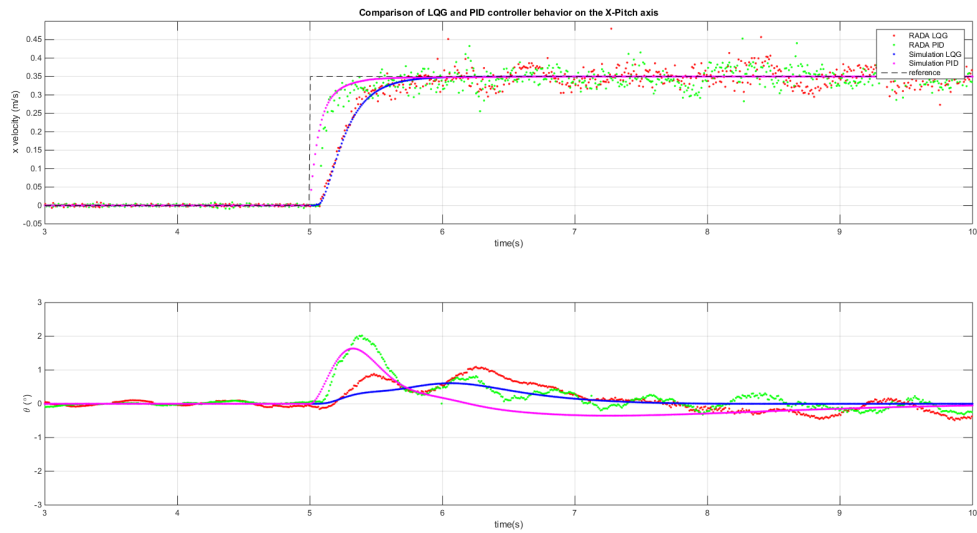


Figure 6.4 Comparison of the LQG and PID in $x-\theta$ when accelerating.

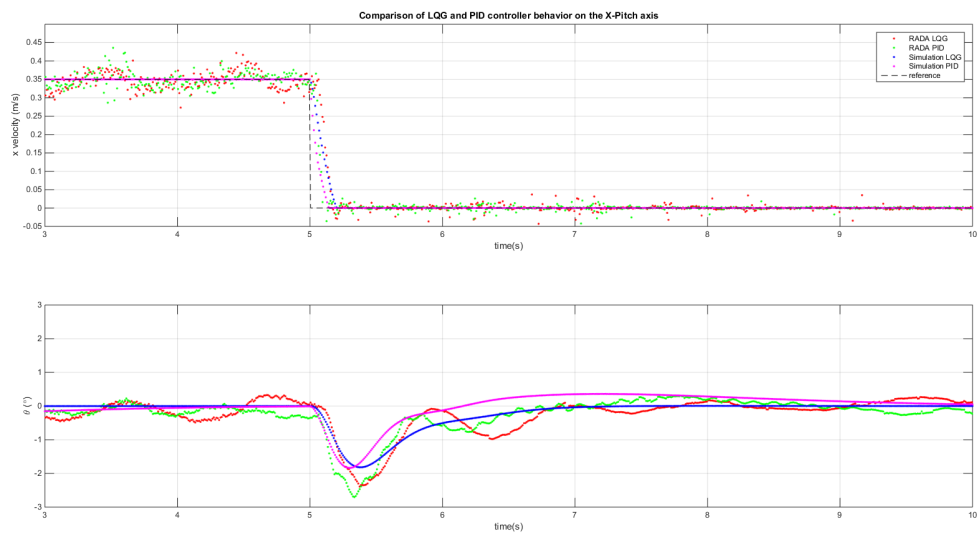


Figure 6.5 Comparison of the LQG and PID in $x-\theta$ when stopping.

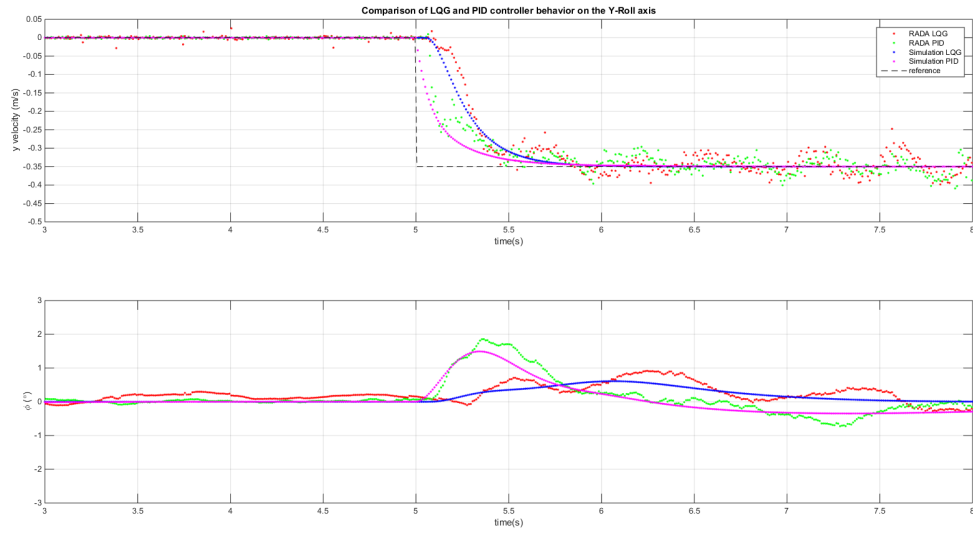


Figure 6.6 Comparison of the LQG and PID in $y-\phi$ when accelerating.

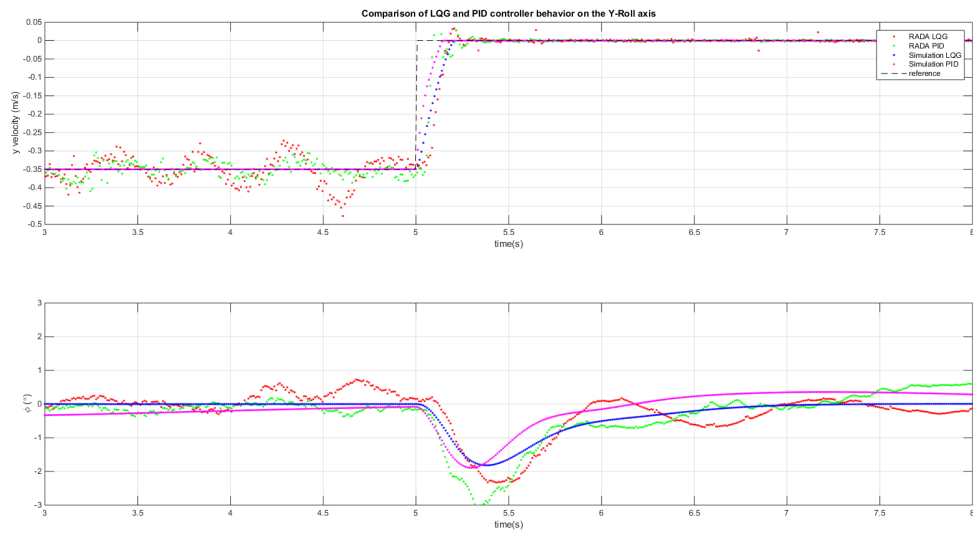


Figure 6.7 Comparison of the LQG and PID in $y-\phi$ when stopping.

CHAPTER 7. SEPARATE LQG CONTROLLER DESIGNS

In Chapter 6 a LQG controller was designed that operated on the whole system at once. There are some situations in which this may not be desirable. Because this system's running time will be limited by the capacity of the quadrotor battery, and there is only so much space available for batteries, it may be desirable to design a system that allow quadrotors to land on top of the sensor arm, articulate its motors to balance the sensor arm and leave when its batteries get too low. This would require controllers to be designed that decouple the ground robot control and pendulum control. One way to accomplish this is to divide the system into the ground robot, the states $(\psi, \dot{x}, \dot{y}, \dot{\psi})$, and the pendulum, the states $(\phi, \theta, \dot{\phi}, \dot{\theta})$. Both of these new systems would only have the inputs related directly to those states removing any knowledge of the cross-coupling of the two systems.

Overall the design of these separate LQG controllers started by splitting the linear system into two subsystems, one for the pendulum and one for the ground vehicle. Then each subsystem is sampled at 100 Hz. Next an LQG controller is designed for the pendulum system. This controller is tested with open loop ground robot commands. Next a LQG controller is designed for the ground vehicle and tested with the LQG controller for the pendulum system.

7.1 Subsystem division

The system is divided into two subsystems, with the linearization of each subsystem done about the same points as in Chapter 6.1.

The first is the pendulum system represented by Equation 7.1. Where pendulum's states are defined by Equation 7.2 and the pendulum's inputs are defined by Equation 7.3.

$$\dot{\Lambda}_{pen} = A_{pen}\Lambda_{pen} + B_{pen}U_{pen} \quad (7.1)$$

$$Y_{pen} = C_{pen}\Lambda_{pen} + D_{pen}U_{pen}$$

$$\Lambda_{pen} = \begin{bmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} \quad (7.2)$$

$$U_{pen} = \begin{bmatrix} u_{\phi} \\ u_{\theta} \end{bmatrix} \quad (7.3)$$

Where A_{pen} is represented by Equation 7.4, B_{pen} by Equation 7.5, C_{pen} by Equation 7.6, and D_{pen} by Equation 7.7.

$$A_{pen} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 7.7252 & 0 & 0 & 0 \\ 0 & 7.7252 & 0 & 0 \end{bmatrix} \quad (7.4)$$

$$B_{pen} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 3.8356 \times 10^{-4} & 6.1693 \times 10^{-6} \\ -6.1693 \times 10^{-6} & 3.8356 \times 10^{-4} \end{bmatrix} \quad (7.5)$$

$$C_{pen} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (7.6)$$

$$D_{pen} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (7.7)$$

The second system is the ground robot system which is represented by Equation 7.8. Where the robot's states are defined by Equation 7.9, its inputs are defined by Equation 7.10, and the disturbance inputs are the same as w originally defined in Chapter 6.1.

$$\dot{\Lambda}_{bot} = A_{bot}\Lambda_{bot} + B_{bot}U_{bot} + W_{bot}w \quad (7.8)$$

$$Y_{bot} = C_{pen}\Lambda_{bot} + D_{U_{bot}}U_{bot} + D_{w_{bot}}w$$

$$\Lambda_{bot} = \begin{bmatrix} \psi \\ \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} \quad (7.9)$$

$$U_{bot} = \begin{bmatrix} u_{\psi} \\ u_x \\ u_y \end{bmatrix} \quad (7.10)$$

Where A_{bot} is represented by Equation 7.11, B_{bot} by Equation 7.12, W_{bot} by Equation 7.13, C_{bot} by Equation 7.14, $D_{U_{bot}}$ by Equation 7.15, and $D_{w_{bot}}$ by Equation 7.16.

$$A_{bot} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -3.9536 & 0 & 0 \\ 0 & 0 & -3.9536 & 0 \\ 0 & 0 & 0 & -11.6817 \end{bmatrix} \quad (7.11)$$

$$B_{bot} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2402^{-4} & 0 \\ 0 & 0 & 1.2402^{-4} \\ 0.0032 & 0 & 0 \end{bmatrix} \quad (7.12)$$

$$W_{bot} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1.2838 & 0 \\ 0 & 0 & -1.2838 \\ -4.1428 & 0 & 0 \end{bmatrix} \quad (7.13)$$

$$C_{bot} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (7.14)$$

$$D_{U_{bot}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.15)$$

$$D_{w_{bot}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.16)$$

With both subsystems defined LQG controllers can now be designed for both subsystems, starting with the unstable pendulum system.

7.2 LQG controller for the pendulum system

The linear pendulum subsystem is digitized using the same process used in Chapter 6.2. For the purposes of this design, the gain G_B is replaced with the gain $G_{B_{pen}}$ shown in Equation 7.17.

$$G_{B_{pen}} = 2607 \quad (7.17)$$

The weighting matrices for the pendulum LQG controller design were initially based off the controller designed in Chapter 6.2, but these weights were able to be strengthened after some experimentation. The weighting matrices used for the pendulum's LQG are shown in Equation 7.18. The resulting controller takes the form of Equation 7.19.

$$Q_{xu_{pen}} = \begin{bmatrix} 1200 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 0 & 300 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q_{wv_{pen}} = \begin{bmatrix} 0.2000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (7.18)$$

$$Q_{I_{pen}} = \begin{bmatrix} 8500 & 0 \\ 0 & 8500 \end{bmatrix}$$

$$\begin{aligned}\Lambda_{pen_{est}} [kT + 1] &= A_{pen_{lqg}} \Lambda_{pen_{est}} [kT] + B_{pen_{lqg}} \begin{bmatrix} Y_{pen_{ref}} [kT] \\ Y_{pen} [kT] \end{bmatrix} \\ U_{pen} [kT] &= C_{pen_{lqg}} \Lambda_{pen_{est}} [kT] + D_{pen_{lqg}} \begin{bmatrix} Y_{pen_{ref}} [kT] \\ Y_{pen} [kT] \end{bmatrix}\end{aligned}\quad (7.19)$$

The matrix $A_{pen_{lqg}}$ is represented by Equation 7.20.

$$A_{pen_{lqg}} = \begin{bmatrix} 0.8808 & 0 & 0.0090 & 0 & 0.0041 & 0 \\ 0 & 0.8808 & 0 & 0.0090 & 0 & 0.0041 \\ -1.2163 & 0 & 0.8039 & 0 & 0.8285 & 0 \\ 0 & -1.2163 & 0 & 0.8039 & 0 & 0.8285 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\quad (7.20)$$

Matrix $B_{pen_{lqg}}$ is shown by Equation 7.21.

$$B_{pen_{lqg}} = \begin{bmatrix} 0 & 0 & 0.1159 & 0 \\ 0 & 0 & 0 & 0.1159 \\ 0 & 0 & 0.5589 & 0 \\ 0 & 0 & 0 & 0.5589 \\ 0.0100 & 0 & -0.0100 & 0 \\ 0 & 0.0100 & 0 & -0.0100 \end{bmatrix}\quad (7.21)$$

Equation 7.22 defines the matrix $C_{pen_{lqg}}$.

$$C_{pen_{lqg}} = \begin{bmatrix} -191457.5248 & 3079.4999 & -51207.6131 & 823.6492 & 215924.1054 & -3473.0326 \\ -3079.4999 & -191457.5248 & -823.6492 & -51207.6131 & 3473.0326 & 215924.1054 \end{bmatrix}\quad (7.22)$$

$$D_{pen_{lqg}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\quad (7.23)$$

This controller can now be tested while giving the ground robot open-loop commands.

7.2.1 Pendulum LQG with open-loop robot results

The system was tested under an open-loop wheel command of 26880. Otherwise the same testing procedure was used from Chapter 5.5.1 in order to compare this test to a corresponding

simulation. The results of this test are shown in Figure 7.1. The peak disturbance in θ is about 2.4° when the system is accelerating from a stop and takes about 3.81 seconds to reject the disturbance. When the cart is stopping the peak disturbance in pitch is about 3.2° and it took the system about 3.17 seconds to reject this disturbance. This is faster than the PID controller but slower than the full system LQG controller. The peak disturbance is larger for this controller than either the PID controller set or the full system LQG controller. It makes sense that a subsystem LQG controller would be slower than the full system LQG controller, as it has less information about the disturbance than the full system controller. It also makes sense that this controller would reject the disturbance caused by the cart stopping faster because the disturbance is a shorter duration. It is hard to judge the peak disturbance of the pendulum under this controller as the cart is going faster than it was with the previous controllers presented in Chapter 5 and Chapter 6. A better comparison will be made after the cart is also controlled with its own subsystem LQG controller. The continued presents of oscillations in the pitch data shouldn't be surprising as splitting the controller doesn't change that the model doesn't perfectly capture every detail of the system.

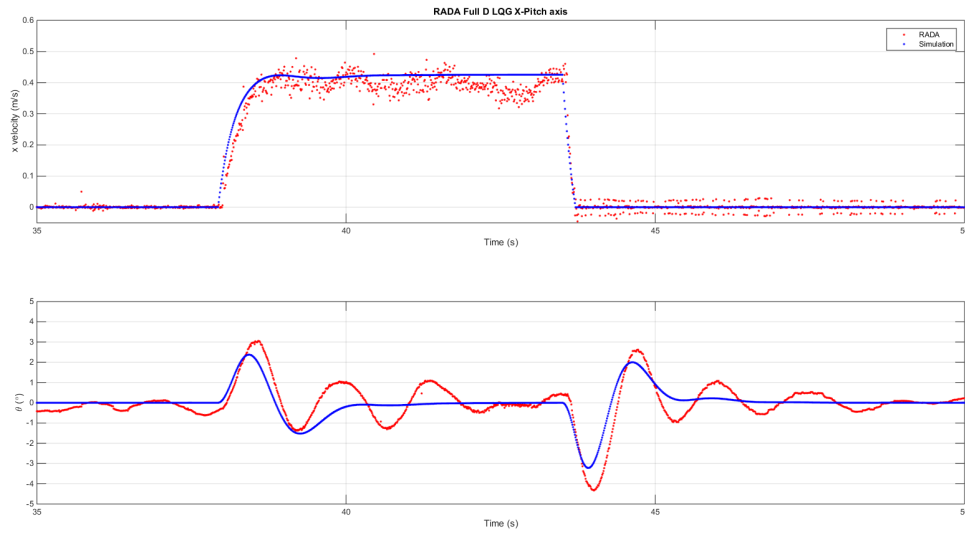


Figure 7.1 LQG control of the pendulum with open-loop cart commands.

7.3 Ground robot LQG controller design

The linear ground robot subsystem is digitized using the same process used in Chapter 6.2. For the purposes of this design, the gain G_B is replaced with the gain $G_{B_{bot}}$ shown in Equation 7.24.

$$G_{B_{bot}} = 8063 \quad (7.24)$$

The weighting matrices for the ground robot LQG controller design was initially based off the controller designed in Chapter 6.2, but these weights were strengthened after some experimentation. The weighting matrices used for the pendulum's LQG are shown in Equation 7.25.

$$\begin{aligned}
Q_{xu_{bot}} &= \begin{bmatrix} \frac{1800}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 700 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 700 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
Q_{wv_{bot}} &= \begin{bmatrix} 0.2000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \\
Q_{I_{bot}} &= \begin{bmatrix} 500 & 0 & 0 \\ 0 & 60000 & 0 \\ 0 & 0 & 60000 \end{bmatrix}
\end{aligned} \tag{7.25}$$

The resulting controller takes the form of Equation 7.26. The matrix $A_{bot_{lqg}}$ is represented by Equation 7.27.

$$\begin{aligned}
\Lambda_{bot_{est}} [kT + 1] &= A_{bot_{lqg}} \Lambda_{bot_{est}} [kT] + B_{bot_{lqg}} \begin{bmatrix} Y_{bot_{ref}} [kT] \\ Y_{bot} [kT] \end{bmatrix} \\
U_{bot} [kT] &= C_{bot_{lqg}} \Lambda_{bot_{est}} [kT] + D_{bot_{lqg}} \begin{bmatrix} Y_{bot_{ref}} [kT] \\ Y_{bot} [kT] \end{bmatrix}
\end{aligned} \tag{7.26}$$

$$A_{bot_{1qg}} = \begin{bmatrix} 0.8273 & 0 & 0 & 0.0086 & 0.0253 & 0 & 0 \\ 0 & 0.5865 & 0 & 0 & 0 & 2.0599 & 0 \\ 0 & 0 & 0.6667 & 0 & 0 & 0 & 2.0599 \\ -3.7496 & 0 & 00.7207 & 4.9664 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.27)$$

Matrix $B_{bot_{1qg}}$ is defined by Equation 7.28.

$$B_{bot_{1qg}} = \begin{bmatrix} 0 & 0 & 0 & 0.1551 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0198 \\ 0 & 0 & 0 & 0.2938 & 0 & 0 \\ 0.0100 & 0 & 0 & -0.0100 & 0 & 0 \\ 0 & 0.0100 & 0 & 0 & -0.0100 & 0 \\ 0 & 0 & 0.0100 & 0 & 0 & -0.0100 \end{bmatrix} \quad (7.28)$$

$C_{bot_{1qg}}$ is defined by Equation 7.29. The matrix is too large to fit on a page so the matrix is further subdivided in to matrix $C_{bot_{1qg1}}$, Equation 7.30, and matrix $C_{bot_{1qg2}}$, Equation 7.31.

$$C_{bot_{1qg}} = \begin{bmatrix} C_{bot_{1qg1}} & C_{bot_{1qg2}} \end{bmatrix} \quad (7.29)$$

$$C_{bot_{1qg1}} = \begin{bmatrix} -114211.0493 & 0 & 0 & -5587.5172 \\ 0 & -225907.8938 & 0 & 0 \\ 0 & 0 & -225907.8938 & 0 \end{bmatrix} \quad (7.30)$$

$$C_{bot_{1qg2}} = \begin{bmatrix} 164133.1219 & 0 & 0 \\ 0 & 169380.7967 & 0 \\ 0 & 0 & 169380.7967 \end{bmatrix} \quad (7.31)$$

$$D_{bot_{1qG}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.32)$$

Finally, $D_{bot_{lqg}}$ is defined by Equation 7.32. With the ground robot's LQG controller defined for both subsystem LQG controllers can be tested together.

7.4 Results

Figure 7.2 shows the subsystem LQG controllers behavior in the x -pitch axis. Figure 7.3 shows the subsystem LQG controllers behavior in the y -roll axis. Overall the pendulum controller is able keep the peak disturbance to about 2.5° when accelerating from zero and when the cart stops the peak disturbance is 2.6° . The pendulum controller about 3.25 seconds to reject both of these disturbances. While it takes the cart about 1.4 seconds accelerate from a stop with an overshoot of 2 cm/s , it only takes 0.21 seconds to stop with no overshoot. Overall the cart accelerates and stops faster when using the subsystem LQG control as opposed to the full system LQG controller but slower than the PID controller set. This increase in speed of the cart controller makes sense because the cart is no longer penalized by error in the pendulum system. The cart subsystem's velocity controller under this subsystem design causes overshoot of the desired velocity. This is worse than both the PID and the full system LQG controller presented in Chapter 5 and Chapter 6. Once again the pendulum rejects the disturbance faster than the PID but slower than the full system LQG controller. This makes sense because the full LQG controller has more information about the disturbance than the subsystem LQG controllers do. The peak disturbance is worse than both the full system LQG and PID controllers. The increase in disturbance magnitude isn't caused by a decrease in propeller motor voltage because the controller was tested under similar battery voltages to previous tests. This is most likely due to the reduction in information that the pendulum LQG controller has compared to the full system LQG controller that knows when the disturbance occurs.

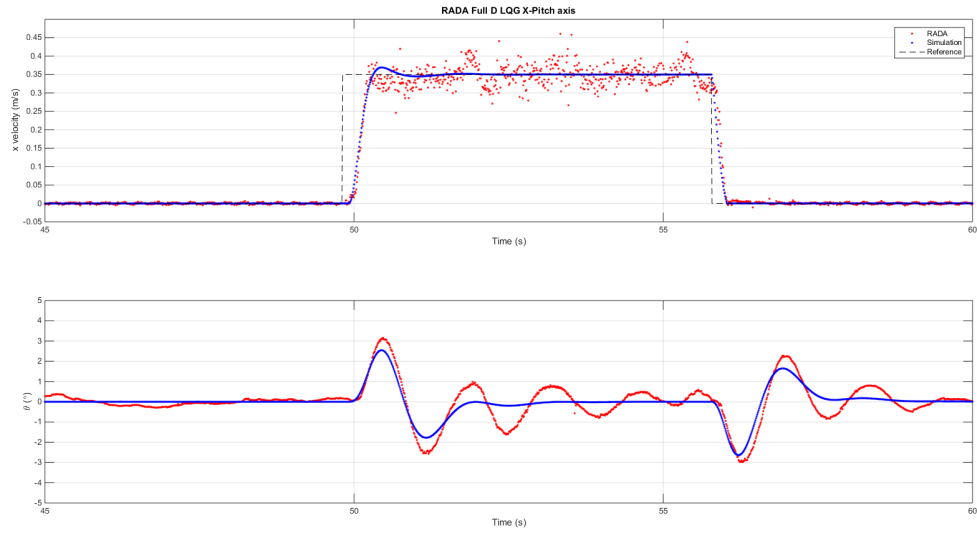


Figure 7.2 Test LQG control of each subsystem independently in $x-\theta$.

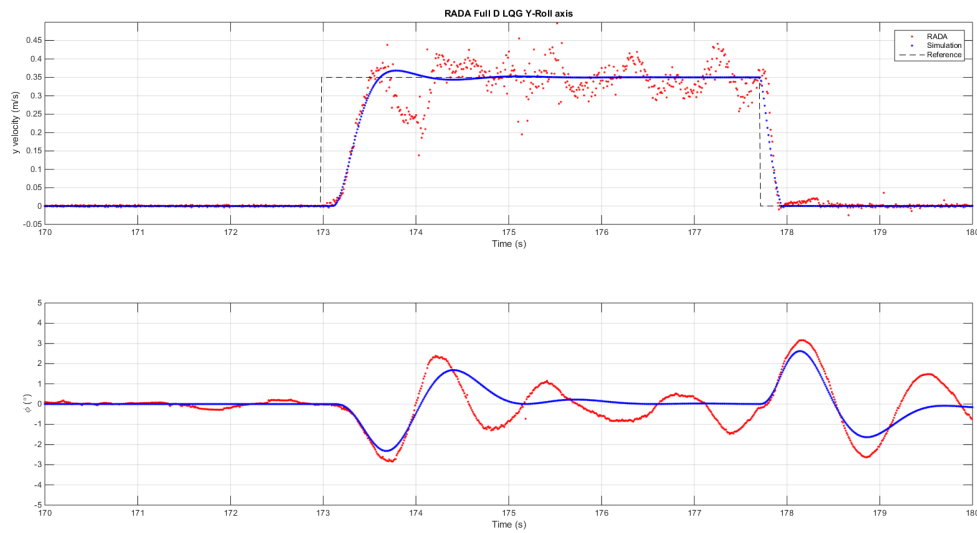


Figure 7.3 Test LQG control of each subsystem independently in $y-\phi$.

CHAPTER 8. CONCLUSION

Modeling of physical systems is always difficult because it is difficult to tell when your model and controllers are good enough to call the model complete. There is always going to be some sort of unmodeled dynamics that can be observed in the physical system that isn't present in the model. The first question that needs to be asked is, how much do these differences cause the controllers designed using the model to become useless or no longer predictive of the physical systems behavior? The next question that needs to be ask is, if the controllers that do work are good enough or if they need to be better in some way?

Overall the model of RADA behaves well enough. The model is not perfect there are still some oscillations that the model doesn't explain but they are fairly small. The disturbance response of the pendulum and the motion of the robot don't match exactly but they are very similar overall. The controller system design behaves well enough. While there are also additional unmodeled dynamics such as the center of mass of the pendulum, vibrations within the pendulum system, sensor noise, and delays are all currently not taken into account. One such issue could be that the wheels are not perfectly mounted to the system causing the actual system to have wheels that rotate like an oval instead of a circle. This could be simulated as a sinusoidal wheel force disturbance at twice the rotational frequency of the wheels. Figure 8.1 compares the RADA model undisturbed in blue to a simple sinusoidal wheel force model shown by the red dots when both systems are controlled with the full system LQG from Chapter 6. Overall this could help to explain some of the noise in linear velocity calculation but it doesn't seem to effect the pendulum much at least not at this amplitude. Other sources of noise like vibration and flex of the pendulum should be investigated in the future and characterized.

Adding disturbances into the model would make simulation match the system better but it wouldn't necessarily make better controllers. The non-linear model of RADA is good enough

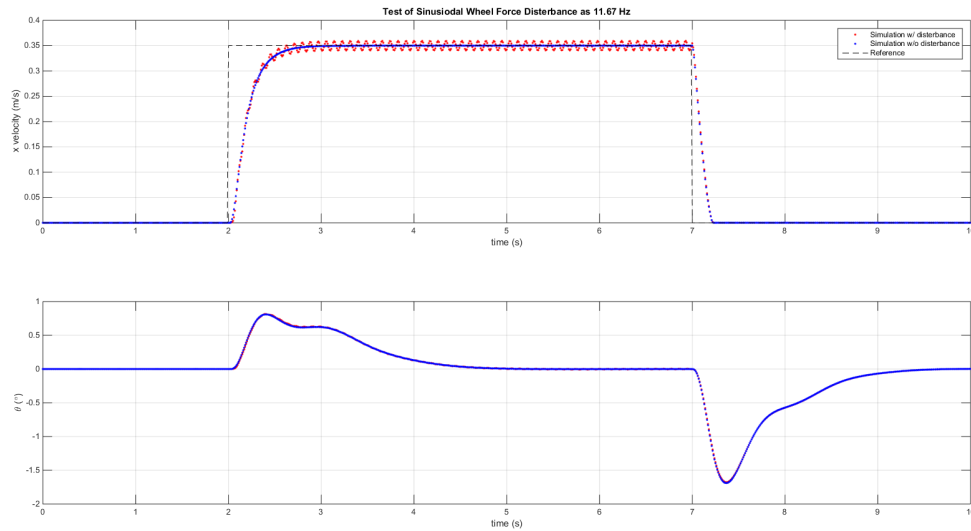


Figure 8.1 Simulation test of a possible wheel force disturbance model.

as it is unless some drastic changes occur to degrade the physical system, that would require additions to the model. The full system LQG controller rejects the disturbances fast enough and accelerates the robot quick enough to meet the goals of RADA. The subsystem based LQG controller design is somewhat better than the PID controller but is not as good overall than the full system LQG controller and would only be beneficial if the system design required the pendulum and cart to be treated as separate systems that couldn't communicate. Otherwise, the full system controller design architecture produces better controllers.

The fact that the vehicle stops quicker and causes larger disturbances to the pendulum is not really too concerning for the use case of RADA. It is meant to continuously move through a row of crops so the initial acceleration and disturbance attenuation is more important for getting the best data. When RADA stops it will be after the end of the row and will need to maneuver the next row so there is plenty of time to attenuate the larger amplitude disturbance.

The model of RADA and the physical system itself could be further refined but the current model reflects the current behavior of the system well enough to serve as a foundation for future controller development and should be capable of being modified as the physical system evolves. Overall the model is not 100% complete but it works well enough to be the foundation for future expansion and improvements to the system.

8.1 Future work

In the future RADA will need to be expanded and improved. The current prototype is functional but by no means ideal. The main issues with the system is that the current pendulum arm is too flexible in its current form. This will need to be remedied especially if an outdoor prototype is going to be produced as the pendulum arm will need to be lengthened up to twice its current length making any vibrations worse. Obviously the drive train needs to be improved for any sort of outdoor prototype. A tank tread like drive train would probably work the best out in the field. In terms of the electronics the FPGA should be replaced as it was designed to be used a motor controller instead of being used like an FPGA. The motor controller portion of current system should be very easy to recreate on a new platform. The wheel motors and gearboxes will need to be replaced if the drive train is changed. The power distribution board also needs to at least be rebuilt seeing as it was photo-etched and has no solder mask to prevent short circuits. Finally, the power cable running up the pendulum needs to have larger wires to power the propellers as there is significant power loss with the current system.

The current prototype also needs upgraded localization sensors. The camera system works well in the lab, but it cannot be taken out into a field. The pendulum requires an inertial measurement unit (IMU) in order to calculate the current orientation without the camera system. The ground robot will also need a global positioning system (GPS) for position and speed and a magnetometer to calculate heading. Another IMU and quadrature motor encoders to dead recon between GPS updates would also be useful for navigating the ground vehicle. These additions are part of the reason for upgrading the FPGA because the host CPU's motherboard doesn't support the communication protocols needed to communicate with these sensors. Without an easily reconfigurable FPGA it will be difficult to integrate these sensors.

RADA can also be used in its current form to test advance control schemes. One example of this would be testing the optimal mean-squared performance for controllers over fading channels developed in [27]. The ability to use the LAN connection to force packet loss will allow for good analysis of how the controller perform under varying amounts of packet loss. Then the same controller can be tested using the wireless connection to test how it performs under a

real fading channel. A second idea is to test co-operative controllers where the pendulum and ground vehicle are split into two systems and tell the pendulum of the vehicle's velocity commands before actually giving the command to the ground robot. This is desirable because even though the propeller dynamics were simplified out of the linear system used for controller design in Chapter 6.1, the propellers are still slower than the disturbance caused by the wheel acceleration so allowing the propellers a few sample periods to spin up before actually seeing the disturbance should help the system perform even better. These are just a few ideas for other controllers that could be demonstrated on this in-lab prototype due to the very powerful processor and camera system available for use with the system.

BIBLIOGRAPHY

- [1] J. Shen, A. K. Sanyal, N. A. Chaturvedi, D. Bernstein, and H. McClamroch, “Dynamics and control of a 3d pendulum,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, Dec 2004, pp. 323–328 Vol.1.
- [2] M. Kumagai and T. Ochiai, “Development of a robot balancing on a ball,” in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, Oct 2008, pp. 433–438.
- [3] M. Hehn and R. D’Andrea, “A flying inverted pendulum,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 763–770.
- [4] L. Freidovich, A. Robertsson, A. Shiriaev, and R. Johansson, “Periodic motions of the pendubot via virtual holonomic constraints: Theory and experiments,” *Automatica*, vol. 44, no. 3, pp. 785 – 791, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109807003676>
- [5] U. Didactic. Rt 122 fuzzy control: Inverted pendulum. [Online]. Available: <http://www.usdidactic.com/html/p4713.htm>
- [6] T. Bak and H. Jakobsen, “Agricultural robotic platform with four wheel steering for weed detection,” *Biosystems Engineering*, vol. 87, no. 2, pp. 125 – 136, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S153751100300196X>
- [7] B. Åstrand and A.-J. Baerveldt, “An agricultural mobile robot with vision-based perception for mechanical weed control,” *Autonomous Robots*, vol. 13, no. 1, pp. 21–35, 2007. [Online]. Available: <http://dx.doi.org/10.1023/A:1015674004201>

- [8] M. Rich, “Model development, system identification, and control of a quadrotor helicopter,” Master’s thesis, Iowa State University, Ames, IA, 2012.
- [9] D. Gransee, R. Larsen, A. D. Martino, I. McInerney, A. Pederson, R. Zambre, and F. Zhu, “Robotic agricultural data acquisition project report,” Iowa State, <http://may1527.ece.iastate.edu/index.html>, Tech. Rep., 2015.
- [10] S. Beinhart, P. Martinson, and J. J. Clausman, “Omni-directional robot,” Iowa State, <http://seniord.ece.iastate.edu/dec1001/>, Tech. Rep., 2010.
- [11] *Pluto*, 1st ed., Diamond Systems Corporation, www.diamondsystems.com, 7 2009, datasheet.
- [12] *4I68 ANYTHING I/O Manual*, 1st ed., MESA, manual.
- [13] *The Pololu Dual MC33926 Motor Driver Carriertitle of the work*, Parallax inc., www.parallax.com, 11 2011.
- [14] *Faulhaber DC-Motors 10 mNm Serise 2232 ... SR*, MicroMo Electronics, Inc., www.micromo.com, 1 2008, datasheet.
- [15] *Faulhaber Planetary Gearheads 0.5 Nm Series 20/1*, Dr. Fritz Faulhaber GMBH & Co. Kg, www.faulhaber.com, 2015, datasheet.
- [16] *FlameWheel450 User Manual*, v1.7 ed., DJI, www.dji.com, 2011.
- [17] “Blheli firmware,” <https://github.com/bitdump/BLHeli>.
- [18] *Tracking Tools 2.4.0 User’s Guide*, OptiTrack.
- [19] “Vrpn 7.31 sorce code,” https://github.com/vrpn/vrpn/releases/tag/version_07.31.
- [20] *Single Inverted Pendulum Control Laboratory Linear Exeriment #5: LQR Control- Student Handout*, 4th ed., Quanser.
- [21] H. Y. R. Geller and L. Ford, *University Physics with Modern Physics*, 12th ed. Pearson, 2008.

- [22] C. C. Ward and K. Iagnemma, “Model-based wheel slip for outdoor mobile robots,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE. IEEE, 2007, pp. 2724–2729.
- [23] *Dual Range Force Sensor*, Vernier, <http://www.vernier.com/>, 2016, dataSheet.
- [24] *LabPro Technical Reference Manual*, Vernier, <http://www.vernier.com/>, 2002, datasheet.
- [25] *LoggerPro 3 Quick Reference Manual*, Vernier, <http://www.vernier.com/>, datashet.
- [26] *HP 54600-Series Oscilloscopes Datasheet*, Hewlett Packard.
- [27] M. Rich and N. Elia, “Optimal mean-square performance for mimo networked systems,” in *2015 American Control Conference (ACC)*, July 2015, pp. 6040–6045.