DESIGN AND PROTOTYPING OF A LOW-COST SPACECRAFT ATTITUDE DETERMINATION AND CONTROL SETUP

A Project

Presented to

The Faculty of the Department of Aerospace Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Anthony Gong

May 2015

The Designated Thesis Committee Approves the Thesis Titled

DESIGN AND PROTOTYPING OF A LOW-COST SPACECRAFT ATTITUDE DETERMINATION AND CONTROL SETUP

by

Anthony Gong

APPROVED FOR THE DEPARTMENT OF AEROSPACE ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2015

Dr. Kamran Turkoglu Department of Aerospace Engineering

ABSTRACT

DESIGN AND PROTOTYPING OF A LOW-COST SPACECRAFT ATTITUDE DETERMINATION AND CONTROL SETUP

by Anthony Gong

In current literature, various ground-based spacecraft attitude determination and control setups are employed to test control strategies and algorithms. However, they typically require relatively large amounts of clearance for rotation and are prohibitively expensive for academic/research use. This paper outlines the development of a 1-D prototype that aims to serve as a low-cost alternative. The goal is to develop an open-source, low-cost platform at minimal cost. The prototype consists of an reaction wheel driven by a brushless DC motor mounted to a plate that is allowed to pivot about a hinge. The prototype has the ability to transfer angular momentum between the wheel and the body as well as balance using reaction torques. PID and LQR controllers for the prototype have been designed to achieve minimal control effort. Preliminary results demonstrate that the LQR is able to achieve energy savings of $\sim$8% over the baseline PID controller with acceptable tradeoffs. Experimental results show that the LQR controller is able to successfully balance and reject external disturbances. The 1-D prototype remains low-cost at $480.

ACKNOWLEDGEMENTS

I would like to sincerely thank Space Systems/Loral, and especially Claudia Lam, for providing financial support for the project and Advanced Motion Controls for donating the B15A8 motor controller used for this prototype.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

Nomenclature

C	dynamic friction coefficient
g	gravitational acceleration
h	angular momentum
I	moment of inertia
K_m	motor torque constant
l	distance from pivot point
m	mass
T	torque
u	input current
ω	angular velocity of reaction wheel
θ	angle or position
$\dot{\theta}$	angular velocity
$\ddot{\theta}$	angular acceleration
$(\ )_b$	body-related parameter
$(\ )_w$	wheel/rotor assembly-related parameter

CHAPTER 1

INTRODUCTION

All spacecraft require some form of stabilization and control due to the external disturbances in space. Stabilization can be achieved through various methods such as gravity-gradient[FW90], spin [DFF77], and 3-axis stabilization [Azo93], and control of the spacecraft is achieved through passive [Bro02] or active[IFS93] actuators such as magnetic torque rods, reaction thrusters, and reaction wheels.

Before deployment in space, spacecraft must be tested extensively through ground-based equipment that attempt to simulate the environment. These setups typically involve a rotating air-bearing table designed to mitigate the effects of gravity.[JT03, PC03, SPH03, Cro11] However, these setups usually require relatively large amounts of space for rotation and are excessively expensive (where $20,000 is considered low-cost).[KVKT03]

There are novel spacecraft attitude control setups currently found in literature. 3D reaction wheel configurations[ITT08] and three axis control via two reaction wheels are some of the recent research efforts.[GJX13, BBC13, XQLQ06] Various configurations with four reaction wheels have also been investigated for lowest energy consumption.[IV10]

Inverted pendulums have been developed to study control algorithms and strategies. Amongst these designs, there are some that utilize reaction wheels and their associated reaction torques to self-erect and balance.[SCL01, GMTD12, MDdC09] It is possible to utilize existing setups in

literature to design one from scratch, which can serve as an experimental, low-cost research and education platform for spacecraft attitude determination and control studies. Previously developed Cubli[GMTD12] is a prime example and the primary motivation for the current design.

The design would need to satisfy several requirements. It must be as unobstructive as possible (can fit on a small desk). It must be affordable such that academic institutions and even students can purchase the components and construct the suggested set-up on their own. The setup should also draw minimal power to be representative of a spacecraft with a limited power budget.

The final product of is envisioned to be a 6-inch cube that contains the attitude determination and control system of a typical 3-axis stabilized spacecraft. Three 1-D (one degree of freedom) prototypes will be mounted orthogonally to provide 3-axis control. The 1-D prototype is first manufactured to evaluate the mathematical model and assess its performance before proceeding to a full 3-D assembly.

The main aim of this paper is to present a low-cost spacecraft attitude determination and control setup which could be utilized in research and education. For analytical purposes, the corresponding equations of motion and an overview of the design is described in Chapters 2 and 3. Chapters 4 and 5 provide an overview of the hardware and software setup. Chapter 7 provides a robustness analysis of the controller to uncertainties. In Chapter 6, various controller designs are investigated and with the experimental results given in Chapter 8, the paper is concluded.

CHAPTER 2

1-D PROTOTYPE DESIGN

2.1 Design Objectives

The goal of this project is to design a low-cost setup that can serve as a spacecraft attitude determination and control platform for research and education. Due to certain drawbacks described in Chapter 1, the design must address the following requirements:

(1) Operate in a *space-like* environment

(2) Remain low-cost and affordable for education

(3) Not require large amounts of space for operation

(4) Act as a visual demonstration of angular momentum conservation

The proposed solution that satisfies the four design requirements is pictured in Fig. 2.1. In short, it is a reaction wheel inverted pendulum. The reasons why such a setup fits the design requirements is described below.

The first constraint ensures that the designed setup operates in such a fashion that it mimics the conditions in space where gravity does not exert external torques on the system. In the vertical, stabilized, equilibrium position, gravity will not apply any torques as the center of mass of the setup is directly over the pivot point. As long as the tilt angle remains small, gravitational effects will be minor, and the setup can be assumed to be operating in a space-like environment. The

requirements that the setup must be low-cost and relatively small are easily met (as described in later sections), and the use of a reaction wheel for control, in conjunction with the use of a impulsive braking mechanism, provides an excellent visual representation of angular momentum conservation.

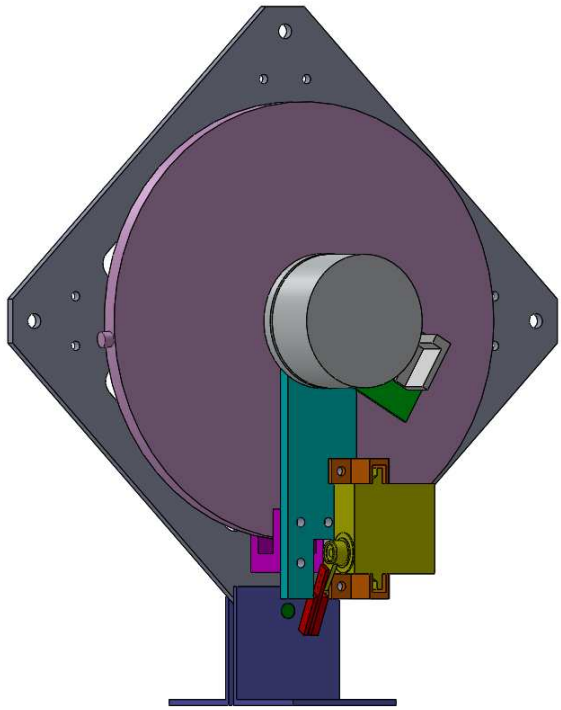

Figure 2.1: CAD drawings of the 1-D prototype.

The prototype consists of a reaction wheel driven by a brushless DC motor that is mounted to the center of a square metal plate (spacecraft body). The self-erecting setup utilizes reaction torques for balancing and a braking mechanism to impulsively transfer angular momentum for the initial jump-up maneuver.

The terms pendulum body and spacecraft body are used interchangeably as well as servo drive and motor controller.

2.2 Prototype Design

As shown in Fig. 2.2, the setup is capable of impulsively transferring angular momentum from the wheel assembly to the body through the use of a braking mechanism. Such a transfer will cause the setup to pop up from its initial resting position and travel 45° to the vertical position, which is depicted in Fig. 2.3.

To accomplish this, careful consideration is taken to design the setup such that the combination of inertia and angular velocity provides enough angular momentum to reach the desired position. Such is done through Eq. (2.1) and is also explained in Gajamohan et al. [GMTD12].

$$\omega^2 = (2 - \sqrt{2})\frac{(I_w + I_b + m_w {l_w}^2)}{{I_w}^2}(m_b l_b + m_w l_w)\ g \tag{2.1}$$

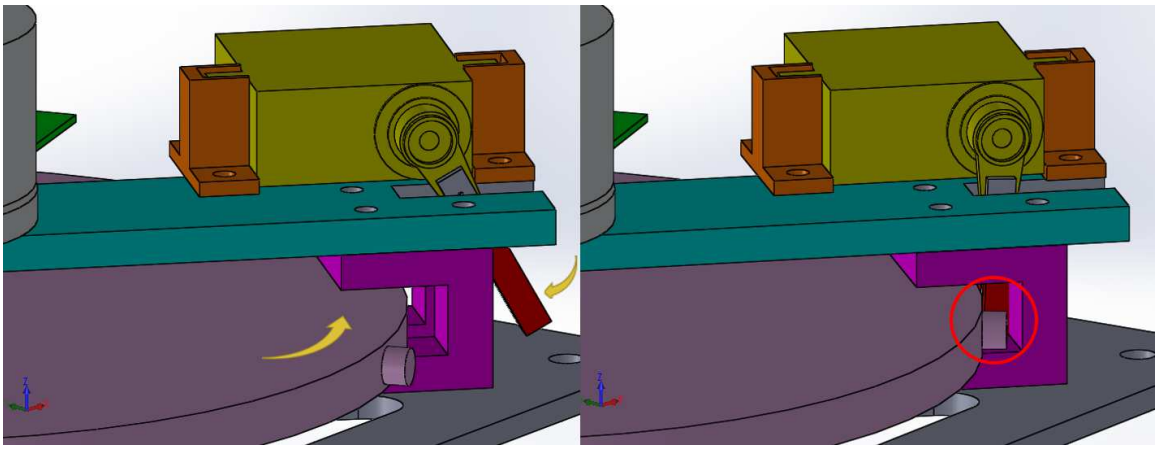

Figure 2.2: Angular momentum transfer achieved by stopping the momentum wheel via a metal barrier actuated by the RC servo.

Low-cost RC servos do not possess the torque required to stop the spinning momentum wheel without damaging the servo. Even if it could, the angular momentum transfer would not be instantaneous and would introduce additional system delays, dynamics, and uncertainties while degrading the setup's self-erecting

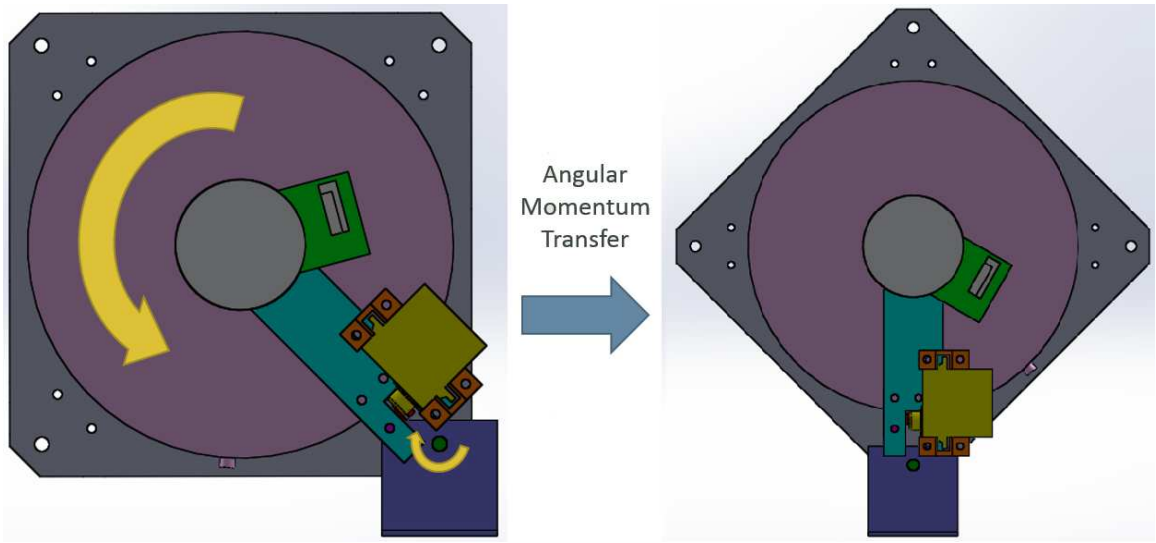

Figure 2.3: Jump-up maneuver accomplished by transferring angular momentum from the wheel to the body and pivoting about the hinge point.

ability. Instead, the impulsive force during momentum transfer is carried by the slotted (purple) metal component in Fig. 2.1, so the servo does not require a high torque rating. The RC servo's task is merely to actuate the (red) metal barrier as quickly as possible. A 0.5-inch 4-40 screw is put into the side of the wheel to create a protrusion on the edge. This creates a contact point for the metal barrier to stop the spinning wheel to achieve momentum transfer.

2.3 Material and Component Selection

Since the main goal is to design a low-cost experimental set-up via commercially available, off-the-shelf products, component selection becomes an important part of the design process and is driven by the cost and availability of the material. Selections are made with the 3-D prototype in mind to reduce the overall cost of the project. Components used in the 1-D prototype are directly carried over to the 3-D configuration to reduce development time and production costs.

Although more expensive than other microcontroller boards, the Arduino Mega 2560 is chosen for its Simulink compatibility and the availability of extensive open-source libraries. The MPU-6050 inertial measurement unit (IMU) is used which houses a 3-axis accelerometer and 3-axis gyroscope. Although a single-axis sensor would be sufficient for this prototype, a 3-axis solution is chosen with the future 3-D configuration in mind. The HSG-5084MG servo is selected for its unique actuation speed (0.07 sec/60°) to minimize associated system delays. The DF45 from Nanotec is implemented for its high torque rating and small form factor. The B15A8 is used since it possesses a current mode that allows for direct control of output torque. Two 12-bit digital-to-analog converters are required to provide an analog voltage output to drive the motor controller since the Arduino Mega does not have an onboard DAC. It is also the reason why the DACs are required because the B15A8 motor controller can only be driven by an analog voltage command. Newer models, especially digital servo drives, can be operated by a variety of input sources including PWM signals (which the Arudino Mega can output without additional hardware), so with the next iteration of the design, a different motor controller will be chosen to reduce the complexity of the setup and eliminate the required DACs completely. However, as a starting point, the donated analog motor controller is sufficient to demonstrate the feasibility of setup and operate this prototype.

Aluminum 6061 is chosen to construct all custom components because of its desirable material properties and cost. It is also easily available in various forms. Custom components are designed using standard dimensions so that purchased material requires minimal machining and fabrication is quick. CNC fabrication can be avoided completely by designing parts that do not require high precision so that the total cost of the project can be further reduced. A complete list of materials and components is given in Table 2.1.

Table 2.1: Cost breakdown of the major components.

Component	Manufacturer	Model Number	Cost
Microcontroller	Arduino	Mega 2560	50
IMU	InvenSense	MPU-6050	10
Brushless DC Motor	Nanotec	DF45	77
Motor Controller	Advanced Motion Controls	B15A8	180
RC Servo	HiTec	HSG-5084MG	30
Raw Material	McMaster Carr	6061 Aluminum Alloy	50
12-bit DACs	Sparkfun	MCP4725	10
		Total*	**480**

*Total includes miscellaneous items, tax, and shipping costs

2.4 Custom Components

The setup calls for five custom designed components, and those can be seen in Fig. 2.1. The first is the (grey) plate (6 x 6 x 0.125 inches) that serves as the main platform and will later serve as a face of the cube on the 3-D prototype. The assembly will pivot about a hole on this plate. The second is a cylindrical (pink) disk (5 inch diameter, 0.25 inch thickness) that serves as the momentum wheel. The third is a (teal) plate (4 x 1 x 0.1875 inches) used to mount the motor/wheel assembly. A fourth (purple) piece holds up the teal plate and provides a slot for the braking mechanism to swing in and contact the stopper at the edge of the wheel. A fifth (red) piece (0.25 x 0.375 x 1.375 inches) will be actuated by the servo and acts as a barrier to stop the spinning momentum wheel.

$$\theta$$

2.5 Experimental Setup Overview

A basic overview of the experimental setup can be seen in Fig. 2.4. The accelerometer and gyroscope provide measurements on the tilt angle and tilt rate of

the setup, respectively. Hall sensors on the DF45 motor provide a measurement on angular velocity. The Arduino Mega 2560 calculates the controller output required to stabilize the plant through the feedback control law. The output is converted to a 12-bit digital command that can be outputted by the digital-to-analog converters as an analog voltage to the motor controller. The motor controller then takes the voltage command and outputs a current command to control the torque output.

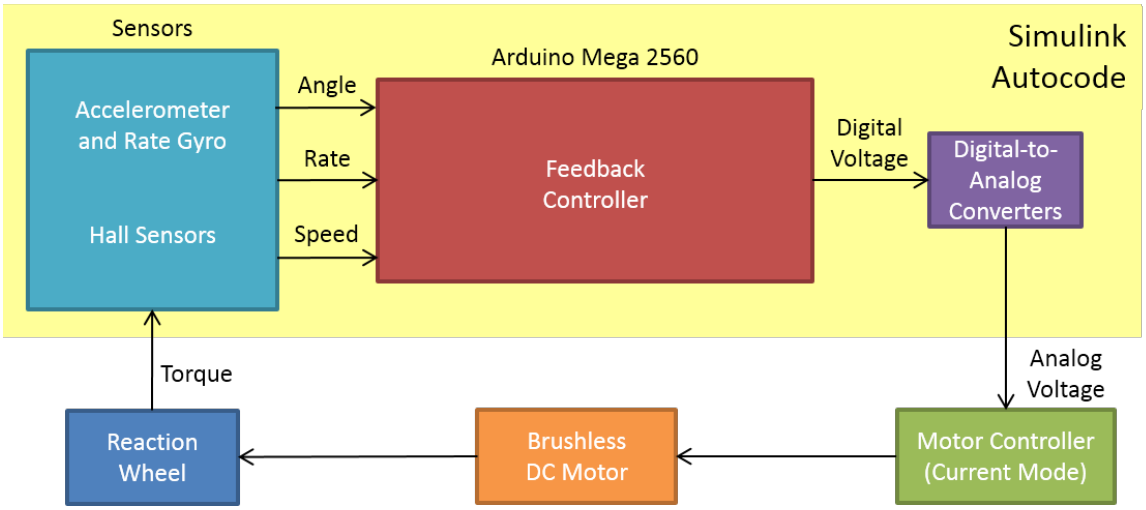

Figure 2.4: Overview of experimental setup

CHAPTER 3

SYSTEM DYNAMICS AND ANALYSIS

3.1 Equations of Motion

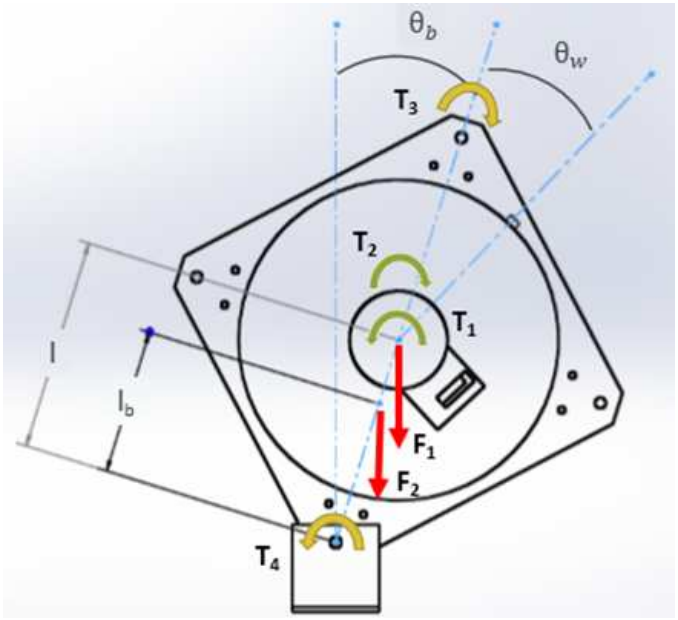

Figure 3.1: Free-body diagram of the 1-D prototype.

Angular momentum and torque can be respectively expressed as,[Sid97]

$$h = I\dot{\theta} = I\omega \tag{3.1}$$

$$T = I\ddot{\theta} = I\dot{\omega} \tag{3.2}$$

the equations of motion, as previously presented[GMTD12], can be derived from the free-body diagram shown in Fig. 3.1 as

$$\ddot{\theta}_b = \frac{(m_b l_b + m_w l_w)\ g \sin\theta_b - T_m - C_b\dot{\theta}_b + C_w\omega}{I_b + m_w {l_w}^2} \tag{3.3}$$

$$\ddot{\theta}_w = \frac{(I_b + I_w + m_w {l_w}^2)(T_m - C_w\omega)}{I_w(I_b + m_w {l_w}^2)} - \frac{(m_b l_b + m_w l_w)\ g\sin\theta_b - C_b\dot{\theta}_b}{I_b + m_w {l_w}^2} \tag{3.4}$$

where m_w is the mass of the reaction wheel and m_b is the mass of the spacecraft body. l_w and l_b are the distances from the pivot point to the center of mass of the wheel and pendulum body, respectively. θ_b is the tilt angle measured from the local vertical and θ_w is the arbitrary position of the wheel relative to some point on the body. Unlike its time derivative, ω, which is the angular velocity of the reaction wheel, θ_w is a variable that will be derived but not used as a state. Here, T_m is the torque of the motor and it is defined in Eq. (3.5)

$$T_m = K_m u \tag{3.5}$$

where u is the current input measured in Amps.

3.2 State Space Representation

From the general form of a dynamic system[Sid97],

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3.6}$$

Eqs. (3.3) and (3.4) are linearized about the vertical equilibrium point,

$$(\theta_b, \dot{\theta}_b, \omega) = (0, 0, 0) \tag{3.7}$$

and the state space representation is given in Eq. (3.8).[GMTD12]

$$\begin{bmatrix} \dot{\theta}_b \\ \ddot{\theta}_b \\ \ddot{\theta}_w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{(m_b l_b + m_w l_w)g}{I_b + m_w {l_w}^2} & -\frac{C_b}{I_b + m_w {l_w}^2} & \frac{C_w}{I_b + m_w {l_w}^2} \\ -\frac{(m_b l_b + m_w l_w)g}{I_b + m_w {l_w}^2} & \frac{C_b}{I_b + m_w {l_w}^2} & -\frac{C_w(I_b + I_w + m_w {l_w}^2)}{I_w(I_b + m_w {l_w}^2)} \end{bmatrix} \begin{bmatrix} \theta_b \\ \dot{\theta}_b \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K_m}{I_b + m_w {l_w}^2} \\ \frac{K_m(I_b + I_w + m_w {l_w}^2}{I_w(I_b + m_w {l_w}^2)} \end{bmatrix} u \tag{3.8}$$

3.3 Plant Dynamics

Eigenvalues of the A-matrix represent the pole locations of the system (roots of the characteristic equation). As expected of a naturally unstable system, there is a pole in the right half plane (RHP). Therefore, a closed-loop feedback controller is required to stabilize the plant. The observeability and controlability are first checked.

3.4 Parameter Estimation

Table 3.1 summarizes the system parameters that have been identified. The lengths and masses are easily determined through simple experiments. The moments of inertia are estimated by Solidworks®[Sol] after applying material properties to each component. I_w is found by summing the moments of inertia of the aluminum disk and DF45. As previous described [GMTD12], the friction coefficients are found through collecting experimental data and Eqs. (3.9)-(3.10).

$$I_w\ddot{\theta}_w(t) = K_m u(t) - C_w\omega(t) \tag{3.9}$$

$$(I_b + I_w + m_w {l_w}^2)\ddot{\theta}_b(t) = -C_b\dot{\theta}_b(t) + (m_b l_b + m_w l_w)g\sin\theta_b(t) \tag{3.10}$$

3.5 Digital Filter Design

It is necessary to calibrate and correct the sensor's measurements for optimal performance. A noisy signal will result in increased energy consumption since the controller will attempt to correct for erroneous angle displacements (assuming the noise itself does not destabilize the setup). Angles obtained from gyroscope measurements will drift over time ($\sim$3 deg/sec) due to integration, so it is unreliable

Table 3.1: Identified System Parameters

Parameter	Value	Unit
l_b	0.077	m
l_w	0.089	m
m_b	0.436	kg
m_w	0.131	kg
I_b	$4.21 \cdot 10^{-3}$	$kg \cdot m^2$
I_w	$0.42 \cdot 10^{-3}$	$kg \cdot m^2$
C_b	$1.94 \cdot 10^{-3}$	$kg \cdot m^2 \cdot s^{-1}$
C_w	$0.07 \cdot 10^{-3}$	$kg \cdot m^2 \cdot s^{-1}$

in the long term. A complementary filter, as shown in Eq. (3.11), is used to correct for the drift of the gyroscope and filter out the noise from the accelerometer.[CJC14] It does so by applying a low-pass filter to the accelerometer and a high-pass filter to the rate gyro.

$$\theta_b = \alpha \ (\theta_b + \dot{\theta}_{b,gyro} \ \Delta t) + (1 - \alpha)(\theta_{b,accel}) \tag{3.11}$$

$$\theta_{b,accel} = \tan^{-1}(\frac{g_y}{g_z}) - \theta_0 \tag{3.12}$$

where $\dot{\theta}_{b,gyro}$ is the angular velocity (deg/sec) output by the gyroscope about an axis. Multiplying the gyroscope output by a time constant (Δt) approximates the integral to provide an angle. $\theta_{b,accel}$ is the angle about an axis computed via the inverse tangent of the accelerometer readings from the other two axes, shown in Eq. (3.12). α is a weighting factor that determines which signal the complimentary filter follows closer. It is found that $\alpha = 0.98$ provides a relatively clean signal that does not drift over time, as shown in Fig. 3.2.

Offsets for the gyroscope and accelerometer readings are found by measuring 1000 data points while the sensor is in a static condition (laying down on a flat surface with the z-axis aligned with the local vertical). Measured values are averaged and compared to known values in such an orientation before being used to

calibrate the gyroscope and accelerometer. θ_0 is set to 45 deg. to offset the measured angle so that 0 deg. is measured at the balancing reference point.

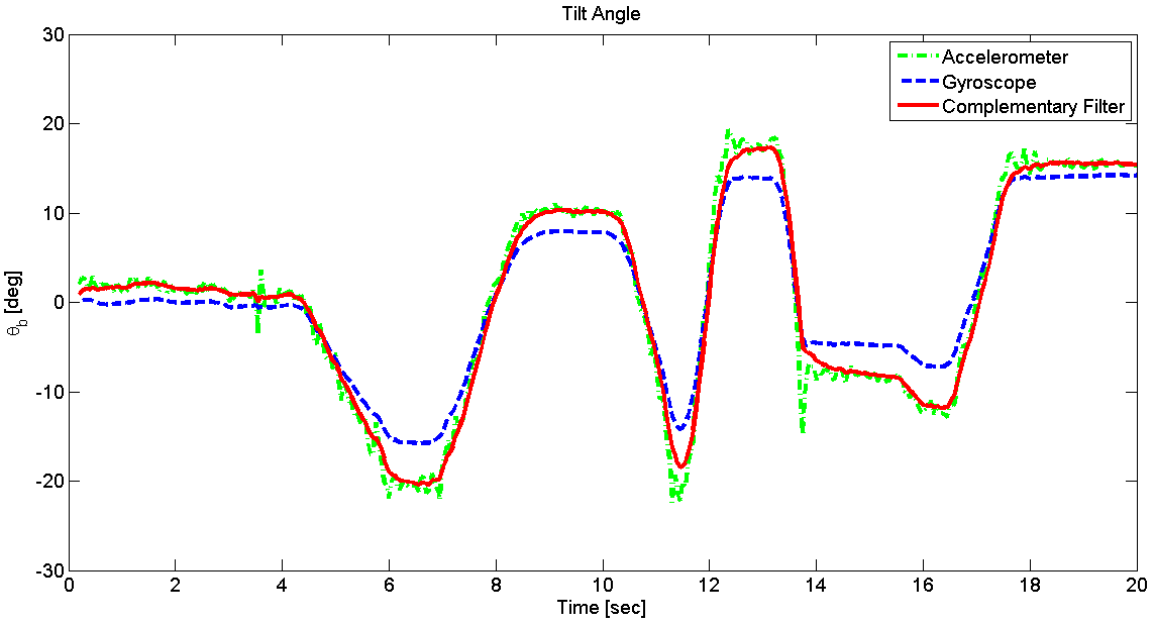

Figure 3.2: Tilt angle measurements obtained via accelerometer, gyroscope, and complimentary filter.

CHAPTER 4

ELECTRONICS SETUP

This chapter details the various aspects involving the electronics of the setup include wiring diagrams and pin-outs. Fig. 4.1 describes how the entire setup is wired together. The following sections describe the details with each component.

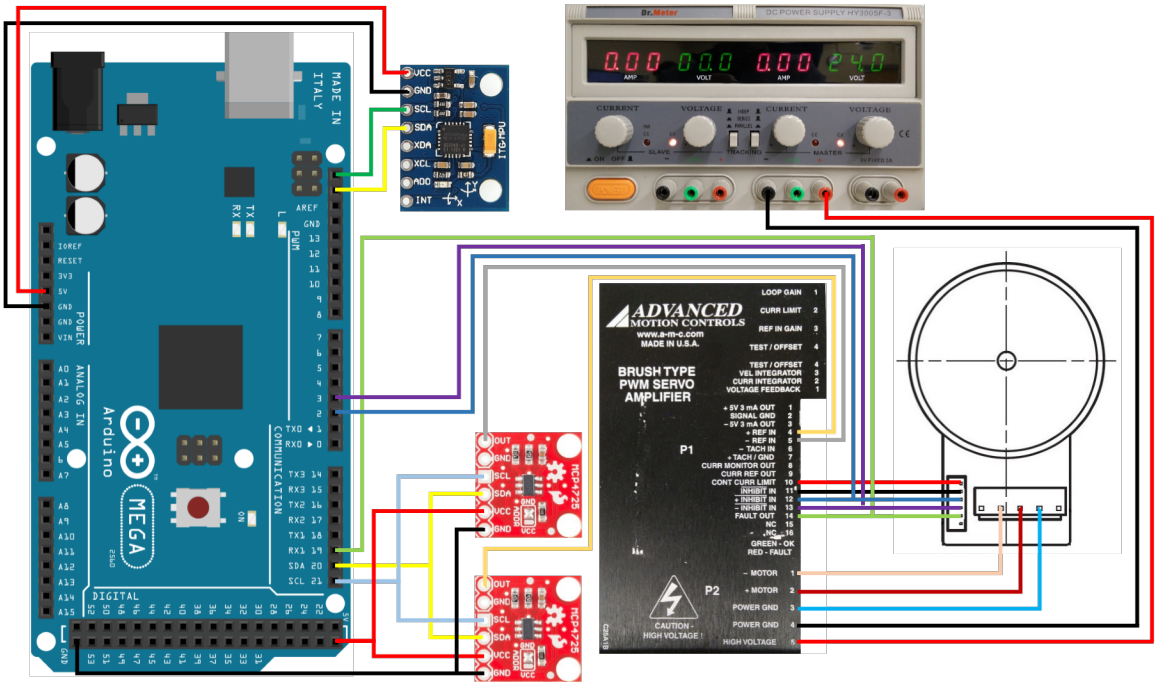

Figure 4.1: Wiring diagram showing connections between all hardware components.

NOTE: Labels on motor controller in Fig. 4.1 are not correct. See Section 4.3 for more information.

4.1 IMU

The MPU6050 communicates with the Arduino over an I^2C line. A 5-volt and ground line from the Arduino powers the VCC and GND lines on the IMU, respectively. A set of SCL (clock) and SDA (data) pins are used to pass the data over I^2C.

4.2 Digital-to-Analog Converters

The digital-to-analog converters also communicate to the Arduino over I^2C. In this case, they are connected together and share the same connections on the Arduino. Since both DACs are identical, the addresses need to differ for the Arduino to differentiate them over the same I^2C line. This is done by de-soldering the default address connection on one of the DACs and re-soldering it to the adjacent set of pads. The I^2C pullups connection on the back of that DAC is also disconnected with a small knife. The two DACs individually output an analog voltage to the motor controller lines (+REF IN and -REF IN).

4.3 Motor Controller

The motor controller is the link between the microcontroller and the brushless DC motor. It is essentially what converts a normal brushless DC motor driving a load to a reaction wheel assembly (RWA). It is courtesy of the internal current feedback loop of the motor controller that allows the microcontroller to directly command the torque output. In current mode (in other words, torque mode, since $T = K_m u$), the motor controller receives a differential analog voltage command from the Arduino and performs onboard commutation to produce the correct waveform to drive the DF45 while feeding back the current to maintain torque output.

Depending on the relative magnitude of the analog voltages from the DACs, the motor controller will turn the motor left or right.

The motor controller is powered by a 24-volt power supply that has been set up with a current limit set at 3 Amps.

NOTE: The motor controller was delivered with the wrong cover case attached (which has incorrect pin labels). However, the manufacturer has confirmed that the model number found under the PCB defines the hardware so the motor controller is, in fact, a B15A8. As a result, the words on the metal case incorrectly label the pins, so they should not be used as reference when wiring the setup. However, the wiring diagram in Fig. 4.1 is still valid and displays the correct pin connections. See Appendix A for the correct pin-out of the motor controller.

4.3.1 Set-up and Tuning

The B15A8 has four DIP switches (SW1-SW4) and four potentiometers (Pot 1 - Pot 4) that allow for configurable hardware settings.

It is desired to put the motor controller in current-mode, so SW1 is turned OFF and SW3 is turned ON. SW2 is turned ON to select 120-degree commutation phasing to match the motor's hall sensor placement. SW4 is turned OFF to set the offset in the differential voltage input signal.

Pots 1 and 3, which control the loop gain and reference gain, are turned fully counter-clockwise (CCW) for current mode.

Pot 2 adjusts the current limit. It is important to limit the current output of the motor controller to protect the motor. As described by the manufacturer [REF], the number of clockwise turns in the potentiometer is used to set the current limit is found with Eq. 4.1, where I_{cont} is the motor's maximum continuous current (3.26 Amps) and I_{max} is the servo drive's maximum continuous current (7.5 Amps), which

is conveniently half the drive's maximum peak current. It is also noted that while the drive's current limits can be changed, the ratio of continuous-to-peak current remains at 50%.

$$\begin{aligned} \frac{I_{cont}}{I_{max}} * 12 + 1 \; turn &= number \; of \; clockwise \; turns \\ \frac{3.26}{7.5} * 12 + 1 \; turn &= 6.216 \; clockwise \; turns \end{aligned} \tag{4.1}$$

Pot 2 is first turned fully CCW (maximum of 14 turns) and then turned roughly 6 times CW.

Depending on SW4's setting, Pot 4 can either be used as an on-board signal source or offset adjustment for the input signal. To find the offset adjustment for symmetrical torque outputs on both directions, chirp signals are used as inputs and reaction wheel angular velocity measurements are collected as outputs. The experiments are repeated several times until a roughly symmetrical torque output is achieved. This tuning procedure is extremely important in order to ensure that the feedback controller produces the same output in both direction. Otherwise, the controller may only be able to stabilize the plant in one of the two directions or produce non-symmetrical responses in the left and right directions.

4.4 DF45

The DF45 has two connectors on it. One of the connectors is used for voltage transfer and the other is for the digital hall sensor.

As a brushless DC motor, only three of the five pins are required for operation (the ground pins are not necessary due to nature of commutation). Pins 2, 3, and 4 which correlate to the three phases (U, V, and W) are connected to Pins 1, 2, and 3 of power connector (P2) on the motor controller corresponding to motor lines A, B,

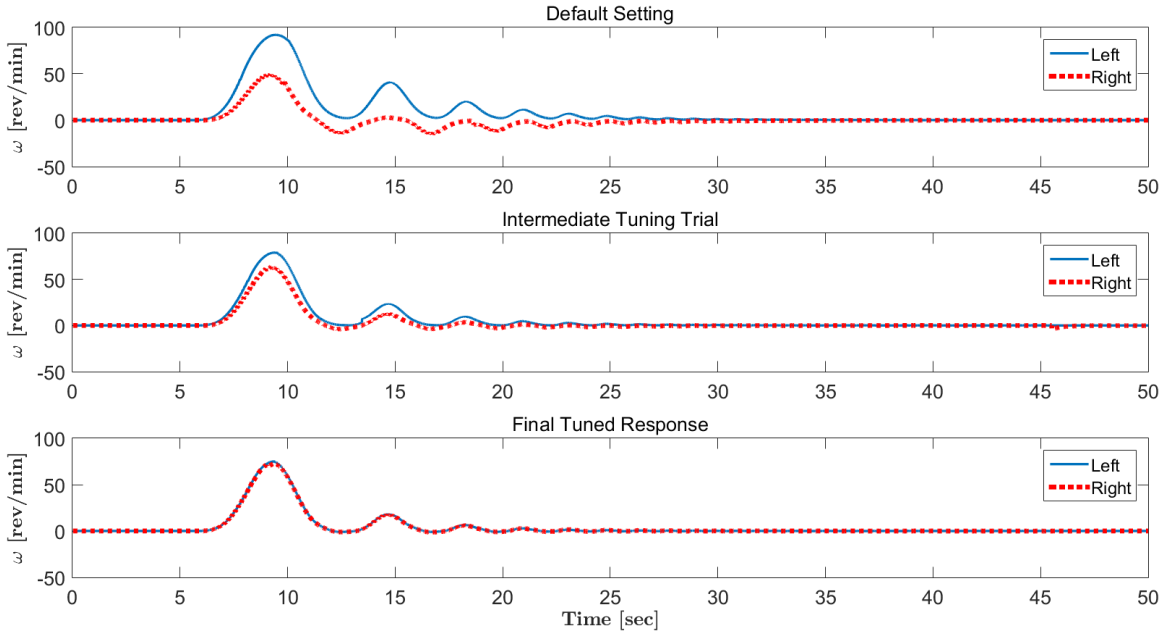

Figure 4.2: Progression of experimental tests to find symmetrical output in both directions.

and C. The order of the phases is important for proper commutation. There are 6 possible combinations of connecting the three phases (only one of which is correct), so a simple trial-and-error method is used to find the correct order. If incorrectly connected, the motor will either not produce any movement or provide erratic, interrupted angular velocity.

The hall sensors from the brushless DC motor are connected to the motor controller in order for it to perform commutation correctly. They are connected to the motor controller on pins 12, 13, and 14. Pins 10 and 11 from the motor controller are also connected to provide the hall sensors with voltage and ground. Only five of the six pins on the hall sensors are used since the sixth is a redundant ground pin.

The motor controller requires the hall sensors for commutation, but the Arduino also needs to have knowledge of the hall sensors' outputs to estimate the

angular velocity and direction of the reaction wheel. Therefore, the hall sensor outputs are connected in parallel and sent to the Arduino.

Both sets of wires leading from the motor controller to the DF45 (5 wires for the hall sensors and 3 for the voltage) are braided, crimped, and connected to corresponding plastic connectors that properly mate with the opposing versions on the DF45. This makes it impossible to improperly connect the wires on the DF45 side, and it also makes connecting and disconnecting the DF45 quick and simple.

CHAPTER 5

HARDWARE-IN-THE-LOOP SETUP

In order to facilitate controller design through Simulink, a hardware-in-the-loop architecture is built to take advantage of the Arduino Mega and Simulink's autocoding capabilities. Each hardware component must be driven by various drivers, so S-Functions (Simulink Functions) are used to adapt Arduino code into Simulink. Wrappers are generated that allow for controllers designed in Simulink (along with all necessary hardware drivers) to be directly built onto the microcontroller. Such an architecture streamlines the development and deployment process of feedback controllers and simplifies the operation of the setup (essentially turning it into a "1-click process").

The use of a microcontroller (which lacks computation power) and the lack of a data acquisition card (DAQ) severely limits the sampling rate of the setup. Typical DAQs allow for sample rates on the order of kHz whereas the current setup operates at 50 Hz. Such a limitation not only limits the amount of data that can be collected, but can directly affect the stability of the setup.

It is desirable to have a crossover frequency that is 2-3 times that of the most unstable pole. The sample rate should be at least 20 times faster than the crossover frequency.

The built-in Simulink support for Arduino through "External Mode" imposes a strict 25 Hz sampling rate limitation. In order to improve the sampling rate, the ability to change the model in realtime is given up. Instead, Simulink models are directly built onto the Arudino and data is collect through a serial line (COM port).

The sample rate is directly affect by the amount of data that is being collected, so if only state outputs are collected, the setup can operate at up to 100 Hz.

5.1 S-Function Development

The development of S-Functions to streamline the controller design process is done through the S-Function Builder in Simulink. The builder compiles the code and generates a set of files that can be deployed onto the Arduino Mega. Specifically, a TLC, MEX, and a wrapper C-file are generated each time. However, several modifications need to be made to the C-file before deployment. Rather than manually making the modifications each time, the function *renc2cpp* can be used on each generated wrapper. Sections 5.1.1-5.1.5 outline what each S-Function accomplishes.

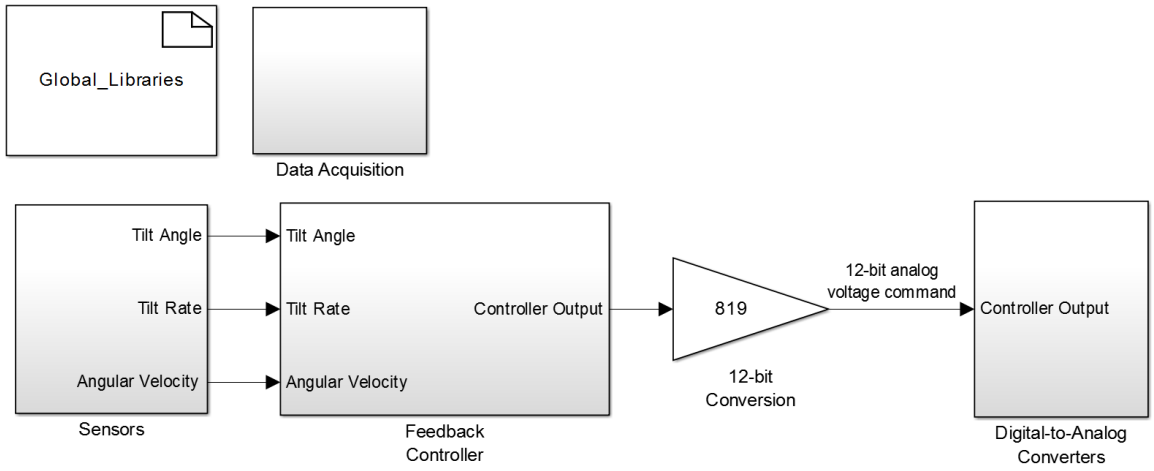

Figure 5.1: Hardware-in-the-Loop setup in Simulink.

5.1.1 IMU

The MPU-6050 defaults to a sleep mode during startup, so the IMU needs to first be turned on. The IMU's settings are configured each time during startup by

accessing the appropriate registers on the IMU and changing the values. This process includes setting the sensitivities on the rate gyro and accelerometer.

The raw data from the sensors are read and offsets are applied to provide correct readings. The offsets are found by collecting and averaging the raw data while the IMU sits on a flat surface.

The complementary filter discussed in the Section 3.5 is applied to the raw readings within the S-Function and various outputs ranging from the raw values to the filtered values are accessible as outputs of the S-function.

Since the start of each sampling cycle begins with the sensor readings, this S-Function is used to define the global sampling rate, and all succeeding blocks inherit the sampling time from this one.

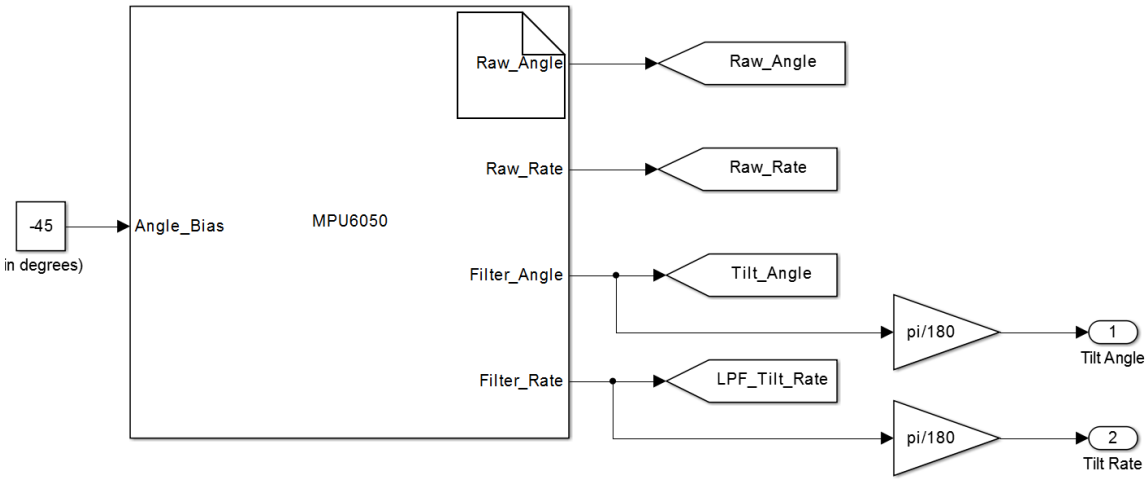

Figure 5.2: IMU (MPU6050) S-Function in Simulink.

5.1.2 Hall Sensors

Each of the three hall sensors are connected to an interrupt pin on the Arduino Mega. The timing and order of the pulses are read and used to determine

both the speed and direction of the reaction wheel. Hall-1 is connected to the Arduino on pin-2 (interrupt 0) and is used solely to determine the angular velocity of the reaction wheel. An interrupt function (triggered on each RISING pulse) uses a timer (in microseconds) to calculate the duration of time that has passed between pulses. Eq. 5.1 can then be used to calculate the angular velocity.

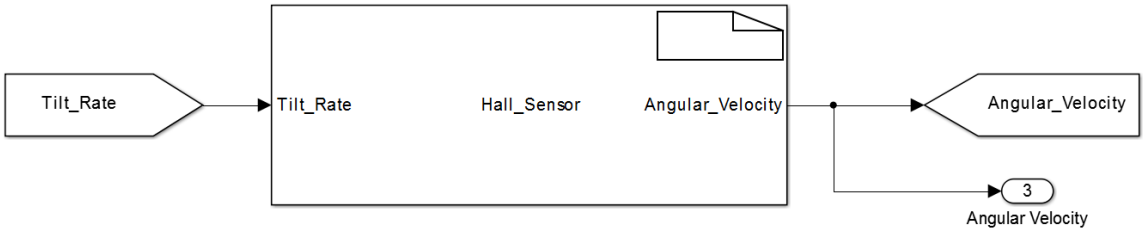

Figure 5.3: Hall Sensor S-Function in Simulink.

$$\omega = \frac{30}{\pi}\frac{1}{n\Delta t}(60 * 10^6)\ rad/sec \tag{5.1}$$

where n is the number of poles in the motor (16 for the DF45) and Δt is the time duration between RISING pulses captured by the interrupt (in microseconds).

Two additional interrupt functions are used to create a quadrature signal that determines the rotation direction via a lookup table. Since the three hall sensors are 90 degrees out of phase from one another, there are only six unique combinations of states that the hall sensors can produce at any given time. The three hall sensor readings are combined to form a binary number that is then converted to a decimal value (Table 5.1.2). In other words, the order of the decimal values is predefined and can be used to determine the rotation direction. For example, if the decimal reading of the Hall sensors is a 6, then the next reading can only be a 2 (left rotation) or a 4 (right rotation).

$$direction = 6 * previous\ decimal\ value + current\ decimal\ value \tag{5.2}$$

Table 5.1: Six possible Hall sensor states as binary and decimal values.

Hall-1	Hall-2	Hall-3	Binary Value	Decimal Value
0	0	1	001	1
0	1	1	011	3
0	1	0	010	2
1	1	0	110	6
1	0	0	100	4
1	0	1	101	5

↑ direction ↓

Table 5.2: Lookup table used to determine rotation direction. CCW rotation is positive and CW rotation is negative.

		Current Decimal Value					
		1	2	3	4	5	6
Previous Decimal Value	1			+1		-1	
	2			-1			+1
	3	-1	+1				
	4					+1	-1
	5	+1			-1		
	6		-1		+1		

Table 5.1.2 is converted into a 1D array (accessed by Eq. 5.2) to be used as a lookup table by the Arduino. The value pulled from the lookup table determines the rotation direction (+1 being clockwise and -1 being counterclockwise).

5.1.3 Digital-to-Analog Converters

The Arduino Mega lacks an on-board digital-to-analog converter, so it is incapable of outputting an analog voltage output. The MCP4725 receives an input voltage command (in 12-bit form) between 0-1023 which it maps to an output

voltage of 0-5 volts. The corresponding voltage command (in volts) is then sent and used to drive the motor controller.

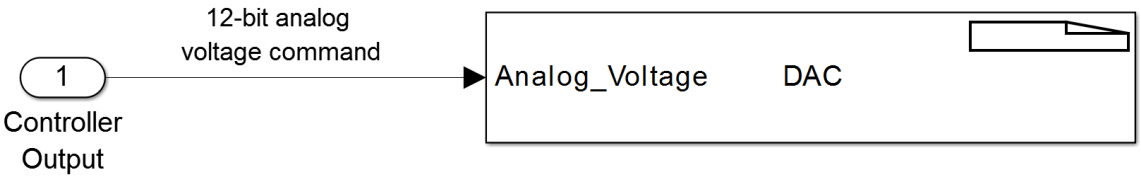

Figure 5.4: Digital-to-Analog Converter S-Function in Simulink.

5.1.4 Data Acquisition

Data is acquired through the Serial Print S-Function. The block diagram is designed with global *Goto* and *From* blocks that allow for information to be collected from anywhere in the block diagram. Desired data is connected to the S-Function as an input and will be output to any serial monitor accordingly. The first output is always a timer (set in microseconds) followed by the desired outputs in the same order as connected in the block diagram.

5.1.5 Library Definitions

Due to the way that the separate autocoded S-Functions are compiled and downloaded onto the microcontroller, library conflicts will (quite commonly) occur (i.e. Two (or more) S-Functions require the same library to function, but will cause an error during the compilation process if included twice (in the respective S-Functions that require the said libraries). As a result, a separate S-Function is included with no inputs or outputs. It's sole purpose is to include libraries that are required by more than one S-Function. It acts as a global source for other S-Functions to locate required libraries. In theory, all required libraries can be

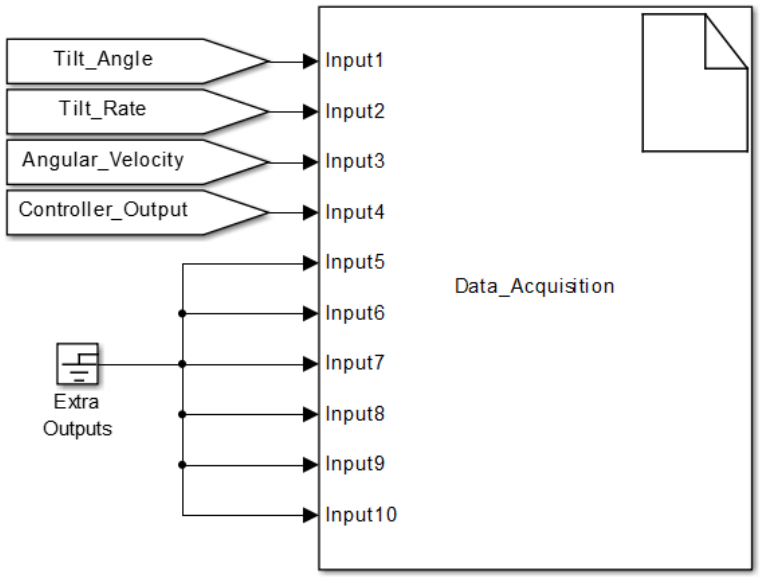

Figure 5.5: Data Acquisition S-Function in Simulink.

included in this block, but it will complicate the debugging process.

CHAPTER 6

CONTROLLER DESIGN

It is important to design controllers with spacecraft performance goals (such as reference tracking, attitude maneuvering, energy consumption minimization) in mind since the setup is to serve as a spacecraft attitude determination and control platform. As such, large emphasis is placed on minimizing controller effort to reduce associated power consumption, thus saving energy and minimize fuel/power consumption. Considering the fact that the life span of satellites is measured with years, even small savings in energy consumption (through reduced control effort) will become substantial over the whole lifetime of a spacecraft. This also has a direct impact on the overall mission objectives and reduces the associated cost of the mission. For the purposes of this setup, a

To address all those concerns, for this study, two controllers are developed for this purpose. The first is a proportional-integral-derivative (PID) controller.[FPEN10]

$$PID(s) = K_p(1 + \frac{1}{T_i s} + T_D s) \tag{6.1}$$

A PID controller is designed to achieve minimal control effort, and will serve as a baseline controller for comparison purposes. A second PID controller is also designed for reference tracking performance. This will provide insight on the performance and reliability of the PID tuner application and process.

A linear quadratic regulator[Oga10] (LQR) is also developed to achieve optimal performance and minimize energy consumption. The LQR controller

minimizes the cost function-J via weighting factors, Q and R, that penalize the transient and control energy, respectively, as also shown in Eq. 6.2.

$$min\ J = \int_{t1}^{t2} (x^T Q x + u^T R u)\ dt \tag{6.2}$$

Two different LQR controllers are also designed for comparison. In the first one, the weighting factors are equally penalizing the transient and control energy, while the second heavily penalizes control energy to minimize the control effort.

6.1 Observability and Controlability

The observability and controlability matrices (Eqs. 6.3 and 6.4) are both checked before controllers are designed. Both matrices are full rank, so the system is both observable and controllable.

$$\begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \tag{6.3}$$

$$\begin{bmatrix} B & AB & A^2B \end{bmatrix} \tag{6.4}$$

6.2 Simulation Results

Preliminary results from both controllers can be seen in Fig. 6.1 and Fig. 6.2. The curves in Fig. 6.2 are numerically integrated to find the total energy consumption of each controller when subjected to the same unitary impulse disturbance. This will be used as a direct measure of the controller's performance. Both the initial and improved LQR controllers provide reductions in energy consumption by $\sim$4% and $\sim$8% over the baseline PID controller, respectively.

The results also indicate PID controllers are difficult to reliably tune for optimal performance since there is no way to ensure that a local minimum has been reached with the current configuration. Various design attempts demonstrated that while the PID can be tuned for best time domain reference tracking performance or minimal control effort, simultaneously optimizing multiple objectives is found to be tedious and time-consuming, therefore impractical. On the other hand, the LQR performed as desired, and will be implemented on the experimental setup as the initial test controller.

Some limitations of the setup are found after the system has been fully identified, where RC servo and DC motor dynamics have yet to be identified, and is excluded from the content of this paper. Results will be included and reported in another future study. The constraints to the maximum tilt angle ($\theta_{b,max}$) will provide bounds for optimal controller design. As long as the maximum tilt angle is not exceeded by a given impulse, the controller can be further improved to lower energy expenditure.

Hardware limitations (such as maximum current or torque) directly influence the setup's dynamics and capabilities. Actuator saturation (primarily the DC motor) is a major constraint that limits the controller's ability to reject disturbances. Such limitations need to be taken into account during controller and model design. Otherwise, even if the controller demands more performance, the hardware will not be able to follow the commanded inputs, and the desired performance will not be achieved.

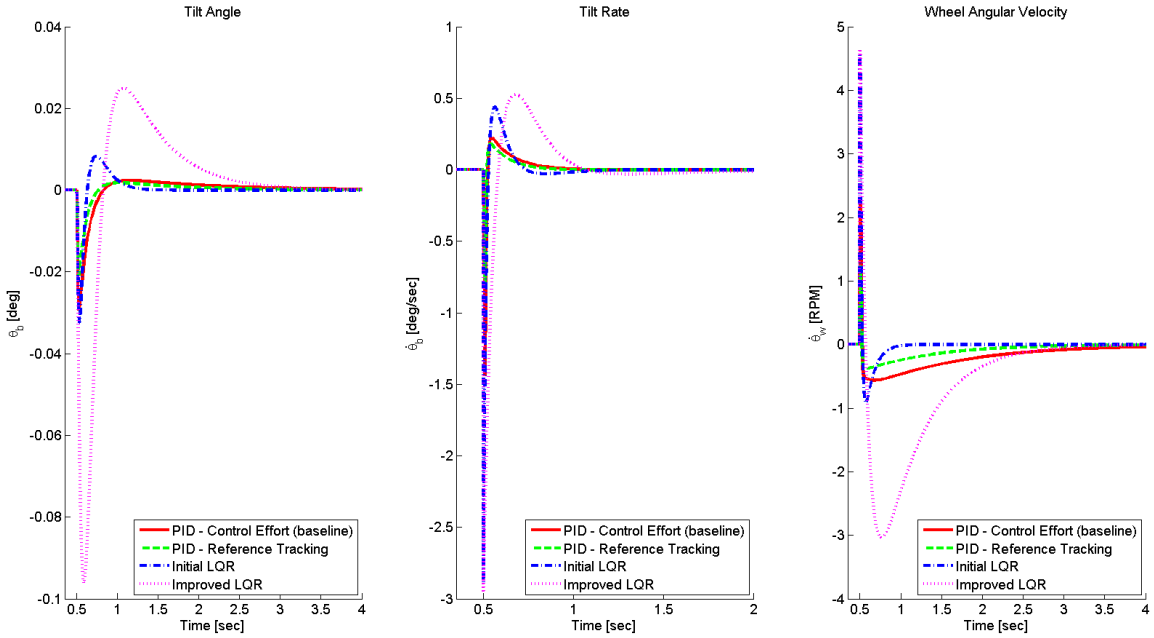

Figure 6.1: Comparison of PID and LQR controllers for the three states.

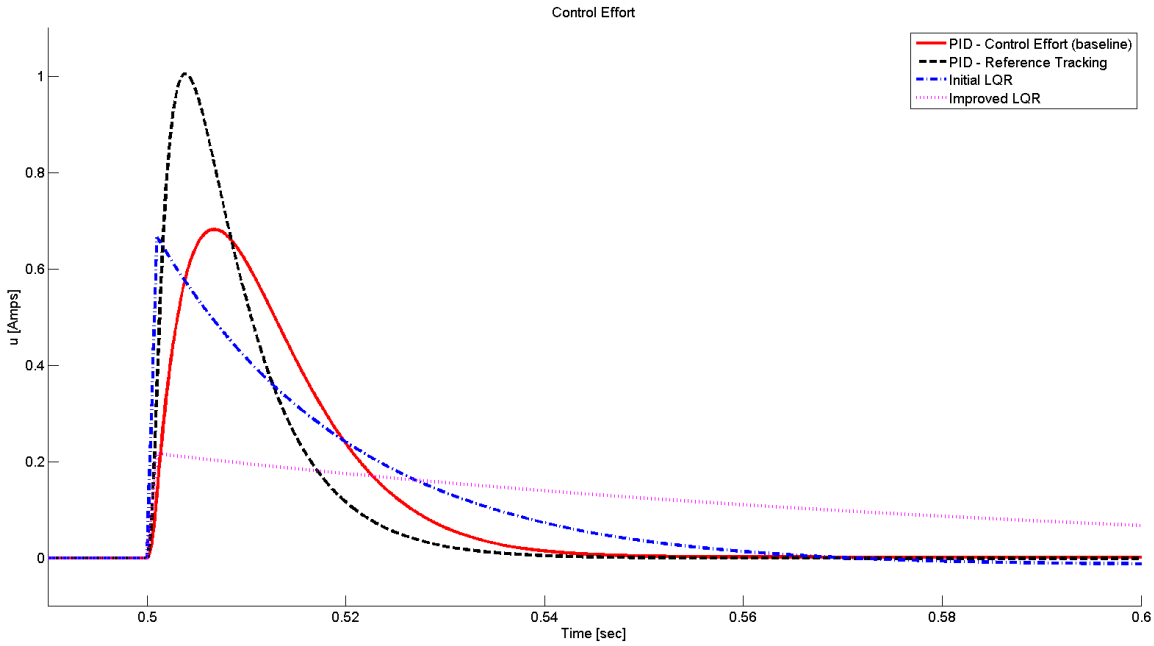

Figure 6.2: Controller effort of various PID and LQR controllers.

CHAPTER 7

ROBUSTNESS ANALYSIS

All models contain some degree of uncertainty. Parametric uncertainties are those associated with uncertainties in the parameters (masses, lengths, etc.) that define the system. There is little uncertainty with masses and lengths since they are typically measured very accurately. Moment of inertia and friction coefficient parameters are less likely be to accurately estimated, with the exception of I_w, so their values are more uncertain. Monte Carlo simulations are performed to test the controller's robustness to additive uncertainties in plant dynamics. Two different cases are run to analyze the robustness of the LQR controller. Case 1 uses more realistic uncertainties while Case 2 explores the effects of larger uncertainties. Fig. 7.1 shows that for reasonable uncertainties, the closed-loop system demonstrates stable behavior. However, for Fig. 7.2, larger uncertainties (particularly with I_b) produce potentially unstable responses. While LQR control guarantees asymptotic stability, there are no guarantees for robustness to uncertainties.

Table 7.1: Percent additive uncertainties used to perform robustness analysis.

Parameter	**Case 1**	**Case 2**
m_b	5	10
m_w	5	10
l_w	10	20
l_b	10	20
I_w	5	10
I_b	25	100
C_w	50	1000
C_b	50	1000
K_b	5	10

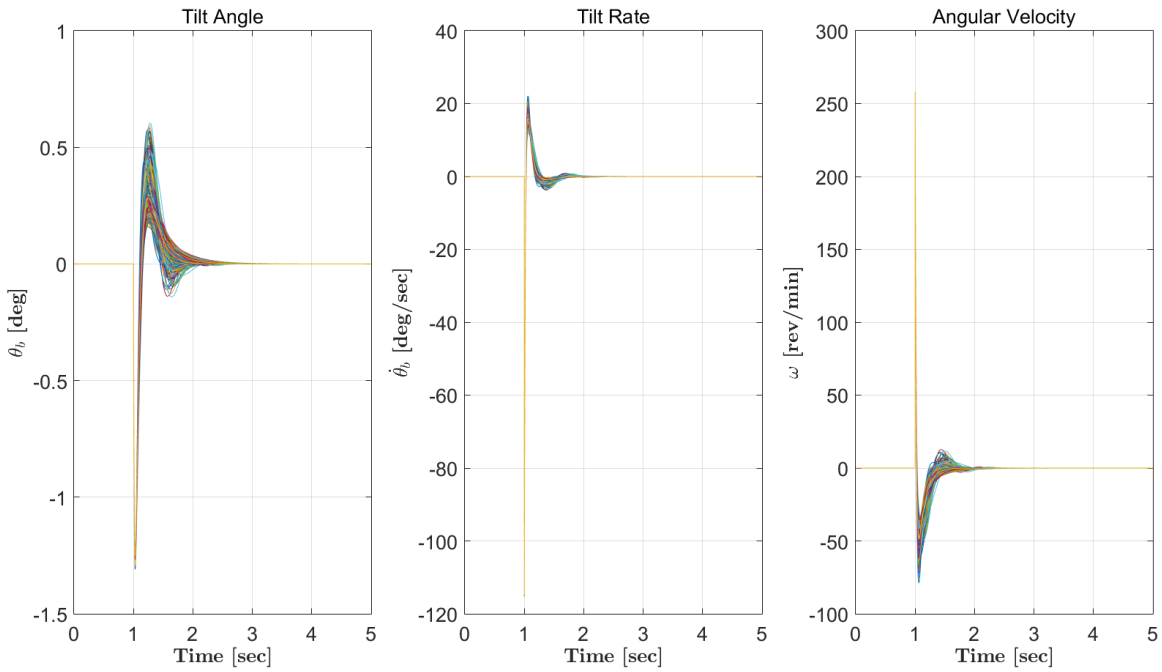

Figure 7.1: Case 1: Robustness analysis results from smaller parameter uncertainties.

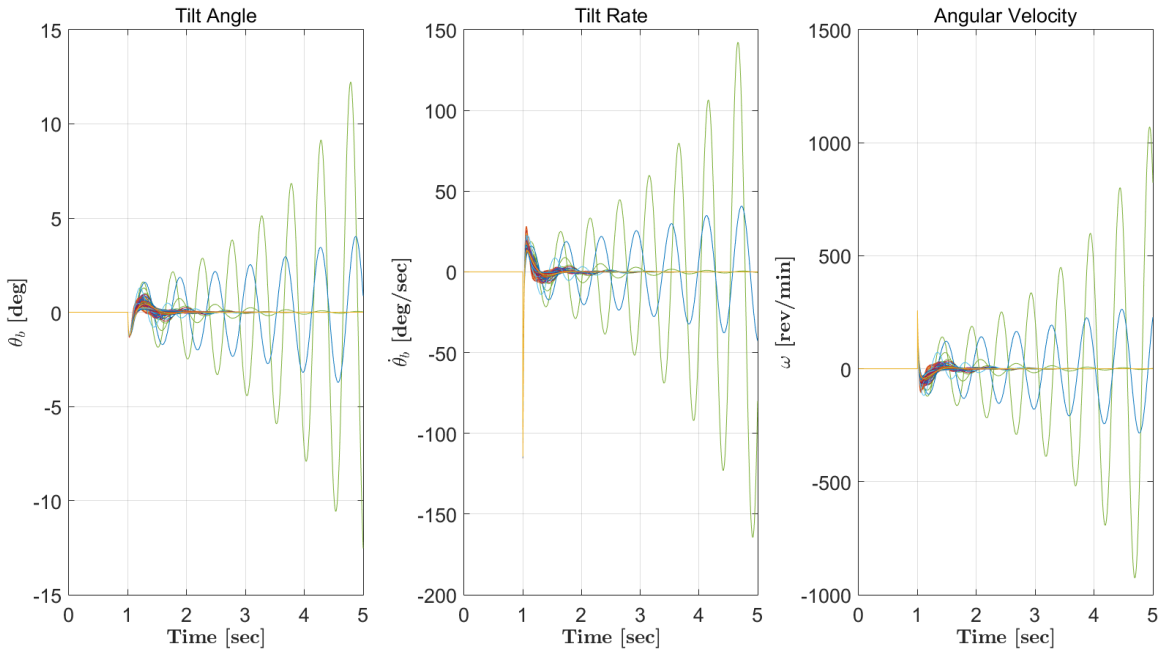

Figure 7.2: Case 2: Robustness analysis results from larger parameter uncertainties.

CHAPTER 8

EXPERIMENTAL RESULTS

For this portion of the study, the state space is discretized (Eq. 8.1) and a feedback control law (Eq. 8.2) is implemented on the Arudino Mega.

$$\begin{aligned} x[n+1] &= Ax[n] + Bu[n] \\ y[n] &= Cx[n] + Du[n] \end{aligned} \tag{8.1}$$

$$u[n] = -K_{LQR}x[n] \tag{8.2}$$

However, initial attempts using Simulink's Arduino environment proved unsuccessful. The 25 [Hz] sampling frequency limitation degraded the performance of the LQR controller to the point that the plant could not be stabilized. Fig. 8.1 characterizes the impact on the controller's performance based on the sampling frequency. In order to improve the performance, Arduino's native IDE is used to program and operate the controller as a standalone unit. The sampling frequency is then able to be increased up to 100 [Hz]. However, to provide extra time for data collection through the serial connection and prevent overrunning, the control loop is run at 50 [Hz].

With the increased sampling frequency, the initial tests of the LQR demonstrated the ability to reject impulse responses. However, as a direct result of an impulse disturbance, the CG of the pendulum shifts and gravity applies a constant torque due to the nonzero tilt angle. The gravity-induced step disturbance generates a nonzero steady state error from the 0-degree (vertical) reference point.

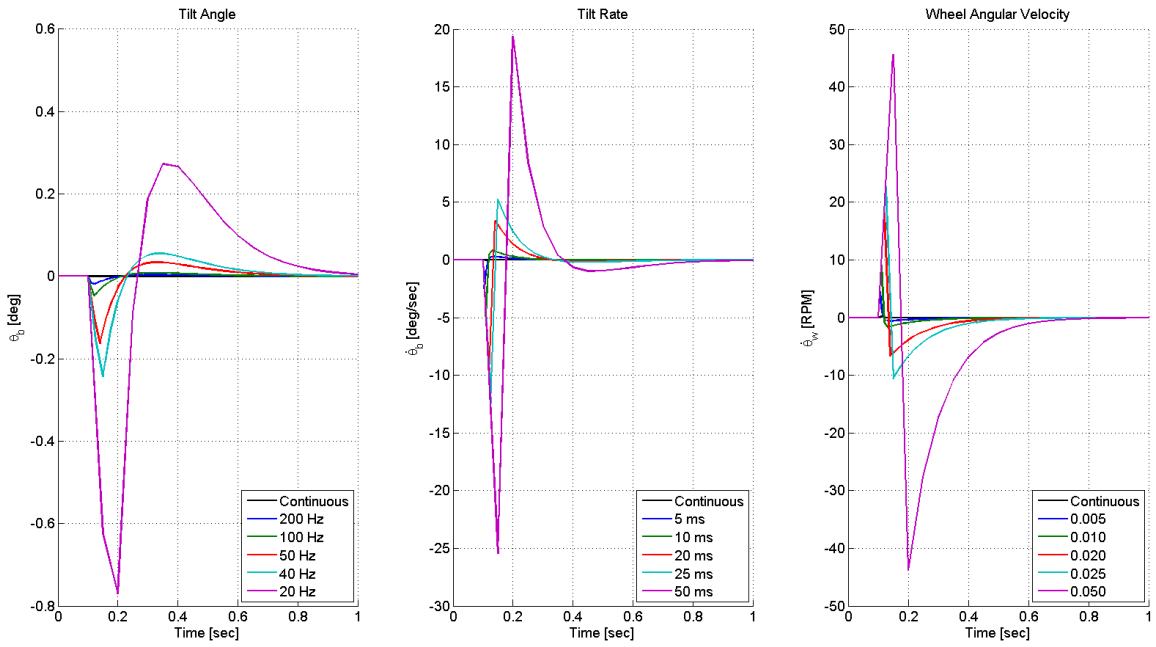

Figure 8.1: Continuous and discrete LQR controller performance for various sampling frequencies.

This drives the DC motor to produce constant angular acceleration to generate the necessary torque to maintain its new (nonzero) orientation. Once the DC motor reaches the imposed saturation limit, the setup will destabilize.

In order to correct the nonzero steady state error due to gravity, the state space representation is augmented to provide integral action. Namely, the A and B matrices are augmented with a fourth state, the integral of error in the tilt angle. The derivative of the fourth state is then simply the error, e, in the tilt angle from the 0-degree reference.

$$e = y - ref = \theta_b \tag{8.3}$$

$$\bar{A} = \begin{bmatrix} A_{3x3} & 0_{3x1} \\ 1 & 0_{1x3} \end{bmatrix}_{4x4} \qquad \bar{B} = \begin{bmatrix} B_{3x1} \\ 0 \end{bmatrix}_{4x1} \tag{8.4}$$

$$\begin{bmatrix} \dot{x} \\ e \end{bmatrix} = \begin{bmatrix} \bar{A} \end{bmatrix} \begin{bmatrix} x \\ \int e \end{bmatrix} + \begin{bmatrix} \bar{B} \end{bmatrix} u \tag{8.5}$$

A prefilter gain is added to facilitate the elimination of any steady state error. The block diagram of the control strategy can be seen in Fig. 8.2.

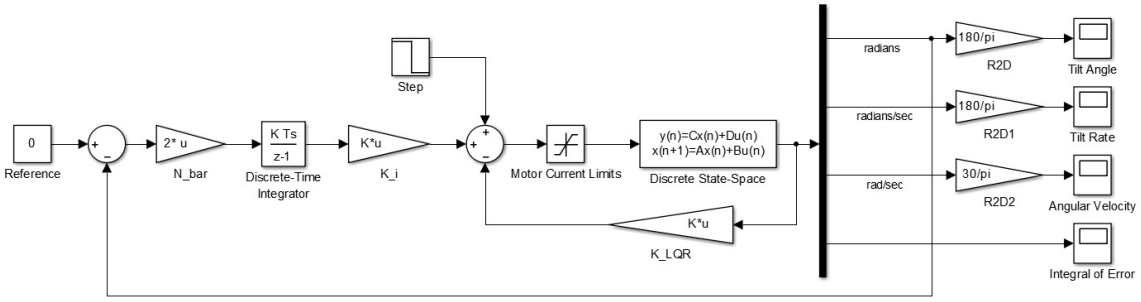

Figure 8.2: Control strategy including prefilter gain and integral action to eliminate the steady state error in tilt angle.

Fig. 8.5 shows the time histories of the three states overlaid to provide an idea of the system response. Two impulse disturbances (-2 and +7 degrees) are injected into the system while the feedback controller is stabilizing the plant at zero degrees (vertical equilibrium) at roughly 1 and 9 seconds. Two of the three states are scaled down (by 10 and 50 times) to provide an intuitive idea of how the system reacts to an impulse disturbance.

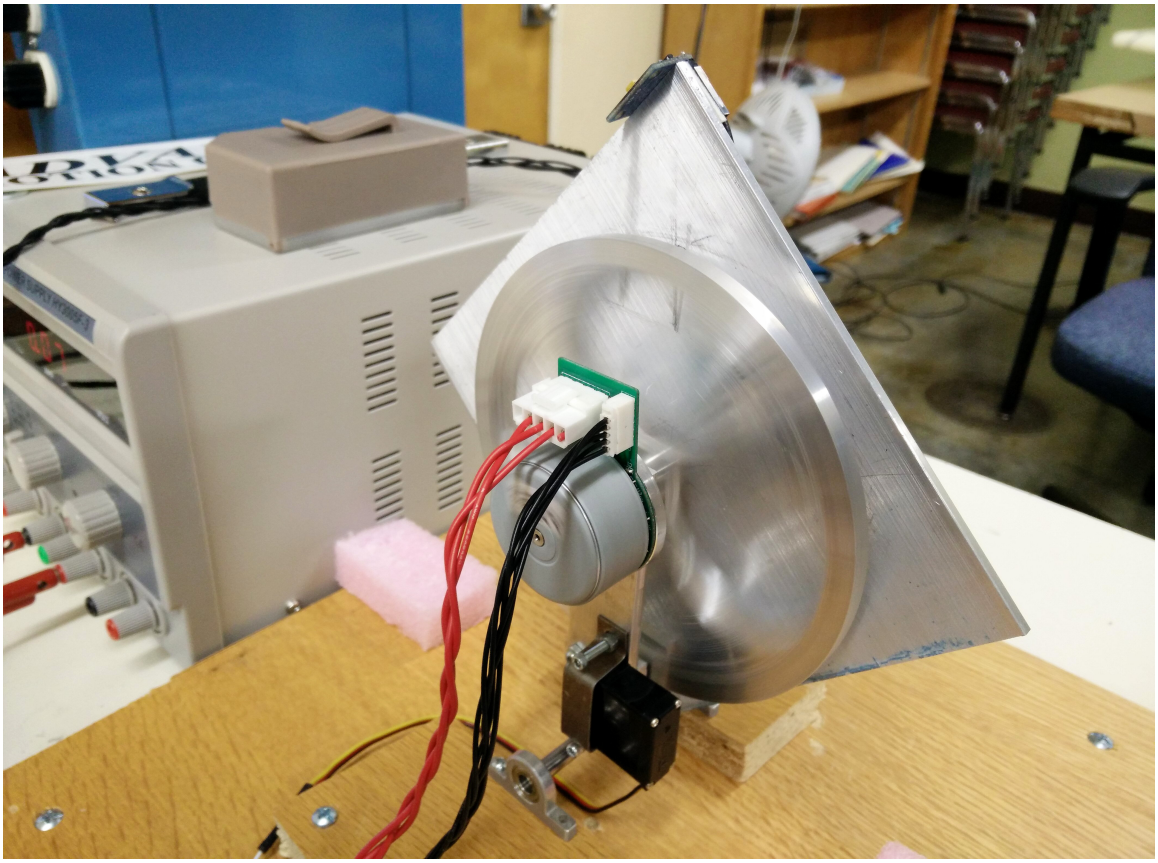

Figure 8.3: Experimental setup during a balancing maneuver.

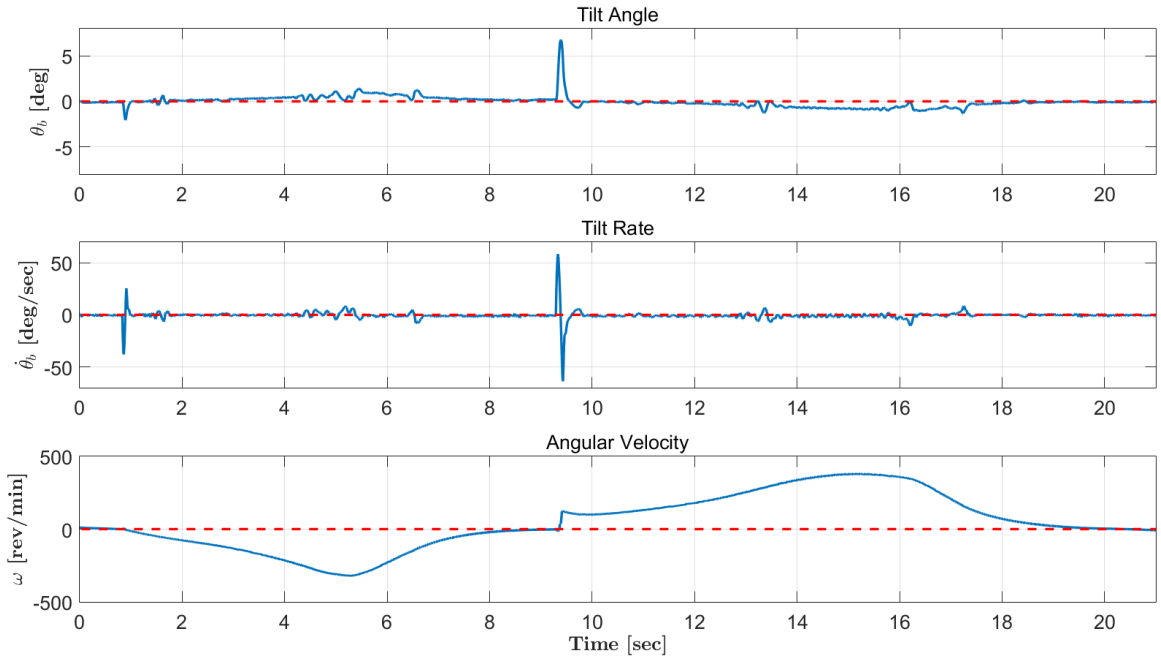

Figure 8.4: System outputs due to negative and positive impulse disturbances at 0.85 and 9.30 seconds.

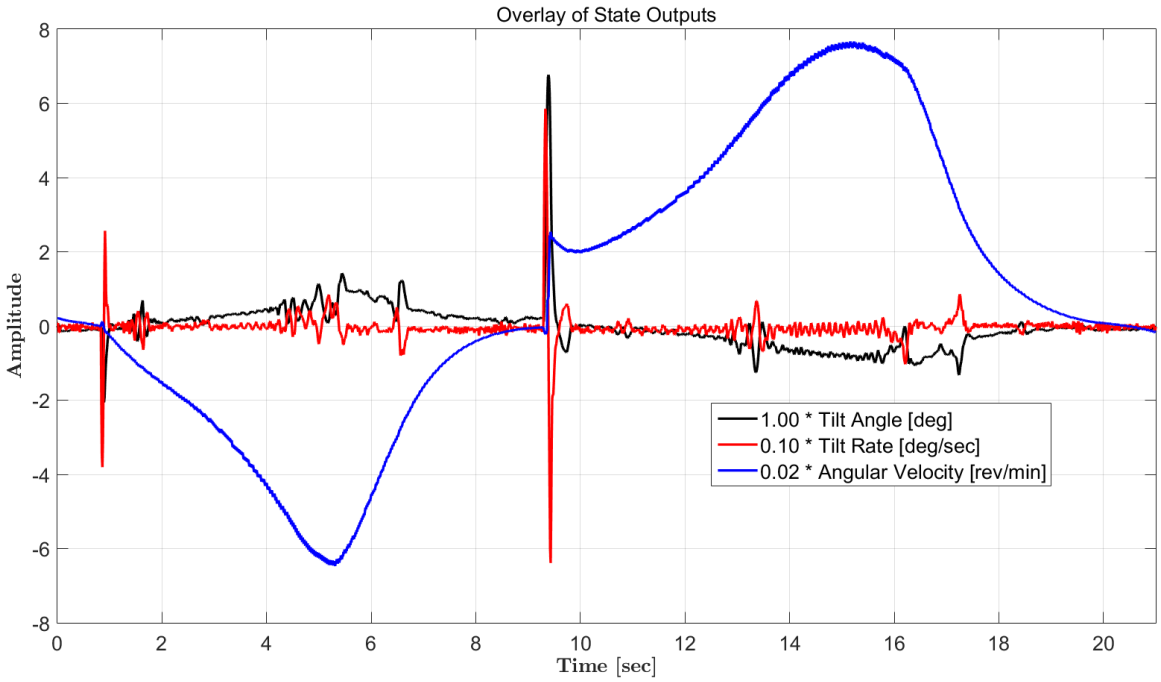

Figure 8.5: Overlay of system outputs due to negative and positive impulse disturbances at 0.85 and 9.30 seconds.

CHAPTER 9

CONCLUSION

In this study, a 1-D reaction wheel inverted pendulum prototype has been designed to serve as a spacecraft attitude determination and control research platform. PID and LQR controllers have been designed in Simulink and analyzed for best performance with minimal control effort. The LQR demonstrated energy savings of up to $\sim$8% over the PID in simulation. However, the LQR's energy savings are attributed to the difficulty of tuning a PID for minimum controller effort whereas it is done optimally by design of a LQR. Initial attempts to streamline the controller design process through Simulink's Arduino External Mode environment was unsuccessful due to sampling time limitations. Instead, Arduino Mega is used as a standalone controller where the autocode is directly loaded form Simulink to overcome the issue. A LQR was implemented on the experimental setup and the setup is able to reject both impulses and step disturbances and balance successfully. Robustness analysis demonstrated the LQR controller performs well (all responses were stable) for acceptable uncertainties. Total (including tax and shipping) cost was kept to a minimum ($480).

CHAPTER 10

FUTURE WORK

System identification will be performed to obtain a more accurate open-loop state-space representation. Various controllers will be tested on the 1-D prototype and experimental results on their energy consumption will be collected for analysis. A more durable braking mechanism is currently under development for the jump-up maneuver. Once complete, the 3-D setup will be manufactured by reproducing the 1-D version and assembling the 3-axis controlled spacecraft.

REFERENCES

[Azo93] Ruth Azor, *Momentum Management and Torque Distribution in a Satellite with Reaction Wheels*, Israel Society of Aeronautics and Astronautics, 33rd Israel Annual Conference on Aviation and Astronautics, February 1993, pp. 339–347.

[BBC13] Ramaprakash Bayadi, Ravi N. Banavar, and Dong Eui Chang, *Characterizing the reachable set for a spacecraft with two rotors*, Systems & Control Letters **62** (2013), no. 6, 453 – 460.

[Bro02] C.D. Brown, *Elements of spacecraft design*, AIAA education series, American Institute of Aeronautics & Astronautics, 2002.

[CJC14] Calvin Coopmans, AustinM. Jensen, and YangQuan Chen, *Fractional-order complementary filters for small unmanned aerial system navigation*, Journal of Intelligent & Robotic Systems **73** (2014), no. 1-4, 429–453 (English).

[Cro11] Corey Whitcomb Crowell, *Development and analysis of a small satellite attitude determination and control system testbed*, Ph.D. thesis, Massachusetts Institute of Technology, 2011.

[DFF77] W. J. Devey, C. F. Field, and L. Flook, *An active nutation control system for spin stabilised satellites*, Automatica **13** (1977), no. 2, 161–172.

[FPEN10] G.F. Franklin, J.D. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*, Alternative Etext Formats, no. v. 10, Pearson, 2010.

[FW90] R. Fleeter and R. Warner, *Guidance and control of miniature satellites*, Automatic Control in Aerospace 1989, IFAC Symposia Series, Pergamon, Oxford, 1990, pp. 243 – 248.

[GJX13] Haichao Gui, Lei Jin, and Shijie Xu, *Attitude maneuver control of a two-wheeled spacecraft with bounded wheel speeds*, Acta Astronautica **88** (2013), no. 0, 98 – 107.

[GMTD12] M. Gajamohan, M. Merz, I. Thommen, and R. D'Andrea, *The cubli: A cube that can jump up and balance*, International Conference on Intelligent Robots and Systems, 2012, pp. 3722–3727.

[IFS93] R. Iwens, A. Fleming, and V. Spector, *Precision Attitude Control with a Single Body-Fixed Momentum Wheel*, Mechanics and Control of Flight Conference, February 1993, pp. 339–347.

[ITT08] Atsushi Iwakura, Shin-ichi Tsuda, and Yuichi Tsuda, *Feasibility study on three dimensional reaction wheel*, Proceedings of the School of Science of Tokai University, Series E 33, 2008, pp. 51–57.

[IV10] Zuliana Ismail and Renuganth Varatharajoo, *A study of reaction wheel configurations for a 3-axis satellite attitude control*, Advances in Space Research **45** (2010), no. 6, 750 – 759.

[JT03] Dongwon Jung and Panagiotis Tsiotras, *A 3-dof experimental test-bed for integrated attitude dynamics and control research*, AIAA Guidance, Navigation and Control Conference, Austin, TX, John Wiley & Sons, 2003, pp. 03–5331.

[KVKT03] Byung Moon Kim, E. Velenis, P. Kriengsiri, and P. Tsiotras, *Designing a low-cost spacecraft simulator*, Control Systems, IEEE **23** (2003), no. 4, 26–37.

[MDdC09] Jordan Meyer, Nathan Delson, and Raymond A de Callafon, *Design, modeling and stabilization of a moment exchange based inverted pendulum*, 15th IFAC Symposium on System Identification, 2009, pp. 462–467.

[Oga10] K. Ogata, *Modern control engineering*, Instrumentation and Controls Series, Prentice Hall, 2010.

[PC03] Mason A. Peck and Andrew R. Cavender, *An airbearing-based testbed for momentum-control systems and spacecraft line of sight*, Proceedings of the 13th AAS/AIAA Space Flight Mechanics Winter Meeting, no. AAS 03-127, 2003, pp. 427–446.

[SCL01] Mark W. Spong, Peter Corke, and Rogelio Lozano, *Nonlinear control of the reaction wheel pendulum*, Automatica **37** (2001), no. 11, 1845 – 1851.

[Sid97] M.J. Sidi, *Spacecraft dynamics and control: A practical engineering approach*, Cambridge Aerospace Series, Cambridge University Press, 1997.

[Sol] *Solidworks 2014, available at http://www.solidworks.com.*

[SPH03] Jana L. Schwartz, Mason A. Peck, and Christopher D. Hall, *Historical review of air-bearing spacecraft simulators*, Journal of Guidance, Control and Dynamics **26** (2003), no. 4, 513–522.

[XQLQ06] Ge Xinsheng, Zhang Qizhi, and Chen Li-Qun, *Optimal motion planning for a rigid spacecraft with two momentum wheels using quasi-newton method*, Acta Mechanica Solida Sinica **19** (2006), no. 4, 334 – 340.

APPENDIX A

B15A8 PIN-OUT

PIN FUNCTIONS

P1 - Signal Connector

Pin	Name	Description / Notes	I/O
1	+10V 3mA OUT	±10 V @ 3 mA low power supply for customer use. Short circuit protected. Reference ground common with signal ground.	O
2	SIGNAL GND		GND
3	-10V 3mA OUT		O
4	+REF IN	Differential Reference Input (±10 V Operating Range, ±15 V Maximum Input)	I
5	-REF IN		I
6	-TACH IN	Negative Tachometer Input (Maximum ±60 V). Use signal ground for positive input.	I
7	+TACH / GND	Positive Tachometer Input and Signal Ground	GND
8	CURRENT MONITOR	Current Monitor. Analog output signal proportional to the actual current output. Scaling is 2 A/V. Measure relative to signal ground.	O
9	$\overline{\text{INHIBIT IN}}$	TTL level (+5 V) inhibit/enable input. Leave open to enable drive. Pull to ground to inhibit drive. Inhibit turns off all power devices.	I
10	+V HALL 30mA OUT	Low Power Supply For Hall Sensors (+6 V @ 30 mA). Referenced to signal ground. Short circuit protected.	O
11	GND	Signal Ground	GND
12	HALL 1	Single-ended Hall/Commutation Sensor Inputs (+5 V logic level)	I
13	HALL 2		I
14	HALL 3		I
15	CURR REF OUT	Measures the command signal to the internal current-loop. This pin has a maximum output of ±7.25 V when the drive outputs maximum peak current. Measure relative to signal ground.	O
16	FAULT OUT	TTL level (+5 V) output becomes high when power devices are disabled due to at least one of the following conditions: inhibit, invalid Hall state, output short circuit, over voltage, over temperature, power-up reset.	O

P2 - Power Connector

Pin	Name	Description / Notes	I/O
1	MOTOR A	Motor Phase A	O
2	MOTOR B	Motor Phase B	O
3	MOTOR C	Motor Phase C	O
4	POWER GND	Power Ground (Common With Signal Ground)	GND
5	HIGH VOLTAGE	DC Power Input	I

Figure A.1: Correct Pinout for the B15A8