2013

Stabilization of an inverted pendulum using control moment gyros

Chris Joel Walck
Iowa State University

Stabilization of an inverted pendulum using control moment gyros

by

Christopher Joel Walck

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Mechanical Engineering

Program of Study Committee:
Greg Luecke, Major Professor
Atul Kelkar
Ratnesh Kumar

Iowa State University

Ames, Iowa

2013

TABLE OF CONTENTS

ABSTRACT

Gyroscopes have played an important role in the science community with uses ranging from simple classroom demonstrations to cutting edge technological advancements. They offer a unique source of torque that has been proven useful in a wide range of applications. One common educational demonstration calls for a person to stand on a pivoting platform, and hold a spinning bicycle wheel with one hand on each side of the axle. As the person rotates their arms, causing precession of the bicycle tire, they begin to spin on the platform. This is due to the dynamic effect of the gyroscope and is a perfect example of a control moment gyro. This paper presents the use of control moment gyros as a compact way of dynamically controlling an inverted pendulum. The dynamic characteristics are derived for a dual-gyroscope configuration that generates torque proportional to the velocity about the gimbal axis. Classical control theory is used to design a controller that not only stabilizes the pendulum, but also controls the gyroscopes to return to a neutral steady state position. Control gains are adjusted to account for noise effects and to compensate for parameter errors, and an accelerometer is used to replace the potentiometer measuring the vertical angle. With the theory and background in place, experimental results are presented to verify the predicted response and validate the control approach. The end result is a stable system that is resilient to a broad range of external influences and erroneous measurements.

CHAPTER 1. INTRODUCTION

The gyroscope is an interesting display of motion and dynamics. The movement of a gyroscope's spin axis causes a torque which results in the precession seen in spinning tops and other gyroscope based applications. With the dynamics of gyroscope motion understood, people began to develop creative ways to use them for practical purposes. Perhaps the first notable implementation of gyro-stabilization was the gyrocar, created by Russian Count Peter Schilovski, in 1912. The car drove on two inline wheels and used a large gyroscope torsion system to balance the car during operation. When the car leaned in one direction, a mechanical component would measure this angle and apply a clutch to use gyroscope momentum to apply torque in a direction that would balance the car. In 1962, a similar implementation was used on a monorail train system, [16] and [20]. Since then, control momentum gyroscopes have had continued use in ship stabilization and in new concept cars that use two inline wheels with a gyro-stabilized chassis. They have also been used to stabilize free space objects in satellite attitude control, [13] [14] [15], and in some early missile guidance systems [12]. More recently, a space ready robotic arm was developed and tested using control moment gyros to control the kinematics, rather than electric motors [23]. This configuration is beneficial because while electric motors cause unwanted moments in the robot chassis in free space, the control moment gyros can drive the motion of the robotic arm without causing these unwanted moments on the chassis.

The idea of using gyroscopes to balance an object is part of a much broader effort to stabilize inverted pendulums. In general, the inverted pendulum can be defined as a pole with

a mass on top, and a pivot point on the bottom. Alone, the pendulum has a vertical unstable equilibrium point, and the challenge is to implement some kind of mechanism that can influence the motion of the pendulum with a control system that stabilizes it about the equilibrium point. The inverted pendulum, originally developed for the stabilization of rockets, is now being applied to a large range of different control applications. The inverted pendulums in [4] [5] [6] [7] and [8] describe a mobile chassis to dynamically balance the pendulum. This is extremely effective whenever a wheeled chassis is available for use. The Segway [19] is an interesting, commercial application of this method of control. It uses two side-by-side wheels on a mobile base to drive a passenger in a vertical orientation. It is an effective transportation device on flat grounds and paved roadways. However, in an environment where control cannot be achieved by chassis movements, other methods were developed. One such solution, the reaction wheel pendulum [1] [2] [3], achieves stability with the torque exerted by an accelerating flywheel. Even more exotic solutions use the manipulation of actuators or arms to cause the momentum and center of gravity imbalances used to stabilize the pendulum. One such implementation, is shown in [9], uses a robotic arm to accelerate the base of an inverted pendulum such that the whole system balances. However, many of these systems are impractical because the weight or shape of the active components will always be a hindrance in a practical application. The problem with using mechanical moments to balance a robot is that the more massive the robot chassis is compared to the mass of the balancing mechanism, the less influence the balancing system has on the body. This results in lower controllable range of motion. Therefore, stabilizing the pendulum would then require more acceleration in the reaction wheel pendulum and balancing arm. The same problem can be solved with control

moment gyros simply by increasing the gyroscope angular velocity or radius. An inverted pendulum stabilized by control moment gyros has some clear advantages because it can provide a large amount of control torque in a small compact package. It can also be significantly more efficient than some of the systems mentioned above because the gyros can spin with a very efficient bearing system using very little power to keep the gyros spinning, and the movements required to create the control torques require almost no work given that the pendulum moves only a few degrees. This is a huge advantage over the reaction wheel pendulum, and the segway, which would require a proportional amount of power per unit torque desired from the balancing mechanism. Therefore, many robotics applications where a light weight active stabilization system is needed could benefit from this research.

The motivation behind this research is to enhance the maneuverability of biped robots. Many biped robots are based on the human being, with a desired motion emulating that of a walking human. The human body can be thought of as an inverted pendulum employing two balancing systems discussed above. The legs stabilize the body by propelling it forward and backward, similar to the Segway operation, while the arms swing from side to side for further stabilization, similar to the self-balancing arm, or reaction wheel pendulum. However modern humanoid robots and bipeds do not have the kinematics or control to achieve such fluid motion as displayed by the human body. Without such advanced control achieved by human walking, the underlying limit of most bipedal robot maneuverability is the size of its footprint. A walking robot is only stable if the normal force vector of the CG is within the footprint. Therefore, in order to achieve stable walking without limiting mobility and agility, the robot must either have extremely accurate kinematic control, or very large feet.

Two commercially available humanoid robots are shown in **Figure 5** and **Figure 6**. They were designed for the toy industry, but since have been modified to fit a range of different hobby applications such as humanoid robot soccer games or humanoid sumo robot matches.

Figure 1. "Robosapien"

Figure 2. "I-SOROBOT"

In general, the reason for large feet is to make it easy to keep the normal force of the CG inside the footprint. These robots serve as examples of the 'large foot' approach to biped robots. These robots have a large range of pre-programmed movements and routines. "Robosapien," for instance, even can dance. With such aggressive movements, especially in an event such as a sumo fighting match, many humanoid robots are made with large feet to accommodate this motion. In these cases, without the luxury of advanced research laboratories, intellect, and a large budget, it is just more economical to give the robot large feet. "Robosapian," for example, has a footprint of about 1/3rd of its body height.

Many efforts have been made to improve the maneuverability of bipedal robots despite their small foot print. Two humanoid robots built by Honda and Toyota are shown in **Figure 7** and **Figure 8**.

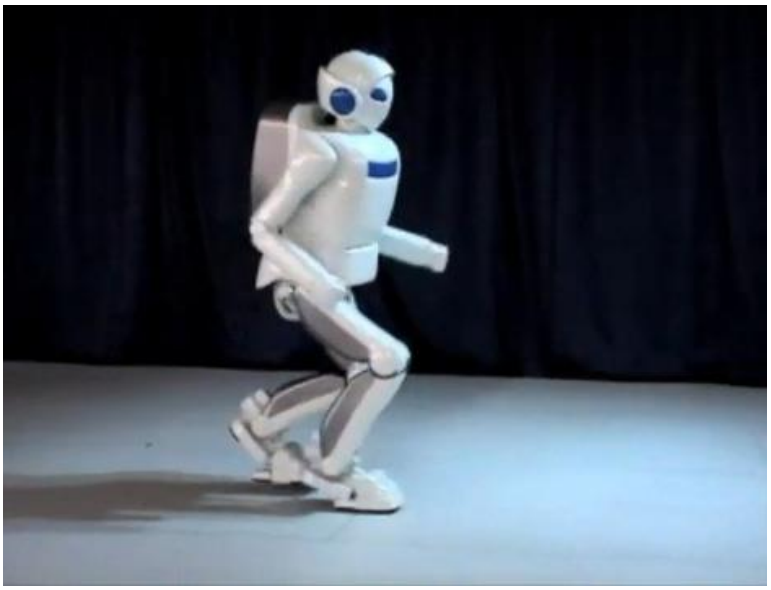

Figure 3. Toyotas Humanoid Robot

Figure 4. Hondas "Asimo" Humanoid Robot

These robots have set a high standard for the humanoid robot industry, and they have some impressive motion capabilities. Honda recently released test results of their humanoid robot “Asimo” having stable bipedal motion at a top speed of 6 Km/h [11]. Toyotas humanoid robot can jog at a speed of 7 km/h [10]. In order to achieve these speeds, very accurate motor control in the leg actuators is needed to achieve a smooth enough motion for these speeds. Millions of dollars were spent developing accurate enough control models, dynamics, and hardware to be able to move at these speeds in the most ideal of environments. These robots are impressive examples of the “use of extremely accurate kinematic control” approach to biped stabilization.

Using an active stabilization system presented in this work, a biped could have very small feet and be stable enough to sprint across the floor, with only a simple model of the robot dynamics. This approach can be considered the middle ground between large feet on bipeds and immensely complex dynamical modeling and control.

This technology can also be extended to other applications. One that comes to mind is a stabilizing backpack for people with trouble walking, or who have lost their sense of balance. It could help people with disabilities increase their mobility, and aid in walking rehabilitation for people with injuries. Stabilizing gyros could also be used in certain instances of haptic feedback, where torque is needed without the use of a kinematic joint creating that torque. These devices could be popular in the video game industry as methods of interaction between the gamer and their virtual reality. Haptic feedback has existed in the form of joystick vibration in some videogame consoles, and adding torques to controllers has the potential to add a new level of interactive gameplay. Some interesting areas of development could include a wireless sword handle that feels like you are hitting the sword of a virtual adversary, or a wireless steering wheel that gives torque feedback based on the turns taken in a road course.

This paper will present a foundation for some of the applications expressed above, as we describe the development of an active gyroscope stabilization system that is used to stabilize an inverted pendulum.

CHAPTER 2. TECHNICAL DEVELOPMENT

A two-gyroscope system was developed to control one axis of an inverted pendulum. The gyros were configured as shown in **Figure 5**. Rotation of each gyroscope by an angle of ϕ causes a torque, τ, that is proportional to the rate of rotation, $\dot{\phi}$.

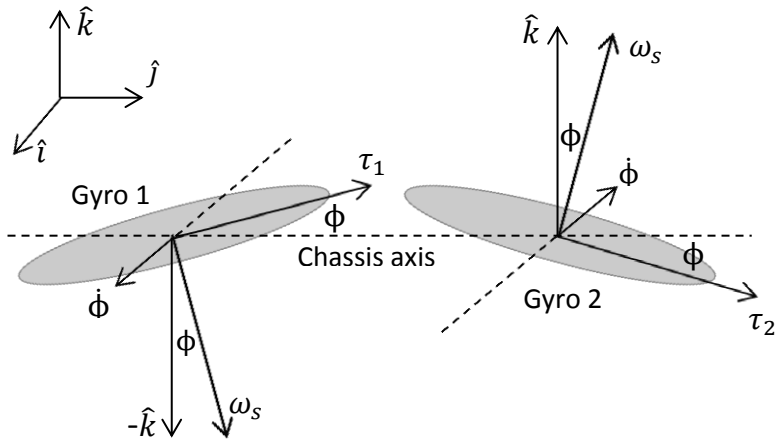

Figure 5. The configuration of the control moment gyroscope system, and its associated torques.

Each gyroscope is mounted on a powered gimbal which rotates the gyroscope about the gimbal axis by an angle ϕ. The rotation of each gyro creates a torque, τ_1 and τ_2, that can be added to obtain the overall torque exerted on the chassis. The torque exerted on the chassis frame is found by:

$$\tau_i = \dot{\phi} \times \mathrm{I}\omega_s \qquad \textbf{Equation 1}$$

For both gyros, the placement of the rotation axis causes the torques to add in only one axis:

$$\tau_{chassis} = \tau_1 + \tau_2 = 2\mathrm{I}\dot{\phi}\omega_s \cos\phi\, \hat{j} \qquad \textbf{Equation 2}$$

Here, ϕ is the gyroscope angle, 'I' is the moment of inertia of each gyro wheel, and ω_s is the angular velocity of each gyroscope wheel.

The chassis is mounted on top of a pole which is free to pivot about an axle mounted to a static base. A complete diagram is shown **Figure 6**.

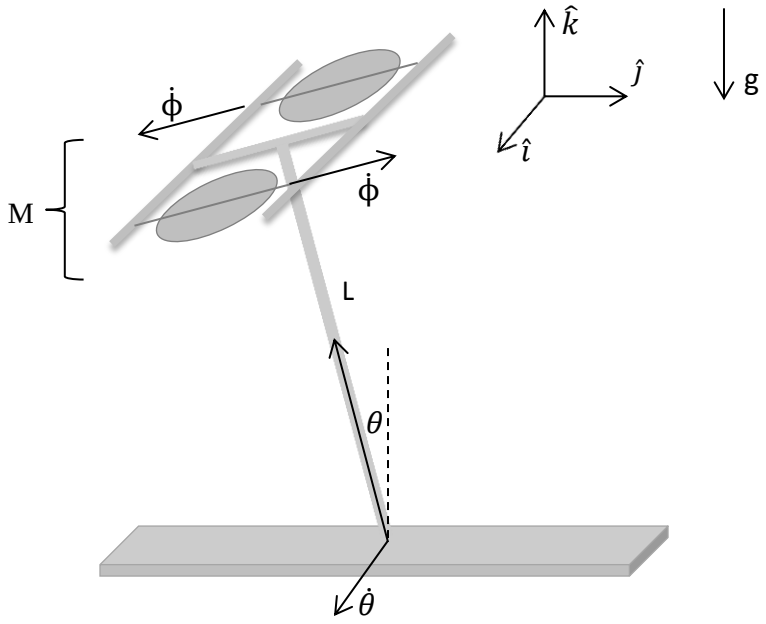

Figure 6. Inverted pendulum setup

In **Figure 6**, the circular objects represent the gyroscopes in the two gyroscope system. Each gyroscope is mounted on a motor and spins with an angular velocity of ω_s. The motor and gyroscope pair is then mounted on a gimbal, which rotates about an axis represented by $\dot{\phi}$. The gimble axis is perpendicular to ω_s and $\dot{\theta}$.

The equations of motion governing the dynamics of this inverted pendulum are given below.

$$\ddot{\theta} = \frac{g}{L}\sin\theta + \frac{2\mathrm{I}\omega_s}{ML^2}\dot{\phi}\cos\phi\cos\theta \qquad \textbf{Equation 3}$$

The linear equations predict the motion for small motions around the angle $\theta = 0$ and $\phi = 0$.

$$\ddot{\theta} = \frac{g}{L}\theta + \frac{2I\omega_s}{ML^2}\dot{\phi} \;\; for \begin{cases} \theta \cong 0 \\ \phi \cong 0 \end{cases} \qquad \textbf{Equation 4}$$

The transfer function relating the pendulum angle and gimbal axis rate is:

$$T(s) = \frac{\theta}{\dot{\phi}}(s) = \frac{\frac{2I\omega_s}{ML^2}}{s^2 - \frac{g}{L}} \qquad \textbf{Equation 5}$$

We linearize the system at $\theta = 0$ in order to use classical control to stabilize the pendulum. Considering this linearized model is a good practical approach, because it allows us to stabilize the system about the unstable equilibrium point at $\theta = 0$. We have also considered any damping forces to be negligible in this model.

The prototype uses a small RC hobby style servo to manipulate each gimbal. As the servo rotates the gimbal, the spin axis of each motor and gyro rotates about the spin axis of the gimbal, as described in **Equation 1**. In order to generate torque from the gyros, an angular velocity is applied on the gyro gimbals. Because the RC servos track a reference position, the dynamic response of the servos is important when a desired velocity is required by the gyro gimbals. In order to achieve an accurate pendulum model, the servo dynamics were determined experimentally. We used a common second order dynamic system to model the

response of the servos. The assumed servo model is shown in **Figure 7** as functional block diagram.

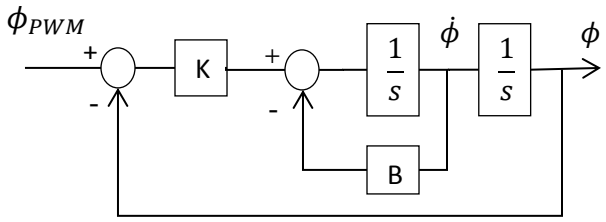

Figure 7. PWM servo system functional block diagram

In **Figure 7**, ϕ_{PWM} is the commanded position sent to the servo as a PWM signal. Experimental measurements of the servos step response were taken to identify the servo dynamics, and will be described in detail in the results section of this paper. **Equation 2** shows that the torque generated by the gyros is proportional to the rate of motion around the gimbal axis. Using the RC servo rate as the output, we can integrate the rate command to obtain a desired servo position. This block diagram is shown in **Figure 8.**

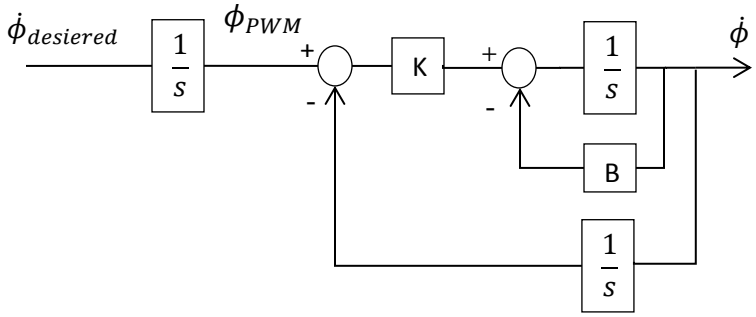

Figure 8. Velocity commanded servo system functional block diagram

Reducing the block diagrams in **Figure 7** and **Figure 8**, results in the same transfer function between the desired and actual rates, and the desired and actual positions. The servo system can then be characterized using a simple second order transfer function.

$$\frac{\dot{\phi}_{desired}}{\dot{\phi}_{actual}}(s) = \frac{\phi_{desired}}{\phi_{actual}}(s) = \frac{K}{s^2 + Bs + K} \qquad \textbf{Equation 6}$$

In **Equation 6**, $\phi_{desired}$ represents the PWM command sent to the servo. The servo then rotates to this position commanded with the dynamics described in **Figure 7**. Although the servo motion is approximated by a second order transfer function, it was found that the specific servos used in this project have a small time delay that accounts for computation and communication protocol between the servo and controller. In order to model this, we use a third order pade approximation of a time delay. Therefore, **Equation 7** shows a more accurate model of the servo system.

$$\frac{\dot{\phi}_{desired}}{\dot{\phi}_{actual}}(s) = \frac{\phi_{desired}}{\phi_{actual}}(s) \approx \left(\frac{1 - \frac{\tau}{2}s + \frac{\tau^2}{10}s^2 - \frac{\tau^3}{120}s^3}{1 + \frac{\tau}{2}s + \frac{\tau^2}{10}s^2 + \frac{\tau^3}{120}s^3} \right) \frac{K}{s^2 + Bs + K} \qquad \textbf{Equation 7}$$

In this equation, τ is the time delay. This perturbation has a negligible effect on the design of the controller, so to start; we will focus on the system without this time delay, and show the effects of the delay later on.

Stabilization of the pendulum angle

The inverted pendulum characteristic equation has one pole in the right half plane, making the system inherently unstable. Stabilizing the system was achieved using a PD controller designed using the root locus method. **Figure 9** and **Figure 10** show the process of controller design using root locus.

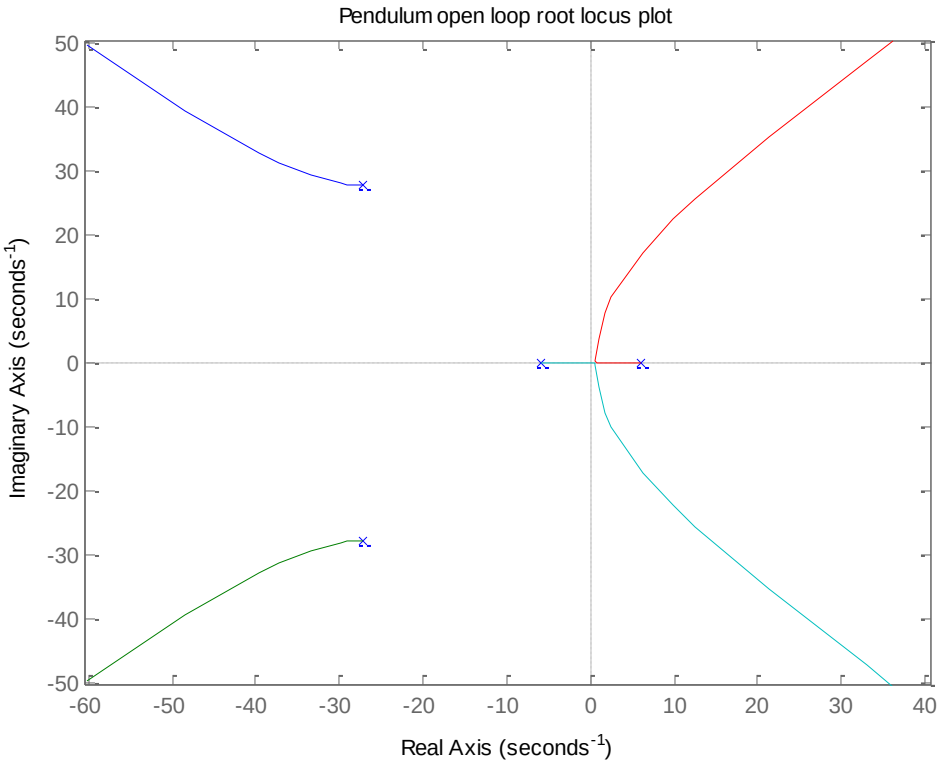

Figure 9. Open loop root locus of the pendulum system

Figure 9 shows that there is no stable region with only proportional control. However adding a zero into the controller transforms the system shown in **Figure 10.**

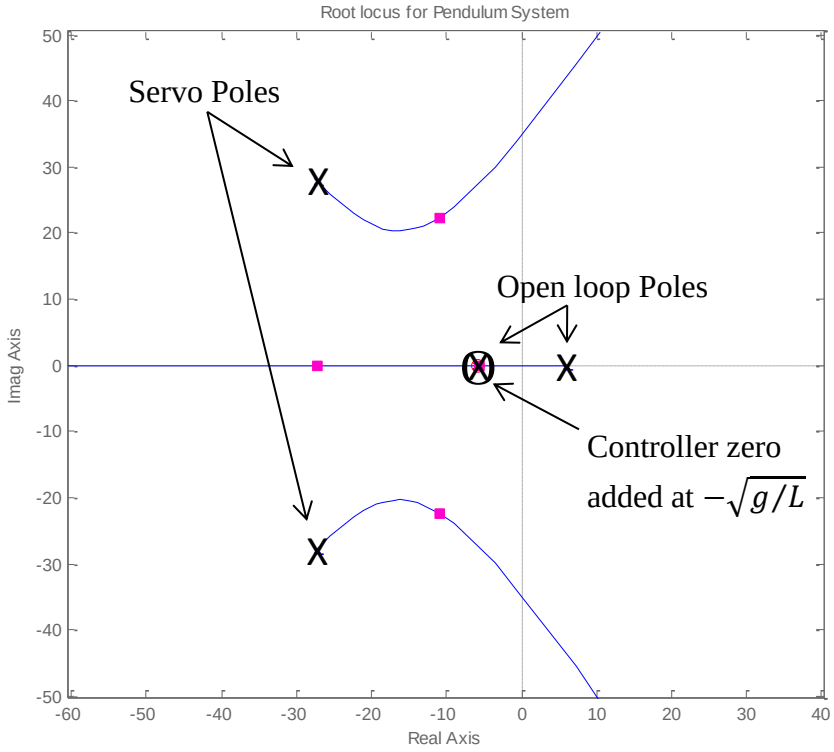

Figure 10. Closed loop poles of Pendulum (zoomed in)

In order to stabilize this system, a zero was placed at $s = -\sqrt{g/L}$, effectively canceling out the left pole on the real axis. The loop gain can then be adjusted to move all four closed loop poles to the left half plane. Given the stability region, the loop gain can be adjusted to obtain the desired response. The resultant controller used to achieve this stability is given by **Equation 1**. A block diagram is included in **Figure 11** to show the overall system stabilized by this controller. This controller will successfully stabilize the inverted pendulum but has some problems with the final value of the gimbal angle, ϕ. Although the rate is zero when the pendulum is balanced, the position is not controlled. This will be solved later on in the paper.

$$D(s) = K_{loop}(s + \sqrt{g/L}) \qquad \textbf{Equation 8}$$

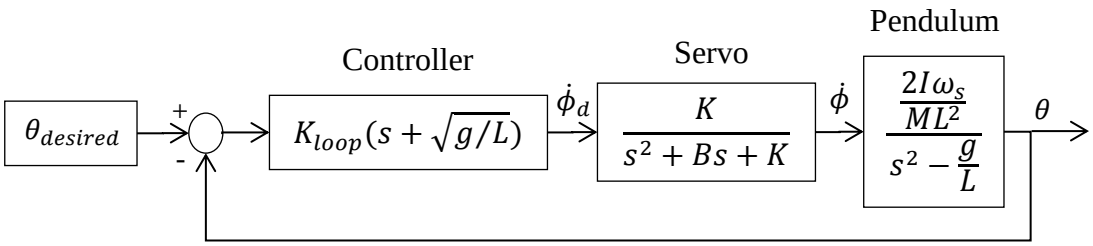

Figure 11. Closed loop system representation of the pendulum stabilization controller

The controller D(s) on its own is non-causal, therefore, its implementation would be impossible. In most control applications, a low-pass filter is added to this controller to retain causality, however the specific physical setup of this system allows for an alternative solution. Because the RC servos already have a closed-loop position controller, the desired rate, $\dot{\phi}_{desired}$ is integrated to obtain the servo position command, $\phi_{desired}$. The servos require an angle command, and the pendulum requires a velocity command. Therefore, the integration term is computed within controller, and is effectively part of the controller. This makes the once derived PD controller, effectively a PI controller. The final controller used to stabilize θ is given in **Equation 9**.

$$D(s) = K_{loop} + \frac{K_{loop}\sqrt{g/L}}{s} \qquad \textbf{Equation 9}$$

With a pendulum controller designed, it is now appropriate to show the effects of the time delay on the system. **Figure 11** shows the addition of zeros and poles due to the Pade approximation.

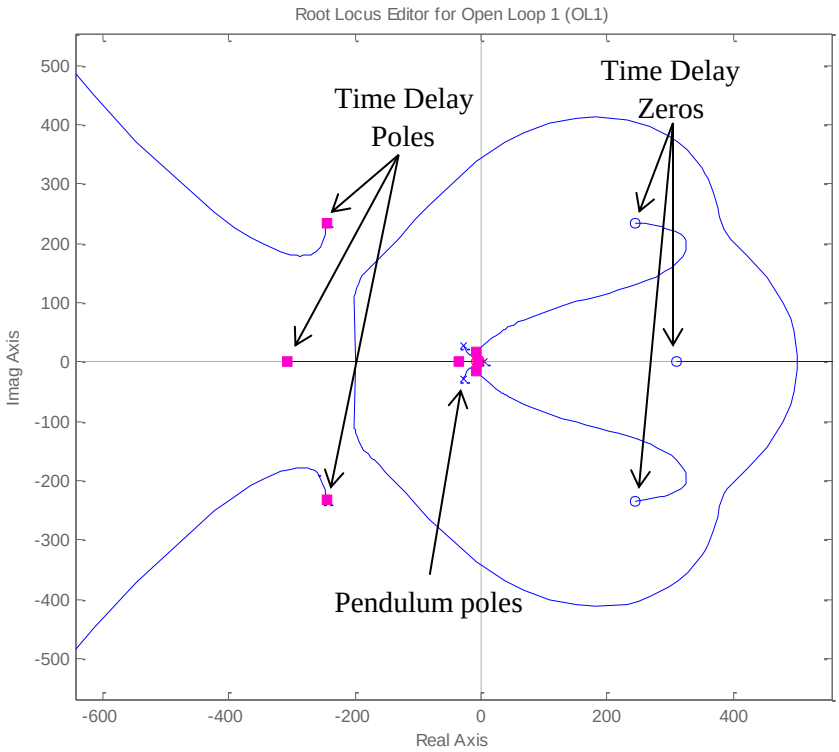

Figure 12. The effect of the Pade approximations on the open loop pendulum system

Figure 13 and **Figure 14** show the open loop and closed loop root locus for the pendulum system with and without the incorporation of the time delay.

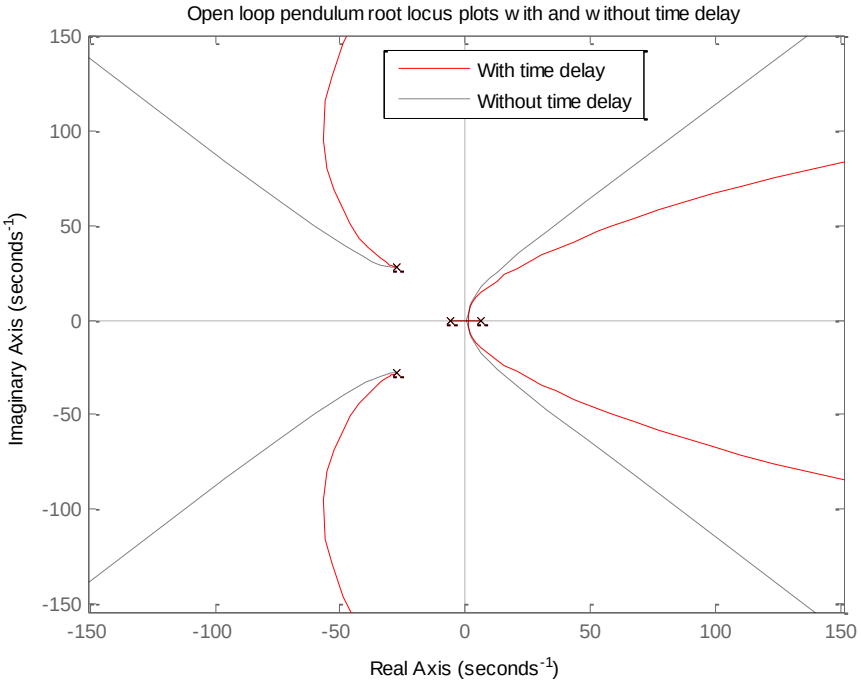

Figure 13. An overlay of the open loop pendulum root locus with the pade approximation, and without the Pade approximation

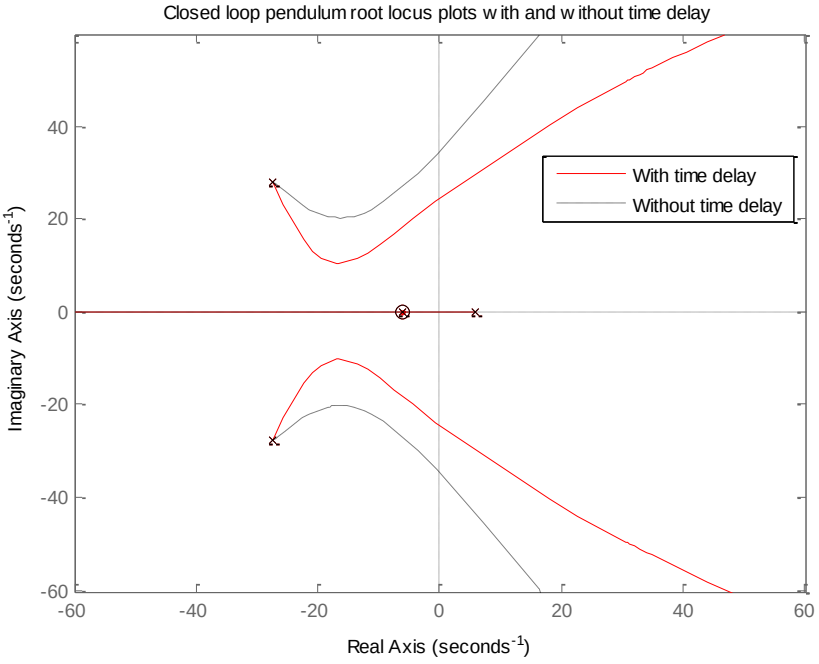

Figure 14. An overlay of the open loop pendulum root locus with the pade approximation, and without the Pade approximation

The time delay causes the pair of complex pole loci to be closer to the imaginary axis. This changes the oscillation component in the final solution. It turns out that the difference between simulations of the system with and without this time delay perturbation is rather large. The time delay in the dynamical system significantly increases the response time and frequency. **Figure 15** shows the simulated response for the system with and without the time delay portion.

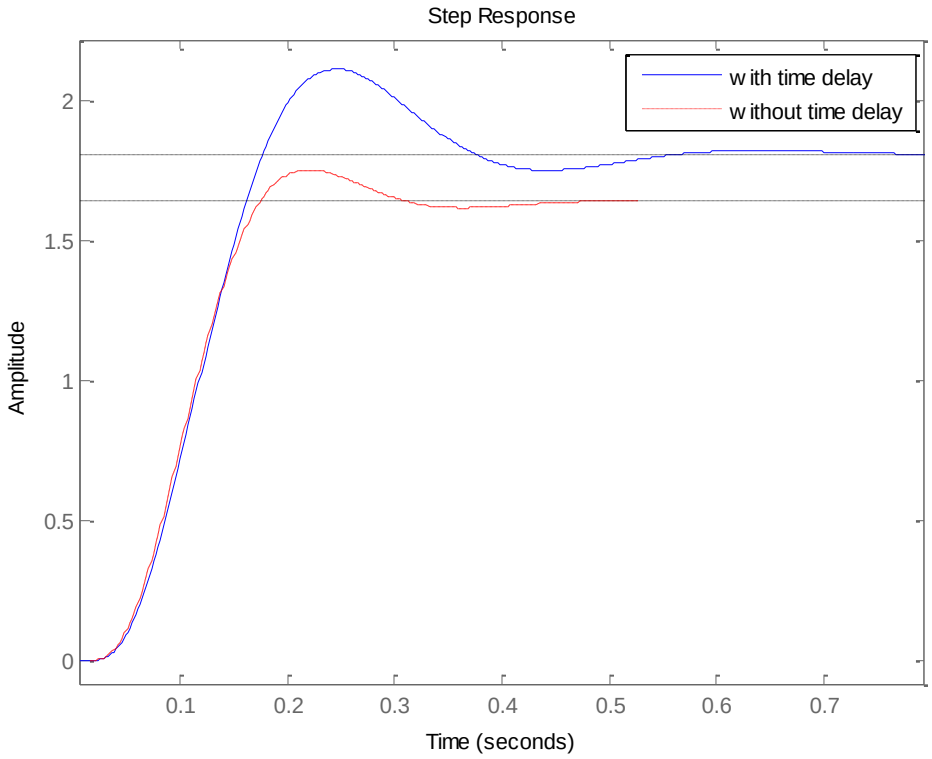

Figure 15. An overlay of the simulated step response for the controlled pendulum system with and without the time delay

These simulations are based on values shown later in the paper. Because of this difference, our data will be compared to the system response with the time delay.

Nonlinear controller

Equation 2 is linearized by the conditions $\theta \cong 0$ and $\phi \cong 0$, and the pendulum controller given in **Equation 9** was designed for these conditions However, it will be shown that the gyro angle, ϕ, is not always near zero. The controller designed for the linear system does a satisfactory job of balancing the pendulum, but the influence of some external force can easily bring the gyroscopes out of the linear range. It is also subject to steady state offsets based on factors explained later in this paper. At this point, the controller is less effective because the controller expects the gyroscopes to apply a torque of $2\mathrm{I}\dot{\phi}\omega_s$ when it is actually applying a torque of$2\mathrm{I}\dot{\phi}\omega_s \cos\phi$. An easy way to compensate for this is to divide the controllers loop gain by $\cos\phi$ making the overall controller given below.

$$D(s) = \frac{K_{loop}}{\cos\phi} + \frac{K_{loop}\sqrt{g/L}}{s\cos\phi} \qquad \textbf{Equation 10}$$

This controller can easily implemented since ϕ is needed for stabilization anyway. With this control equation in place, a singularity is formed at $\phi = \pm \pi/2$, where an infinite gyroscope velocity is needed, so it is important to limit the physical range of the gyroscopes far enough away from these points. This controller was implemented and tested, but is outside the scope of this research, and will not be further discussed in this paper since all data and stabilization techniques were developed and implemented in the linear region.

Stabilization of the gyroscope angle

The above control system is an effective strategy for the immediate stabilization of the pendulum balancing angle; however, it is not a sufficient controller for the overall system. The

current system lacks a way to control the gyroscope angle. As explained above, twisting the gyroscope spin axis at some velocity results in a torque described in **Equation** 2. The current controller essentially commands this velocity to achieve the appropriate torque needed to balance the pendulum. Therefore, the application of any balancing torque causes an inevitable displacement on the gyroscope axis angle. This displacement is a problem because the dynamic equations are derived on the assumption that ϕ is small, and larger values of ϕ will result in inaccurate control and a smaller region of controllable balancing space.

It is apparent that the stabilization of ϕ is also important in this system. If ϕ is made controllable, the system will be able to minimize the gyroscope angle along with the stabilization of the pendulum. With the addition of such a controller, the balancing robot would be more robust, and resilient to external influences on the robot.

The stabilization of ϕ can be achieved by measuring the gyroscope angles, and adjusting the desired balancing angle of the pendulum. When the balancing angle is nonzero, the gyroscope axes spin to counteract the torque caused by gravity. Therefore, controlling the desired angle allows for the control of the gyroscope angles.

To design a gyroscope controller, we first reformulate the control system from **Figure 11** to have the desired balancing angle be its input, and the gyroscope angle, its output. The block diagram in **Figure 16** shows this formulation.

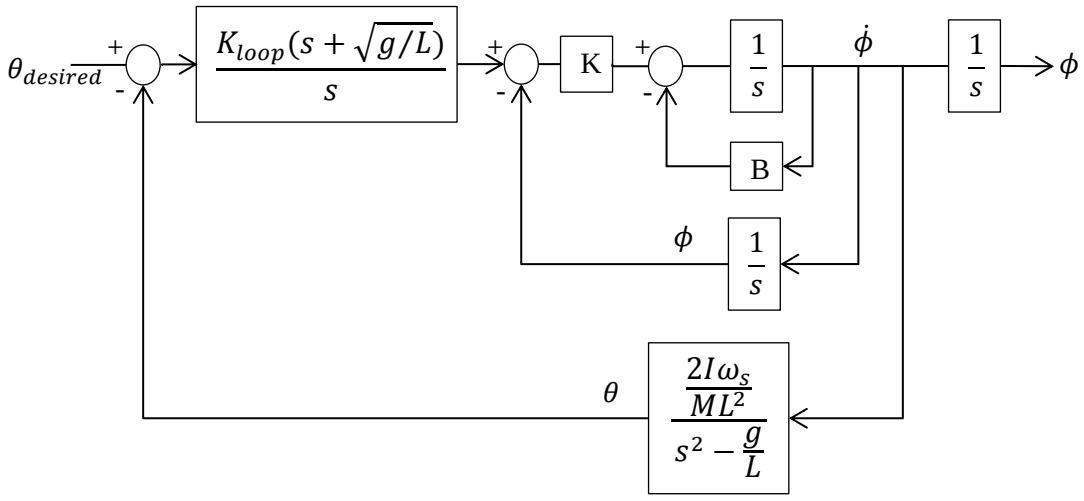

Figure 16. A reformulation of the stabilized inverted pendulum in terms of gyroscope angle.

Using this block diagram, it can be shown that the transfer function for the entire system is given by **Equation 11.**

$$\frac{\emptyset}{\theta_{desired}} = \frac{K_{loop}K\left(s^3+\sqrt{\frac{g}{L}}s^2-\frac{g}{L}s-\left(\frac{g}{L}\right)^{3/2}\right)}{s\left(s^4+Bs^3+\left(K-\frac{g}{L}\right)s^2+\left(K_{loop}K\frac{2I\omega_s}{ML^2}-\frac{Bg}{L}\right)s+K\left(K_{loop}\sqrt{\frac{g}{L}}\frac{2I\omega_s}{ML^2}-\frac{g}{L}\right)\right)} \quad \textbf{Equation 11}$$

By simple inspection, it is apparent that the best that this system can be is marginally stable because of the pole at zero. The Root locus in **Figure 17** shows that there is no stable region for a positive gain value.

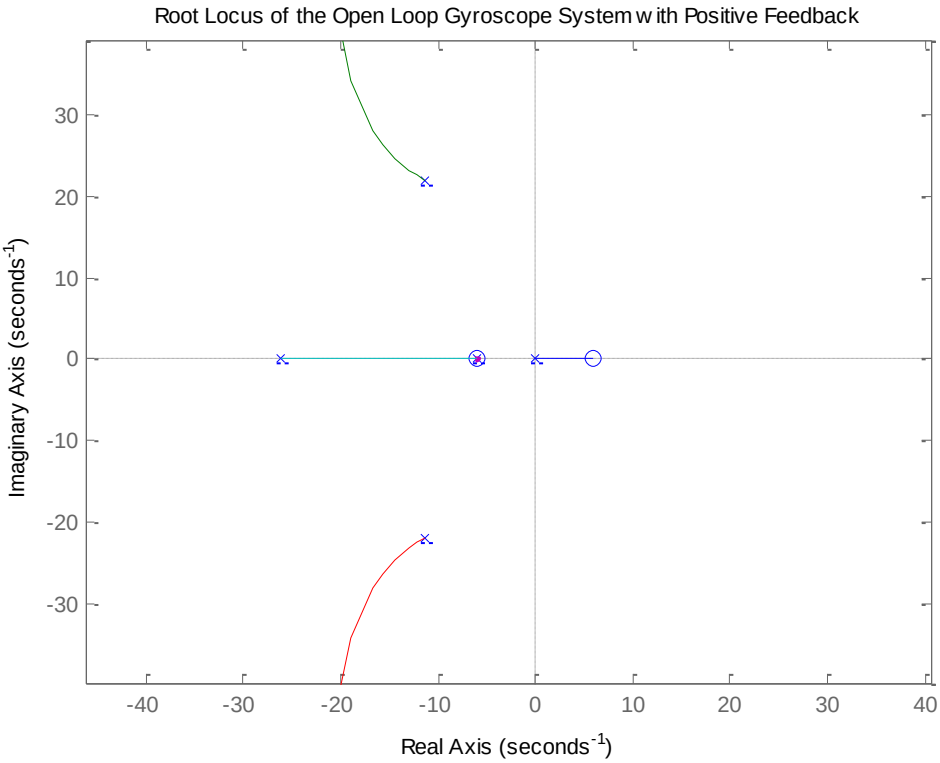

Figure 17. Root locus of the open loop gyroscope system with positive gain

Furthermore, there is very little that exists at all to stabilize this system with a positive feedback using classical control techniques. However, it turns out that the system can easily be stabilized using negative feedback. With a negative loop gain, two different controllers are designed that are very effective in stabilizing ϕ, based on the desired characteristics of the response. The two controllers designed include a proportional controller, and a filtered proportional controller. The root locus plots in **Figure 18** and **Figure 19** show the stability region for a negative proportional controller, and a negative filtered controller.

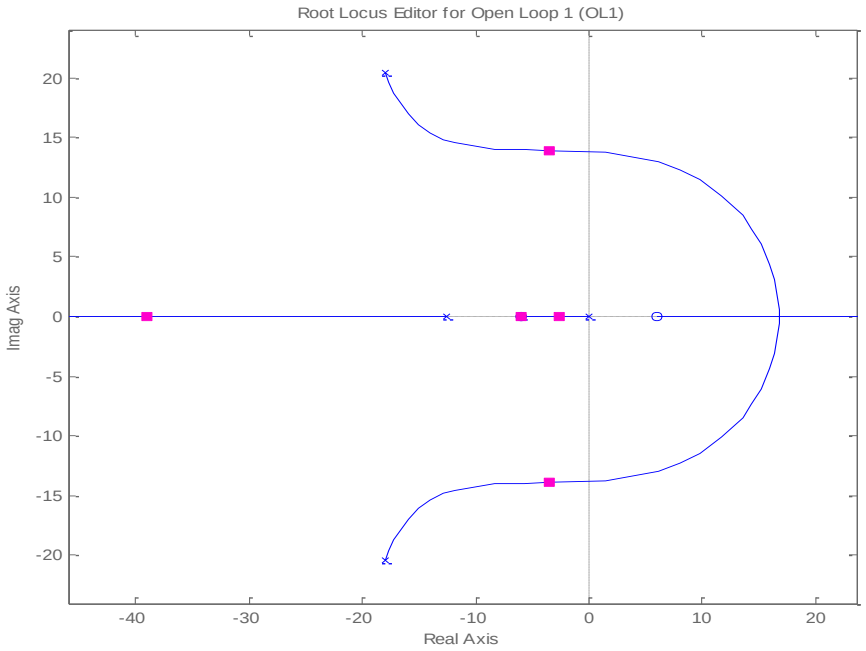

Figure 18. Root locus plot of the open loop gyroscope system with negative feedback.

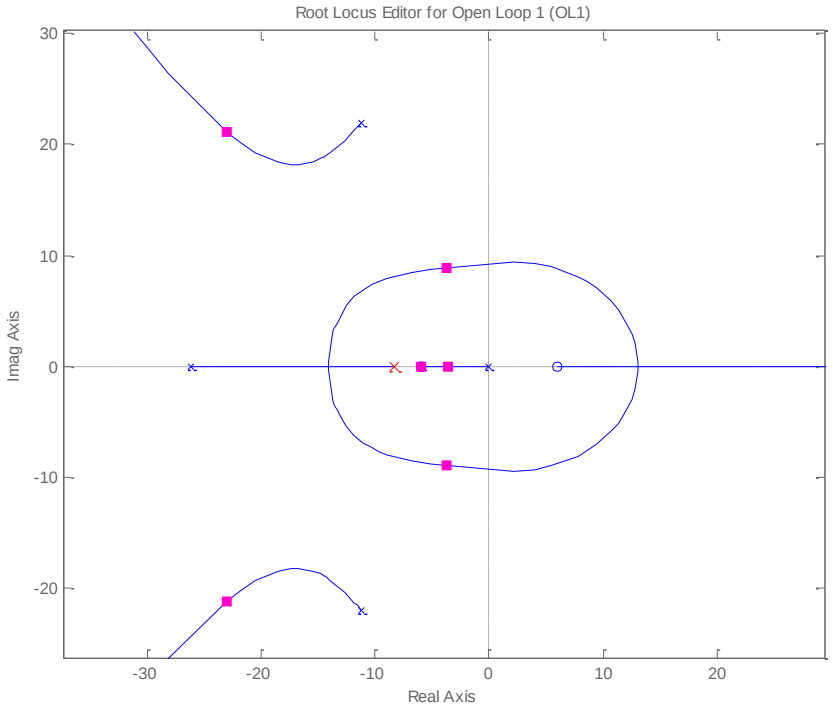

Figure 19. Root locus of the open loop gyroscope system with a filter pole added and negative feedback.

Figure 18 and **Figure 19** above show the stability region for each controller design option. These two options lead to a much different dynamic response that will be discussed in the next section. **Figure 20** shows a block diagram showing the closed loop system with negative proportional and negative filtered control systems.

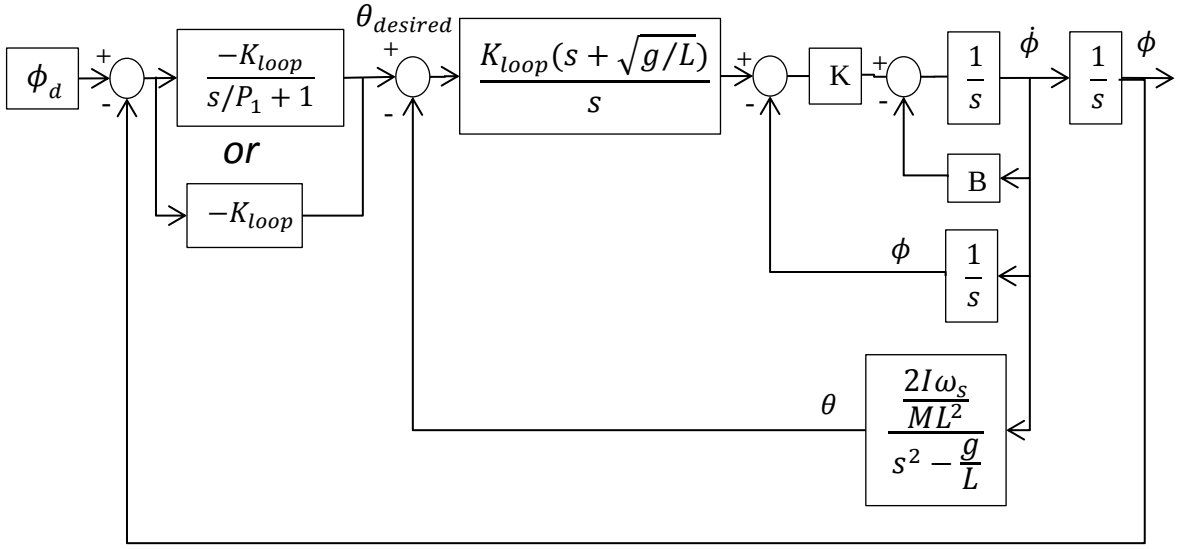

Figure 20. overall controller to stabilize both balancing angle and gyroscope angle.

Now with the control theory in place it is now important to show the perturbations on these dynamics due to the servo time delay. **Figure 15** shows that as we consider the effect of the time delay on the pendulum controller, the systems response time increases. Since the gyroscope controller is based on the responses of the controlled pendulum system, we can expect the gyroscope controller response time to change as well. **Figure 21** and **Figure 22** show the root locus plots of the controllers with and without time delay to show the difference in pole locations.

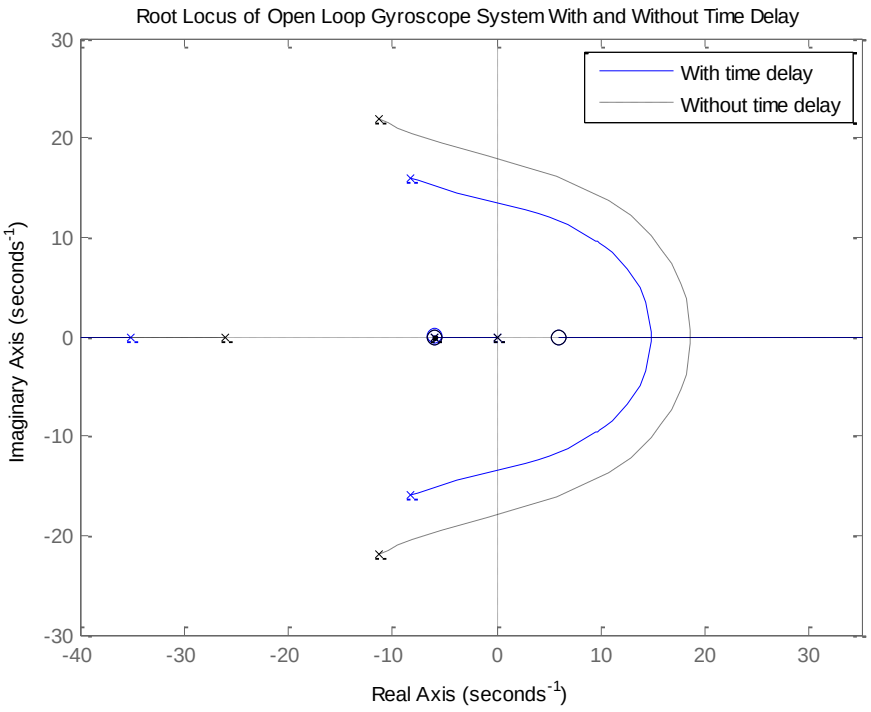

Figure 21. An overlay of root locus plots for the open loop gyroscope system with and without time delay

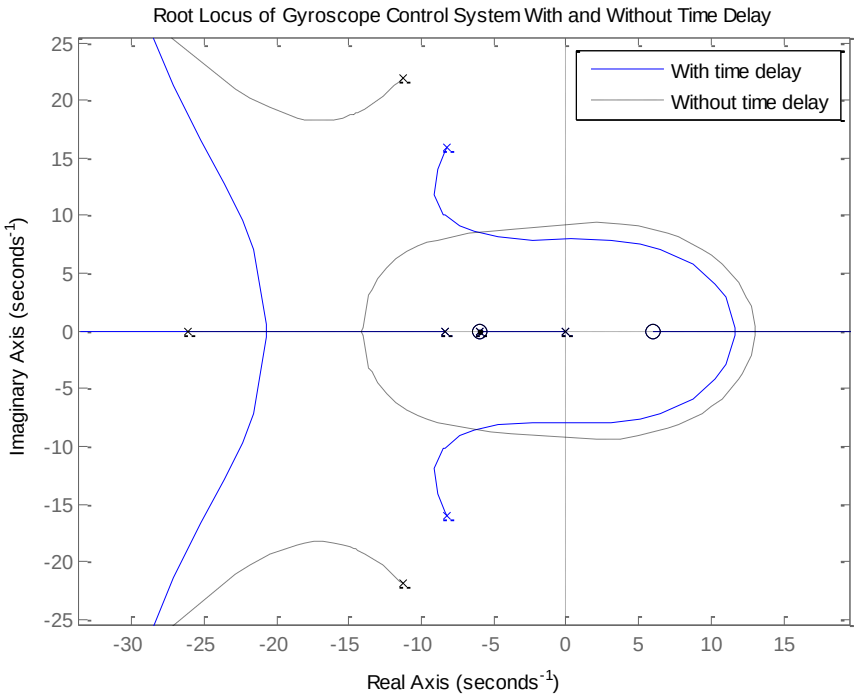

Figure 22. An overlay of root locus plots of the open loop gyroscope system with and without time delay. both plots have a controller pole placed at s = 8.3 and are using negative feedback.

Figure 23 shows a plot of the system response with and without the effect of the time delay.

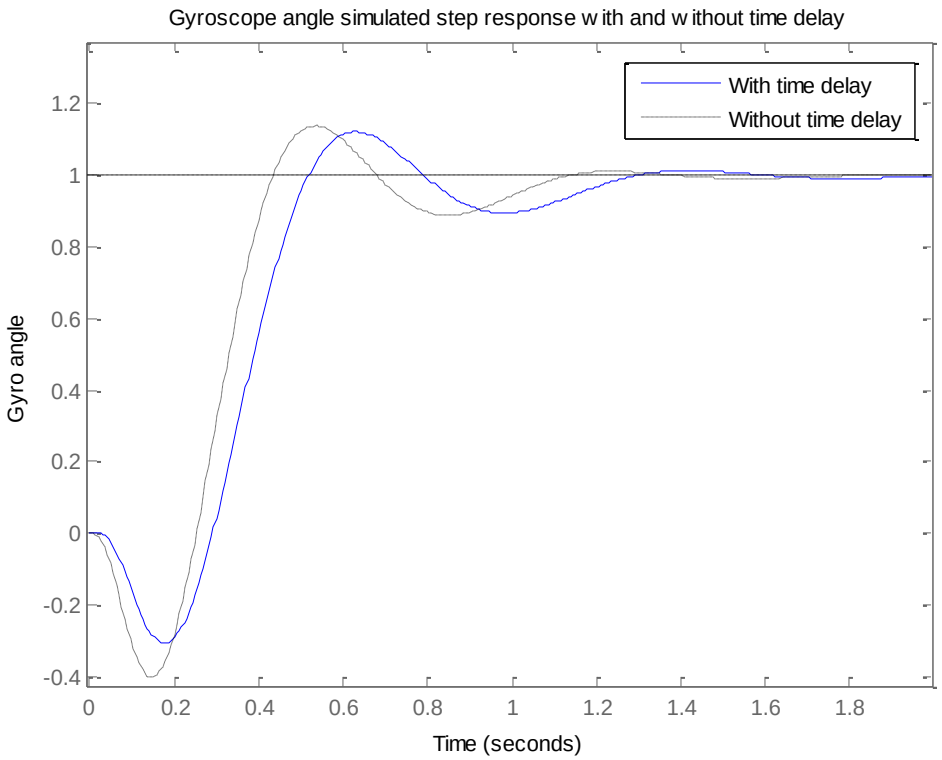

Figure 23. An overlay of the closed loop gyroscope system with and without time delay. both plots are simulations of the gyroscope system controlled with a low pass filter and negative feedback

Because of this difference, our data will be compared to the system response with the time delay.

CHAPTER 3. SIMULATION RESULTS

To implement the proposed stabilization system, a prototype was built to the following specifications.

Overall subsystem	component	value
Gyroscopes	Depth	8.89×10^{-3} m.
	Radius	0.0241 m.
	Mass	0.041 Kg
	Estimated moment of inertia	1.15×10^{-7} Kg*m^2
	Estimated angular velocity	15,000 RPM
Powered gimbals	Servo top speed	3.49 rad/sec
	Angle range	$-\pi/2 - \pi/2$
	Torque	.015 Kg-m
	Angle measurement	potentiometer
	Time delay	0.015 sec
Robot chassis	Mass	0.3 Kg
	Pole length	0.3 m
	Balancing measurement	10kΩ potentiometer or IMU filter
Servo characteristics	K value	1521
	B value	54.6

Table 1. Part specs used in a dynamic model of the systems explained above.

These values were used to model the dynamics of the physical system and the controllers. To begin, the controller to stabilize the pendulum was designed using the root locus method explained in the technical development section. **Equation 12** shows the pendulum controller based on the physical parameters given in **Table 1**.

$$D(s) = 8.7(1 + \frac{5.96}{s}) \qquad \textbf{Equation 12}$$

Figure 24 shows the system step response of this controller.

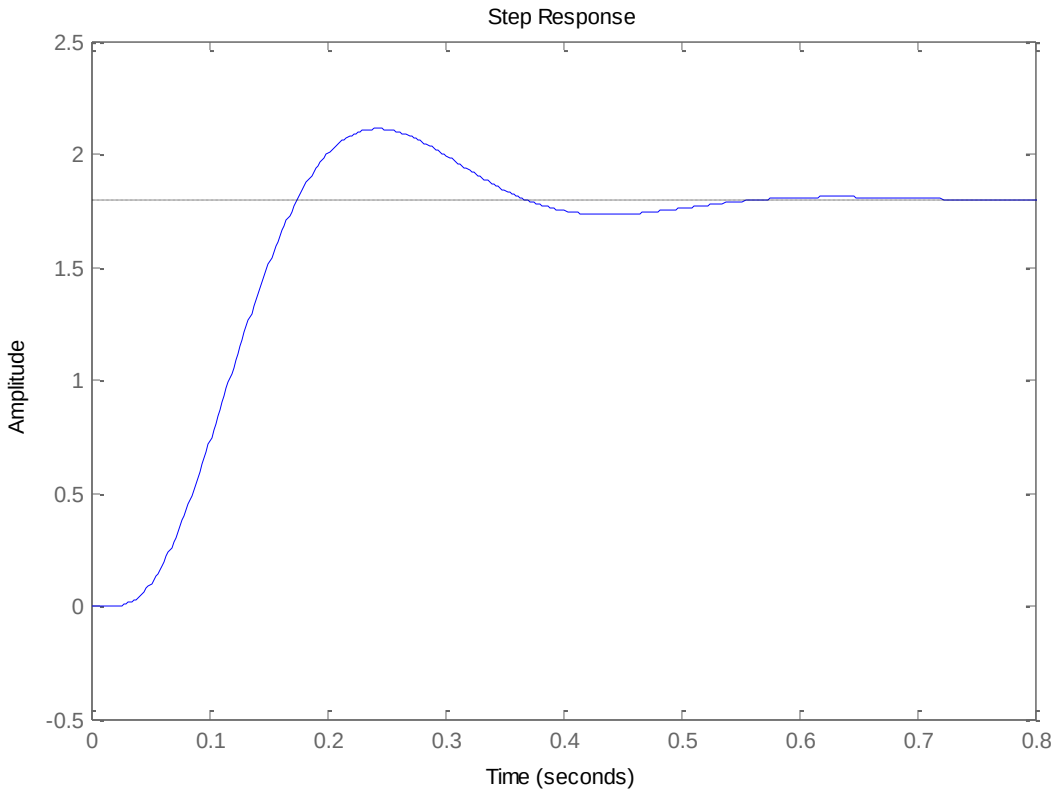

Figure 24. Step response of the stabilized inverted pendulum.

Now, with a closed loop model of the stabilized pendulum, it is time to focus on controlling the gyroscope angle. With the pendulum model in place, we move on to evaluate the gyroscope control systems.

The first controller explored was the proportional controller. **Figure 18** shows that the system has a stability region with negative feedback. Adjusting the loop gain of the negative proportional controller changes the response time of the system, but also changes the amount

of oscillation in the closed loop response. The **Figure 25** shows some of the possible loop gain settings. If the loop gain is larger, the system will react faster, but there will be a large component high frequency oscillation in the system. This happens when the pair of poles on the complex loci gets close to the real axis. A smaller loop gain gets rid of this oscillation for the most part, but significantly increases the response time of the system. If this control method is used, it is up to the operator to decide what gain fits their own physical requirements.

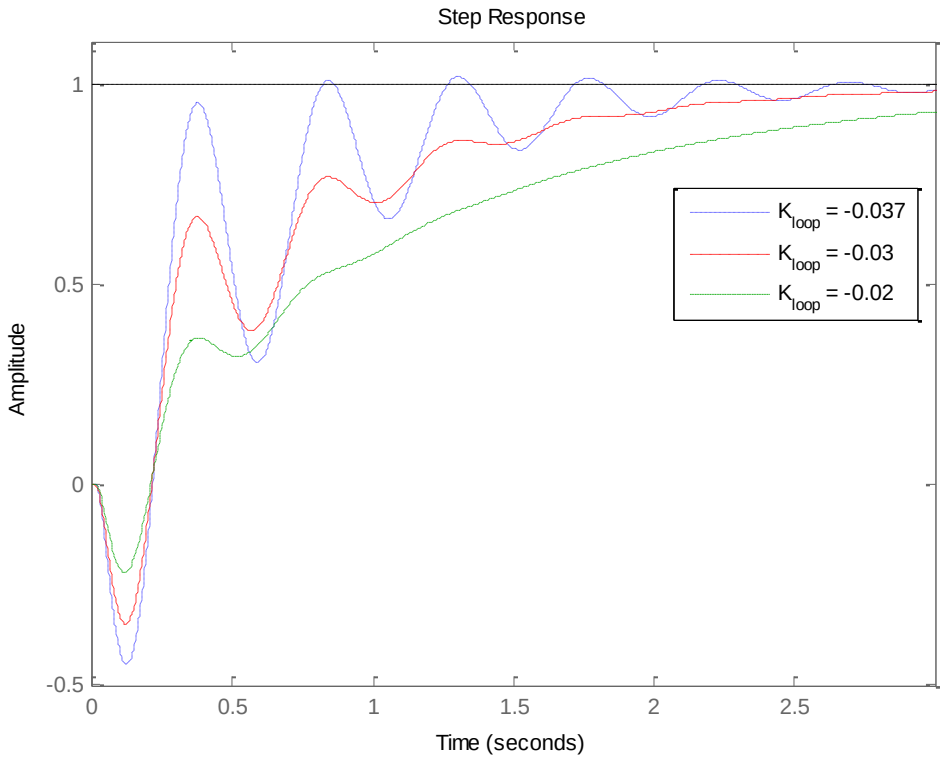

Figure 25. Step responses of various loop gains of the closed loop system.

In this case, the responses shown by the proportional control seem to have either an undesirable amount of oscillation or a slow response. However, it is up to the user to decide whether this dynamic response acceptable for their own application. If a smoother operation is desired, a low pass filter may be added to smoothen out the response motion. The addition of

a filter to the proportional controller, shown in **Figure 19**, changes the response to a more desirable motion. Using the root locus in **Figure 19**, a pole was added at s = -8.33. **Equation 13** shows the resulting controller from this design.

$$D(s) = \frac{-K_{loop}}{(\frac{s}{P_1} + 1)} = \frac{-0.052}{(\frac{s}{8.33} + 1)} \qquad \textbf{Equation 13}$$

Where P_1 is the pole added as the first order filter. The controller pole location can vary somewhat to yield a range of different response options. We chose a pole at s = -8.33 to be an optimal location because it mixes a fast response time with a good amount of overshoot and damping possibilities. A controller pole at zero, and a zero can also be added to the controller on the imaginary axis, making the controller resemble a filtered PI controller. This essentially adds an integral gain component to the controller, and can be shown to have a stable region using the root locus method if the additional controller zero is very close to the origin in the left half plane. The closer this added zero is to the origin, the smaller the integral component. The advantage of this integral component will be discussed later in the paper, but was not implemented in this project. **Figure 26** shows the step response of the closed loop system is shown below.

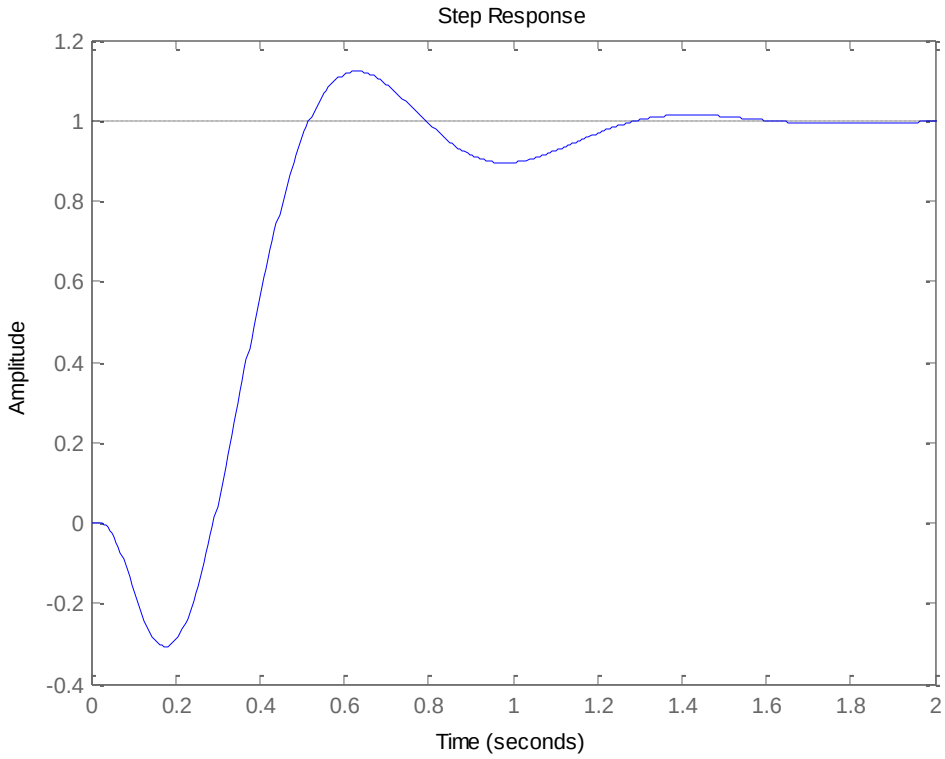

Figure 26. predicted step response of the gyroscope angle control loop

CHAPTER 4. EXPERIMENTAL SETUP AND RESULTS

The prototype is constructed out of machined aluminum. Each gyroscope is mounted on a small dc motor, which is mounted on a swivel platform powered by a 9 gram hobby style RC servo. The servo is able to spin the gyro axis at an angular rate of 200 degrees/second. Each gyroscope is independently controlled by its own servo, and it is assumed that both gyros are at the same angle at any given time because the servos themselves include a closed loop feedback system to command an angle. The gyro angle is measured using a potentiometer, and for initial tests, the balancing angle is also measured using a potentiometer. The whole robot is controlled using an Arduino microprocessor. They are a good prototyping tool because they are inexpensive, light, and can handle a large range IO and communication protocols. A further benefit of using an Arduino is that it is an open source product, and a large open source community of Arduino users has become a strong resource for this hardware. The servos are controlled by a Pololu 'Mini Maestro' servo controller. This is a serial controller that sends pulse width based digital commands to the servos at a 0.025 degree command resolution. The system sample time is about 4 milliseconds.

The controllers were first developed in labview, using the 'Labview Interface for Arduino' toolbox and firmware. This setup programs the arduino to function as a data acquisition board, sending values over a serial line to the labview application. This allowed for quick prototyping of controller designs and convenient data acquisition. In later work, an IMU is used with a complimentary filter to estimate the orientation, and the whole controller was

then programed onto the Arduino for a faster sample time and to allow standalone performance. **Figure 27** shows a picture of the prototype.

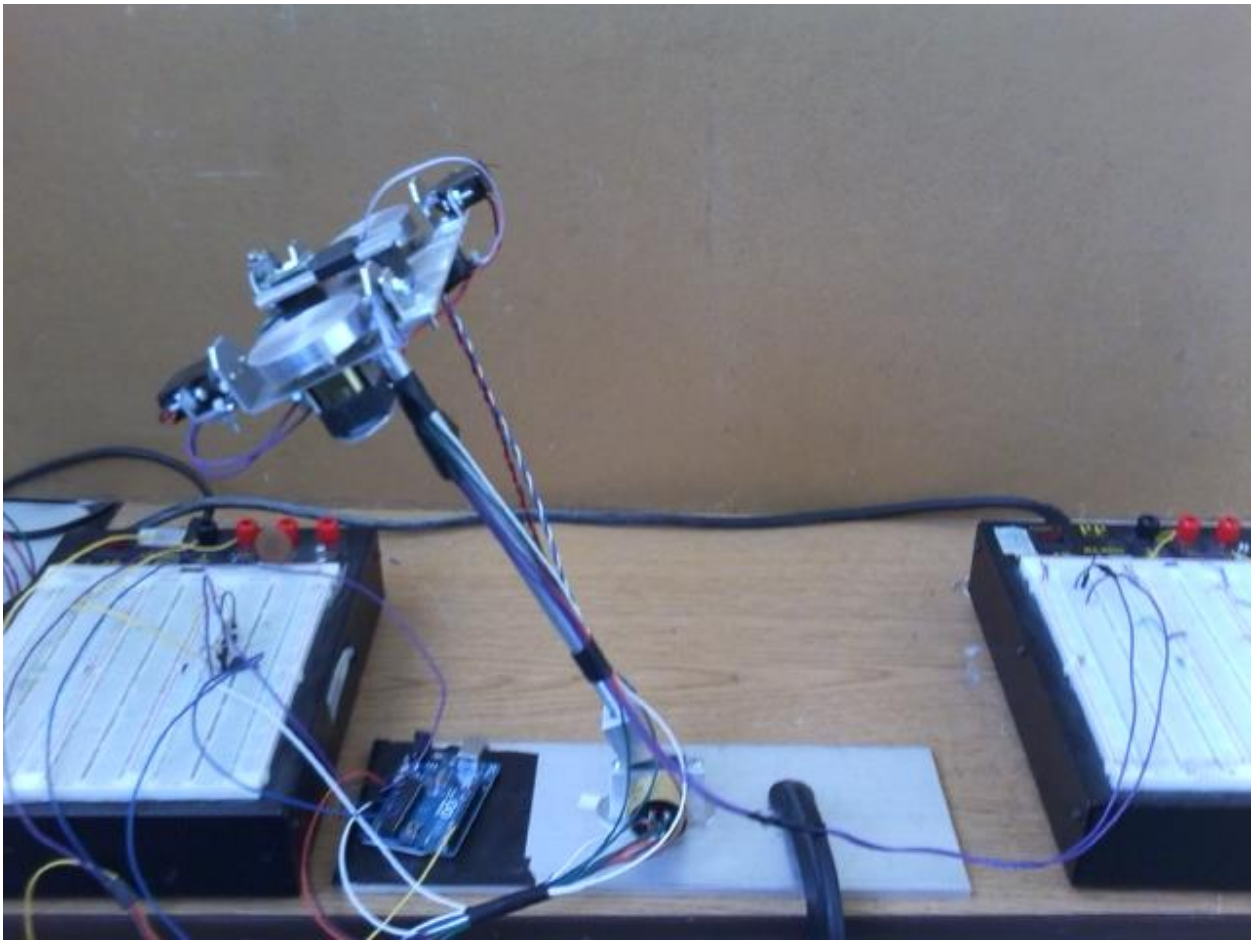

Figure 27. prototype of the Inverted pendulum

Servo calibration

The proposed servo dynamic equation shown above is commonly used to model damped harmonic motion. However, damped harmonic systems are more conveniently expressed in the following form.

$$\frac{\dot{\phi}_{desired}}{\dot{\phi}_{actual}}(s) = \frac{\omega_n^2}{s^2 + 2\omega_n \sigma s + \omega_n^2}$$ **Equation 14**

Where ω_n is the systems natural frequency, and σ is the damping factor. Keep in mind that our particular servos have a small time delay that we model using a Pade approximation. We compared the measured data from the servo step response with the above dynamic model, and found the physical system to have a natural frequency of 39 rad/sec and a damping factor of 0.7. Using the model given above, we can conclude that K = 1521 and B = 54.6. The servo time delay is approximately 0.015 sec. **Figure 28** and **Figure 29** show the servo step response along with the simulated system response based on the above values.

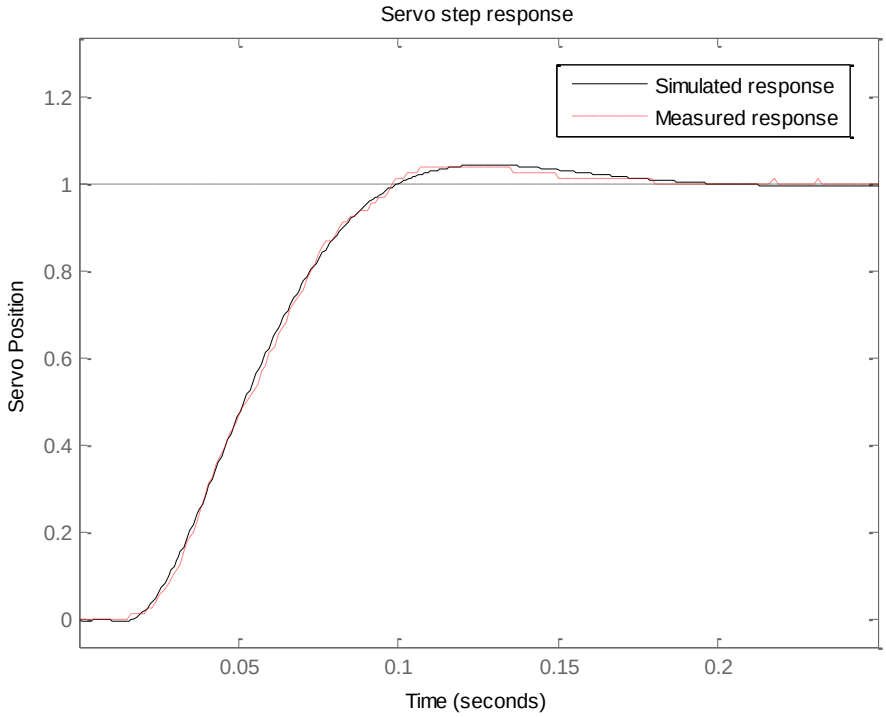

Figure 28. An overlay of the measured servo step response, with a simulation of a model that approximates its motion.

Magnifying the first 30 milliseconds of the servos response shows the effect of the Pade approximation on this servo model, along with the time delay of the servo.

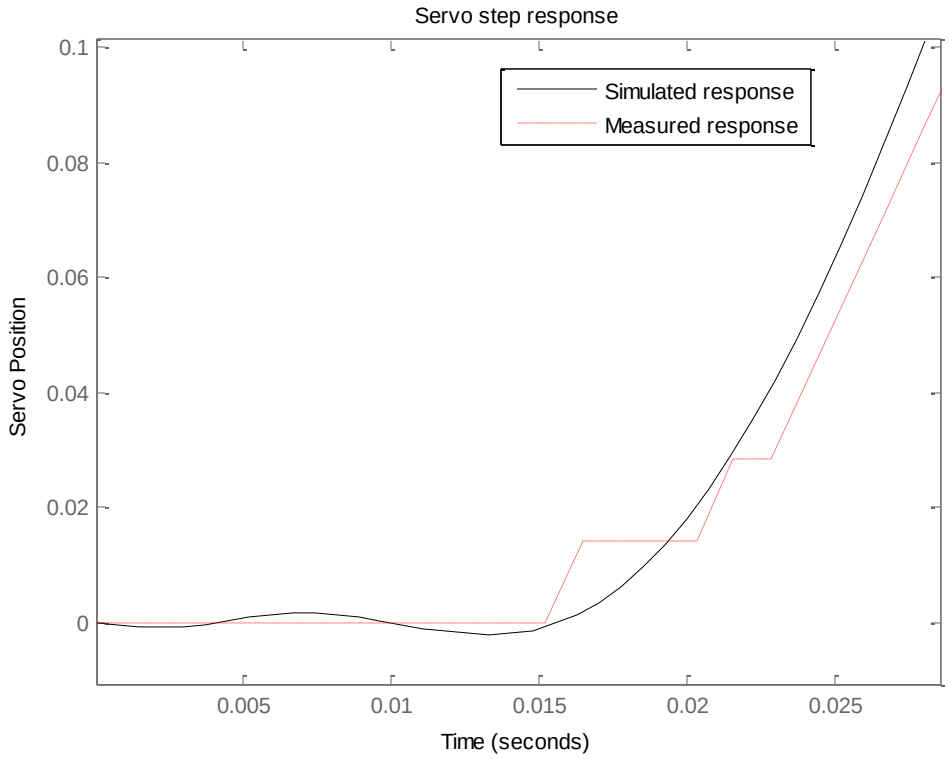

Figure 29. Time delay region of servo step response

Pendulum controller results

The specs of the robot were used to create a dynamic model of the system. The root locus method outlined above was used to design a controller for the robot. Measurements of the pendulums step response were taken by allowing the pendulum to lean approximately -2.4 degrees against a rigid body. Once the gyroscopes were up to speed, the controller was enabled, and results were recorded while the pendulum would stabilize about $\theta = 0$. The first implementation of the controller appeared to stabilize as expected, but the data suggests a small inaccuracy in the pendulum controller. **Figure 30** shows the step response of three common runs along with the predicted step response.

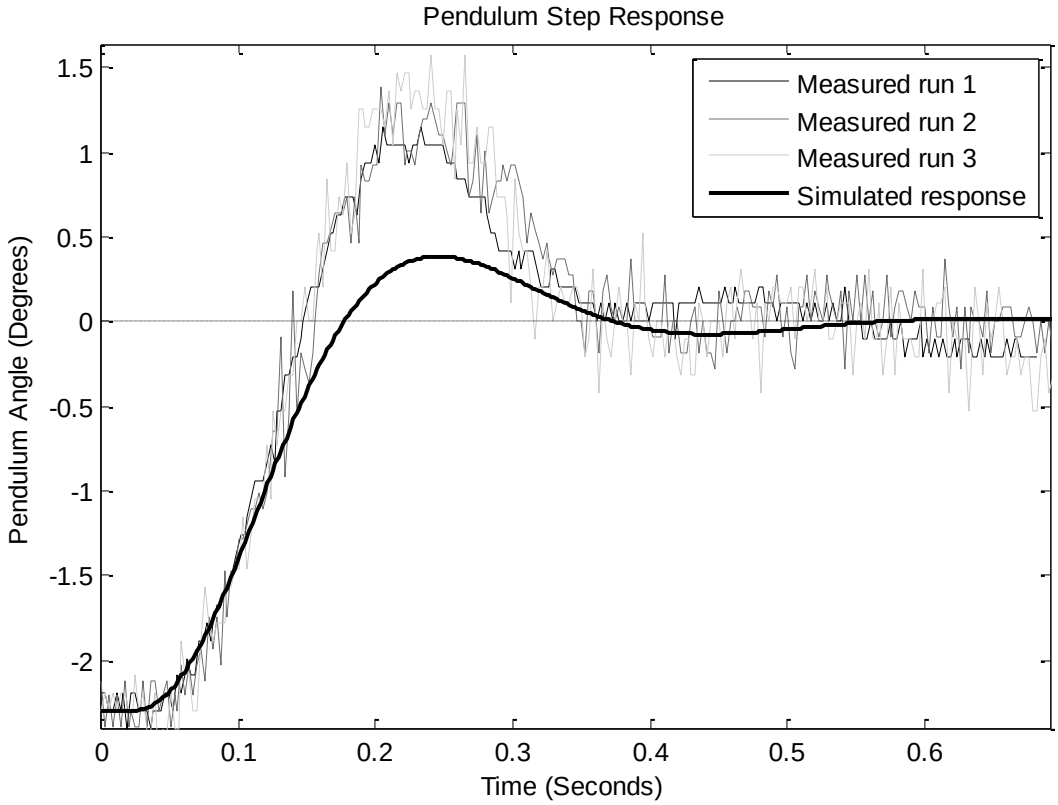

Figure 30. An overlay of the simulated pendulum step response vs. actual controllers step response

In these results, the most noticeable deviation is the pendulums overshoot. However, examining the root locus plot of the pendulum system exposed a possible reason for this deviation in motion. Recall that to stabilize the system a controller zero was placed at $s = -\sqrt{g/L}$, and the loop gain can be designed to obtain the desired response shown above, and the value of $s = -\sqrt{g/L}$ was chosen because it canceled out the pendulums pole. Adding the controller zero was chosen because it resulted in a desirable response motion, however if the zero was placed at $s > -\sqrt{g/L}$, the response approached the desired value with more of an asymptotic motion, and the wavelength of any oscillation would decrease as S approaches 0. Whereas, if $s < -\sqrt{g/L}$, the model would have a larger overshoot and increased response

wavelength. **Figure 31** shows the step response of the system with controller zeros placed at various locations of $s < -\sqrt{g/L}$.

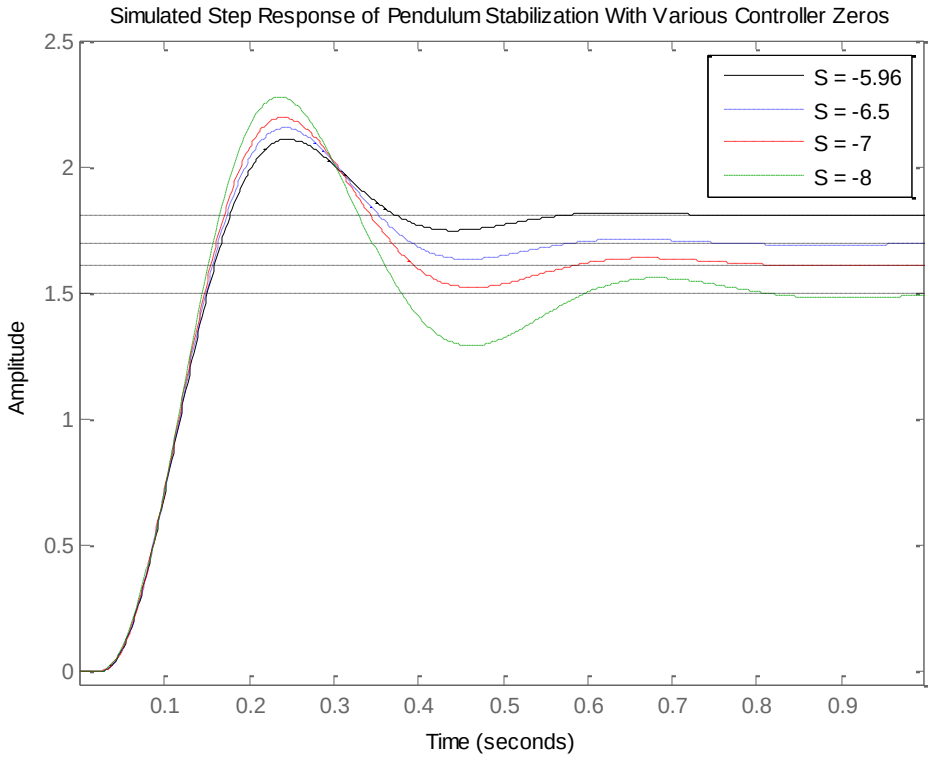

Figure 31. Variations of system step response using different pole values in the controller defined by Equation 9.

Given that the data has a larger overshoot and a slightly increased wavelength, we can conclude that the controller pole is slightly larger than the physical pole on the robot. This can be expected since the dynamics outlined in **Equation 5** assumes the robot to be a point mass. Therefore, it can be expected that the theoretical pole does not accurately represent the actual pole of the pendulum. Based on this information, one can conclude that decreasing the controller zero to accommodate the actual pole of the system will better result in the motion predicted by the models above. In order to pinpoint the pole, controller zero was adjusted until

the data better matched the simulated response. It was found that the pole is located somewhere around s = -5.3, where the optimal response was achieved. **Figure 32** shows three typical responses with the "corrected controller" against the modeled response.

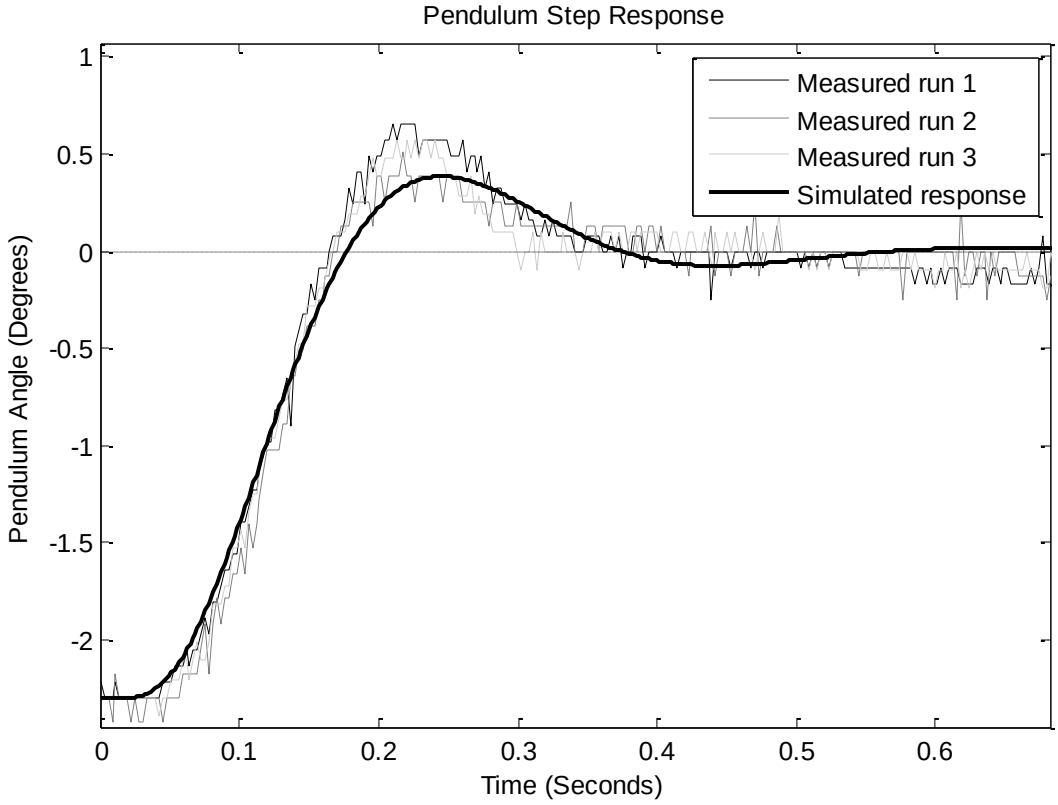

Figure 32. Tuned controller vs. predicted system step response

Gyroscope controller results

Out of the two gyroscope controllers described in the "Technical development" section, we chose to implement the filtered controller because it would provide a smoother response requiring less work on of the servos and less friction in the gyro motor axles. Measurements were taken by allowing the gyro controller and pendulum controller to stabilize about a commanded gyroscope angle of -5.5°, then commanding a +5.5° desired angle, and recording

the results. Because our balancing controller dynamics were not exact, the phi control had some discrepancies as well, but stabilizes nonetheless. **Figure 33** shows the simulated response along with three typical runs.

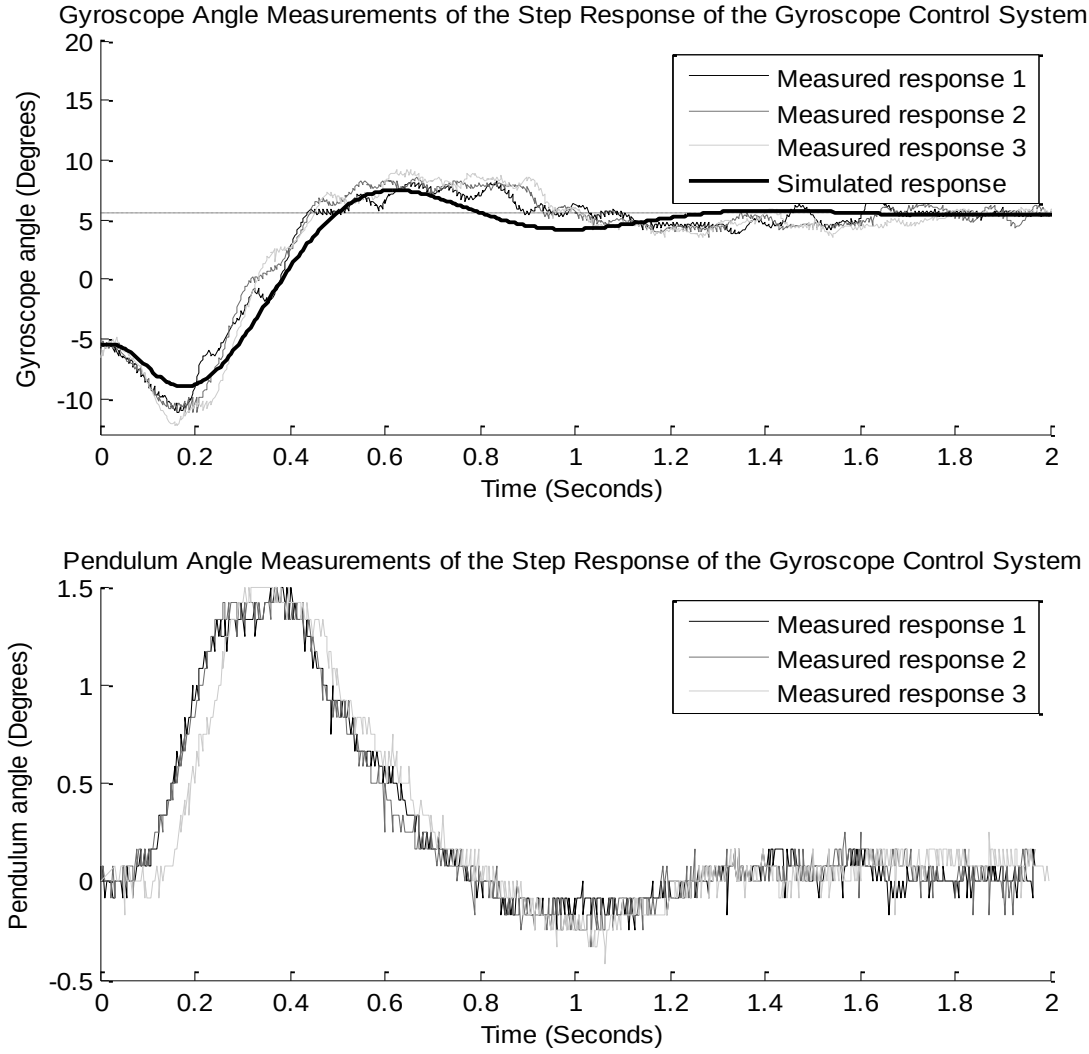

Figure 33. Final results of the Gyroscope controller step response. The top graph shows an overlay of the simulated response compared to three typical machine runs. The bottom graph shows the pendulum behavior as the gyroscopes control system responds to a step input command. This confirms the pendulum stability with the addition of the gyroscope controller.

Standalone controller

In order for this prototype to be useful in real world applications it must be developed into a standalone controller. The main obstacle in this process is measuring the pendulum angle. In a laboratory setting, the pendulum angle was a measurable quality of the pendulum because a potentiometer could be easily attached to measure between the pendulum and the base. However, this is only useful in a lab because the potentiometer only measures the pendulum angle with respect to the base. In a standalone application, the base will be removed and the pendulum angle is no longer a measureable parameter of the pendulum. Therefore, if we want to use the control design outlined above, we must find a way to estimate this quantity. Recall that the object of the inverted pendulum is to balance the pendulum at a vertical unstable equilibrium point. This orientation is reference to the earth's gravitational field, so we can estimate the pendulum orientation by measuring the earth's gravitational field. The earth's gravitational field can be measured using a three axis accelerometer, and these measurements can be used to estimate the normal force vector in the reference frame of the pendulum. It turns out that these measurements can be combined with the pendulums angular velocity, measured by a rate gyro sensor to create the orientation filters described below. The estimate will be used to replace the potentiometer and provide a pendulum angle estimate.

Many existing real world applications combine measurements of angular velocity of an inverted pendulum with gravitational field measurements to estimate orientation. Two common methods of this sensor fusion include Kalman filters, and complimentary filters. We chose to

use a complimentary filter because it is simple, easy to use, and will ensure a fast time step due to the small amount of computation needed to calculate an estimate.

Complimentary filters essentially combine data from an accelerometer and gyroscope to estimate orientation more accurately than just the accelerometer or gyroscope alone. An accelerometer can be used to calculate an absolute measurement of attitude, but has a lot of noise in its data. Gyroscopes on the other hand, output a velocity, which must be integrated to calculate position. This results in less noise, but introduces drift upon numerical integration. Therefore, a simple low pass filter is applied to the accelerometer estimate, and is combined with the integrated gyro estimate. Therefore, the gyro provides a quick, real time estimate of the attitude, while the accelerometers absolute measurement slowly corrects for the gyroscopes drift. **Equation 15** and **Equation 16** show this system.

$$pitch_{accelerometer\ estimate} = atan2(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}) \quad \textbf{Equation 15}$$

$$pitch(t) = \delta(pitch(t-\tau) + \omega\tau) + (1-\delta)(pitch_{accelerometer\ estimate}) \quad \textbf{Equation 16}$$

Where a_x a_y a_z are the 3 accelerometer measurements, t is the unit of time and τ is the sample time of the system. δ is the filter coefficient, and is set to 0.998 in this system. ω is the angular velocity gyroscope measurement in the y-axis. A 2-axis version of this complimentary filter also exists, and code for a few different types of complimentary and kalman orientation filters can be found on many open source websites. The system defined by **Equation 15** and **Equation 16** refer to [21]. To find the accuracy of the orientation filter, angle measurements from

both orientation filter and potentiometer were taken simultaneously, over a range of different angles. The results are shown in **Figure 34**, **Figure 35**, and **Figure 36**.

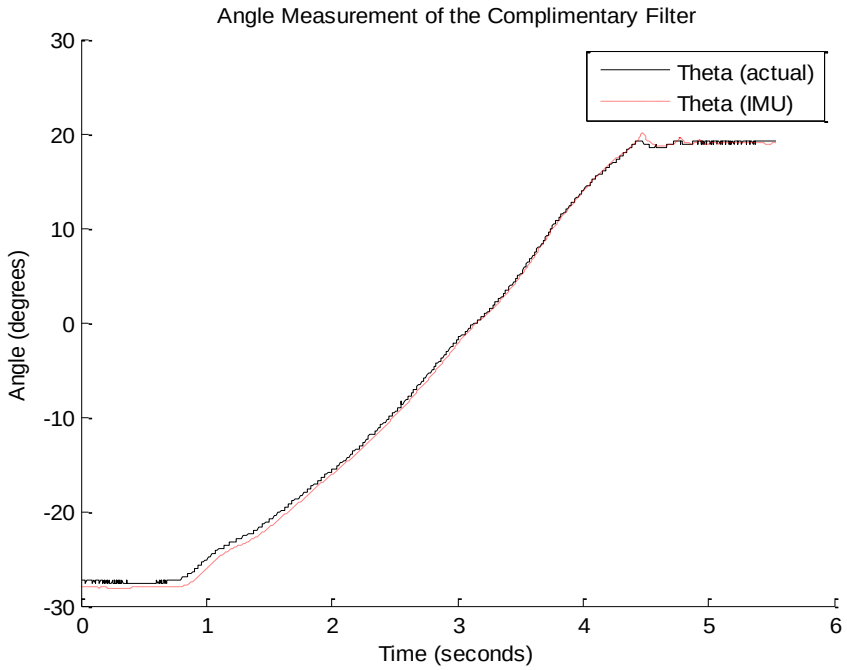

Figure 34. Orientation Sensor vs. Potentiometer readout over full range of motion

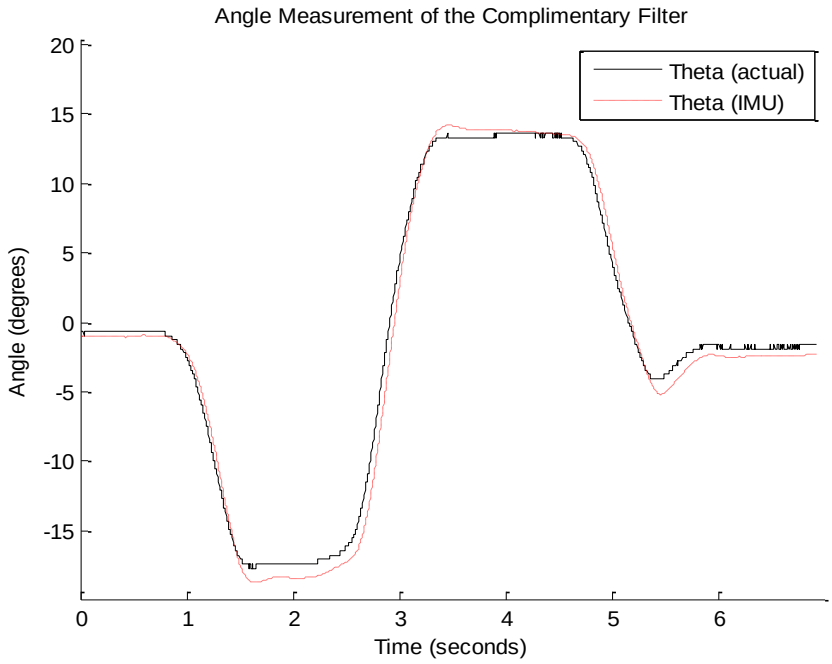

Figure 35. Orientation Sensor vs. Potentiometer readout undergoing step responses.

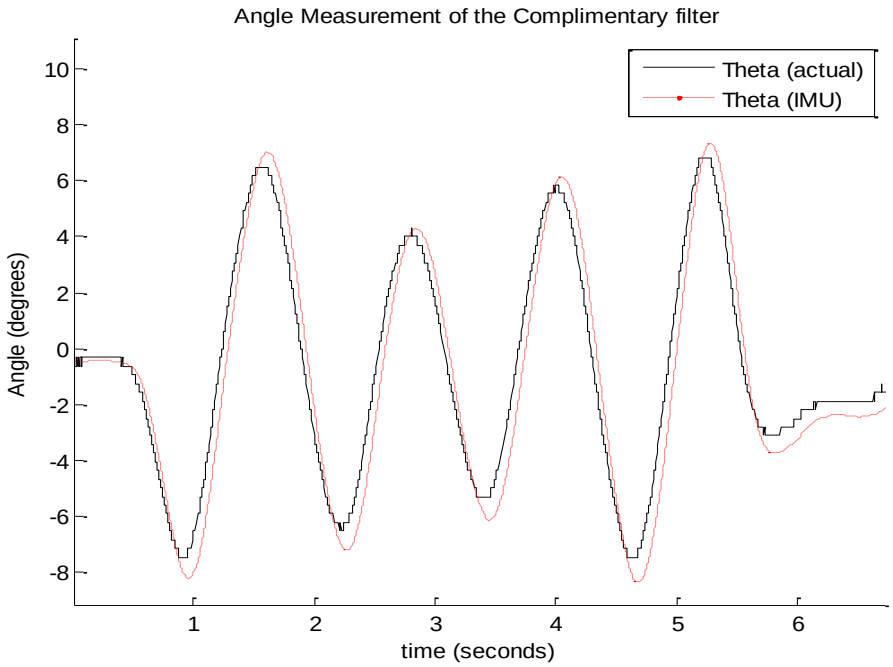

Figure 36. Orientation Sensor vs. Potentiometer readout through "sine wave" motion

The data above show that there is some error in the angle measurement of the orientation filter. Some uncertainty is added because the IMU is not mounted on the robots axis of rotation. Therefore, when the robot begins to pivot, the IMU measures the acceleration to the side as well as an upward acceleration due to gravity. Since the orientation filter measures normal force, anytime the robot pivots, a centripetal acceleration component is added to a_z. Also, when robot experiences angular acceleration, an acceleration component is added to a_x. These discrepancies caused by the motion of the robot result in error in the orientation estimate.

Also, since the accelerometer estimate is the result of a low pass filter, the orientation sensor is also a dynamic system. In fact, in the event of a gyroscope error, the system has the same dynamic response as that of a low pass filter. Not only does this add error to the

orientation estimate, but it adds new dynamics to the pendulum system. Despite these inaccuracies, the pendulum system is able to stabilize with the same controller. **Figure 37** shows the systems response.

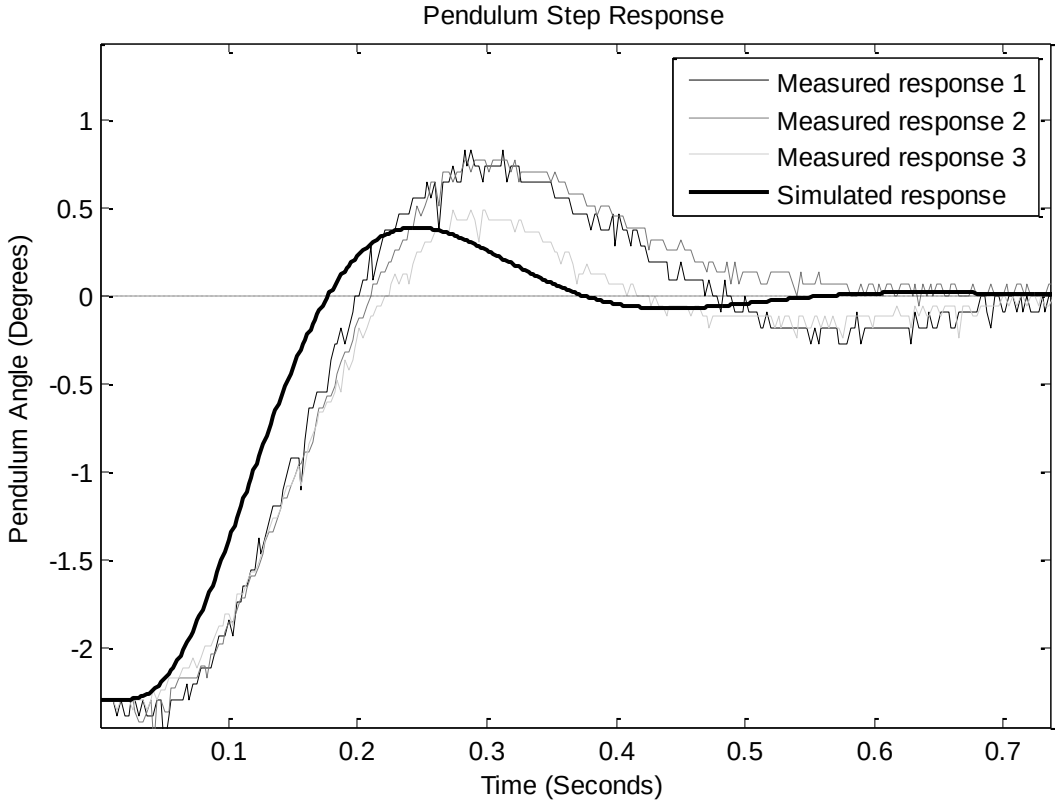

Figure 37. Step response of standalone theta controller

As **Figure 37** shows, the pendulum dynamics have changed slightly with the use of the orientation sensor. In relation to the initial (potentiometer based) results, the standalone system has more overshoot, and a slower response time.

In evaluating these discrepancies, we know that as the pendulum accelerates towards its point of stabilization, the angular acceleration causes error in the orientation sensor, as explained above. But since there is no way to measure gyroscope sensors drift or accuracy, we don't know how much of an impact the filtered nature of the accelerometer estimate has.

Although, it is quite apparent that it does have an impact because the systems response time is slower. Once again, we can confirm this using the root locus plot of the pendulum system. Adding a low pass filter to the root locus plot of the open loop system will simulate the low pass filter of the accelerometer estimate. Decreasing the magnitude this pole increases the response time of the system, while still allowing the system to be stabilized by the original controller. This resembles the motion illustrated by **Figure 37**.

Although the balancing angle is somewhat inaccurate, the system is stable regardless. It turns out that the proportional and filtered proportional controllers are able to accommodate inaccurate orientation measurements. If the orientation measurement is offset from actual zero, the robot initially stabilizes to this angle, but to maintain the angle, the theta controller rotates the gyroscopes axes at the velocity needed to stabilize at that point, against the pull of gravity. The resulting displacement of the gyroscope axis is fed back to the proportional or filtered proportional controller which adjusts the desired balancing angle to counteract the orientation error, until a net zero is found in the pendulum controller. Therefore, in steady state, an inaccuracy in the balancing angle estimate is corrected by a steady state phi offset. Furthermore, the inconvenience of having a steady state offset can be fixed as well. Recall, the "Simulation Results" section, mentions that an extra controller pole and zero can be placed in the left half plane of the phi controller, turning it into a filtered PI controller. When a controller pole is put at zero, and a controller zero is put very close to zero in the left half plane, a small integral component is added to the controller. Although this was not explored in the scope of this project, it can be used to further enhance the robustness of the controller. In this case, an inaccurate balancing measurement again causes the displacement of the gyroscopes causing

the robot to stabilize with gyro angles at an offset. But now, with the integral component in the controller, the offset between desired and actual phi cause integral windup which causes phi to approach zero. This turns out to be a useful feature of the phi controller because it increases the robustness of the controller, and can be extended to stabilize the robot influenced by external forces. The pendulum, for example leans into wind, and presses back against an external touch.

The standalone controller was implemented with the orientation sensor. **Figure 38** and **Figure 39** show the step response of both balancing angle, and gyroscope angle stabilization.

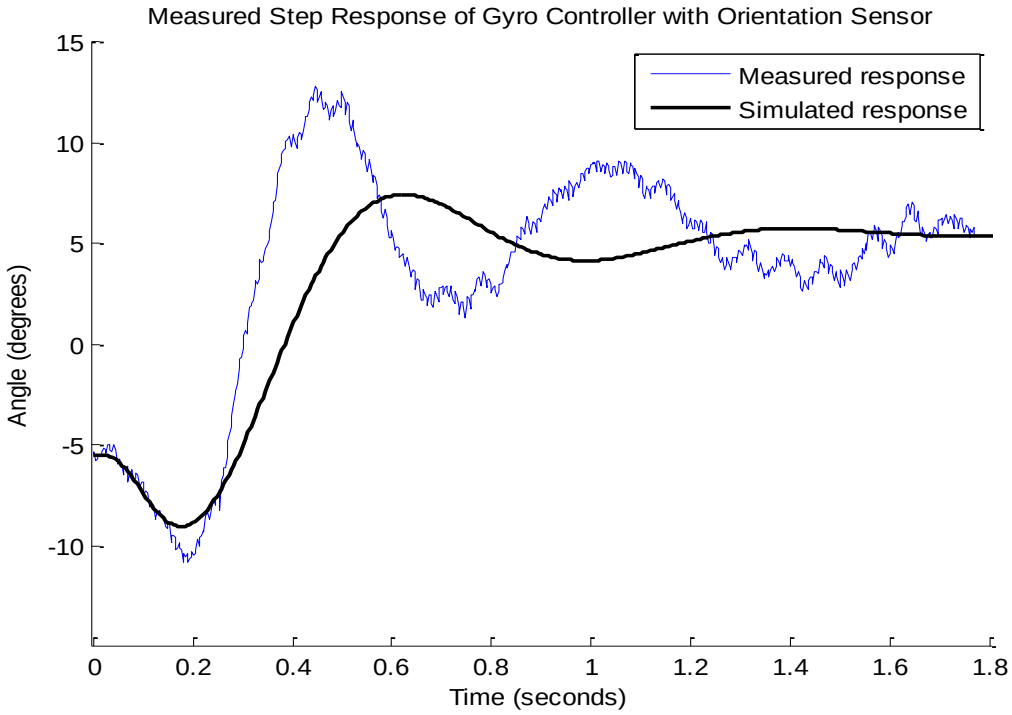

Figure 38. Gyroscope data – Initial stabilization of Standalone system, compared to the theoretical model.

Due to the inaccuracies of the orientation sensor, and its dynamics, the systems motion differed from the simulated motion. In **Figure 38**, it is apparent that the loop gain is too high.

We found that adjusting the gain from 0.052 to 0.04 at least helped correct the overshoot. Although the response time was still not quite right, the system stabilizes. **Figure 39** shows three common responses of the system along with the simulated response.

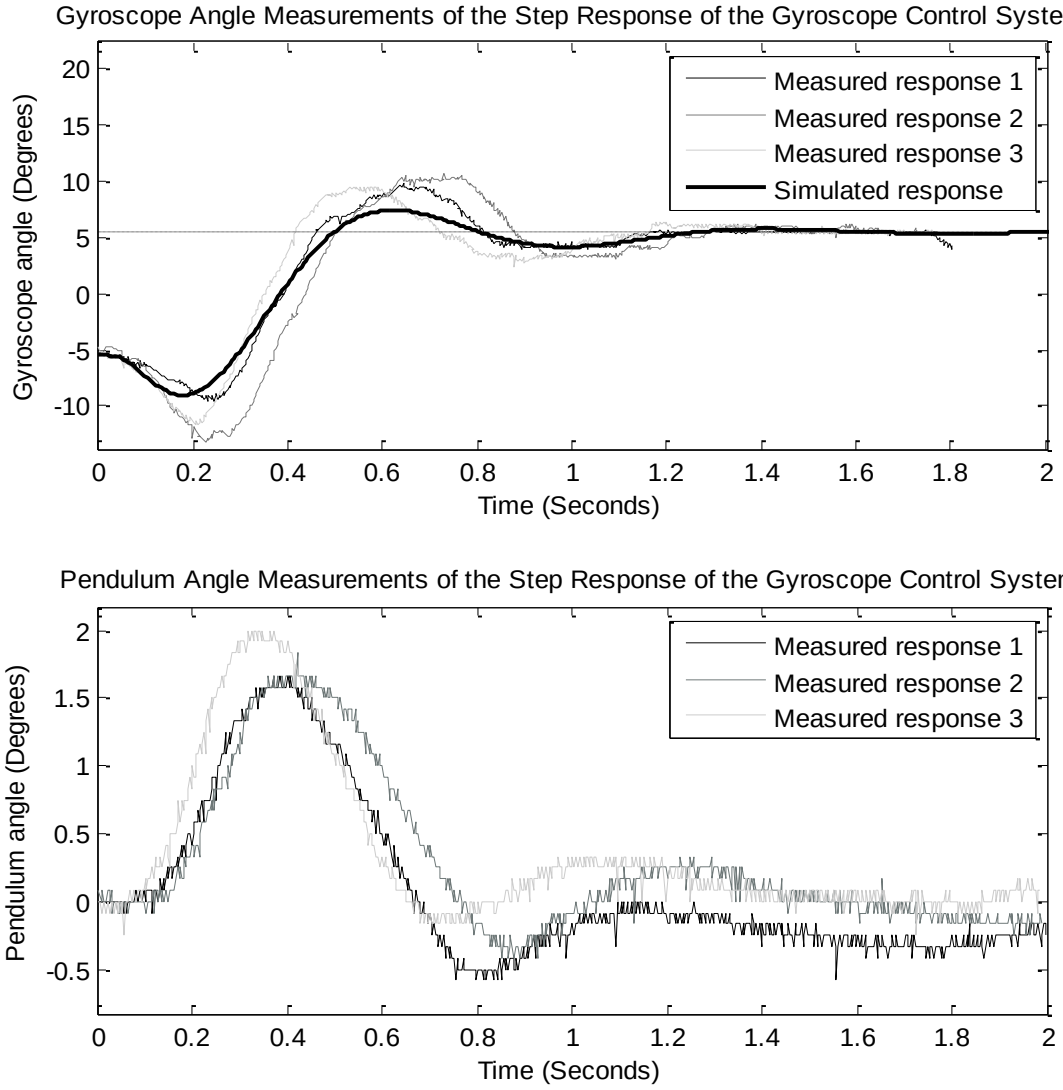

Figure 39. Final results of the gyroscope controller step response with the addition of an IMU orientation sensor. The top graph shows the simulated step response of the gyroscope control system along with three typical runs. The bottom graph shows the pendulum motion as the gyroscope control system responds to the step command. This graph shows that the pendulum is still stable with the addition of the gyroscope controller and IMU orientation sensor.

CHAPTER 5. CONCLUSIONS

The results of this study have shown an effective way to stabilize an inverted pendulum using control moment gyros. The stabilization of both balancing angle and gyroscope angle allows for a robust control system that can withstand a large range of external stimuli and erroneous orientation measurements. For instance, if the normal vector is inaccurate, the robot will still stabilize with only an offset in the equilibrium gyroscope angle. If the robot is tapped by an external force, it will stabilize in less than a second. Furthermore, the nature of the gyroscope angle stabilization loop allows the robot to lean towards a wind or in the direction of acceleration if the entire pendulum is pushed in a certain direction.

With the information found in this research, the control moment gyro system offers some clear advantages over other balancing methods. The linearized dynamic model can be stabilized with a very simple controller design. The control moment available in this system is available in a small compact space, where the amount of controllable moment is increased by gyroscope angular velocity, or moment of inertia. This can be done without adding any weight to the system. A set of two gyros can also be configured to stabilize a robot with two axes of freedom if each gyro is mounted on a two axis powered gimbal.

This system, if stabilized in only one axis, is also advantageous in the interest of energy usage. When the robot stabilizes about a point the gimbal servos only experience resistive torque when the pendulum angle changes. Therefore, if the pendulum stabilizes with very small angle fluctuations, the gimbal servos require very little energy, and experience very little deflection, due to back torque of the gyros. This is opposite to other common mechanisms

such as the reaction wheel pendulum, or Segway, which use shear torque exerted by their motors to provide the accelerations needed to stabilize the body. In an ideal environment, with no friction, the gyro system could use almost no energy to stabilize.

At the same time, the system has some clear disadvantages as well. The robot must maintain continuous rotation of its gyroscopes. Although a properly balanced gyro wheel with a well-designed axle and motor system could be very efficient, this can't be achieved without high precision machining and fabrication methods. Furthermore, an unbalanced gyro can add a lot of vibration into the system, which can add noise to the IMU data. This was a problem in our prototype because the hand crafted gyroscopes were not completely balanced. While running, the chassis vibrated strongly, deemed the IMU useless. This was fixed by mounting the IMU on a floating surface, coupled to the robot by an acoustic damping material. This couple was absorbent enough to reduce most of the noise in the IMU data. With these ideas in mind, our robot was not designed well. The gyros being mounted on small hobby style DC motors put considerable wear on the motor shafts due to the shaft torsion caused when rotating the gyroscopes. All of this friction and vibration absorbed a lot of energy out of the gyro momentum. An improved model will use an axle and bearing mounting both sides of the gyroscope.

With these strengths and weaknesses in mind, it is important for an engineer to evaluate the needs of the specific application before deciding the use of this system.

Future work

There are many areas that would be interesting to continue exploring with this system. The next logical step would be to control this system using full state feedback. This wouldn't be too difficult since the 4 states $\phi, \dot{\phi}, \theta$, and $\dot{\theta}$ can all be measured fairly easily. I would also be interesting to design a true Kalman filter to estimate θ, and design a linear quadratic regulator to maximize performance. It would also be interesting to develop a robot and controller that can stabilize the robot in two axes of motion. If this is achieved, it may be possible to experiment with some of the applications mentioned in the introduction.

With these ideas in mind, a new robot is already being fabricated, and will have some nice improvements. First of all, the gyros are machined to high precision and have very little vibration. They are supported on both sides of the gyro axel, which is integrated onto a brushless motor shaft. Both gyro gimbal axes are coupled together by a spur gear, so only one servo system is used to drive both gyroscopes. This will be nice because it forces the gyroscopes to be at equal and opposite angles at all times. The centers are milled out of the gyroscopes so the moment of inertia to mass ratio is almost doubled. The new chassis will have two sets of gyroscope systems so both forward and side to side leaning can be influenced by the system. The system uses high torque servos that may be modified by a microprocessor to have a more accurate response. The whole robot will be controlled by an MBED microprocessor, which has some impressive specs in relation to the arduino used in the current system, and would make a great controller for this system. The gyroscope to chassis weight ratio is greater, which will lead to increased performance and robustness. All of these improvements should lead to a drastically improved prototype.

REFERENCES

1. Bapiraju, Bonagiri, and Laxmidhar Behera. "On Balancing Control Strategies for a Reaction Wheel Pendulum." *IEEE* (2004): 199-204. Print.

2. Hernández, Victor M. "A Combined Sliding Mode-Generalized Pi Control Scheme For Swinging Up And Balancing The Inertia Wheel Pendulum." *Asian Journal of Control*5.4 (2003): 620-25. Print.

3. Garabedian, Bruno, Michel Benoit, and Sebastien Krut. "A Futuristic Monorail Tramway Stabilized by an Inertia Wheel." *IEEE International Conference on Control and Automation* (2007): 1581-586. Print.

4. Salerno, Alessio, and Jorge Angeles. "A New Family of Two-Wheeled Mobile Robots: Modeling and Controllability." *IEEE Transactions on Robotics* 23.1 (2007): 169-73. Print.

5. Nawawi, S. W., M. N. Ahmad, A. R. Husain, and M. F. Abdollah. "Controller Design for Two-wheels Inverted Pendulum Mobile Robot Using PISMC." *Student Conference on Research and Development* (2006): 194-99. Print.

6. Li, Jingtao, Hueshan Gao, Qiang Huang, and Osamu Matsumoto. "Controller Design of a Two-Wheeled Inverted Pendulum Mobile Robot." *IEEE International Conference on Mechatronics and Automation* (2008): 7-12. Print.

7. Mao-Lin, Chen. "Analysis and Design of Robust Feedback Control Systems for a Nonlinear Two-Wheel Inverted Pendulum System." *International Symposium on Computer, Consumer and Control* (2012): 949-53. Print.

8. Muhammad, M., S. Buyamin, M. N. Ahmad, and S. W. Nawawi. "Dynamic Modeling and Analysis of a Two-Wheeled Inverted Pendulum Robot." *Third International Conference on Computational Intelligence, Modelling & Simulation* (2011): 159-64. Print.

9. Awtar, S. "Inverted Pendulum Systems: Rotary and Arm-driven - a Mechatronic System Design Case Study." *Mechatronics* 12.2 (2002): 357-70. Print.

10. Singularity Hub. "Toyota Humanoid Robot Runs At 7 Km/hr (4.35 M/Hr) - Video // Current TV." *Toyota Humanoid Robot Runs At 7 Km/hr (4.35 M/Hr) - Video // Current TV*. N.p., 29 July 2009. Web. 27 Nov. 2012. <http://current.com/135im4c>.

11. "Honda Worldwide | ASIMO | Running at 6km/h (00:35)." *Honda Worldwide | ASIMO | Running at 6km/h (00:35)*. Honda Motor CO., Dec. 2005. Web. 27 Nov. 2012. <http://world.honda.com/ASIMO/video/2005/run-6kmh/index.html>.

12. Alexander, J. D., Dr., and K. D. Dannenberg, Dr. "MISSILE CONTROL EMPLOYING CONTROL MOMENT GYROS." *IEEE* 16 (1977): 220-25. Print.

13. Fan, Jixiang, and Di Zhou. "Nonlinear Attitude Control of Flexible Spacecraft with Scissored Pairs of Control Moment Gyros." *First International Conference on Pervasive Computing, Signal Processing and Applications* (2010): 719-22. Print.

14. Steyn, W. "A Dual-wheel Multi-mode Spacecraft Actuator for Near-minimum-time Large Angle Slew Maneuvers." *Aerospace Science and Technology* 12.7 (2008): 545-54. Print.

15. Xu, Kai, Guang Jin, and Changqing Chen. "United Steering of Control Moment Gyros for Small Satellite with Limited Gimbal Rate." *IEEE International Conference on Information and Automation* (2010): 1489-493. Print.

16. Newkirk, Burt L. "Nutation and the Monorail Car." *Journal of the Franklin Institute* 174.3 (1912): 265-78. Print.

17. Ferreira, Enrique D., Shu-Jen Tsai, Christiaan J. Paredis, and Benjamin Brown. "Control of the Gyrover: A Single-wheel Gyroscopically Stabilized Robot." *Advanced Robotics*14 (2000): 459-75. Print.

18. Lam, Pom Yuan. "Gyroscopic Stabilization of a Kid-Size Bicycle." *IEEE 5th International Conference on Cybernetics and Intelligent Systems* (2011): 247-52. Print.

19. "Segway - Simply Moving." *Segway â" The Leader in Personal, Green Transportation*. N.p., 2001. Web. 02 Dec. 2012. <http://www.segway.com/>.

20. Moffett, Cleveland. "TRANSPORTATION AND THE GYROSCOPE LOUIS BRENNAN'S MONO-RAIL CAR." *Catskill Archive*. Timothy J. Mallery, n.d. Web. 2 Dec. 2012. <http://www.catskillarchive.com/rrextra/ODGYRO.Html>.

21. "Accelerometer & Gyro Tutorial." *Instructables*. N.p., n.d. Web. 7 Dec. 2012. <http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/>

22. Colton, Shane. *The Balance Filter*. Tech. N.p., 25 June 2007. Web. 7 Dec. 2012. <http://web.mit.edu/scolton/www/filter.pdf>

23. Carpenter, Michele D., and Mason A. Peck. "Dynamics of a High-Agility, Low-Power Imaging Payload." *IEEE Transactions on Robotics* 24.3 (2008): 666-75. Print.