**December 4, 2016**

University of Houston

Electrical Engineering Department

4800 Calhoun Road

Houston, TX 77004

**Dear ECE Department:**

Attached is a report pertaining to the design and implementation of the Ball Balancing Robot that Team 2 has built. Team 2 consists of Jitin George, Christian Carlin, Zachary Freeman and Leonardo Han Lee. This report contains a detailed discussion of the way Team 2 has approached the design of the Ball Balancing Robot, including its specifications and constraints. Also included in the report is the Goal Analysis which is a graphical representation of the individual goals that have been met in order to meet the final goal of building a robot that attempts to balance on a ball.

**Sincerely,**

**Team 2**

Advanced Ball Balancing Robot

University of Houston

Facilitators:

Dr. Jose L. Contreras-Vidal

Dr. Shin-Shem Steven Pei

Sponsor: Self Sponsored

Jitin George

Christian Carlin

Zachary Freeman

Leonardo Lee

ECE 4336

12/04/16

**Abstract**

The purpose of this project was to create a robot that could balance on a ball without outside assistance in order to lay out a foundation for the construction of an omnidirectional surveillance bot. The ball bot operates off of an input voltage of 11.1 volts supplied by a Turnigy 4000mAh 3s 30-40 [C] LiPo battery. The physical structure of the robot consists of a plastic base which holds up the three sets of brushed dc motors and omni-wheels along with the Tiva-C microcontroller responsible for all of the robot’s functions. Currently the robot has the ability to connect to a PC via Bluetooth up to a range of 100 feet. The robot uses an IMU to read in accelerometer and gyroscope values for use in the control algorithm; these values are passed through a Kalman filter to increase reliability and stability and a control algorithm was built based off these values to make the robot balance on the ball. In its current state, the robot balances on a ball without outside assistance; while this is a great accomplishment for our team there is still much that can be done with this project. The construction of this robot has laid an excellent groundwork for any future team to pick up where we left off; and in future development it is recommended that the omni-wheels be replaced with rubberized omni-wheels and the current motors be replaced with motors that have increased torque for better micro-corrections on the ball.

**Table of Contents**

[Introduction and Background 5](#_Toc368387411)

[Statements of Goals 6](#_Toc368387412)

[Specifications 7](#_Toc368387413)

[Constraints 9](#_Toc368387414)

[Design 10](#_Toc368387411)

[Methodology 12](#_Toc368387411)

[Results 15](#_Toc368387412)

[Conclusions 23](#_Toc368387413)

[Recommendations 23](#_Toc368387414)

[Financial Summary 24](#_Toc368387415)

**Introduction and Background**

The purpose of this report is to discuss the design, testing, and overall success Team 2 has experienced while building a robot that can balance on a ball without outside assistance. This document not only includes details about the design, specifications and constraints of the robot but also contains information about the obstacles that were encountered along the way and the steps that were taken to overcome them.

The goal of our project was to take preliminary steps in the advancement of omnidirectional movement for surveillance type robots. When we thought of traditional remote controlled surveillance robots, movement and maneuverability is usually limited due to the minimum turning radius of the robot wheels. This greatly reduces the ability of these robots to traverse tight spaces and hard to reach areas thereby limiting their ability to provide surveillance of these regions. We intended to take the first step in solving this problem by building a robot that could balance on a ball. Now that this goal has been accomplished, the robot can be further improved in future semesters by making it drivable and provide surveillance of the areas traversed. Making the robot drivable while balancing on a ball will allow it to fit into smaller areas and tight spaces with ease due to its 0 degree turning radius. Such a robot could be used to provide surveillance of crowded parking lots or even small alleyways in a crowded street where the 0 degree turning radius of the robot would come in handy to navigate through tight spaces without crashing into obstacles.

We built the robot with 3 omni-wheels and DC brushed motors that help the bot to balance on a ball. The purpose of the omni-wheels is to move based on the movement of the ball in order to prevent the robot from falling over. The movement of the omni-wheels is controlled by the TI microcontroller which serves as brain of the robot which is also used to interface with the gyroscope and accelerometer. The data from the gyroscope and accelerometer is then be passed through a Kalman filter which outputs the relative angle of the robot with respect to the X and Y axis. This data is then used to adjust the robot’s motors which allows the robot to climb back to the top of the ball where the Kalman filter reads a 0 degree angle with respect to the X and Y axis.

**Statements of Goals**

The following will discuss the goals that were set for our group throughout the duration of this project beginning with those put into place during the first semester and concluding with those that were set for the final semester. All goals set out to be completed are shown below in Figure 1.

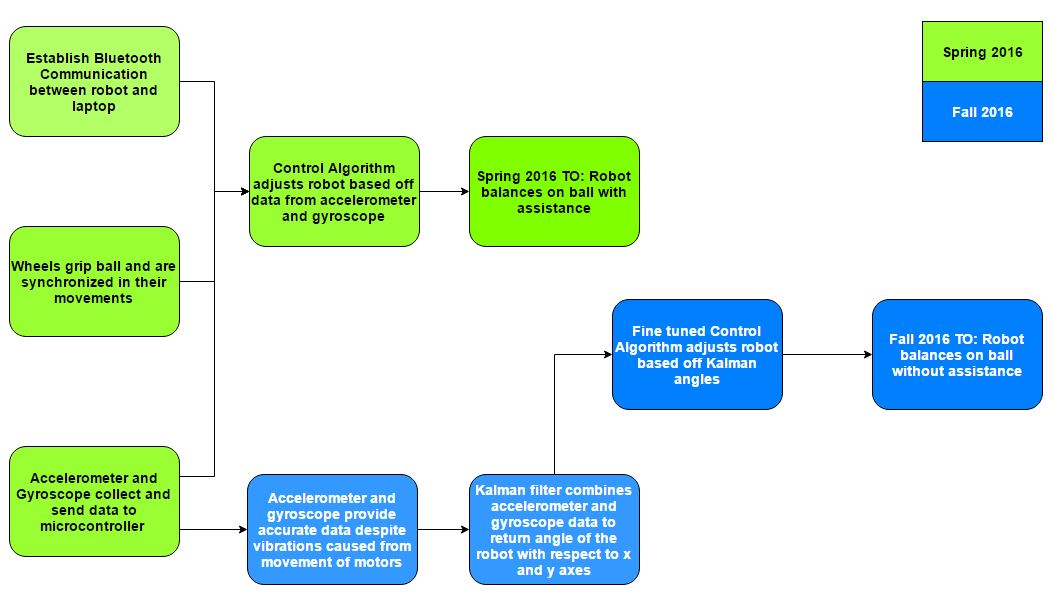


Figure 1: A diagram which shows all presented goals for the project

The first goal of the spring semester was to establish a Bluetooth connection with our robot using the BlueSMiRF gold Bluetooth modem. Once that was established, the team would then need to ensure that all the motors were working and receiving commands; this would be tested by sending various commands to the motors through the previously established Bluetooth connectivity. Next the team would interface the IMU with the Tiva C and then scale those values read in by the IMU to get meaningful data for the accelerometer and the gyroscope. These values would then be passed through a Kalman filter which should output an angular value for the robot’s position in space with respect to the X and Y axis. Because of issues with accelerometer data being distorted through the tactile vibration of the motors being powered on an anti-vibrational apparatus would be built and tested for successful vibrational mitigation. Once this was complete the team would implement and refine a working control algorithm; this control algorithm would take in values given by the Kalman filter and the robot would correct off of those given values. Finally, once the control algorithm had been polished the robot would be able to balance on the ball without outside assistance.

**Specifications**

**Establish Bluetooth communication between robot and laptop:**

The robot will sync and communicate with a PC up to a range of 100 feet with a 0% allowance for failure/error.

**Wheels grip ball and are synchronized in their movements:**

Wheels will respond to given commands through Bluetooth with 100% accuracy with a 0% allowance for failure/error.

**Accelerometer and Gyroscope collect and send data to the microcontroller:**

MCU operates off a 40 MHz clock frequency and communicates with IMU using I2C communication which gives accelerometer and gyroscope data with a +/- 5% error tolerance.

**Accelerometer and Gyroscope data combined through the use of a Kalman filter:**

Using a Kalman filter both gyroscope and accelerometer data are combined to output an angular reading with respect to the X and Y axis with a +/- 2 ° error tolerance.

**Accelerometer distortion from vibrational interference mitigated:**

The anti-vibrational apparatus that was constructed to mitigate the vibrations felt by the accelerometer will dampen 75% or more of vibrational interference.

**Control Algorithm adjusts robot based off data collected from the gyroscope and accelerometer:**

The control algorithm will correct in accordance with the Kalman filter’s output data within 90% accuracy.

**Robot balances on the ball:**

The robot will continuously maintain its balance on a ball and will have the ability to correct against a moderate manual offset given by the user.

**Constraints**

One of the major constraints for this project is the cost. Team 2 is self-sponsored which in turn restricts us from buying expensive power sources and high efficiency motors that could have been used to improve its stability while trying to balance on the ball. Weight is another constraint for the robot. More powerful motors would be required in order to adjust the robot quick enough for it to balance if the robot were to be too heavy. This would incur a higher cost and as mentioned before, we are constrained by cost. In order to avoid having to purchase very expensive motors, the weight of the robot will be limited to 7 kilograms. The robot will only be able to operate on a flat surface. Stairs and slopes are constraining factors for the robot as it will not be able to traverse these obstacles.

**Design**

For reference in the following description of the design choices made in the construction of this robot an image of the ball balancing robot with each of its pertinent parts labeled will be displayed below in Figure 2.

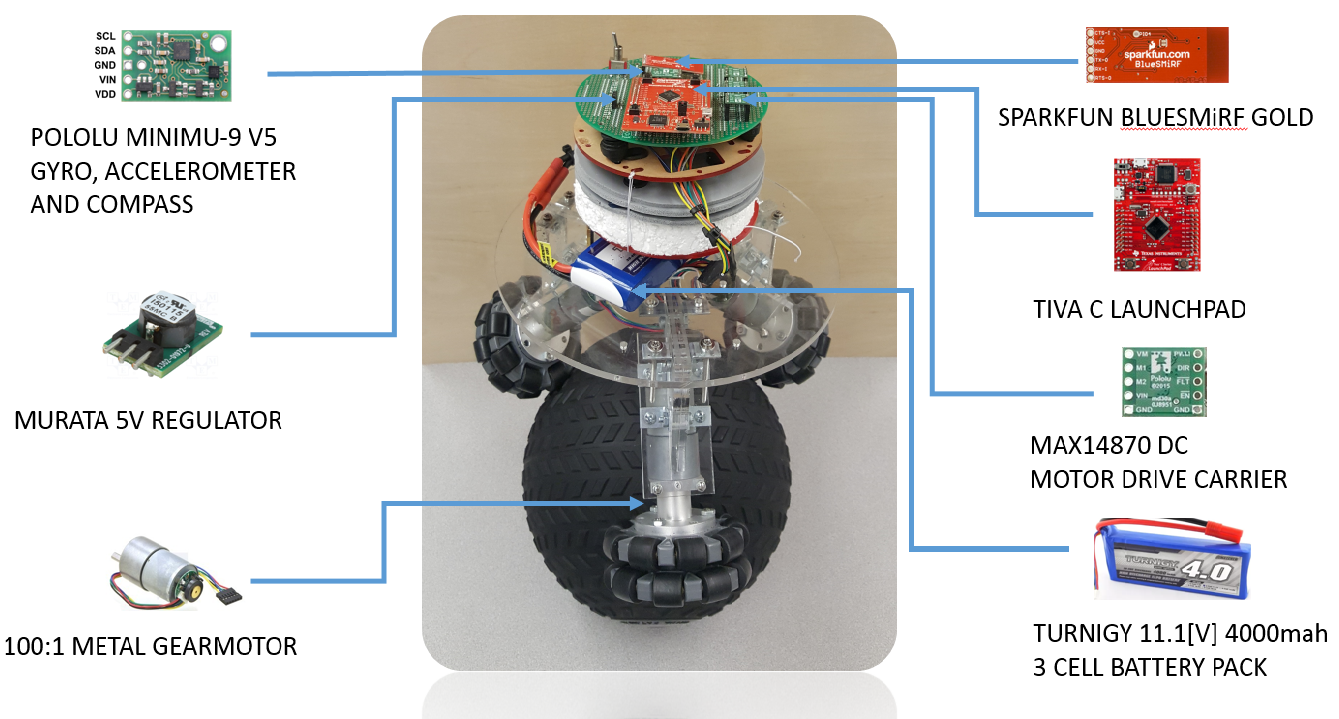


Figure 2: Overview Diagram for the Ball Balancing Robot.

**SparkFun Bluetooth Modem - BlueSMiRF Gold:**

The BlueSMiRF Gold Bluetooth modem has the ability to communicate with a device at extended ranges of up to 100 feet. The silver edition of the same Bluetooth modem is indeed cheaper, but has much more limited range capabilities. With the expectations of this robot becoming mobile in the future, the gold edition was chosen for the enhanced range of connectivity.

**Tiva C Series TM4C123G LaunchPad:**

With such a large variety of microcontroller on the market for use in small scale robotics applications like this one there was a lot of consideration that went into the selection of our microcontroller. Other microcontrollers were considered for use in our project such as Arduino’s line of microcontrollers which hold high esteem in the current market for DIY projects. However, the Tiva C was chosen not only because it was the most familiar device to work with for every member of the team, but also for its ability to execute multiple threads simultaneously.

**MAX 14870 DC Motor Drive Carrier:**

In the early stages of development each motor was connected to a VNH5019 Motor Drive Carrier. This chip was causing issues which led to decreased response from the motors after having swapped directions; and so after multiple failures in testing of the motor’s ability to switch direction back and forth quickly the chip was swapped out for the MAX 14870 DC Motor Drive Carrier instead.

**Turnigy 11.1 [V] 4000 [mAh] 3s 30-40 [C] Battery Pack:**

The team previously assigned to this design project was using a 7.4 [V] 2s “Venom” brand battery; they stated that the motors they were using weren’t giving enough of a response due to the low gearbox ratio in the motors. This turned out to be untrue, the actual cause of the “slow” motors was actually just low voltage to the motors. Our motors require at least 5 [V] to be powered on and can sustain up to 12 [V] of power. Because our group wanted to get the most out of what hardware we had, a new Turnigy 11.1 [V] 4000 [mAh] 3s 30-40 [C] battery pack was selected and purchased for further testing.

**Murata 5 [V] Regulator:**

In the early stages of development the team used a S7V7F5 5 [V] voltage regulator chip from Pololu. This chip worked perfectly fine while the original 9 [V] battery was in use; but in the second semester of development this regulator failed due to the fact that our updated battery discharged too much current for it to handle. The S7V7F5 has a current limit of 500 [mA] which was not enough for our new battery. This led to the selection of the Murata 5 [V] regulator due to its 1 [A] current limit.

**100:1 DC Brushed Metal Gear Motors and Omni-wheels:**

The motors and wheels that were used throughout the duration of the project were left by a former team as the project was passed on. These wheels and motors did perform decently and we were able to achieve our goals using them; however it is recommended that in the future these omni-wheels be swapped our for rubberized omni-wheels and the motors be swapped out for higher torque motors instead. This will lead to better micro-corrections while on top of the ball.

**Methodology**

**Establish Bluetooth communication between robot and laptop:**

Team 2 used the BlueSMiRF Gold Bluetooth module that once connected through Bluetooth can transmit and receive to and from GPIO pins on the microcontroller and enable the user to connect via laptop up to 100 ft. away. Once 3.3 [V] was connected from the microcontroller to the module, all we had to do was search for Bluetooth connections on either a phone or laptop, and find the specific name attributed to our module. Once connected, TeraTerm was used to retrieve and send commands directly to the module, and the robot itself. The Bluetooth module has an LED that lights up green once connected for verification.

**Wheels grip ball and are synchronized in their movements:**

The lab bench power supply was used to hook up each motor individually to test for reliability. This needed to be done in order to thoroughly test the motors since they were not brand new. After verifying the motors themselves worked just fine, each motor was connected to a VNH5019 Motor Driver Carrier which is connected logically to the microcontroller. Now that the motors were hooked up to the microcontroller and Bluetooth communication was set up, different commands could now be sent [ startMotorClock(), startMotorCounter(), stopMotor() ] to each motor to test the ability to control the motors separately and in unison. One problem that was encountered was that when going from clockwise to counterclockwise rotation the motors wouldn’t give a strong response. This was discovered to be due to the Motor Driver Carrier itself, so the MAX14870 Single Brushed DC motor drive carrier was used instead.

**Accelerometer and Gyroscope collect and send data to the microcontroller:**

The minIMU that is being used requires communication between the microcontroller and the sensors, which is a protocol that no one on Team 2 had any experience with before. Nearly a week was spent researching and experimenting with this new concept before the sensors were properly connected and initialized. Once everything was initialized the accelerometer and the gyroscope data was the now available through the chip sensors to be used in the control algorithm to make the ball balance.

**Accelerometer and Gyroscope data combined through the use of a Kalman filter:**

Because accelerometer noise interference can cause accelerometer data to become unreliable and gyroscope data steadily becomes more unreliable as time goes on a countermeasure was needed to ensure reliable data was being given for extended periods of time. Thus a Kalman filter was developed and implemented which allows the users to sustain reliable and usable data for any given period of time. This filter works recursively, combining values read in by the accelerometer and gyroscope to output a steady angular value in degrees with respect to the X and Y axis.

**Accelerometer distortion from vibrational interference mitigated:**

Vibrational noise experienced from the tactile feedback of the motors being powered on caused the accelerometer data to spike up and down uncontrollably. This caused extremely unreliable data when it came time for the robot to correct and had to be nullified immediately. An anti-vibrational apparatus was constructed using Styrofoam, silicone, “Moon gel”, and rubber gimbals which separated our IMU from the vibrational feedback of the motors.

**Control Algorithm adjusts robot based off data collected from the gyroscope and accelerometer:**

In order for the robot to balance on top of the ball a robust control algorithm had to be designed which accounted for the weight of the robot, the weight of the ball, the size of the ball, etc.. Multiple control algorithms were created to manipulate both the speed of the motors and the location of the robot in space.

**Results**

**Objectives Accomplished**

**Establish Bluetooth communication between robot and laptop:**

BlueSMiRF Gold Bluetooth module was used to accomplish this goal. Once wired to the microcontroller, TeraTerm was used to communicate directly to the Bluetooth modem connected to the robot. A toggle LED command was sent to the microcontroller using TeraTerm to test the Bluetooth connection. The LED did toggle; therefore, this goal was considered accomplished.

**Wheels grip ball and are synchronized in their movements:**

Each motor was connected to the microcontroller through MAX14870 Single Brushed DC Motor Drive Carriers. Different commands were then sent [RunMotorRight(), RunMotorLeft(), StopMotor()] to each motor as a test. The motors responded to the commands sent from the microcontroller and rotated right, left or stopped accordingly. Thus this goal was considered accomplished.

**Accelerometer and Gyroscope collect and send data to the microcontroller:**

Programming the minIMU 9v5 accelerometer and gyroscope chip was a challenge for the team because it required knowledge of I2C communication protocol. Since no one on the team was familiar with this, a lot of time was spent conducting research and learning about the I2C protocol. Once programming of the accelerometer and gyroscope was complete, the minIMU was able to collect and send data to the microcontroller which could now be used by the control algorithm to make the robot balance. Figure 3 below shows the readings from the accelerometer and gyroscope when the robot was held stable on a flat surface.



Figure 3: Accelerometer and Gyroscope readings when robot was held stable on a flat surface.

Gys, Gxs and Gzs in Figure 3 stand for scaled Gyroscope readings along the Y axis, X axis and Z axis respectively. Similarly, Axs, Ays and Azs stand for the scaled Accelerometer readings along the X, Y and Z axis. It can be clearly seen from figure 4, that when the robot is held stable on a flat surface, the gyroscope and accelerometer reads 0 for all axis. The only exception is Azs which is approximately -9.8 ms­-2which is the acceleration due to gravity acting on the robot. Our next test was to take these readings while tilting the robot in different directions. The result of this test is shown in Figure 4.

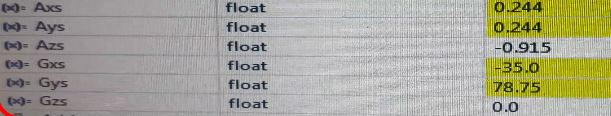


Figure 4: Accelerometer and Gyroscope readings when robot is being tilted.

It can be seen that the readings from the accelerometer and gyroscope change based on the movement of the robot.

**Kalman Filter combines accelerometer and gyroscope data to return angle of the robot with respect to the X axis and Y axis:**

Having failed to balance the robot on the ball without assistance after multiple attempts of fine tuning the PID, the team realized that in order to get the PID to adjust the robot correctly, the accelerometer and gyroscope readings would have to be combined to make it more dependable. We had determined last semester itself that the gyroscope readings were not dependable because of the drift that it would experience after a period of time which would in turn sway the values. This had led us to use solely the accelerometer readings for the PID to make adjustments to the robot. This had its own drawbacks. Although, the accelerometer provides exact readings for extended periods of time, it is susceptible to noise. In order to overcome these obstacles, we decided to incorporate a Kalman filter that would combine the accelerometer and gyroscope data to output a dependable angular value. Once this filter was programmed and incorporated into our project, it returned the angle of tilt of the robot with respect to the X and Y axis. This angle returned by the Kalman filter can now be used to make the necessary adjustments to bring the robot back atop the ball to reach the 0 degree state with respect to the X and Y axis. The results from the Kalman filter are discussed below:

Figure 5 shows the accelerometer and gyroscope readings as well as the output from the Kalman filter when the robot is held stable on a flat surface.

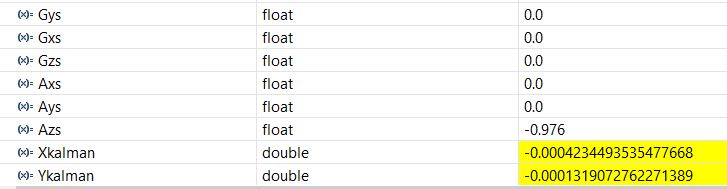


Figure 5: Kalman filter output when robot is held stable on a flat surface.

Xkalman stands for the angle of tilt of the robot along the X axis, while Ykalman stands for the angle of tilt of the robot along the Y axis. It can be clearly seen from figure 9 that when the robot is held on a flat surface, the Kalman filter outputs an angle of 0 for both axis.

Figure 6 shows the output of the Kalman filter when the robot is tilted in the negative X direction.

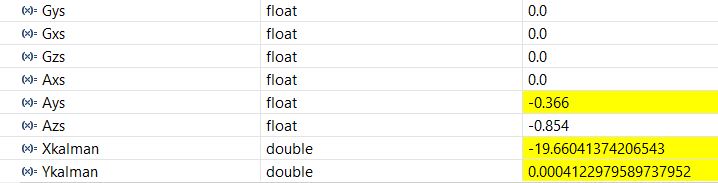


Figure 6: Kalman filter output when robot is tilted in the negative X direction.

Since the robot is tilted in the negative X direction, the angle Xkalman is -19.6 degrees and Ykalman is 0 degrees since there is no movement along the Y axis.

Figure 7 shows the output of the Kalman filter when the robot is tilted in the positive X direction.

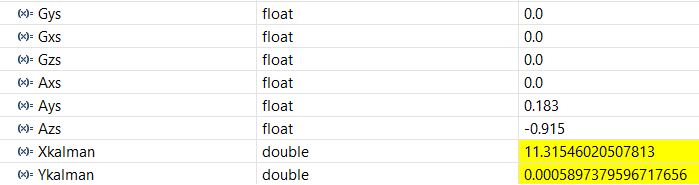


Figure 7: Kalman filter output when robot is tilted in the positive X direction.

Since the robot is tilted in the positive X direction, the angle Xkalman is 11.3 degrees and Ykalman is 0 degrees since there is no movement along the Y axis.

Figure 8 shows the output of the Kalman filter when the robot is tilted in the negative Y direction.

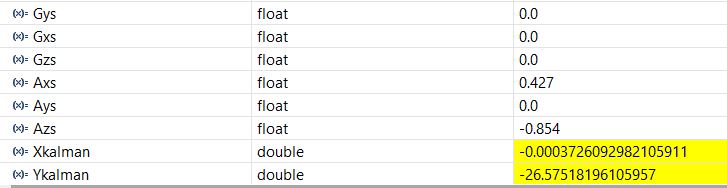


Figure 8: Kalman filter output when robot is tilted in the negative Y direction.

Since the robot is tilted in the negative Y direction, the angle Ykalman is -26.5 degrees and Xkalman is 0 degrees since there is no movement along the X axis.

Figure 9 shows the output of the Kalman filter when the robot is tilted in the positive Y direction.

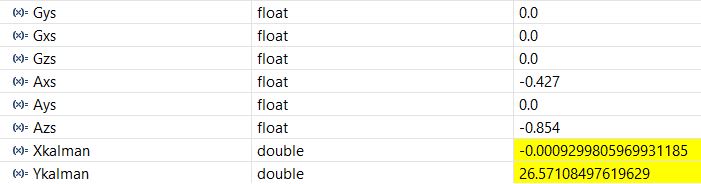


Figure 9: Kalman filter output when robot is tilted in the positive Y direction.

Since the robot is tilted in the positive Y direction, the angle Ykalman is 26.5 degrees and Xkalman is 0 degrees since there is no movement along the X axis.

Figure 10 shows the output of the Kalman filter when the robot is tilted along the xy plane.

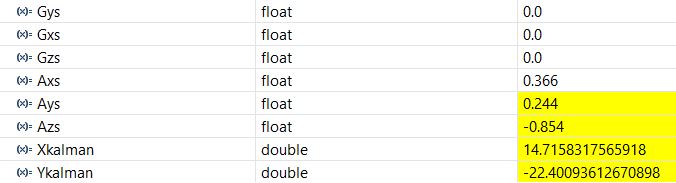


Figure 10: Kalman filter output when robot is tilted along the XY plane.

Since the robot is tilted along the XY plane, the angle Xkalman is 14.7 degrees and Ykalman is -22 degrees.

**Accelerometer distortion from vibrational interference mitigated:**

After the creation and implementation of the Kalman filter, the angular readings that were at their base rooted to the accelerometer readings were still spiking, not by as much, but the distortion was still there. This was due to the vibrational interference coming from the tactile feedback of the motors being powered on. The only way for this to be avoided was to implement some sort of anti-vibrational apparatus or by separating the board from the robot completely which was not a viable option. Thus, we combined Styrofoam, silicone, “Moon gel”, and rubber gimbals to form our anti-vibrational rig. A seismometer was used to measure the vibrations that the device experienced while the motors were powered on. The readings of the seismometer without our anti-vibrational hardware are shown below in Figure 11.

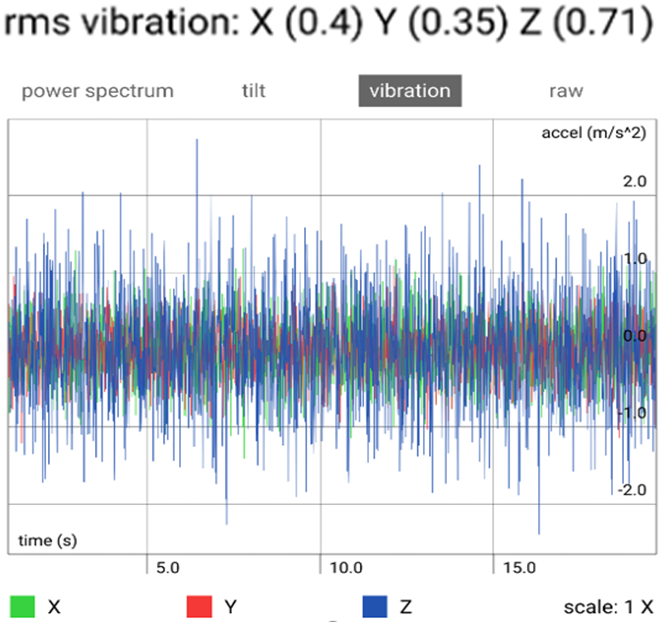
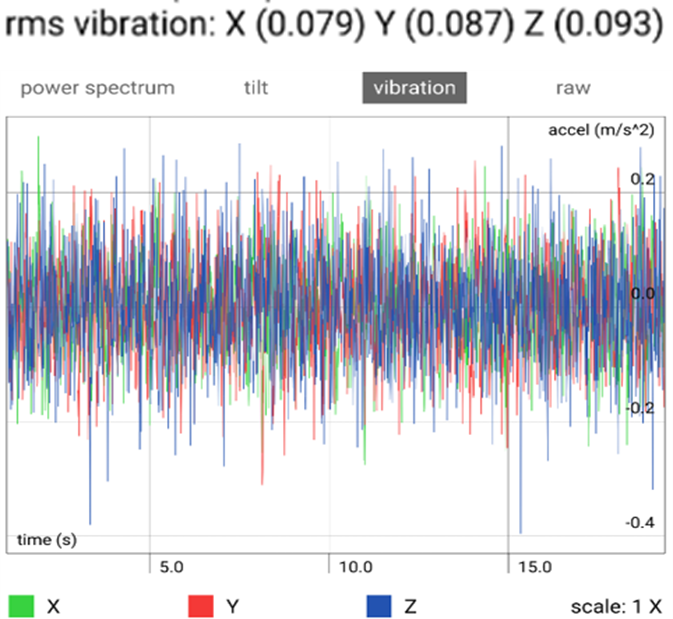


Figure 11: Seismometer readings without anti-vibrational apparatus.

As shown by the readings in Figure 11, the vibrations were throwing off the accelerometer readings by quite a bit in some cases by 4 or 5 times what they should have been. Then the vibrational dampening materials were added to the robot and the seismometer test was run again; the results of that test is shown below in Figure 12.

 Figure 12: Seismometer readings with anti-vibrational apparatus.

As shown by the data in Figure 12, the vibrational acceleration was now reduced by more than 75% which led to much more reliable data only ever spiking by +/- 3°.

**Control Algorithm adjusts robot based off data collected from the gyroscope and accelerometer and the robot balances on top of a ball without outside assistance:**

In the final stages of the project the only thing remaining was the development and refinement of our PID controllers. The control algorithm had been the biggest challenge for our team throughout the duration of the project. The team initially used two PID controllers in order to get a working control algorithm. One PID was controlling the speed of the wheels while the other controlled correction in each of the various directions. The two respective PIDs once optimized would make the robot balance on the ball. However, upon implementation of these controllers an issue was experienced with the DC motors that had been used for the extent of the project. These DC motors were fine for the majority of our corrections and worked well, but when the robot was at the top of the ball between +/- 10° from the origin it would attempt to make micro-corrections which would cause the motors to stall out due to a lack of torque in the motors. In order to alleviate this issue, the duty cycle was ramped up to overcompensate for these problems; this led to the system being unstable. The final decision by the group was to instead of trying to change the motors at the last minute, change the way the system corrected. The team created a new controller that was indeed not the PID that had been recommended to us by our advisors, but a different type of system that worked for our low torque motors. By essentially establishing quadrants for the system to correct off of the robot would know what direction it was falling in. Simultaneously in a separate subroutine the controller would change the speed the motors corrected at using the Kalman filter’s angular outputs as a guideline. This led to a more responsive controller which allowed the robot to balance on top of a ball without outside assistance.

**Conclusions**

Over the last semester our team has completed all of the initially established goals. We have overcome the unreliable accelerometer data that affected our controller in the spring semester through the use of the Kalman filter in concurrence with our anti-vibrational apparatus. We have fully designed and optimized a controller that works and balances the robot on top of the ball without assistance. Most importantly, we have created a solid foundational project to be carried on by a future senior design team.

**Recommendations**

The construction of this robot has laid a groundwork for any future team to pick up where we left off. With another year of work from an equally qualified design team this robot has the potential to become remote controlled and pilotable which would realize our initial vision of a drivable omnidirectional ball bot. However, if this project is to be carried on in the future there are a few technical recommendations that our team has including replacing the omni-wheels with rubberized omni-wheels for increased grip on the ball; and replacing the current low torque motors with higher torque motors in order to eliminate stalling during micro-corrections. The combination of these two hardware replacements will lead to a much more stable balancing robot. In addition to these hardware recommendations our team also recommends possibly replacing the IMU with one that communicates through SPI bus or CAN bus instead of using I2C communication as it is fairly noisy to begin with.

**Financial Summary**

The cost of each different part purchased is shown below in table 1.

Table 1. Parts List with cost.



The total cost of labor is shown below in table 2.

Table 2. Labor Cost.



The total advisor pay is shown below in table 3.

Table 3. Total Advisor Pay.



The total expenditure for the project is shown in table 4.

Table 4. Total Expenditure for Ball Balancing Robot.

