NAIT

Edmonton, Alberta

**3-axis Self-Stabilizing Camera Mount**

As a submission to

Kelly Shepherd, Instructor

English and Communication Department

Submitted by

Ervin Hernandez, Student

CMPE 2960

Computer Engineering Technology

April 29, 2016

8703 24 Ave NW

Edmonton, AB T6K 2W3

April 29, 2016

Mr. Kelly Shepherd

Instructor, English and Communications Department

NAIT

11762 – 106 St NW

Edmonton, AB T5G 2R1

Dear Mr. Shepherd

I am submitting the report 3-axis Self-Stabilizing Camera Mount, as you requested, for your evaluation. This report partially fulfills my obligations towards the requirements of CMPE 2960: Computer Engineering Capstone.

This report details the design and construction of the self-stabilizing camera mount. It goes over the details of how the process works and how it is useful to users and businesses.

I’d like to thank all of my instructors at NAIT for their dedication to work for the success of their students. Their passion for what they teach is a great motivator in the classroom.

Sincerely,

Ervin Hernandez

CNT Student

Contents

[1.0 Introduction 3](#_Toc448396723)

[2.0 Overview 3](#_Toc448396724)

[2.1 History 4](#_Toc448396725)

[3.0 Process 4](#_Toc448396726)

[3.1 Design 4](#_Toc448396727)

[3.1.1 Parts 4](#_Toc448396728)

[3.1.2 Materials 6](#_Toc448396729)

[3.2 Building 7](#_Toc448396730)

[3.2.1 Cases 7](#_Toc448396731)

[3.2.2 L-brackets 9](#_Toc448396732)

[3.2.3 Power Supply 9](#_Toc448396733)

[3.3 Software 9](#_Toc448396734)

[4.0 Finding and Problems 11](#_Toc448396735)

[4.1 Structure 11](#_Toc448396736)

[4.1.1 Scenario #1 - Servo Capability (No Load). 11](#_Toc448396737)

[4.1.2 Scenario #2 - Servo Capabilities (With Load). 12](#_Toc448396738)

[4.1.3 Scenario #3 – Power Supply Test 13](#_Toc448396739)

[4.2 Code Problem 14](#_Toc448396740)

[4.2.1 Scenario #4 – Sensor Reading Accuracy 14](#_Toc448396741)

[5.0 Project Result 15](#_Toc448396742)

[6.0 Conclusion 15](#_Toc448396743)

[References 15](#_Toc448396744)

**List of Figures**

[Figure 1: The IDE (Integrated Development Environment) of the OpenSCAD 7](#_Toc448398572)

[Figure 2: Virtual Rendering of a Servo 8](file:///C:\Users\ervin\Documents\Ervin%20Final%20Report.docx#_Toc448398573)

[Figure 3: Wireframe case 9](#_Toc448398574)

[Figure 4: IMU Wire Hook Up 11](file:///C:\Users\ervin\Documents\Ervin%20Final%20Report.docx#_Toc448398575)

**Abstract**

A camera mount that uses gyroscopes and accelerometers to self-stabilize from any rotational movement along the 3-axes. Parts consist of the mount, three Hitec HS-485HB analog servos, LSM6DS3 IMU Board, which housed the accelerometers and gyroscopes, and an Arduino Mega 2560 microcontroller. Each part of the mount was constructed with the MakerBot Replicator. Then it was assemble where the three servos are stack on top of each other, from bottom to top, Yaw, Roll and Pitch. The data gathered from the sensor, sends data toward the Arduino Mega 2560 Microcontroller. From the Arduino, it uses the data to calculate forces and change of angle from the sensor where it translate it to analog volt for the servos.

# 

**3-axis Self-Stabilizing Camera Mount**

# 1.0 Introduction

As technology grows in today, mimicking human movements become more prevalent with human and technology interaction. One of the ways to mimic human movement is using accelerometers and gyroscope. These sensor can help track rotational movement of a human and translate into data where companies like Oculus and HTC can create the latest VR technologies. The purpose of this report is to walk through the process of designing and building a 3-axes camera mount with accelerometers and gyroscope to self-correct any human rotational movement along the 3-axes. The report will demonstrate the designing and building the prototype, our finding and problem that arise, and how did we solve these problems. Overall this report will reveal the application of a 3-axes camera mount in today market and the future it could hold when implemented correctly.

# 2.0 Overview

The 3-axes self-stabilizing camera mount uses two essential sensors. An accelerometer, which gather data of force applying to the sensor, and a gyroscope, a sensor that reads the change of angle applying to the sensor. Once the sensor gather the data, they send it to the Arduino board which calculate the data and translate the code into analog output where 3 servos, each mounted on different axes, position itself to those coordinates. This technology is used in Military, for plane and ships, the smart phone industry, space exploration and the robotics industry.

## 2.1 History

The concept of using gyroscopes as a way to stabilize an object on an unleveled surface dates back to Ancient Greek, Chinese and Romans. Where they try to stabilize objects on ship to traverse across seas. It was not until the early 19th Centuries where scientist develop tools and reduce the amount of variables for traveling through choppy or fog seas. A French scientist name Fleuriais created the first gyroscope by a top that was continuously blow by air jets which maintain the top stability (Range & Mullins, 2011). By the 1898, Austrian Ludwig Obry patented a gyroscopic inertia which is the base to all modern gyroscope.

# 3.0 Process

## 3.1 Design

My partner, Gabriel Natividad, and I discus ideas of designing the structure and the location of each motor. We debated on several design of the prototype. One design was L shape bracket similar to the camera mount used in the film industry, but the problem was the size and weight of the camera mount. So we discover a design where each of the motor were stack on top of each other. This design make so that the weight is more center on the middle where the camera is rather the motor orbiting around the camera. The feature that was to be implemented were, portability, a tracking capability, and manual control over the camera mount.

### 3.1.1 Parts

Once we agreed on a design we did some research on what type of motor that the camera mount will uses? We narrow it down to two type of motor: stepper, or servos.

#### 3.1.1.1 Steppers

Stepper is an electric motor the uses digital pulses which rotate a mechanical shaft into divided step. Unfortunately the motor require separated pulse for each step. To be able to move the motor by step, the motor require a drive chip to send separate pulses to the motor a step (JColvin91). Not only this could be a problem in maintaining a simple design, the stepper motor are quite heavy. And when introduce in violent rotational movement, the stepper motors could lock up because of the sudden change of movements by an individual.

#### 3.1.1.2 Servos

Servos is a small motor that has a positional shaft. Meaning, this shaft, if receive a specific angular position, the shaft can accurately move to the desire angle. The servos require a coded analog signal to the shaft could rotate to the angle. Not only the servos are small, they are light and could output strong torque depending the amount of amps it consumes. Which is a problem when having multiple servos running at the same time.

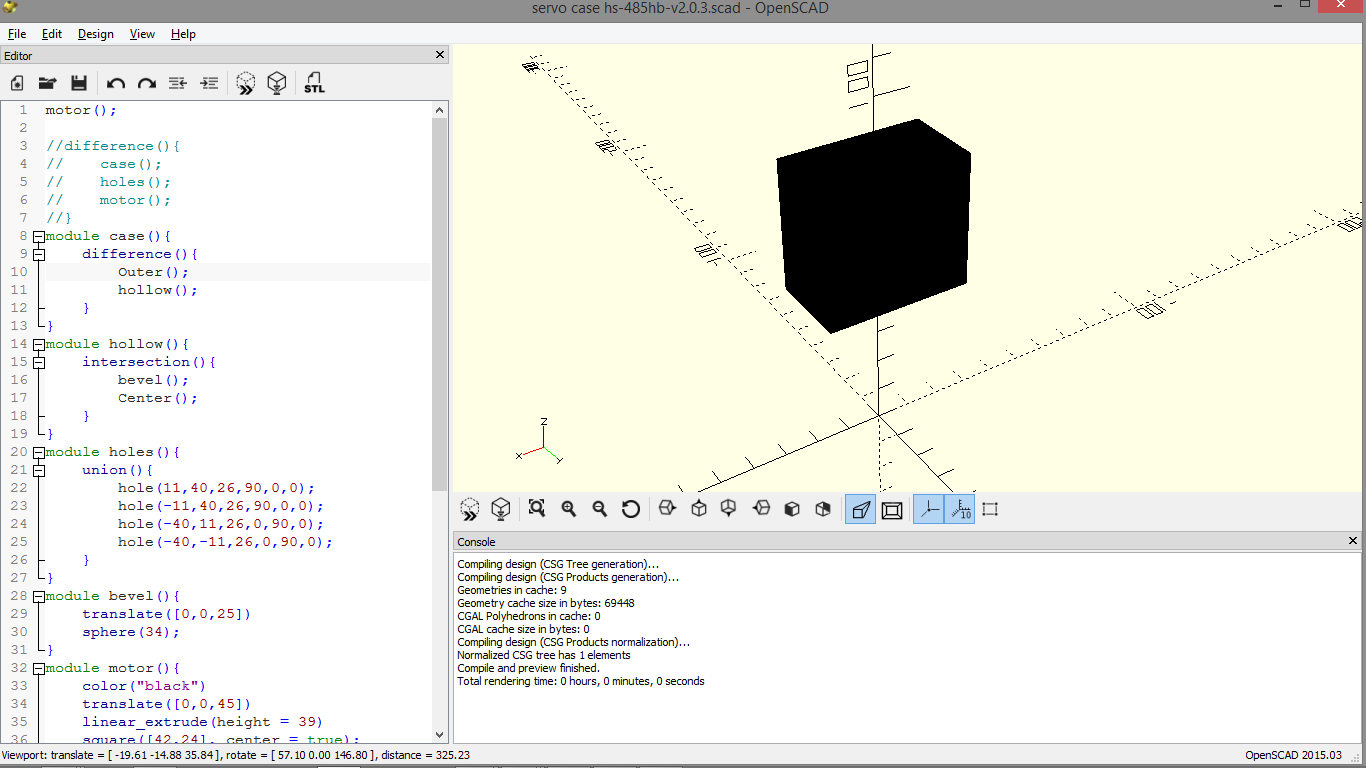
We concluded that the best type of motor for the project was to use servos. Reason is they are light, powerful, cheap and plentiful in Edmonton. The servos we use was Hitec HS-485HB Analog servo motor, which provide plenty of torque and won’t grind the gears out since they were made out of carbonite plastic (Robotzone, LLC, n.d.).

With the servos finalize we search for a sensor that could read force movement and positional movement. Gabriel and I spent at least a couple days researching type of sensor for the project. We concluded on an IMU broad from SparkFun because it has both an accelerometer and gyroscope in one sensor. SparkFun also provided a general code libraries for the accelerometer and gyroscope. Thus it was easy to translate the code they provided into the micro board, which was the next problem.

The micro board we were about to use was the school provide micro board. But we encounter a problem. The board provided by NAIT was using 5.5 volts to all of the sensor on the board. The IMU board that Gabriel and I chose was only 3.3 volts restricted (MTaylor). Unfortunately, we order the IMU board before consulting with our instructor about the board. I spent the next couple of day searching a suitable board that is compatible to the IMU board. The board that was used for the project was the board recovered from a previous project I build a couple years back. I tested the Arduino Mega 2560 to check if there was any defect on the board. The Arduino Mega 2560 has a 5.5 volts supply and a 3.3 volts supply built into the board. The school provide board require a leveler component to re-adjust the power level to 3.3 volt for the IMU board.

### 3.1.2 Materials

Designing the structure for the project was finalize when my partner and I confirm on the type of motors, sensor and micro board for the capstone. I took the dimension and weight of the servos and design a frame for the three servos. During the creation of the prototype, I’ve encounter a problem with the servos. They tend to draw more current from the source when the servos are under heavy load. I conclude that the frames needs to be built from lighter material so it will reduce the amount of current drawn from the servos. After researching I found out that the MakerBot Replicator – a 3d printer provided by our instructor – creates a lightweight objects that uses a plastic filament. The problem was I didn’t have access to a 3d modeling program that could export STL files needed for the 3d printer. So I spent a couple days finding a free open source program that could create a STL file. After a sufficient amount of research, I settle with OpenSCAD (Kintel). I chose OpenSCAD over other programmes because: it’s free, it directed toward a programming style of modeling, and it could export into a STL file. Shown in figure 1 is a sample of the interface of the OpenSCAD when designing the case. Once the parts and general idea of creating the prototype, we began building. I work on designing and building the frame and cases for the servos, and Gabriel worked on the software component.



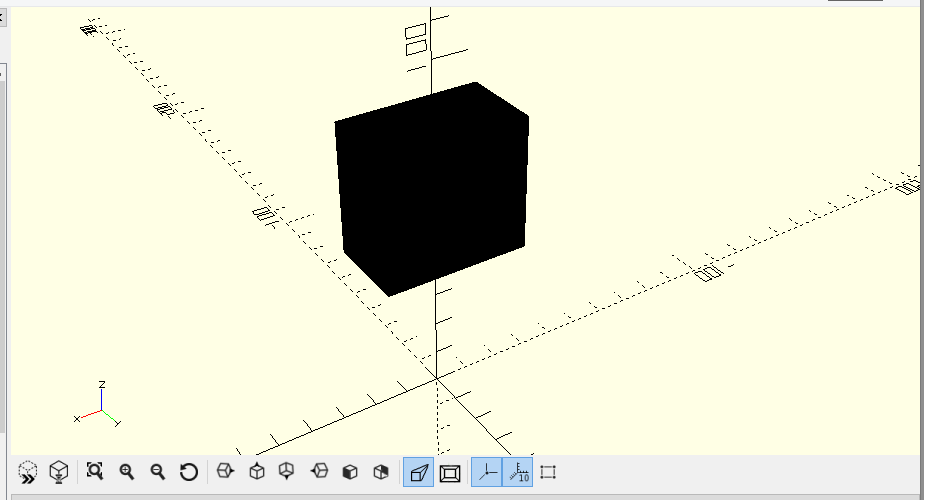
**Figure 1: The IDE (Integrated Development Environment) of the OpenSCAD**

## 3.2 Building

When building the cases of the prototype, design problem arise. How big the case for each servos? How are they being attach? Where to place each part? I tackle each question like I tackle code problems, breaking them into smaller, modular components.

### 3.2.1 Cases

I measure the Hitech HS-485HB Servos and created a virtual version of the servo where I could create a case around the Hitech servo. The black box, demonstrated in the figure 2, represent the general outline or size of the servo.



**Figure 2: Virtual Rendering of a Servo**

Once I created a virtual servo, I built a square block around the servo and adjust so the servo is peeking above the frame. While the servo is above a square block, I hollowed out the square block which became the case for the servo. Unfortunately the wall of the case were too thick. With that I created another object which reduce the thickness of the wall in the case. I use a sphere to reduce the wall thickness because it will bevel the corner in the case. The bevel inside the case, in figure 3, will help to strengthen the structural integrality.

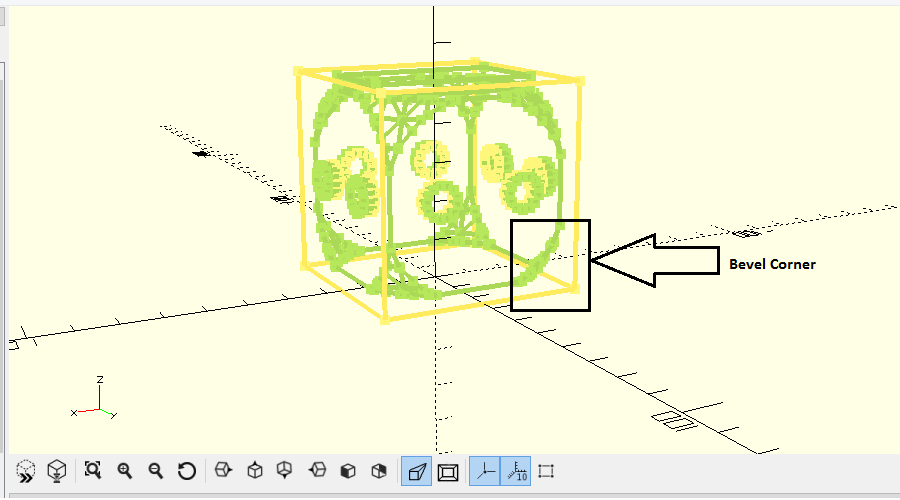


Figure 3: Wireframe case

After getting the correct dimension of the wall, I add two 5mm holes around the faces of the case. The holes around the faces help facet the case into L bracket, which it is also being design later in the process. When I build the prototype, I notice that the frame dimension where larger than the require specification. Even with sanding out some of the hole to be able to fit the servo, it was penetrating into the honeycomb casing of the frame. I conclude that I didn’t account for the size of the filament, thus I adjusted the frame and holes size to compensate for the filament error. The holes on the faces have increase to 8mm from 5mm.

### 3.2.2 L-brackets

Learning about this offset, l apply the offset to all of my design and re-adjust the L-bracket design. The L-bracket was design with two thin square plate with a thickness of 5mm. when creating the screw holes for the L-bracket so they are able to be mounted on the servos aluminum horns, I added a small pit for the screw head can be flush with the surface of the bracket. The reason being was that the screw head won’t obstruct the cases when being mounted. This will help securing the bracket, servos and cases tightly. Again from the cases, I implemented a bevel corner to help strengthen the structural integrality of the brackets. Next I center the top holes of the bracket so that the center of weight of the servos remain near the middle or a line together making it compact.

### 3.2.3 Power Supply

Initially we were implementing an outside source to power the servo the servos. It was successful, until the servo started to slow down because the weight problems. As the prototype continue to run, the servos started to drain more and more power overtime (soshimo). One of the way we try to reduce the power consumption was giving each a specific order from bottom to top. The yaw was at the bottom, the roll was in the center and the pitch was at the top due to the IMU board limitation (MTaylor). We consulted with our instructors if there is any way to avoid the problem. He suggested to use a power supply that could output 10 amps at 5 volts. It was perfect. I solder leads to a power cord so it is able to obtain power from a wall outlet. And hook up the power and ground to the circuit. Once hookup the prototype responded very quickly because of the excess amount of amps provided by the power supply.

## 3.3 Software

While the parts where being built by the MakerBot Replicator, I started setting up and testing the Arduino Mega 2560. I hook up one of the servo to the Arduino and another part hook up to a joystick. With both the Arduino, servo and a joystick, Gabriel was able to build a simple test code to run them. It was a success. With one of the servos connected we attempted to add additional servos to the Arduino. First with an outside power source, which unfortunately gave an irregular current to the servos that gave out jitters during the run. Second, the Arduino powering the three servos. It work, but the Arduino didn’t provide sufficient amount of current to the servos.

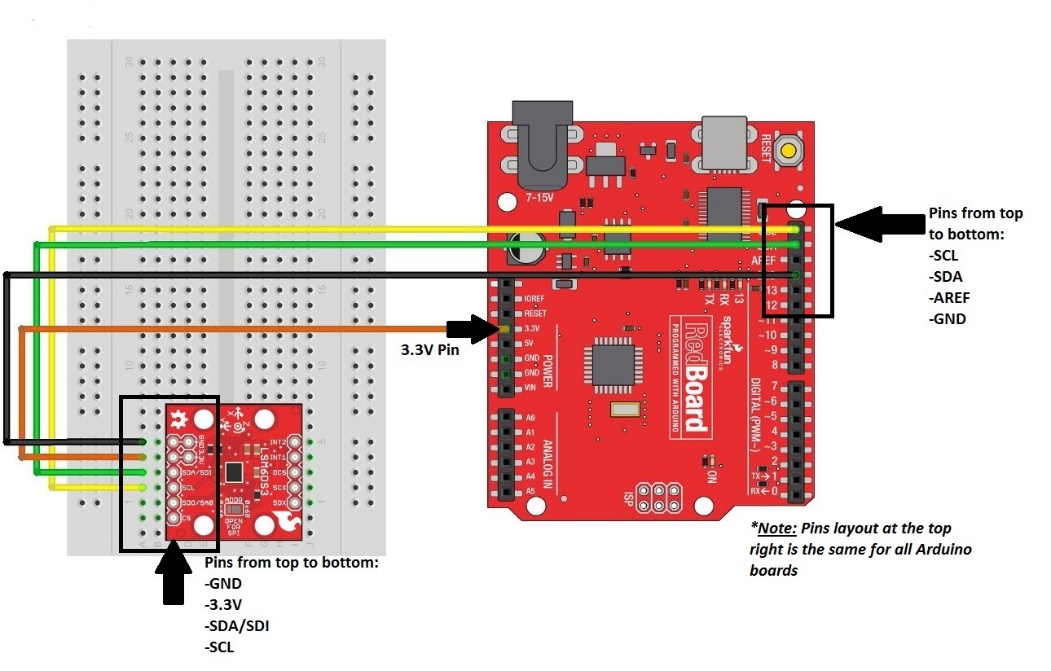


Figure 4: IMU Wire Hook Up

Once the servos were running with the Arduino board. We integrated the IMU board with the Arduino and servos. We hooked up the IMU board via I2C connection (MTaylor). The IMU board have four pins on the left side of the board. From top to bottom, we hook up the ground, 3.3 volts, SDA/SDI and SCL pins to the Arduino, in that same order shown in figure 3. Using the provided libraries, we successfully retrieved data from the IMU board and send it to the Arduino board. Gabriel was able to start creating the code and calculation for the Arduino.

# 4.0 Finding and Problems

Once we were able to assemble the camera mount, we started creating scenarios and tests to help reveal any problems or ways to optimize the prototype. Each scenario have a goal and purpose to test the capabilities of the prototype.

## 4.1 Structure

The first test was the structure and servos capability when put in stressful scenarios. These tests reveal multiple problems about the servos power consumption and the sensitivity of the IMU board.

### 4.1.1 Scenario #1 - Servo Capability (No Load).

The prototype was assembled and was prepared to test the servos without any load. The goal of the test was to observe the voltage and amperage consumption of the servos. There will be two tests: One with the servos stationary on stable ground and the second will be when the servos are programmed to do sweeping movement at angles between 0 and 180 inclusive for a duration of 5 minutes.

When the servos were powered and left idle on flat and stable ground, the servos started to vibrate. The vibrations were minimal at the base but were amplified and more obvious at the upper-half of the prototype. The servos didn’t consume any significant current during the 5 minutes of idling on a stable surface. Unfortunately the vibration from the servos started to affect the IMU board, and given that servo movement is based entirely on the board, the servo vibrations increased.

When the servos were programmed to do sweeps for 5 minutes, the amperage consumption started to increase over time. With each sweeping motion, the multimeter readings indicate that the servo current consumption was around 0.5 A to 1.2 A. This posed a problem concerning runtime because the increase amount of current consumption drained the supply at a faster rate.

After the two tests, we concluded that the problems encountered during the stationary test can be solved with some software adjustments. Unfortunately the sweep tests demonstrated a big flaw in the prototype design. With each sweeping motion, the servos were required to lift a significant amount of weight, specifically the center servo which carries one servo and potentially, the camera. There were multiple suggestions to solve the problem. One was to use a better power supply, one that had a higher amperage rating in order to provide the power the servos needed. Another suggestion was to change the overall structure of the prototype, make it compact and keep the center gravity of each servo motor close to each other. More testing is required to make a final decision

### 4.1.2 Scenario #2 - Servo Capabilities (With Load).

The goal of this test was to observe the voltage and amperage consumption of the servos when it’s required to lift certain amounts of weight. Similar to the previous scenario, there were two tests. Both tests involved the mount carrying a 50g load and a 100g load. The first test had the servo-controlled mount stationary on a stable surface, and the second had the servos do sweeping motions at angles similar to the previous scenario.

When the prototype was stationary with the 50g load on top, there were small vibrations moving across the prototype. The vibrations weren’t as big as the vibrations found in the previous scenario (camera mount subject to no load) so the IMU broad didn’t detect it. The 100g test showed the same result

When the prototype was programmed to sweep for 5 minutes with the 50g load, there were large amounts of power consumption during the test. Similar to the last scenario, as the servos were swinging around, the servos drew more power than expected. The servos consumed around 1 to 1.75 A, with the amperage spiking up during 45-degree angles. At 100g, the load was too heavy for the servos to lift, resulting to complete failure.

After the two tests, the overall outcome was not satisfactory. The stationary test was a great success in that the weight of the load muffled the vibration by a small amount. As soon the prototype was required to move, the prototype failed to meet expectations. Not only was the consumption of current amplified, the 100g load revealed that the servos required more power to lift the load.

### 4.1.3 Scenario #3 – Power Supply Test

The goal of this test was to observe the movements of the prototype with a new power supply that gave 5 volts at 10 amps of current. Similar to the previous scenario, there will be a test when the servos are motionless and a test when the servos are moving. The only difference is there will be a 50g web camera mounted on top of the prototype. No 100g load tests were done given that a 50g webcam can serve as the camera component of the prototype.

When the prototype was stationary on a stable surface, small vibrations reverberated towards the sensor and, similar to the previous scenario, affected the whole prototype. As the vibration continued, the video from the webcam was also affected, resulting to a shaky and jittery video feed.

When the prototype was lifted and rotated around the different axes, the prototype was able to self-correct and minimize the vibrations that were observed at the stationary test. Although if the prototype was held stationary at any angle, the vibrations returned and started affecting the whole system.

When kept at constant movement, the vibrations from the prototype were minimal and negligible. Since the vibrations are due to excess outside force being introduced to the sensor by the motors, a possible fix was to distance the sensor itself from the motors. Another fix was to pad the sensor with a cushion at its base to soften the vibrations coming from the motors. Ultimately, it was decided that a demo for each situation needed to be done. Further tests are required to determine the best possible solution.

## 4.2 Code Problem

The next tests were about the code. Most of the problem with the code was more focus on the calculations and estimation of the servo. Each revision of the code for the prototype did little or no effect on structure or power consumption because it was more related to the positional and coordinates of the IMU board.

### 4.2.1 Scenario #4 – Sensor Reading Accuracy

Test code was written in C programming language to check the accuracy of the sensor readings, both long-term and short-term duration. During earlier test and research the gyroscope reading tend to drift into random values overtime, these tests code will attempt to correct of reveal any signs of removing the drift readings.

While the prototype was running for a minimum of 3 minutes. The sensor retains the correct orientation and was able to correct all – if any rotational movement force on the prototype. Thus remaining accurate.

After it pass the threshold, the z – axis (yaw) started to drift in one direction of an angle. As it prolong itself, the offset drift accumulate into large sums of offset. Thus the servos with the z – axis will product inaccurate data.

We concluded that by trying to extend the code delay of the drift, he was able to extend the threshold to approximate to 4 minutes.

# 5.0 Project Result

In the end, the prototype was able to function well within a span of 5 to 10 minutes. Some of the feature set in the proposal wasn’t really met. We weren’t able to add those feature due the vibrations and instability of the camera, and the yaw drift that accumulated over time. But the fundamental component of the project where completed

# 6.0 Conclusion

Throughout the capstone project my partner and I have learned a lot on how to manage time, deconstructing the project into manageable assignment and maintaining communication and feedback between us. But overall the project was a success. Maybe the accuracy of the yaw wasn’t correct, or the vibration of the servo reverb through the system, or the servo we choose drawn too much amps from the power supply, but these obstacle didn’t deter from the original goal set from the start of the capstone: creating a self-correcting camera mount in any rotational movement.

# References

JColvin91. (n.d.). *How to use a Stepper Motor.* Retrieved from Instructables: http://www.instructables.com/id/How-to-use-a-Stepper-Motor/

Kintel, M. (n.d.). *OpenSCAD The Programmers Solid 3D CAD Modeller.* Retrieved from OpenSCAD documentation: http://www.openscad.org/documentation.html

MTaylor. (n.d.). *LSM6DS3 Breakout Hookup Guide.* Retrieved from SparkFun: https://learn.sparkfun.com/tutorials/lsm6ds3-breakout-hookup-guide?\_ga=1.132588302.1084261190.1452808600

Range, S. K., & Mullins, J. (2011, February 7). *Brief History of Gyroscopes.* Retrieved from NASA Web Archive: https://web.archive.org/web/20150710113230/http://solarsystem.nasa.gov/scitech/display.cfm?ST\_ID=327

Robotzone, LLC. (n.d.). *HS-485HB Servo*. Retrieved from Servo City Web site: https://www.servocity.com/html/hs-485hb\_servo.html#.Vw\_Dp\_krK02

soshimo. (n.d.). *Serial Servo Controller w/Arduino - Control Up To 12 Servos At Once Using the Arduino and a USB Connection.* Retrieved from Instructable: http://www.instructables.com/id/Serial-Servo-Controller-wAduino-Control-Up-To-1/