**Final Project Report**
**MEAM 445/446: Senior Design**
**April 30, 2017**

# Team Stabilize

## Active Camera Stabilization System for Legged Robots

September 2016 – April 2017

Department of Mechanical Engineering & Applied Mechanics
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA

**Team Members:**

Devin Caplin-Munro
decamun@gmail.com

Sean Cohen
seancohen33@gmail.com

Mitch Fogelson
mfogelson630@gmail.com

Langston MacDiarmid
langstonmacdiarmid@gmail.com

Jared Sobel
jsobel9@gmail.com

Ilana Teicher
ilanamichelleteicher@gmail.com

**Faculty Advisor:**

Dr. Bruce Kothmann
Mechanical Engineering and Applied Mechanics, University of Pennsylvania
kothmann@seas.upenn.edu

**Technical Advisor:**

Dr. Aaron M. Johnson
Mechanical Engineering, Carnegie Mellon University
amj1@cmu.edu

## Abstract

The rising trend of autonomy in robotics opens up exciting new applications. However, legged robots have proven more difficult to automate because their jerky motion can cause problems for cameras they carry. RHex is a widely used six legged robot that Professor Aaron Johnson at Carnegie Mellon University hopes to automate. Unfortunately, like most legged robots, RHex’s motion makes it difficult to implement computer vision with onboard cameras. Stabilize provides a mechanical solution to bridge this sensor gap.

Stabilize is a 4 DoF camera stabilizer that integrates non-invasively with existing legged robots. Much like a chicken isolates its head from the motion of its torso, Stabilize seeks to isolate an onboard camera from the motion of the robot. It is also designed to be robust and energy efficient so that it does not limit the distance or terrains the RHex can traverse.

Stabilize uses high precision motor controllers and brushless direct drive motors to actively stabilize and track Roll, Pitch, Yaw (RPY) and Z translation of the sensor. The 3 motors controlling the RPY are axially aligned to reduce power consumption and a spring in parallel with the Z axis motor compensates for gravity and helps to passively stabilize at higher frequencies.

The system is validated by measuring the reduction in maximum rotational and translational velocity of the sensor. This in turn reduces frame loss and motion blur, two key factors in vision performance. Systems like Stabilize will be instrumental in allowing legged robots to work autonomously in chaotic environments.

# Table of Contents

# 1. Executive Summary

The rising trend of autonomy in robotics opens up exciting new applications. However, legged robots have proven more difficult to automate because their jerky motion can cause problems for cameras they carry. RHex is a widely used six legged robot that Professor Aaron Johnson at Carnegie Mellon University hopes to automate. Unfortunately, like most legged robots, RHex's motion makes it difficult to implement computer vision with onboard cameras. Stabilize provides a mechanical solution to bridge this sensor gap.

We set out to design a system the would mitigate the rotational and translational velocities caused by the motion of RHex.  Our customer Professor Aaron Johnson defined our constraints so that we did not affect the dynamics of the robot.  This included a small form factor that could fit on RHex, light weight so that it did not disrupt body motion, able to handle all the disruptive vibrations and able to communicate with RHex.

We came up with Stabilize, a 4 DoF camera stabilizer that integrates non-invasively with existing legged robots, such as the RHex. Much like a chicken isolates its head from the motion of its torso, Stabilize seeks to isolate an onboard camera from the motion of the robot. It is also designed to be robust using materials such as strong aluminum alloys, high strength steel and carbon fiber paneling. Stabilize is energy efficient by utilizing a spring in parallel with the Z axis motor, acting as an isoelastic spring. Stabilize uses high precision motor controllers and brushless direct drive motors to actively stabilize and track Roll, Pitch, Yaw (RPY) and Z translation of the sensor. The 3 motors controlling the RPY are axially aligned to reduce power consumption and a spring in parallel with the Z axis motor compensates for gravity and helps to passively stabilize at higher frequencies.  The position of the camera can be sent via serial to the RHex platform.  In figure 1.1 below, two images, both a side and front view of Stabilize on RHex can be seen.

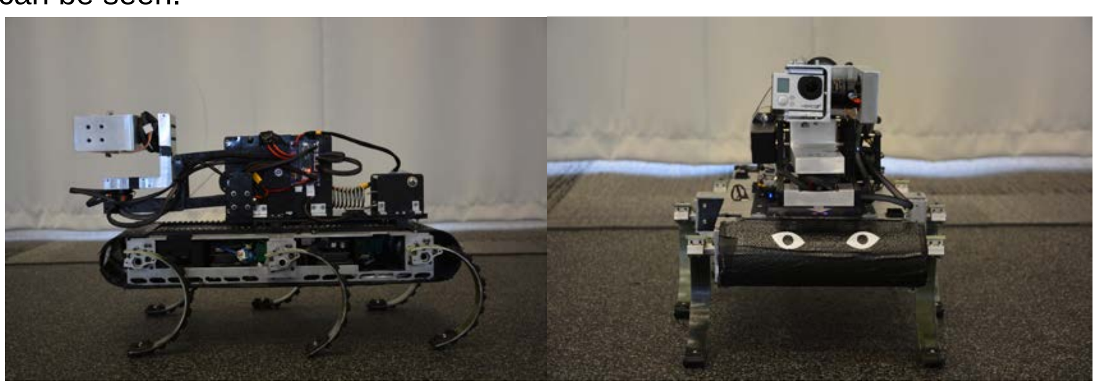


**Figure 1.1: Stabilize System on the RHex Platform**

We tested the system both on the robot and by hand to help mimic the possible environments it would experience when being used by Professor Johnson.  The system is validated by measuring the reduction in maximum rotational and translational velocity of the sensor. This in turn reduces frame loss and motion blur, two key factors in vision performance.

Our initial goals were to reduce these metrics of motion blur and frame loss by 50% and 95% respectively. From testing our system we were able to reduce motion blur by 40% and frame loss by 85%.  Although we came short of our metrics we are confident that our system demonstrates the potential for integrated camera stabilizers. We believe systems like Stabilize will be instrumental in allowing legged robots to work autonomously in chaotic environments.

# 2. Statement of Roles and External Contributions

## 2.1 Team Members

**Devin Caplin-Munro**
*Responsibilities:* Controls, electrical design, and software

**Sean Cohen**
*Responsibilities:* Analysis, manufacturing, and testing

**Mitch Fogelson**
*Responsibilities:* Controls, electrical design, and software

**Langston MacDiarmid**
*Responsibilities:* Z-actuation mechanical design and fabrication

**Jared Sobel**
*Responsibilities:* Manufacturing, project management, and systems architecture

**Ilana Teicher**
*Responsibilities:* Gimbal mechanical design and fabrication

## 2.2 Advisors

*Faculty Advisor*
Dr. Bruce Kothmann
Mechanical Engineering and Applied Mechanics, University of Pennsylvania
*Expertise*: Control Systems
kothmann@seas.upenn.edu

*Technical Advisor*
Dr. Aaron M. Johnson
Mechanical Engineering, Carnegie Mellon University
*Expertise*: Computer Vision and Legged Locomotion
amj1@cmu.edu

*Technical Advisor*
Dr. Matthew Piccoli
Co-Founder IQinetics
*Expertise:* Brushless Motors/Anti-cogging
matt@iqinetics.com

*Technical Advisor*
Dr. Jianbo Shi
Computer and Information Science, University of Pennsylvania
*Expertise*: Computer Vision
jshi@seas.upenn.edu

*Technical Advisor*
Dr. Camillo J. Taylor
Computer and Information Science, University of Pennsylvania
*Expertise*: Computer Vision
cjtaylor@cis.upenn.edu

*Course Professor*
Dr. Graham E. Wabiszewski
Mechanical Engineering and Applied Mechanics, University of Pennsylvania
grahamw@seas.upenn.edu

*Teaching Assistant*
Tyrell McCurbin

## 2.3 Third-Party Hardware/Software

Software:

- Arduino IDE
    - Libraries
        - UM7 by Foster Collins
        - Teensyduino by PJRC
        - SD by Arduino
        - SPI by Arduino
        - MemoryFree by Neil McNeight
        - Bipbuffer by Matthew Piccoli
        - Byte_queue by Matthew Piccoli
        - Communication_interface by Matthew Piccoli
        - Crc_helper by Matthew Piccoli
        - Packet_finder by Matthew Piccoli
        - Generic_interface by Matthew Piccoli
        - Complex_motor_control_client by Matthew Piccoli
- Matlab
    - Computer Vision Systems Toolbox ™ by MathWorks

# 3. Background

## 3.1 Summary of Need

RHex is a hexapedal robot (as seen on the right in Figure 2) created as a DARPA funded project by a consortium of research groups. The robot is able to traverse rough terrain like rocks, mud, and sand using its powerful, independently controlled legs, and is being used to research legged robots, as well as explore exotic terrain.**[1]** Currently, RHex is controlled via remote control, but recent efforts by researchers, such as those at Kod Lab at Penn, include developing features for autonomous motion through visual navigation and obstacle avoidance.

**Figure 3.1: RHex photographed by Kodlab**

Currently, researchers like Professor Aaron Johnson of Carnegie Mellon University are working on visual odometry, in which the robot measures its position and orientation using a visual sensor such as a camera. Objects such as sharp corners are identified and perceived as high frequency objects, whereas curved items are perceived as low frequency items. Visual odometry is especially difficult for the RHex, since as a legged robot its motion has jerky dynamics. The RHex body's motion causes the camera mounted to its back to experience vibrations, distorting the camera's visuals through frame loss and motion blur. After extensive research and consultation with experts in the field, we have not encountered any similar products attempting to solve these issues of legged robot camera stabilization.

### 3.2.1 Frame Loss

Due to the dynamics of legged robots, vision systems can be moved between frames, making it difficult to keep important features within consecutive shots. This loss of visual data due to translation and rotation of the vision system is called frame loss, and can lead to inaccuracies within the context of computer vision. Solutions to this problem include simply buying a camera with a larger frame size or adding multiple cameras and digitally stitching the images together to create a larger frame. In the context of legged locomotion on rough terrains, a single wide angled camera is often not sufficient to solve the problem of frame loss on its own. Through our simulation outlined below, even if a 100 degree, ultra wide angle camera is mounted on the RHex, the camera experiences an estimated frame loss of 16%, far more than Professor Johnson's desired frame loss of 5% or less. Thus, in order to achieve the necessary frame size for this and many other cases of legged locomotion on rough terrain, multiple camera shots must be stitched together. However, this process can be expensive, computationally intense, and most importantly, difficult to execute. A mechanical stabilization system, on the other hand, could reduce frame loss in a simpler way-- stabilize a narrow field of view rather than expand a moving one.

### 3.2.2 Method for Quantifying Frame Loss

In order to quantify frame loss, we considered how the field of view of the camera on RHex would change frame to frame based on its position and orientation over time. In other words, if RHex rotates and translates in some way from time $t_1$ to time $t_2$, frame loss will be the percentage of the camera's frame at time $t_1$ that does not overlaps with the frame at $t_2$.

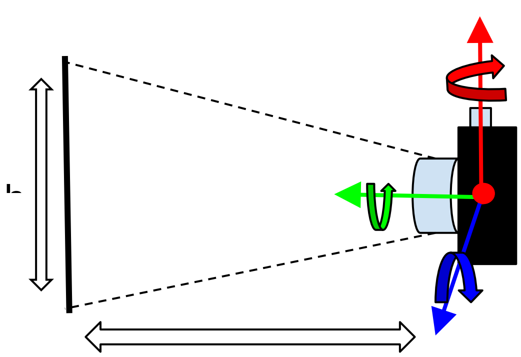


**Figure 3.2: Diagram of camera axis image geometry**

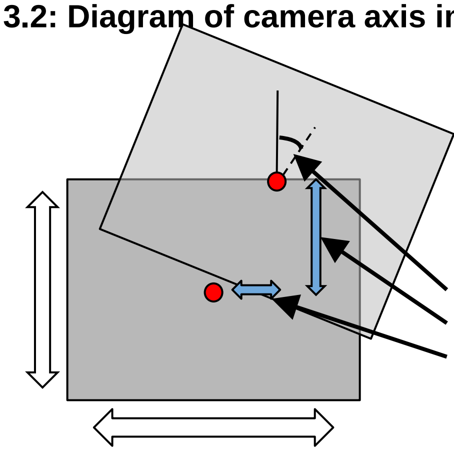


**Figure 3.3: Representation of Frame Loss**

The orientation of the camera's local reference frame, using the standard definitions of roll, pitch, and yaw, is defined in Figure 3.2 above **[2]**. Thus, if the camera rotates in some arbitrary way over a given time interval, the camera's new frame relative to its previous frame will be given by Figure 3.3, also shown above. Usually, the distance from the camera to the object of concern (*d*) is much larger than the camera's own vertical and horizontal field of view (*h* and *w* respectively). Therefore, it is reasonable to assume the small angle approximation, and all lengths can be normalized by *d*. Thus, in this dimensionless space, we derived that the camera frame rotates by the roll angle, translates by the pitch and yaw angles, and has a width and height given by the camera's angular horizontal and vertical field of view, respectively. With this knowledge, the camera's frame can then be divided into a grid of evenly spaced points (with the center of the frame as the origin) and each point can be translated and rotated according to the following transformation equations **[3]**:

$$x_{new} = \theta_{yaw} + x_{old}cos(\theta_{roll}) + y_{old}sin(\theta_{roll})$$ **[Equation 1]**

$$y_{new} = \theta_{pitch} - x_{old}sin(\theta_{roll}) + y_{old}cos(\theta_{roll})$$ **[Equation 2]**

Each point of the new camera frame is then tested to see if it falls within the bounds of the previous frame. Thus, the percentage of these new points that do not fall within the bounds of the previous frame represent the frame loss experienced by the camera during that time interval.

### 3.2.3 Frame Loss on the RHex Platform

Frame loss on the RHex platform was quantified using the above calculations in MATLAB. Using the Qualisys motion tracking system and the RHex's own IMU, position and orientation data was collected during RHex's locomotion on various terrains. This data was subsequently fed into a MATLAB function (Appendix 1) that provided an estimate of the camera's frame loss over time.

According to this analysis, RHex's camera experiences frame losses of various magnitudes depending on the terrain being traversed and the leg speed of RHex's gait. These frame losses are plotted over time for each case in Figures 3.4 through 3.6.

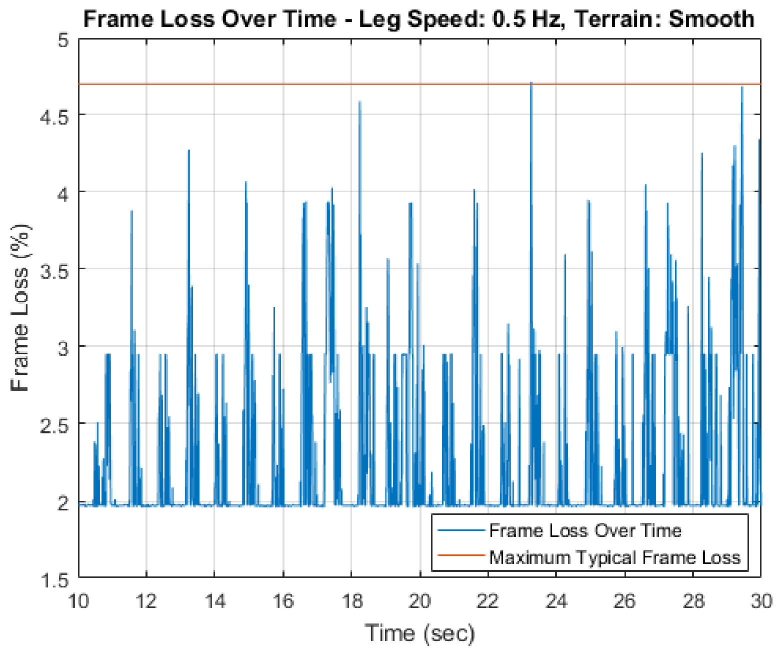



**Figure 3.4: Frame loss over time for 0.5 Hz leg speed and smooth terrain**

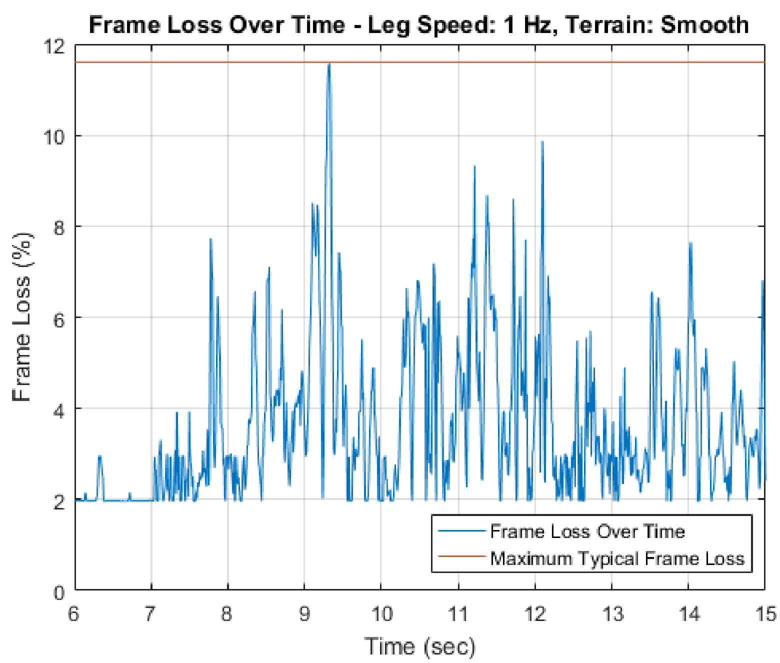



**Figure 3.5: Frame loss over time for 1 Hz leg speed and smooth terrain**

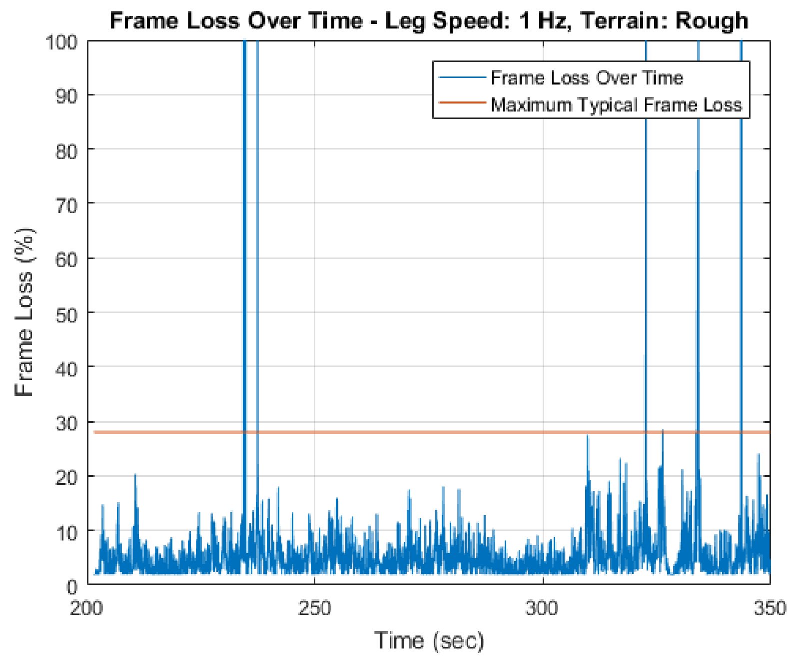



**Figure 3.6: Frame loss over time for 1 Hz leg speed and rough terrain**

Given that on rough terrain, the RHex at times must perform extreme movements such as jumping over obstacles or flipping over, the stabilize system will be mostly concerned with preventing frame loss during typical locomotion of walking and moderate climbing. In fact, as shown by Figure 3.6, these extreme movements can even cause the frame to be entirely lost, a problem that is too difficult to be fixed by the stabilize system. Thus, the primary frame loss metric for the stabilize system will be the maximum *typical* frame loss experienced by RHex, excluding these extreme movements. These maximum typical frame losses are plotted in red Figures 3.4 through 3.6 and summarized below in Table 3.1.

**Table 3.1:** Maximum Typical Frame Loss for Various Terrains and Leg Speeds

| Leg Speed (Hz) | Terrain | Maximum Typical Frame Loss |
|---|---|---|
| 0.5 | Smooth | 4.7% |
| 1.0 | Smooth | 11.6% |
| 1.0 | Rough | 28% |

As shown in the above graphs and tables maximum typical frame loss tends to increase with higher leg speed and rougher terrain. In fact, at the 0.5 Hz leg speed on smooth terrain, the camera requires no stabilization at all, having a maximum typical frame loss below 5%. While this is fortunate, given that the RHex is used primarily on rough outdoor terrains, a stabilizer would no doubt be needed to mitigate the 28% maximum typical frame loss experienced on rough terrain.

As outlined above, frame loss is a central problem for successful computer vision on legged platforms like RHex. By reducing the vibrations experienced by RHex's camera during locomotion, the stabilize system should help RHex and other legged platforms to more successfully identify and interact with objects in their environment; however, due to the complex nature of computer vision, no one factor can be fixed to definitively allow for computer vision applications.

## 3.3 Motion Blur

### 3.3.1 Motion Blur Introduction

In addition to frame loss, "motion blur" is another cause of image distortion that introduces difficulties to visual odometry. Motion blur, a well-known occurrence in the film industry, is the apparent streaking of an object in a still image. This image effect results from an image changing while being recorded by a single exposure. The degree of distortion depends on several factors, such as the velocities of the camera and objects being recorded, shutter speed, and distance of the object from the camera. **[4]**

The distortion from motion blur reduces the signal to noise ratio of the image, meaning it reduces the amount of information the image holds. **[5]** This can be thought of as a reduction in the usable resolution of the image - the blurrier the image, the less usable information it contains. Also, unless the camera is very stable, or otherwise very close to the object of focus, the effect of rotation on motion blur far exceeds the effect of translation.

To compensate for motion blur, computer vision specialists like Professor CJ Taylor of the University of Pennsylvania use different transform algorithms to predict the camera's orientations and vibrations and reverse the effects of motion blur. **[6]** Unfortunately, this method only works when the camera dynamics are easily predicted, and is also computationally intense and an approximation, subject to error. A mechanical stabilization system, on the other hand, could reduce motion blur more effectively by mitigating the vibrations that cause them instead of allowing the vibrations to occur and then attempting to computationally diminish them.

### 3.3.2 Method for Quantifying Motion Blur on the RHex platform

Many metrics exist to determine the amount of motion blur in an image, but it is very difficult to quantify the exact amount of information lost. This is due to the fact that there are many other factors that can affect the information in an image, and estimates of Signal to Noise Ratio are not guaranteed to be accurate.

The motion blur from RHex walking with a GoPro camera causes about 50% reduction in the useful information in the image. This value is found using a measure of complexity called the Fourier Power Spectrum of the image, which indicates how detailed the image is. If the image has more motion blur, it will have less detail, so as RHex walks, the variation in Power Spectrum corresponds to variation in motion blur. **[7]**

We use the following method to calculate Power Spectrum: **[7]**

- Take an image: $I(x, y, 3)$
- Convert to grayscale: $I(x, y)$
- Calculate the Fourier Transform of the Image: $F(x, y)$
- Calculate the Energy of the Fourier Transform using Parseval's Theorem: $P = \sum\sum|\{i * F(i,j), j * F(i,j)\}|^2$

Unlike the Power Spectrum in the aforementioned paper, the transform we use takes into account that higher frequency waves correspond to higher detail in the image than lower frequency waves. **[8]**

For the purposes of visualizing what the Power Spectrum measures, the highest and lowest scoring frames in a five second clip of the RHex robot in motion are shown below in Figures 3.7 and 3.8, respectively:

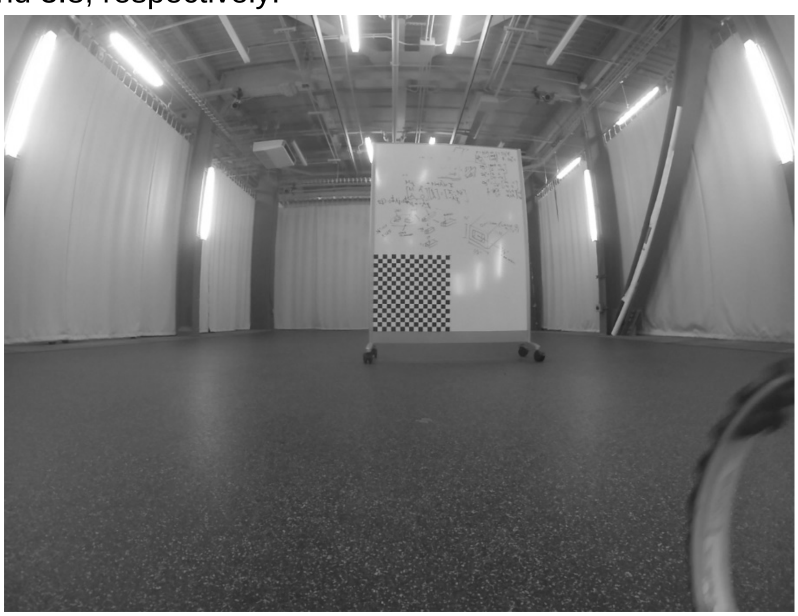


**Figure 3.7: Best Scoring Image: P = 2.17 X $10^8$**

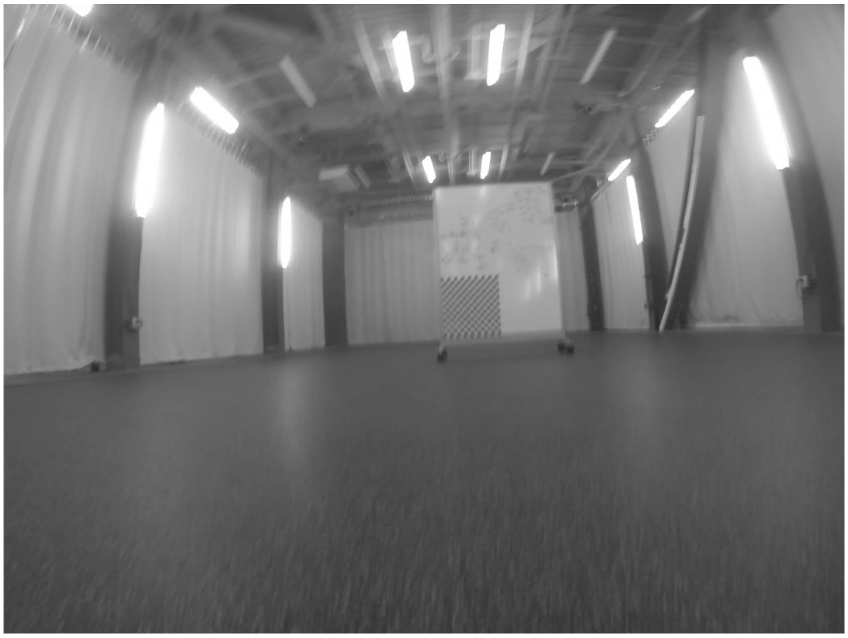


**Figure 3.8: Worst Scoring Image: P = 6.04 X $10^7$**

The Power Spectra for visibly blurred images are lower than those of visibly sharp images. This provides an intuition for the real meaning of the Power Spectrum.

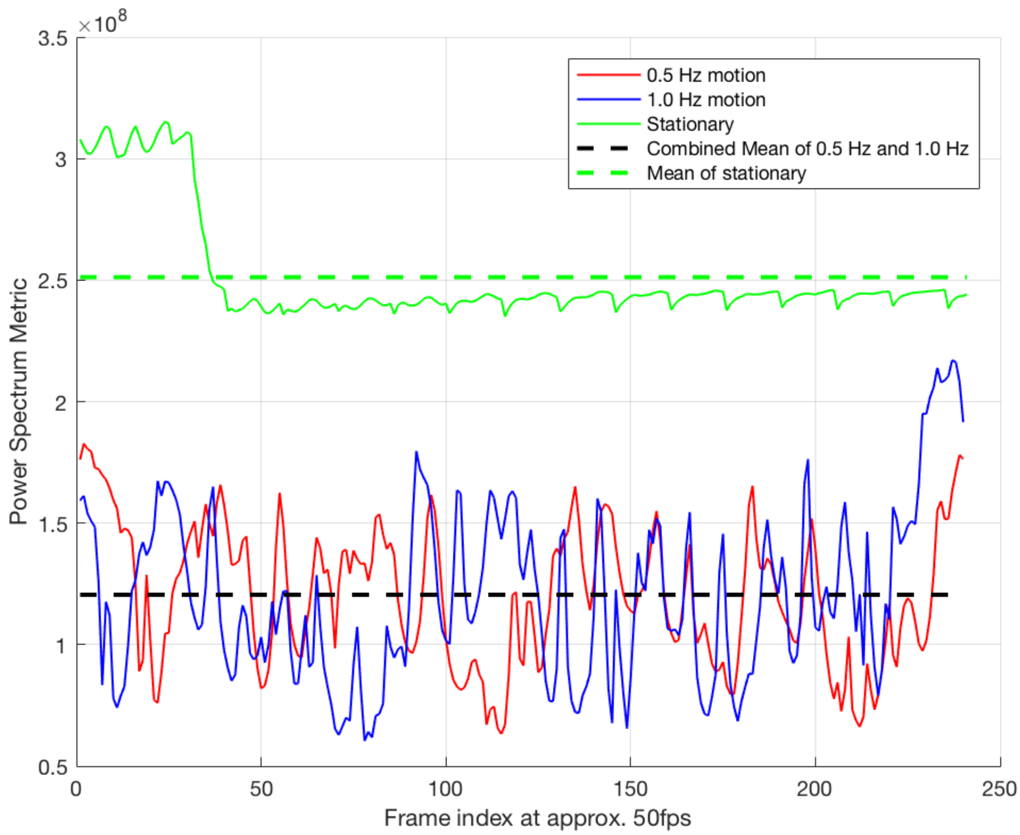



**Figure 3.9: Power Spectrum for Five Seconds of Video (approx. 50 fps)**

Figure 3.9 above shows that the power spectrum plotted by frame is consistently lower for the moving robot than the stationary robot, and also shows clear and regular drops as the robot is jarred by its own gait. In the moving tests, the robot is constantly in motion, so it never reaches stable Power Spectrum levels even between steps. Additionally, the data taken for 1.0Hz and 0.5Hz motion had nearly identical mean values (within 0.005%). This similarity is unexpected, though it may be due to the fact that sample leg frequencies only differ 0.5 Hz.

Most importantly, the above analysis in Figure 3.9 indicates that the RHex gait reduces the usable resolution of the Go-Pro camera by over 50% ($2.512\times10^8$ to $1.205\times10^8$). This is seen in the approximate linear correlation between the Power Spectrum of an image and the information it contains. **[8]**

In other words, a system that stabilized the camera and prevented motion blur could as much as double the usable information yielded as RHex walks. However, as stated previously, due to the complexity of computer vision, any improvement in the native conditions should help RHex accomplish tasks within its environment, but it is impossible to be completely sure.

## 3.4 Existing Solutions

Current solutions that exist in the marketplace consist of off the shelf, three degree-of-freedom gimbals that are generally used for capturing smooth, hand held video. These solutions face two main problems that are essential for visual odometry and other computer vision tasks: position tracking and vertical displacement. Vertical displacement caused by legged robot’s jerky motion is a large cause of frame loss, and any system that does not address this problem is inadequate. In addition, in order to perform visual odometry, the robot needs to know where the camera is in relation to itself. No current gimbals provide this full position feedback.

## 3.5 Expanded Project Reach

Camera isolation to improve vision on the Rhex platform for visual odometry purposes is merely the first step. Motion blur and frame loss pose problems for many other applications in computer vision as well. Our system will allow for legged robots to be able to do SLAM (simultaneous localization and mapping) in more chaotic environments. This stabilization system for high performance vision on legged robots has clear improvements for the future such as decreasing form factor, payload, and response time. The problem of high performance vision on a legged platform has obvious future improvements such as decreasing form factor, payload and response time. Our system also integrates with a variety of other legged robots such as the

“Minitaur” platform by Ghost Robotics **[9]**. Both Professor Shi and Professor Taylor expressed that adding a vision isolation system will allow them to be able to do more complex behaviors with vision on quadrotors. The ability to increase the useful information from cameras will always be an issue in robotics and will allow for many more interesting behaviors and applications of robots in the future.

## 4. Objectives

Our objectives were determined by our customer Professor Aaron Johnson at Carnegie Mellon University **[10]**:

| **Objective** | **Metric** |
|---|---|
| Form Factor | Should Fit within the Rhex platform *(540mm x 390mm x 127mm)* |
| Camera Support | Should support camera's of GoPro form factor *(Camera that Professor Aaron Johnson is using in his application)* |
| Response Directions | Should correct for rotations in Roll, Pitch and Yaw and translations in the Z direction |
| Frequency Response | Should be able to correct for 0.1 - 10 Hz vibrations in Roll, Pitch, Yaw and Z direction *(Based on vibrations experienced by RHex platform)* |
| Communications | Should be able to communicate to RHex platform over serial |
| Mounting | Should be able to integrate with picatinny rails on RHex platform |
| Weight | Should be under the RHex platforms max payload of 8kg |
| Miscellaneous | Should have rugged design to mimic RHex platform |
| Motion Blur | 50% Reductions from initial testing on RHex platform *(Educated guess to perform application by Professor Aaron Johnson)* |
| Average Frame Loss | 95% Reductions from initial testing on RHex platform *(Educated guess to perform application by Professor Aaron Johnson)* |

# 5. Design and Realization

## 5.1 System Level Concept

As discussed in the objectives, our goal was to create a four-degree of freedom camera stabilization system for legged robots. To approach this problem, we decided to design a 3DOF gimbal to control roll, pitch, and yaw, an isoelastic spring system to control the vertical displacement, and an adjustable camera holder for GoPro or similar cameras.

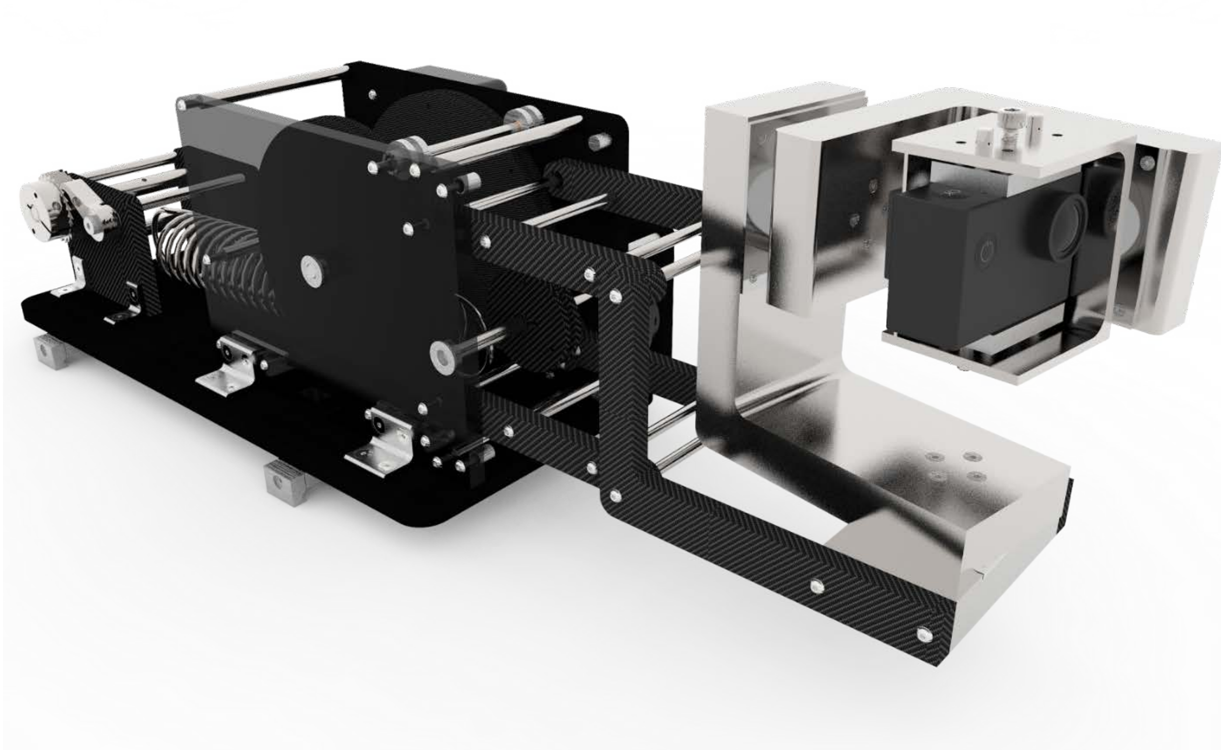


**Figure 5.1: Entire Stabilize System (Rendering)**

In brief, an EMax GB4006 gimbal motor with IQnetics motor controller embedded in a hollow aluminum arm controlled each of the gimbal's three degrees of freedom. The gimbal was attached to the z actuation system, which controlled vertical displacement both actively and passively by a geared-down isoelastic spring and motor. A slightly larger GB4008 motor used position data from an attached IMU[1] and a PID algorithm to stabilize the entire gimbal vertically. Keeping vertical acceleration to a minimum also made it easier for the gimbal to remain steady by minimizing inertial torques about the roll and pitch motor axes.

[1] Inertial Measurement Unit

## 5.2 Design Constraints

Designing a camera stabilization system for legged robot platforms like the RHex and Minitaur using Emax GB4006 motors introduced multiple constraints to the system. As the RHex robot can flip 180 upside down, the stabilization system needed to be less than 5 inches tall, the height of the legs above the surface of the robot. Regarding weight, the RHex can carry an **8** kg load, the Minitaur can carry a **2.25** kg load, and the motors can only manipulate torques up to **.125** Newton-meters, so the entire system was required to be under **2.25** kg, and provide less than **.125** Nm of torque on the motors. Furthermore, the connections between the different motors and the camera mount must be rigid, so that the motors can accurately control the camera's position without interference from bending or vibrations.

In addition, the rugged environments in which legged robots voyage require that the stabilization system be strong enough to withstand impact from branches, rocks, and other obstacles in the environment. The system must also be closed so that sand, dirt, or other contaminants cannot enter and interfere with the electronics, and the electronics (motors, wiring, etc.) must be fully encased.

As robotics researchers frequently use GoPro cameras for their portability and ease of use, the system should be able to accommodate a variety of existing GoPro sized cameras, with depths ranging from 21-36.1 mm, heights from 37.9-44.4 mm, and widths of 37.9-61.7 mm. **[11]**

Lastly, our system must be within our budget, manufacturable, and easy to assemble and disassemble for uncomplicated testing and iteration.

## 5.3 Electronics/Software

Stabilize utilized new technologies that allowed for simpler implementation while maintaining fast processing and high accuracy. Stabilize used a MCU, four motors and an IMU. Below are a series of decision matrices and explanations about the chosen hardware.

***Table 5.1: MCU Decision Matrix***

| Controller | Speed | Communication | Other Notes | Cost |
|---|---|---|---|---|
| Arduino Uno | 16 MHz | Only 1 serial TX and RX | - Have experience with it<br>- Very easy to work with<br>- Doesn't support all C++ | $25 |
| Raspberry pi zero | 900MHz | 1 UART but has USB that can be a UART as well | - Some experience<br>- Lots of developer support<br>- Very small | $5 |
| Arduino Mega | 16Mhz | 4 UART ports | - Have experience with it<br>- Very easy to work with<br>- Doesn't support all C++<br>- Quite large | $46 |
| BeagleBone Black | 1GHz | 6 UART ports | - Plug and play (pre-installed debian)<br>- Has onboard flash memory<br>- Somewhat large | $55 |
| Teensy 3.6 | 180 MHz | 6 UART and 4 I2C and 2 SPI | - Uses the Arduino IDE<br>- Lots of documentation and pre-existing libraries | $29.25 |

The teensy 3.6 microcontroller is a development friendly board using the Arduino IDE. The primary reason the teensy 3.6 was decided upon, was the six serial communication lines and ease of use due to experience with Arduino. This was important for our design because of the need for two way communication with four motor controllers and two IMUs without having to delegate communication lines and ability to prototype quickly.

**Figure 5.2: The teensy 3.6 MCU**

***Table 5.2: Motor Decision Matrix*** **[12]**

| Type | Pro's | Cons |
|---|---|---|
| Servo | - Cheep<br>- Light weight<br>- can have high torque at times | - NOT BACK DRIVABLE EASILY, SLOW REACTION TIME. |
| Brushed | - Cheap<br>- Easy to use | - Can't get accurate position data.<br>- Not backdrivable with gear box easily. |
| Brushless | - Fast reaction<br>- Smooth movement<br>- Back driveable<br>- All new technology and research uses them | - Requires separate microcontroller and sensor board to be able to work |

It seemed apparent that to be successful in this project since brushless motors are used in most commercial and all high end gimbals. However, brushless motors need individual motor controllers so below is another decision matrix on which motor controller to use.

***Table 5.3: Motor Controller Decision Matrix*** **[12, 13]**

| Motor Controllers | Pros | Cons |
|---|---|---|
| Gimbal | - Cheep<br>- Integrated<br>- Easy to use | - Cannot receive position feedback<br>- Inefficient |
| Ghost Robotics | - Smooth transitions for low speeds<br>- Current control for torque output<br>- Position sensing<br>- Serial communication<br>- High bandwidth<br>- Unlimited warranty and support<br>- Can perform at higher power level | - Expensive<br>- Has a lot of useless firmware that we are paying for<br>- Large form factor |
| IQnetics | - Small<br>- Cheap<br>- Anti cogging<br>- Smooth transitions for low speeds<br>- Current control for torque output<br>- Position sensing<br>- Serial communication | - Can't handle powerful motors<br>- Can't communicate with all boards, needs a controller capable of multiple serial communication lines |

For our motors, stabilize investigated a new motor controller company founded by a UPenn alumna Matthew Piccioli, IQinetics. These motor controllers were designed to have simple communication and integrated controls, theoretically allowing the brushless motors to be controlled as easily as a stepper motors. However, due to their early development they came with a variety of bugs. We utilized the IQinetics motor package which came with a brushless DC gimbal motor and the IQinetics motor controller. The IQinetics motor controller has a high precision encoder and integrated position or torque control. Standard gimbal controllers do not use position control, making them not effective for our application, and other motor controllers with similar features were more expensive. For our application we needed high resolution for accurate rotations within one revolution, with which the IQinetics controller was very successful.

**Figure 5.3: IQinetics Motor Module**

The UM7 is a high precision IMU with an integrated Extended Kalman Filter, which combines the information from all the sensors and process the information allowing for accurate heading and orientation.  We decided upon this IMU due to a recommendation from Foster Collins, a former MEAM student at UPenn.


**Figure 5.4: Pololu UM7 IMU**

Wire routing was also significant since the gimbal is designed to have 360 degrees of rotation and fully concealed cabling within the linkages. 28 AWG stranded ribbon cable was used on the motors and IMUs for flexibility, small form factor and cleanliness of routing. Due to the low current draw of the 3 motors controlling the gimbal as well as the IMU we could use a very low gauge wire, however for the Z actuator motor 24 AWG wire was used due to the higher power. For the power connector and switch we used 20AWG stranded cable. Cable socks were then added to help with improving strength of the cable and maintaining cleanliness of routing.

Finally, we designed a PCB to allow for a cleaner aesthetic and smaller package. We used EAGLE a circuit designing software to design the board and then OSHPark to have the boards manufactured.

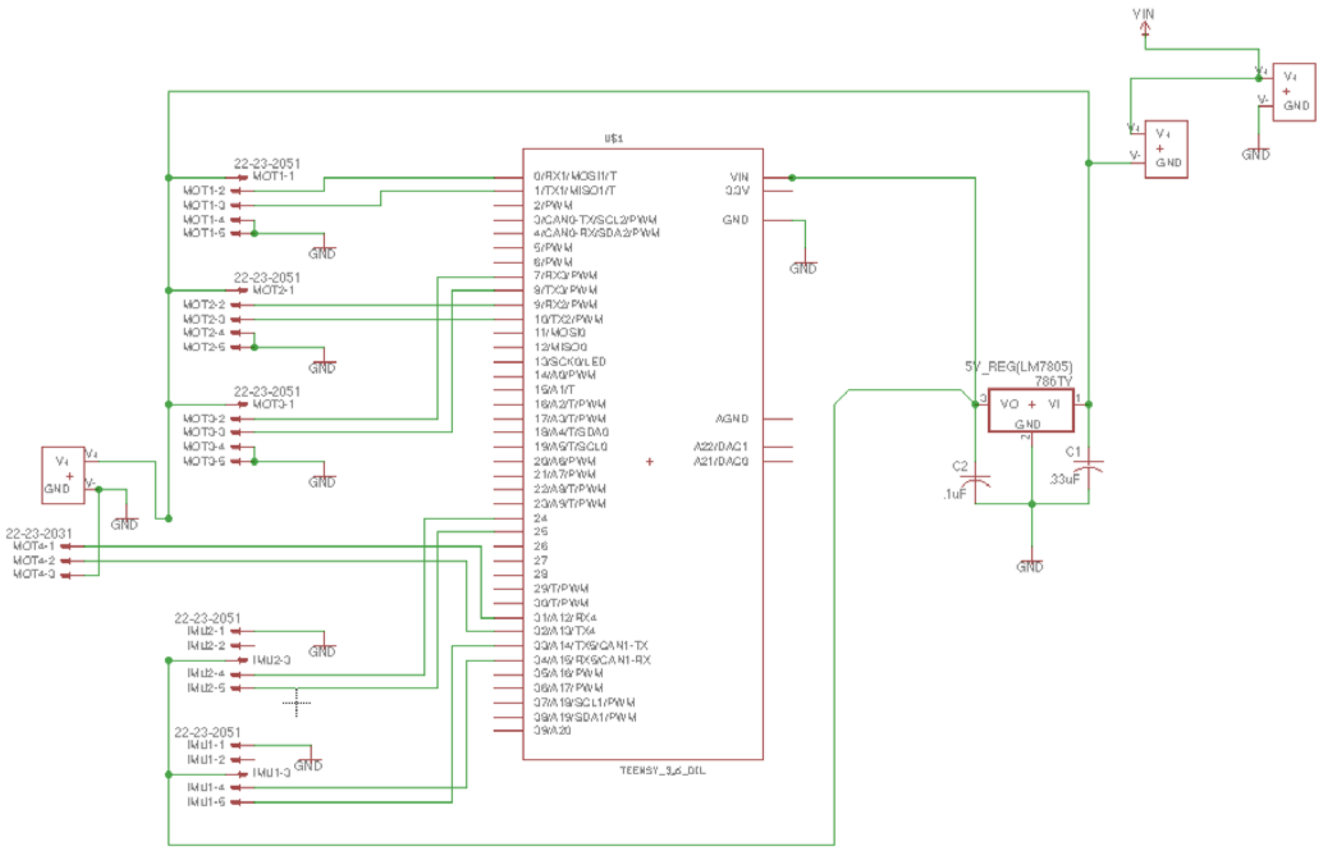


**Figure 5.5: Circuit Diagram**

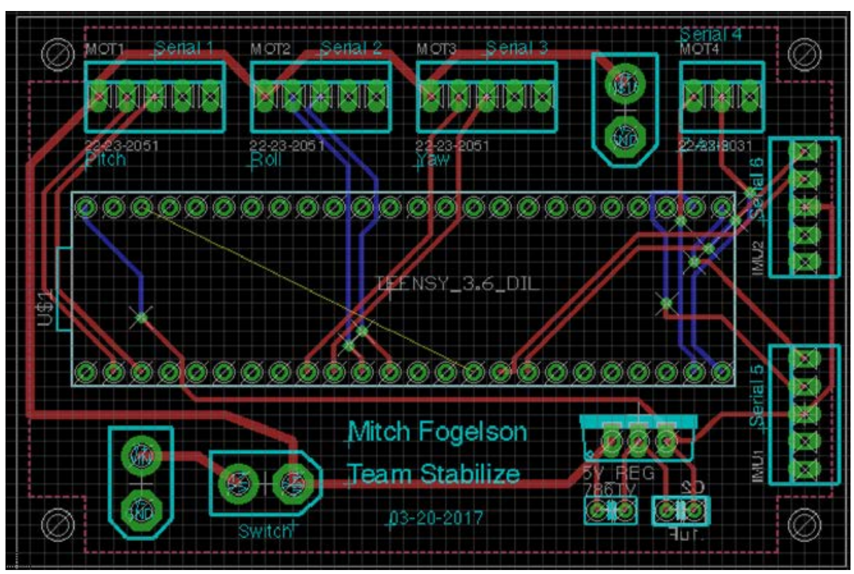


**Figure 5.6: PCB Traces Diagram**

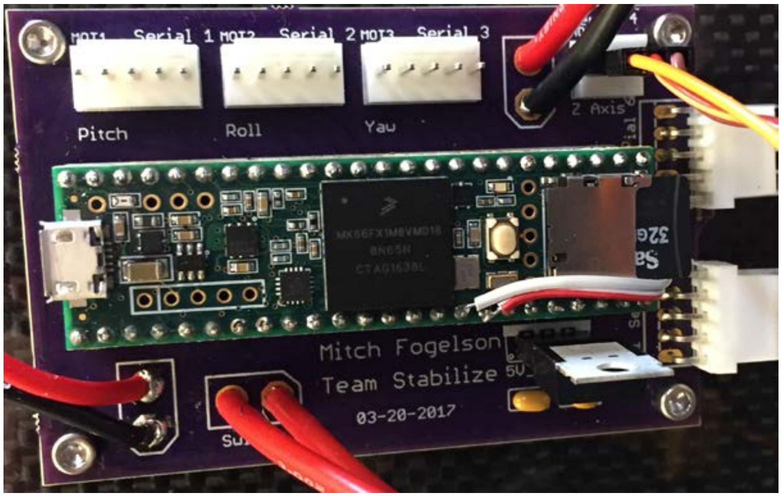



**Figure 5.7: Stabilize PCB**

**Software:**

The Teensy 3.6 can be programmed using the standard Arduino IDE with added support packages available on the website www.pjrc.com. All software for Stabilize is written in Arduino.

The Stabilize software performs four main functions:

1. Reading translation and rotation estimates from IMU
2. Calculating compensation angles (controls and kinematics)
3. Communicating with motors
4. Logging motor positions for odometry analysis.

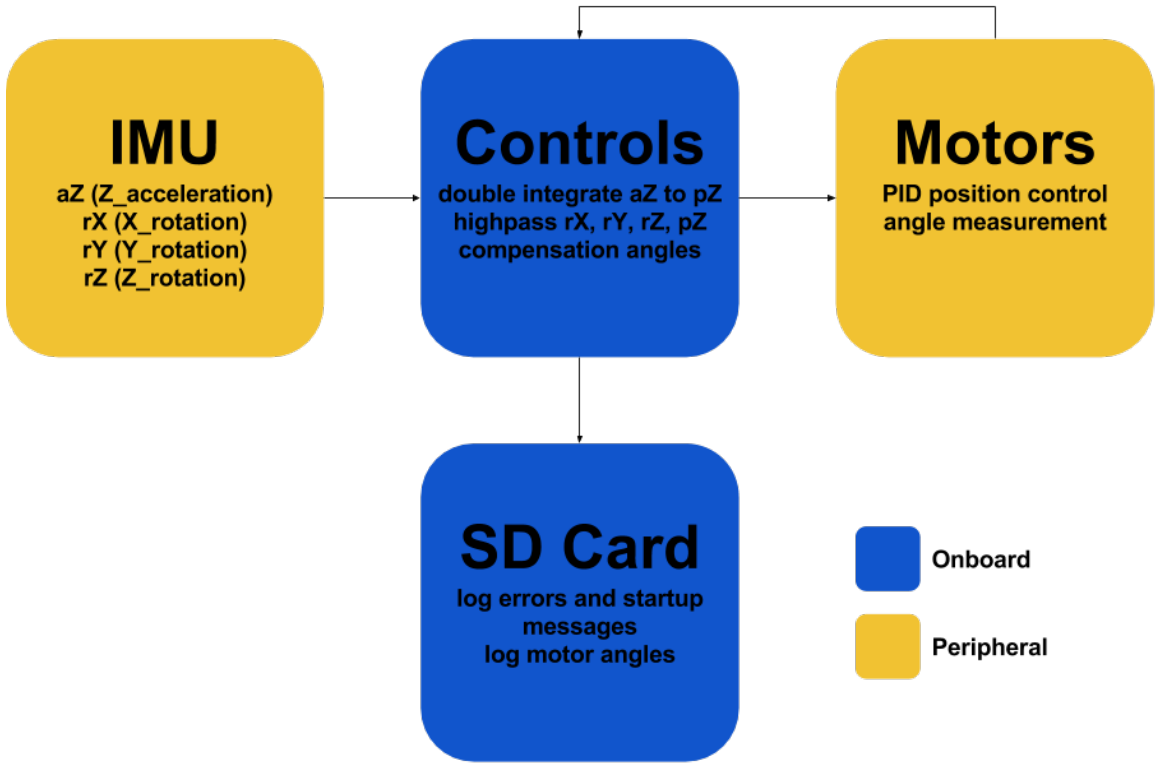



**Figure 5.8: Code Flow**

In order to interface with the UM7 IMU we used an open source library called UM7-Arduino, further modified by Penn student Foster Collins during work on ScubAssist. The UM7 library reads estimated euler angles Roll, Pitch, Yaw (e.g X, Y, Z). These values are the output of an internal Extended Kalman Filter on the UM7. Thanks to Foster's modifications, the library is also able to read the filtered acceleration estimates in X, Y, and Z.

It is important to note that the UM7 must be calibrated using the Redshift UM7 calibration software before it outputs accurate values.

Once read, these values are used to calculate the compensation angles to be sent to the motors. High pass filters with relatively low frequency cutoffs between 0.01 and 0.25 are applied to the euler angles in order to ensure that the camera tracks with the robot as it turns. These filters also mitigate drift over time due to sensor noise and filter inaccuracy. Frequency cutoffs can be changed to suit the nature and task of the robot. High pass filtered Euler angles with the proper sign change are assigned to corresponding gimbal motors.

The Z acceleration is numerically integrated twice in order to find a position estimate, which is filtered similarly to insure proper tracking with the robot and mitigate drift over time. In addition, the controller calibrates the Z acceleration estimate using a weighted average if the first 10 values acquired from the IMU. Z position is converted to an angle by dividing by the length of the linkage arm attached to the Z motor.

Angle commands are sent to the motors using a modified version of the standard IQinetics motor communication library. Because the IQinetics communication library was in development concurrently with our project, the library we used is somewhat out of date with the currently published library, and contains some bugfixes made to our individual codebase by Matthew Piccioli. It is important to note that actual PID control is done by the motor controllers, and our controls code only needs to set the PID gains at startup and send angle commands.

The Teensy syncs with each motor in turn during each controls loop. Syncing with a motor sends all commands, including parameter updates and command angles, and reads measured position and other metrics from the motor.

These measured position values are recorded to the onboard microSD card using an open source library developed by SparkFun. Values are saved in CSV format including millisecond timestamp so that they can easily be imported to Excel or Matlab. A new file is created each time the system starts up.

Our software contains a known bug that causes the Teensy 3.6 to crash and become unresponsive at irregular intervals. The exact cause of the problem is unknown, although it appears to come from the motor communication library. In the limited time available to us, we fixed this problem using the Teensy’s built in Watchdog timer. The timer must be activated using system flags. **[14]** Once activated, the timer will automatically reboot the Teensy after about 3 seconds unless it is reset. We also modified our code to reset the timer after every loop, so that if it ever crashed and stopped looping, the watchdog would cause the system to reboot. The full code base can be found in Appendix E.

## 5.4 Gimbal Design

### 5.4.1 Gimbal Design Decisions

Based on the weight and strength requirements, our system needed to be made of a material that was low density (light), and strong (stiff and non-brittle). Since the system would need to resist breaking upon impact, we opted for a metal. Upon comparing different options, seen below, and keeping manufacturability and cost in mind, we chose Aluminum 7075 T651, whose strength is comparable to steel (Ultimate Tensile Strength of 510–540 MPa, and Yield Strength of at least 430–480 MPa) and a density of 2.810 g/cm³.

**Table 5.4:** Strength to weight ratio of various alloys under consideration **[15]**

| Alloy | Temper | Strength to Weight Ratio |
|---|---|---|
| 7075 | T6 | 148.3870968 |
| 7075 | T651 | 179.0035587 |
| 7075 | T7351 | 124.516129 |
| 6061 | T6511 | 100 |
| 6061 | T6 | 100 |
| Titanium | Grade 5 | 187.1040724 |
| Titanium | Grade 2 | 60.97560976 |

The gimbal arms, which connect the camera mount to the motors, were also made of Aluminum 7075 T651. Since the motors and wiring needed to be fully enclosed, the arms were hollow. Being hollow has the added advantage of decreasing weight. After evaluating similar hollow aluminum parts shown by Pete Szczesniak of the Precision Machining Laboratory at the University of Pennsylvania, a wall thickness of 0.050 inches was decided upon, as this would be rigid such that the motors could accurately control the position of the camera, as well as minimize weight and size.

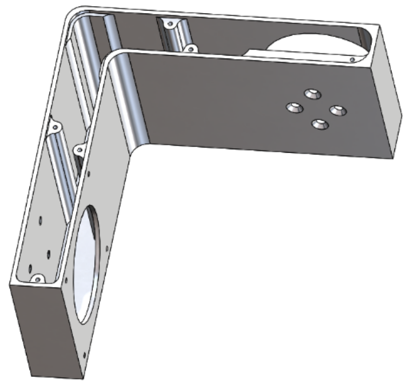


**Figure 5.9: Gimbal arm that connects pitch and yaw motors**

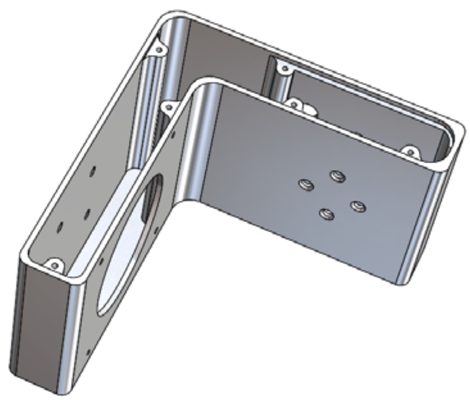


**Figure 5.10: Gimbal arm that connects the roll and pitch motors**

The minimum height of the entire gimbal system is thus mainly defined by the size of the camera mount and the motor assemblies, with the arms adding 0.250 inches as well as 0.10 inches of clearance in total. Unfortunately, the size of the motors, and the adjustable camera mount (which will be discussed further on) were too large, such that the gimbal would not fall under the height requirement. We opted to maintain our design and prioritize performance, and justify the concept of a 4DOF camera stabilization system, and in the future version work to reduce height.

The depth of the arms was defined by the diameter of the motors plus the wiring, which would be wrapped around the motors with additional slack so that there would not be friction while rotating. The placement of the motors along the arms was decided by aligning them with the center of mass of the camera. The overall size of the arms was as small as possible, and the general shape of the arms connecting the roll and pitch motors, and the pitch and yaw motors, were right angles. The inner corners were filleted with a 0.125” fillet so that a quarter-inch end-mill could cut the parts.

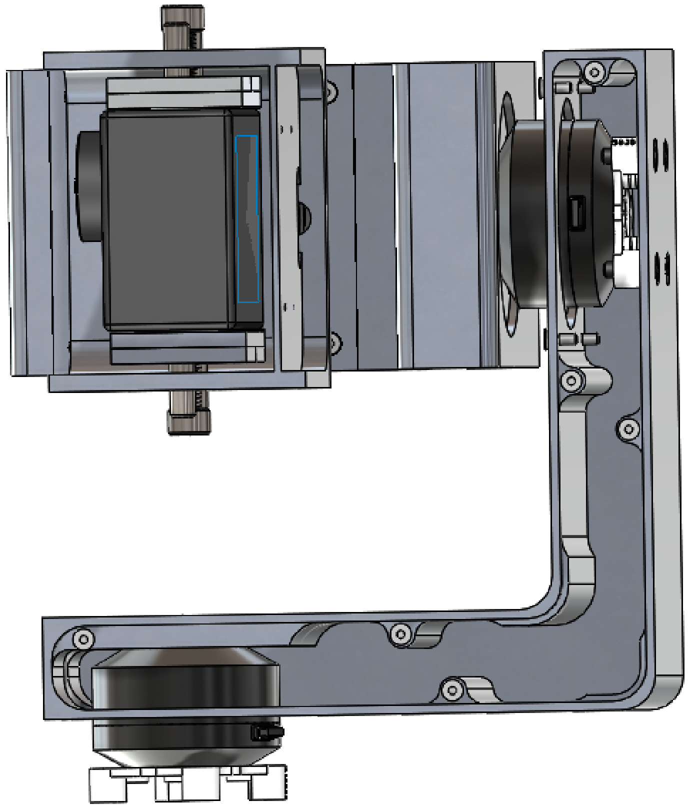


**Figure 5.11: Side view of gimbal: motors embedded within arm**

The hollow arms had 0.050” thick covers that would embed into the arms to completely close the arms and encase the electronics. The covers had 0.001” of tolerance between their walls and the arms, and were sealed down with M2 screws, since a press-fit was determined to not be strong enough to keep everything enclosed in the case of impact. Flathead M2 screws were used, as the head height is 0.0472 inches, such that it would fit flush in the 0.050” thick cover.

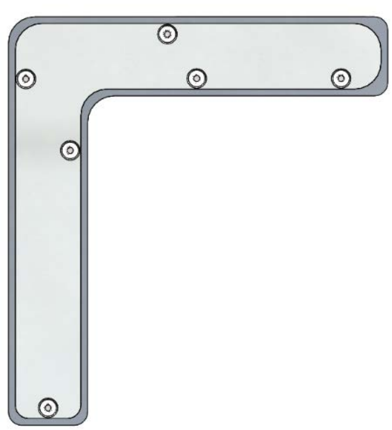


**Figure 5.12: Gimbal Arm with Cover**

The hollow arms and covers allow access to the wiring and motors, for easier assembly and wrapping of the wires. In addition, Allen key-sized holes opposed the motor screws, so that the gimbal could be assembled.

Aluminum ring-plates were also used to enclose the electronics—plates screwed down onto the arms, and the hollow ring enclosed the motor and surrounding wires, leaving room for the wires to wrap and unwrap without introducing friction. The top of these parts embedded into a cylindrical depression in the next arm, so that contaminants could not get in, while have 0.002” of clearance between the cylinders so that there would be no friction upon rotation, and 0.005” of clearance between the top of the ring and the following arm, also to remove the possibility of friction.

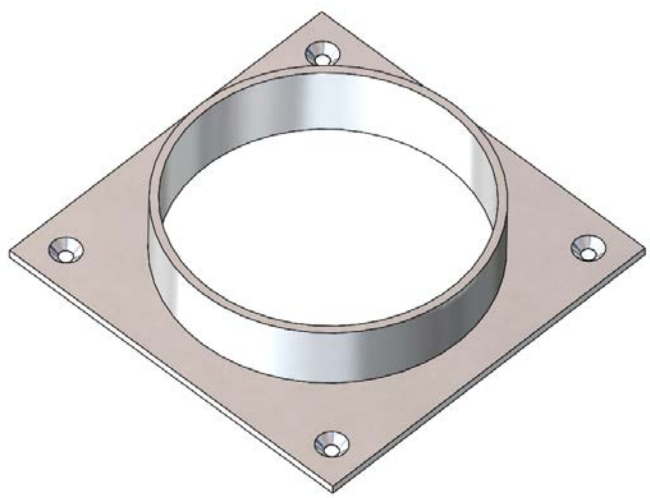


**Figure 5.13: Ring Plate**

The camera mount needed to be able to fit different sizes of GoPro cameras, keeping the camera clamped and unmoving, as well as in line with the motor axes (each of the three sides aligned with roll, pitch, and yaw). In addition, the mount should be adjustable by hand for easier assembly. Thus, a vise-style mechanism was decided upon. A general C shaped holder would attach to the roll motor and contain the camera and camera clamping system. On two opposite sides of the holder, holes would be tapped for thumbscrews, which could be screwed by hand for tool-free camera mounting. These screws would be constrained inside plates such that rotating the screws would cause the screws to rotate freely inside the plates, but translate the plates as they screwed down the walls of the holder. These plates would be held parallel with dowel rods press-fit into the plates that are slip-fit into the holder. 1/16” thick rubber padding was epoxied onto the plates to add friction so that the camera would not move while clamped.

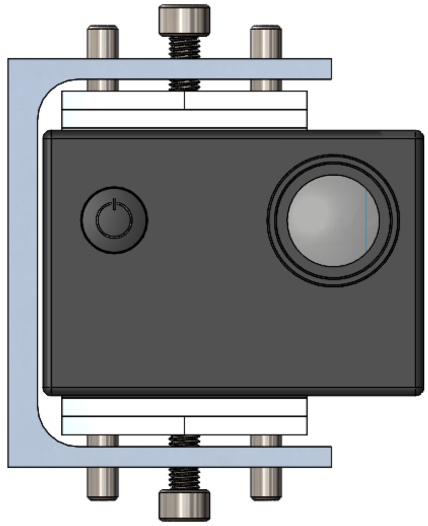


**Figure 5.14: Camera mount system**

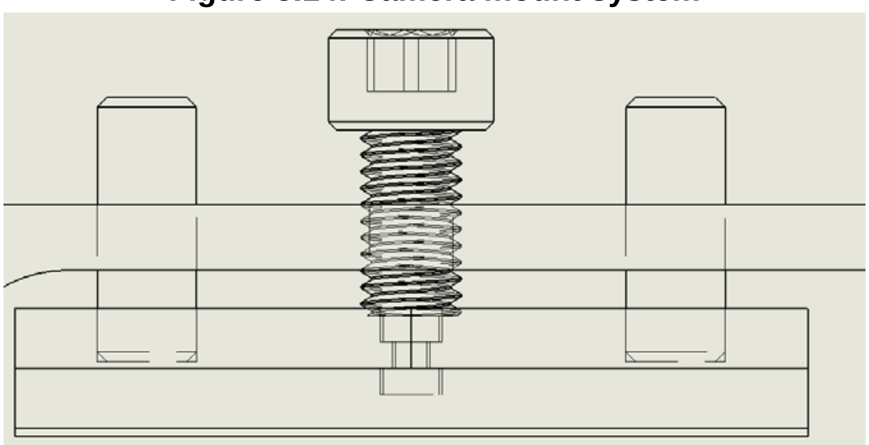


**Figure 5.15: Line Drawing of Vise-style Clamp, showing the mechanism for translation and parallelism**

### 5.4.2 Gimbal Prototyping

After sketching the system and making the above design decisions, the system was designed on SolidWorks with solid arms. The CAD assembly was mated with realistic mates, to test for interferences between parts.

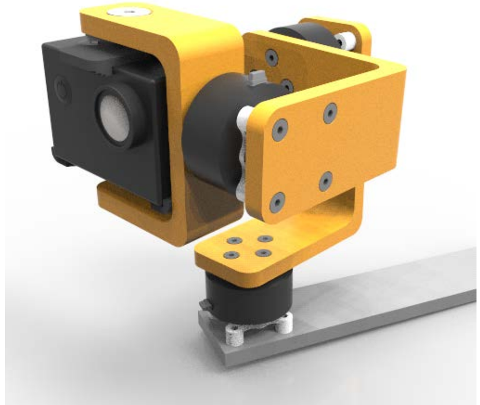


**Figure 5.16: Rendering of first CAD of gimbal with solid arms**

The first prototype was 3D printed on the Makerbot 5, which simply included a rigid mount for the camera and a solid arm connecting the roll and pitch motors. This setup was attached to a laser-cut acrylic board with handles, so that the electronics team could test motor capabilities of controlling position and controlling 2 degrees of freedom simultaneously. This prototype informed on fit of camera for future iterations, as well as the need for larger clearance between holder and the camera.


**Figure 5.17: Second Gimbal Prototype**

The second prototype was 3D printed out of ULTEM on the Fortus, with 0.005" tolerance. This prototype included a static camera holder, and the two hollow arms that connected the roll and pitch, and the pitch and yaw motors, with their covers. The ring-plates were printed on the Makerbot 5. This prototype was used to test interferences, if there were room for wires to wrap the motors, and the ability to and ease of assembling the system. This prototype informed the need for wider openings for the wire-surrounded motors, and confirmed that the gimbal worked and could be assembled without difficulty.  It also showed that a plastic version of this system would require ribs and gussets to increase its strength while maintaining extremely thin walls, as this plastic version bent under the weight of the attached motors and camera.


**Figure 5.18: Manufactured Gimbal and Camera Holder**

### 5.4.3 Gimbal Manufacturing and Assembly

The arms were machined from a 6” x 6” x 2.5” block of 7075 T6 Aluminum. First, the block was squared up on the Prototrak mill. Next, the vertical band saw was used to cut the L-shapes of the arms. The rough sides of the arm that connected roll and pitch motors were squared on the Prototrak. The dimensions of this stock was inputted into SolidWorks CAM, and g-code was created to cut the arms from this oversized stock on the Haas Mini-Mill. A half inch end-mill with 3 inches of cutting length and a quarter inch end-mill with 1.5 inches of cutting length were used to cut the inner and outer profiles, and a center drill and drill bit were used to cut the holes that would later be tapped for the M2 screws to attach the cover. A quarter inch of excess stock remained on the bottom of the arms for the vise to hold onto, and was later removed by a face mill in the Prototrak. The arms were flipped multiple times for postoperative processes on the Prototrak for threading, milling, and drilling various holes.

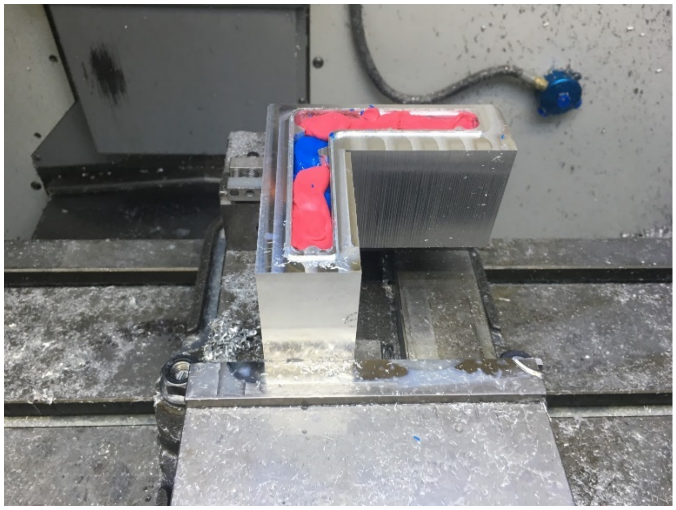


**Figure 5.19: Arm during outer profile machining on the Mini-Mill, filled with clay to minimize vibrations**

The covers were machined by attaching 6” x 6” x 0.065” blocks of Aluminum 7075 T6 to fixtures, and using Prototrak software to cut the correct height of the covers and then drill and chamfer the holes for the M2 screws. The plates were then clamped to the fixture using these screws, and the outer profiles of the covers were cut using Prototrak software.

Using the same stock as the arms, the remaining stock was squared up for the camera holder. In a similar process to the arms, the camera holder’s outer and inner profiles were milled on the Haas Mini-Mill, and extra stock was left on the bottom to later be faced off. In addition to the C-shape discussed above, a plate with holes for an IMU was milled so that the holder was shaped like four sides of a hollow cube, with a hole for the wiring to travel to the roll motor, and be encased by a rubber tube. The extra stock was removed on the Prototrak with the face mill, and the holder was flipped multiple times for post-operation processes on the Prototrak for threading, milling, and drilling various holes.


**Figure 5.20: Finished camera holder**

The black-oxide, 10-32, low-profile, knurled thumbscrews for the camera mount were bought and post-processed. These screws were turned on the manual lathe, so that the ends were circular (not threaded), and of various diameters so that they would remain constrained in the plates.

The plates were machined in three parts on the Prototrak using Aluminum 6061, as these plates would not be as susceptible to damage from the environment. The bottom piece was a solid, squared up block with threaded holes to attach it to the other parts of the plate, and a through-hole for the end of the thumbscrew. The top pieces were each solid, squared up blocks with a half circle cut on the side edge, of two different diameters, to attach around the thumbscrews and hold them in place, and through holes and chamfers for screws to attach these upper plates to the lower plates.

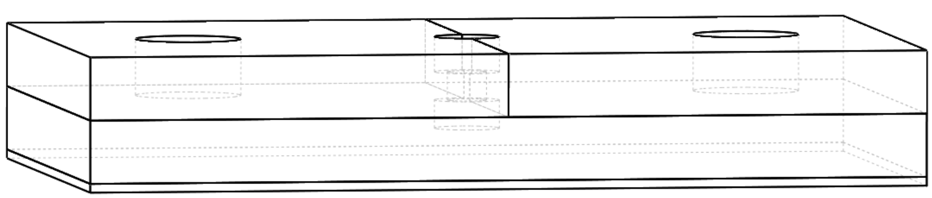


**Figure 5.21: Drawing of three part plates, inner geometry visible**

Lastly, the ring-plates were machined from 7075 T6 Aluminum on the Prototrak. First, blocks were squared up of the outside dimensions of the parts. Holes were drilled in the center, and the stock was attached with these holes to a fixture and the outer profile of the ring was cut. The parts' outer holes were then cut and attached to the fixture using these holes, and the inner profiles of the rings were cut. The gimbal bill of materials are found in Appendix A and the gimbal engineering drawings are found in Appendix B.

### 5.4.4 Gimbal Future Design Improvements

After building the gimbal system, integrating the circuitry and controls, and combining it with the Z actuation system, we came out with a few important alterations for future iterations. First, the motor mounts that contained the motor controllers broke, these were parts that came with the motor module and not manufactured ourselves.  I Second, because of the precision of the motor controllers, the IMU on the camera mount was unnecessary, and its panel could be removed, thus saving material and manufacturing time with a smaller, less complicated part. Third, we learned that the motors struggled with the torque of the system, and needed to be aligned along the center of mass of the system so that there would be no added torque on them, and not along the center of mass of the camera. Thus, the arms lengths would need to be redesigned such that the motors were along the center of mass of the parts that each would manipulate. Fourth, the wiring used was thicker than expected, and the rings caused the wiring to bind. The system could be tested with thinner wires that were unshielded, or the rings could be made larger and the arm width adjusted respectively. Lastly, the height of the gimbal was above our maximum, meaning that if the RHex flipped, the gimbal would impact the ground. To correct for this, smaller gimbal motors could be tested to see if they would provide the adequate torques for our system. Another fix for this could be swappable camera holders for the different cameras, so that there would not be added height on the camera mount.

## 5.5 Z-Actuation

### 5.5.1 Z-Actuation: Design Decisions

As previously mentioned, unlike existing gimbals used in the camera industry, the Stabilize system was designed to provide stabilization not only in the angular roll, pitch, and yaw directions, but also linearly in the vertical, or z, direction. In order to provide this extra degree of freedom, many system level designs were considered.

Three primary options were considered for z-actuation: a linearly actuated system, a cantilevered system, and an isoelastic spring system. Linear actuators, though intuitive, had several drawbacks: speed, cost, size, and latency. Most linear actuators move slowly, and those that move quickly are either expensive or imprecise. Because the design of a linear actuator requires enclosure of the arm, they tend to be more than twice as long as their stroke length, which was not advantageous for our space constraints. Finally, because linear actuators contain a worm gear, there is inherently latency between movement of the motor and movement of the arm. Because we aimed to design a transparent controls system, this was also undesirable. Because of these drawbacks, we decided against a linearly actuated system.

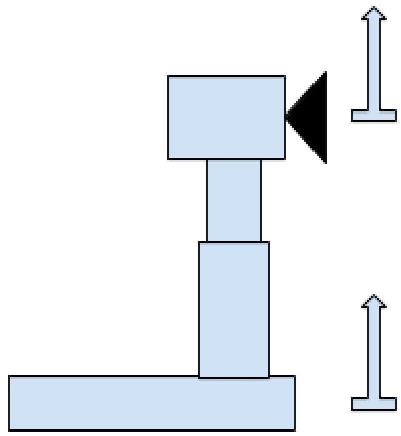


**Figure 5.22: Sketch of Linear Actuator System**

Next, a cantilevered design was considered. Here, the camera and gimbal system would be supported by a short cantilevered arm that would pivot at the body of the robot and thus move the camera vertically in z. As long as the arm's angle of rotation remained relatively small (within approximately plus or minus 30 degrees as to satisfy the small angle approximation), the system would remain nearly linear in nature, greatly simplifying active control via the z-axis motor. This design also had the added benefit of efficiently utilizing vertical space, allowing the system to have adequate z-axis movement without exceeding the vertical space constraint. A sketch of this system can be seen below.

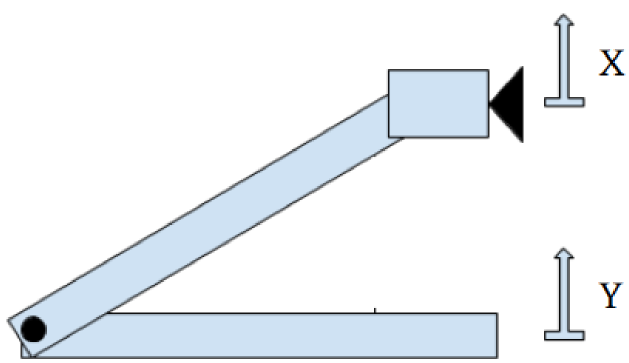



**Figure 5.23: Sketch of Cantilevered System**

However, the primary drawback of this cantilevered system was that it required a high nominal torque from the motor. Given that the entire weight of the camera and gimbal system was supported at a single pivot, even if the system as a whole were at rest, the motor would have to exert a high torque simply to counteract the force of gravity. Thus, even if the z-axis motor were properly sized to handle these torques, the vast majority of the electrical power going to the motor would be dedicated not to actively stabilizing the system, but rather to simply fighting gravity, quickly draining the onboard power supply.

Thus, in order to make this cantilevered design feasible, an isoelastic spring system was added to counteract the gravitational force on the camera and gimbal. A strategy commonly used in current mechanical camera stabilization systems, isoelastic spring systems allow the weight of the camera to be supported by a tensioned spring that, via a gear or pulley system, does not change substantially in length when the system moves. Given that a spring's force depends nearly linearly on its length, this results in a nearly constant force output, regardless of the system's position. Thus, if this nearly constant force is equivalent to the weight of the camera and gimbal system, then the actively stabilizing motor can operate as though it were in approximately zero gravity conditions. This both reduces the amount of electrical power needed by the motor and allows a lower torque motor to be used. A sketch of this system can be seen below.

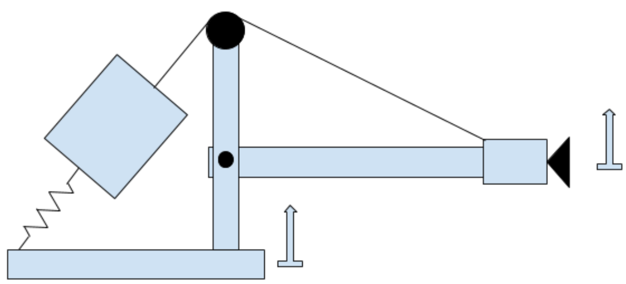


**Figure 5.24: Sketch of Isoelastic System**

Furthermore, the isoelastic spring system has the added benefit of not only counteracting gravity, but also passively stabilizing the system at higher frequencies. Due to vibrations concepts that will be discussed in detail in section 5.5.2, adding a passive spring system reduces the amplitude of high frequency vibrations experienced by the system, allowing the motor to focus on stabilizing mainly lower frequency vibrations.

Therefore, in order to effectively stabilize in the z-direction, the Stabilize system used a cantilevered design in combination with an isoelastic spring system. In this design, the camera and gimbal system is supported by a cantilevered arm, being actively stabilized by a motor at the arm's pivot and passively stabilized by a nearly constant force spring that supports the system's weight.

**5.5.2 Z-Actuation: Optimization of Isoelastic Spring System**

Once it was clear that the z-actuation system required both active stabilization via the z-axis motor and passive stabilization via the isoelastic spring system, the task arose for optimizing the system's response via design and spring selection. This mainly involved using central vibrations concepts to determine an optimal ideal system and then combining these concepts with space and material constraints to converge on an optimal design for our system.

As mentioned previously, the isoelastic spring system serves the dual purpose of both counteracting gravity as well as passively stabilizing the system. Thus, consider a simple cantilevered beam supported directly by an ideal linear spring of stiffness constant k with minimal damping, depicted below in Figure 5.25.

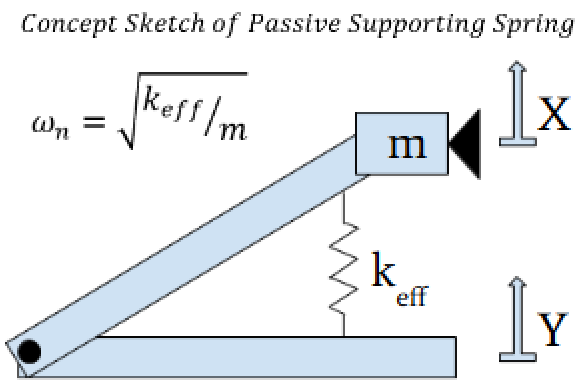



**Figure 5.25: Sketch of Ideal Spring System**

In order to counteract gravity nearly constantly across all ranges of the system's motion, the spring must have a relatively low stiffness constant. By definition, the restoring force provided by the spring is linearly related to its displacement by the stiffness constant k. Thus, if k were large, as the cantilever moved positions, the force provided by the spring would change dramatically. This would defeat the purpose of an isoelastic spring as the spring would provide a force far greater than gravity at some positions and far less than gravity at others.

Similarly, given the springs other purpose of passively damping the system, it is again clear that a small stiffness constant k is optimal. Consider the Steady State Displacement Transmissibility Plot shown below in Figure 5.26.

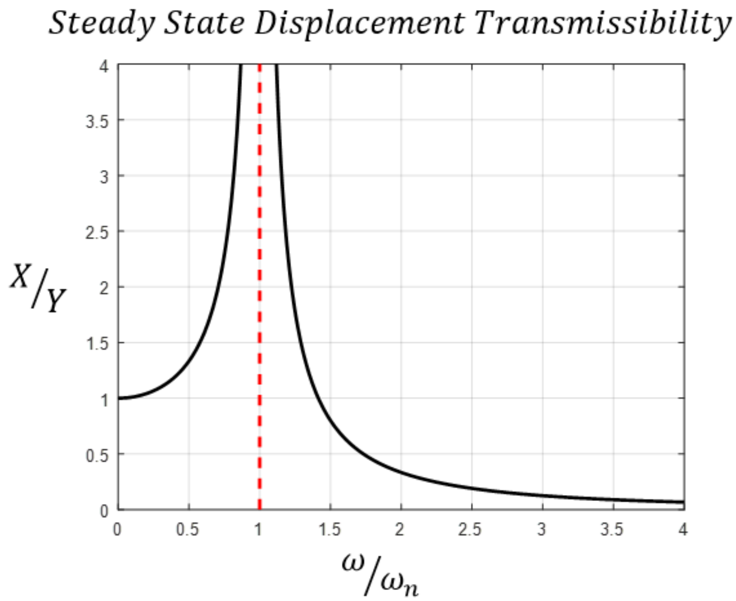



**Figure 5.26: Displacement Response of Lightly Damped System Across Various Frequencies**

This dimensionless plot shows the steady state amplitude response of the system at various input frequencies. A value of one on the x axis corresponds to when the system is shaken at its natural frequency while a value of one on the y axis corresponds to equivalent output and input amplitudes. As one would expect, near the natural frequency of the system, the ratio of X to Y explodes as the system experiences resonance. However, at frequencies far larger than the natural frequency, the opposite effect occurs, with X/Y approaching zero as frequency increases. This corresponds to nearly complete stabilization at these frequencies given that the output amplitude is much smaller than the input amplitude in steady state. Thus, if the system has a much smaller natural frequency than any input frequency it experiences, the system will be nearly entirely passively stabilized. Natural frequency is given by the square root of spring stiffness over mass, where mass is fixed by the weight of the gimbal and camera system. Thus, once again a smaller k value is optimal as the system is more passively stable if the spring stiffness is lower.

The next step in optimization of the spring system thus became taking these lessons from vibrations analysis and best approximating ideal conditions given material and space constraints. From a materials perspective, the primary metric for spring selection is the ratio between yield stress and Young's Modulus. As will be shown in section 5.5.3, a spring can be geared up to reduce its effective spring constant, although this also results in a larger nominal load that it must bear without plastically deforming. Thus, a stiffer material can be a better material choice than a less stiff material if the more stiff material can be geared up to a far higher load without plastically deforming.

Out of the materials generally used for mass produced metal springs, steel has the highest yield stress to Young's Modulus ratio, making it the best material choice for conventional, mass produced springs.

The next design constraint was the limited space with which the spring could operate. For a given material, a longer spring can have a lower stiffness and bear the same maximum load without plastically deforming. Thus, a very long spring could serve as a perfect isoelastic spring as it would be capable to bear large loads while also having a relatively low k value. Practically, however, space on board a legged robot is limited and very long springs cannot be used. Thus, for the Stabilize system, the isoelastic spring needed to remain within 12 inches when tensioned during use.

Applying these space and material constraints, we selected an off-the-shelf stainless steel spring from Lee Springs with a rest length of 5.5 inches and a maximum load of 37 pounds with a stiffness of 4.89 lbs/in. With a safety factor of 1.5, this was the commercially available spring with the lowest k value that fell within the system's space constraints. If the spring were customized, a potentially more optimal spring shape and length could be used. However, after getting numerous quotes from spring providers, the cost appeared be about 10 times larger and the k value only about 5 percent better.

This final optimized spring resulted in a gear ratio of 1:7 (as will be discussed in detail in Section 5.5.3) for the pulley system and gave the isoelastic spring as a whole a natural frequency of approximately 0.9 Hz. Given our space, material, and budget constraints, this was the lowest achievable natural frequency.

### 5.5.3 Z-Actuation Prototyping

After deciding on an isoelastic spring design, three prototypes were manufactured. The first prototype was constructed from acrylic plastic and medium density fiberboard (MDF), and used a 4:1 gear ratio (as seen in the figure below).

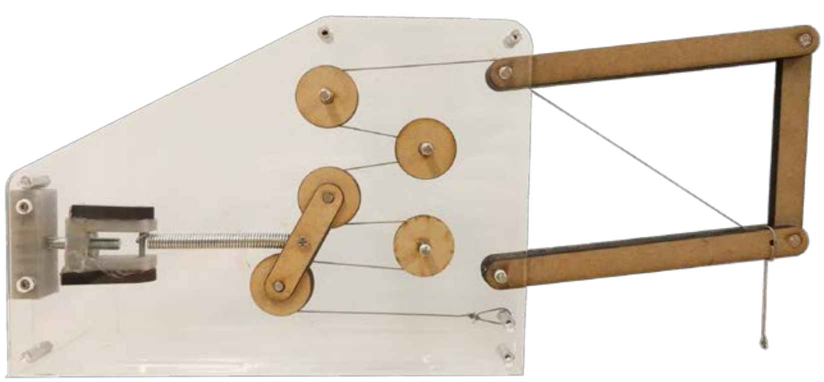


**Figure 5.27: First Prototype of Isoelastic Spring System**

Based upon existing steady cam systems, pulleys were used to apply this gear ratio. Although the system could support a 1kg weight in a variety of positions, the pulley system did not work properly because of excess friction, reducing the effective gear ratio from 4:1 to 2:1. The MDF isoelastic arm also produced more friction than desirable, and did not provide a location for gimble or z-motor mounting. The spring was chosen based on convenience, and was not optimized for the system.

The second prototype was manufactured entirely from acrylic plastic to reduce friction produced by MDF connections. Bearings were also used instead of bushings to reduce friction, and idlers pulleys were used to reduce friction in the wire. Due to the prior pulley difficulties, and concern that suspended pulleys would be compromised by aggressive movements of legged robots, a new gearing method was implemented. The new method used a system of three connected pulleys, two large and one small. The small pulley (diameter 1") connected to the spring, and the large pulley (diameter 4") connected to the isoelastic arm. This produced a 4:1 gear ratio with minimal moving parts. Finally, to aid stability and provide an attachment point for the gimbal, the isoelastic arm was redesigned as two arms (each attached to a large pulley) connected by aluminum standoffs. The z-actuation motor was affixed to one side of the arm to allow actuation. Prototype 2 had significantly less friction than Prototype 1, and gave us more faith in our isoelastic spring design. Initial vibration tests also produced encouraging results. Prototype 2 can be seen in the image below.

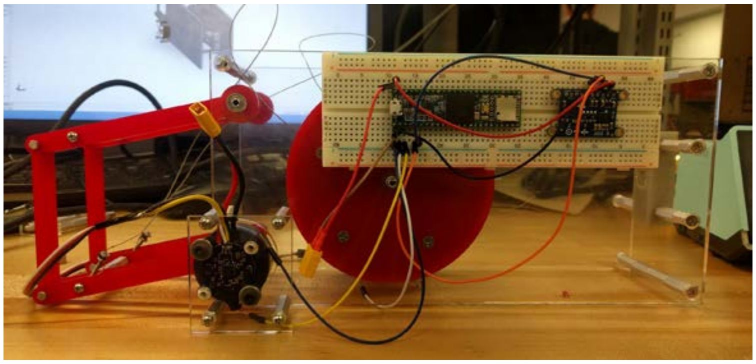


**Figure 5.28: Image of Second Prototype**

Prototype 3 was manufactured to give us the opportunity to use our final design specifications and components before final manufacturing was complete. We used our final spring and an optimized gear ratio of 7:1. The prototype re-used the isoelastic arm from Prototype 2 to reduce manufacturing time. We performed significant testing on this prototype to verify our final design decisions.

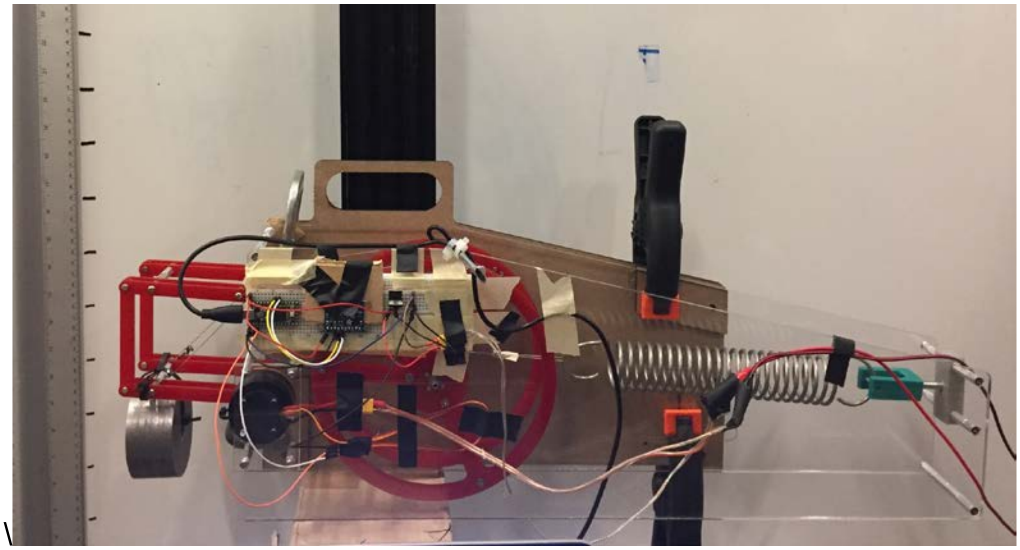


**Figure 5.29: Image of Third Prototype**

### 5.5.4 Z Actuation Manufacturing and Assembly

The z-system was manufactured primarily from carbon fiber laminate to reduce system weight. Carbon fiber with foam core was chosen over a balsa wood core so that the structural integrity of the system would not be damaged by water. For 0.125” components, solid carbon fiber was used because of the prohibitive cost of 0.125” laminate. Carbon fiber components were manufactured in sheets of several components on a ProtoTRAK CNC mill to reduce cutting time. Components were cut using a 0.5” carbide end mill to reduce tool wear, and respirators were used to protect against carbon fiber dust. A vacuum was also run continually to prevent spreading of carbon fiber dust and machine damage. Carbon fiber laminate parts were cut in two passes, one to penetrate the top layer of carbon fiber laminate, and the second for the bottom layer of laminate. Higher RPMs (2000-3000) produced slightly better finishes than lower RPMs.

After cutting was complete, approximately 0.25” of foam was removed from the edges of the carbon fiber laminate components. These gaps were then sealed with epoxy to increase durability and edge strength, and to improve surface finish. The carbon fiber was then covered with masking tape and the edges were spray-painted black. For 0.125” solid carbon fiber parts, edges were sanded with 600 grit sandpaper, and then sealed using cyanoacrylate.

For final manufacturing, the tensioner was changed from the linear tensioning screw design used by Prototypes 1-3 to a ratcheting system. The ratcheting system reduced the overall length of the system to bring it within length constraints. It also simplified tensioning of the spring, and allowed the entire process to be performed with a flat head screwdriver. This required the manufacture of a ratchet pawl from alloy steel. An additional pulley stage was also added to the final design to allow the spring to pass under the pulley system, thereby reducing its impact on overall system length.

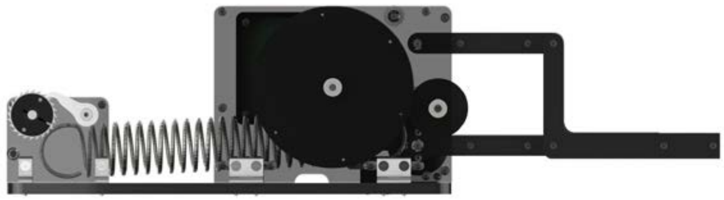


**Figure 5.30: Complete Z-Actuation System Render**

Most metal components were simple rods, which were cut to length and tapped on a Hardinge manual lathe. Support rods were machined from 6061 aluminum, while function-critical rods were machined from ground 303 stainless steel. These metals were chosen for their strength, machinability, and rust-resistance. The gimbal interface plate was machined from 6061 aluminum on a ProtoTRAK CNC mill. Fasteners, bearings, and other stock components were purchased from McMaster Carr. See Appendix A for bill of materials, Appendix C for full part drawings, and Appendix D for assembly procedure.

### 5.5.5 Z-Actuation Design Improvements

Sealing the edges of the carbon fiber laminate components with epoxy significantly increased their weight. The estimated weight of the system was 1.1kg, but the final weight was approximately 1.5kg. For future iterations, this weight could be reduced by removing less of the foam core. Approximately 0.25” was removed, but 0.1” was sufficient. This would have significantly reduced the amount of epoxy required. The surface finish of the parts could also be improved by using Bondo (an auto body filler) instead of epoxy. Bondo is similarly adherent to foam, but is easier to apply and sand.

Weight of the system could also be improved by more significantly optimizing the design of components. Rod sizes were chosen to meet rough force requirements, and to make attachment and perpendicularity easy. By further optimizing these rod sizes with structural analysis, significant weight could be cut from the system, particularly on spring tensioning components. 0.125” Carbon fiber components were also designed for ease of manufacturing and assembly, and were significantly overbuilt. By water-jet cutting the parts and performing finite element analysis, their weight could be significantly reduced with selective cutouts. Weight of 0.25” carbon fiber laminate parts could also be reduced through optimization, but because of the low density of the laminate, these weight savings would likely be insignificant.

Finally, the size of the z-system could potentially be reduced by further exploring torsional springs. Because of the ease of use of linear springs, and our greater familiarity with their use, we prioritized their use over torsional springs. However, by placing a torsional spring on-axis with the pulleys, the length of the system could be significantly reduced, though width would be increased.

# 6. Validation and Testing

## 6.1 Z-Actuation Prototype Motor Selection Validation

Before final construction, we constructed a prototype isoelastic spring to the initial specifications we calculated during design. The main purpose for this prototype was to validate that the selected motor would be able to stabilize the mass of the gimbal when combined with the passive stabilization from the isoelastic spring.

In order to validate the ability of the motor to control the mass, we tested the passive frequency response of the spring-mass system. We conducted this test by mounting the prototype z-actuator on a linear slide and manually oscillating it.

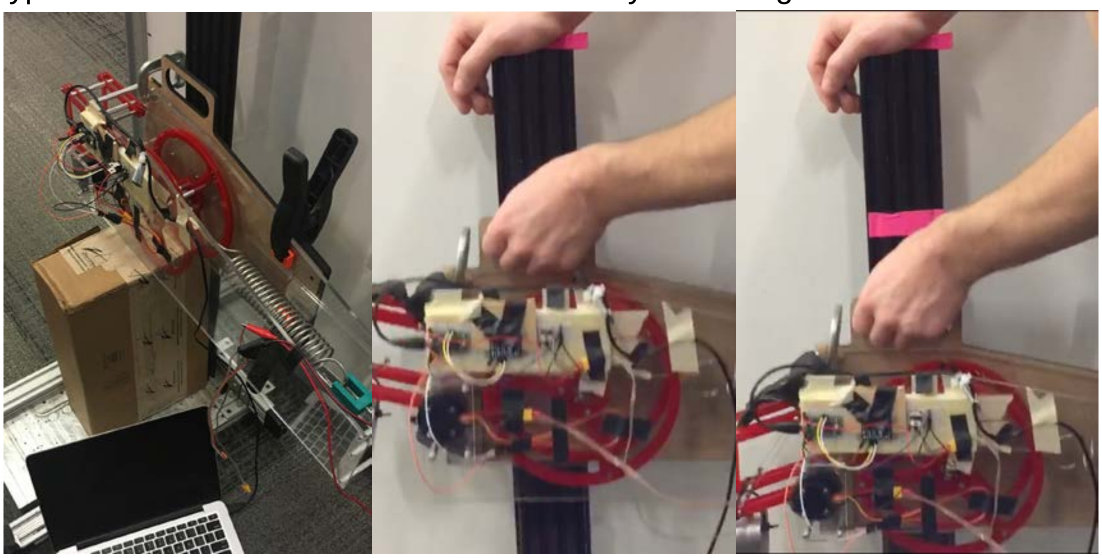


**Figure 6.1: Z-Actuation Test Setup**

A metronome was used to insure accurate frequency generation, and amplitude was standardized using two tape markers on the linear slide. Amplitudes of both (one) and (another) were tested. System response was recorded using the motor as a sensor (but not applying torque) and the same logging system used in the final design.

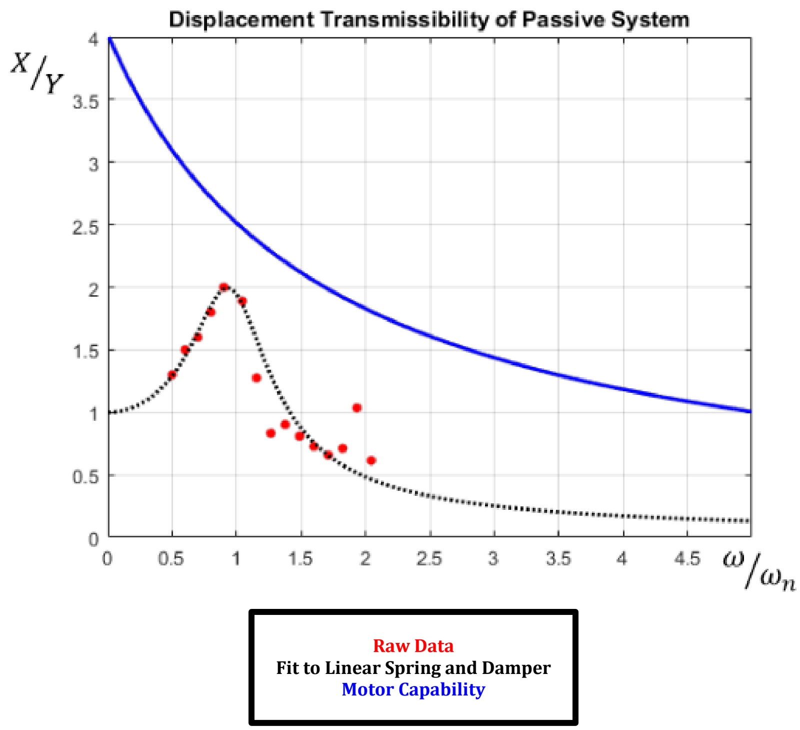



**Figure 6.2: Displacement Transmissibility of Passive System (Prototype)**

Analysis showed that the selected motor would be capable of controlling the mass in conjunction with the passive system. Creating a non-dimensional plot of input frequency versus output amplitude (shown above in Figure 6.2), the raw data points collected from the oscillation test were fit to those predicted by a linear spring and damper system. Since the system's spring constant was already known, the damping coefficient could be determined by which value fit the data best. In this case, we found the damping coefficient to be 0.3, resulting in an r-squared value of 0.99 for the raw data fit. Once these constants were calculated, the motor's capability curve was calculated at various input frequencies using its rated maximum torque output and the new linear equations of motion for the isoelastic spring. The result was that, on a non-dimensional plot, the motor capability curve was always greater than the isoelastic spring output curve. This proved that our system should be capable of stabilization across all frequencies. In addition, rather interestingly, the isoelastic spring system passively damps high frequency vibrations while the motor actively damps low frequency vibrations.

## 6.2 System Analysis of Frame Loss and Motion Blur

There are two main metrics that our system was designed to correct, frame loss and motion blur. To validate our system we had two cameras mounted on our stabilize system, one that was rigidly attached to the base and one that was within our stabilize system.

For frame loss, since we were unable to test with the RHex on non-engineered surfaces, we decided to use our hands to move the system in all four degrees of freedom as well as a combined motion that was meant to mimic the robot in a more chaotic environment.  We then modified a Matlab script to analyze both of the videos and calculate the frame loss.  In our initial tests, we achieved a decrease in average frame loss by 85%.  Also, we reduced the large spikes where nearly 100% of the frame was lost to under 40%.

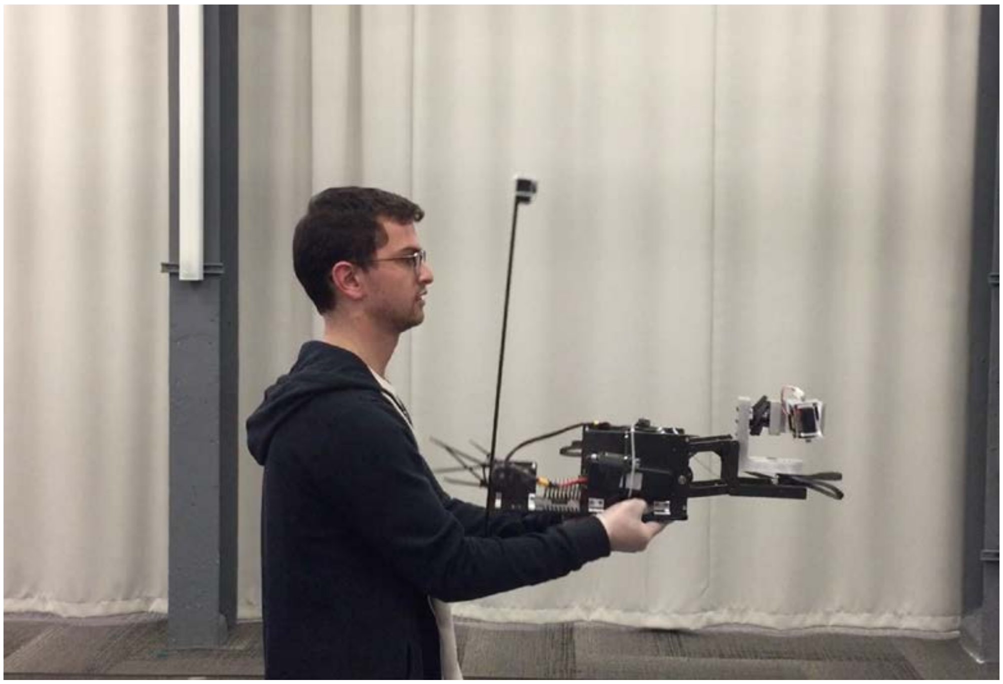


**Figure 6.3: Image of Frame Loss Test Setup**

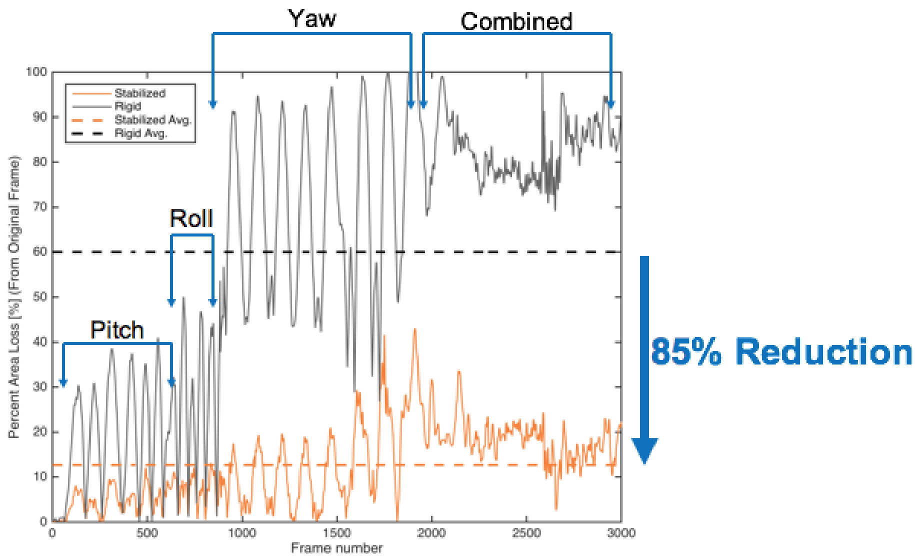



**Figure 6.4: Frame Loss Video Comparison Test**

We used a similar setup for motion blur however this time we tested on the Rhex platform, since motion blur can occur in typical walking gaits inside. We then created a matlab script to analyze the videos and calculate the power spectrum, which correlates to motion blur.

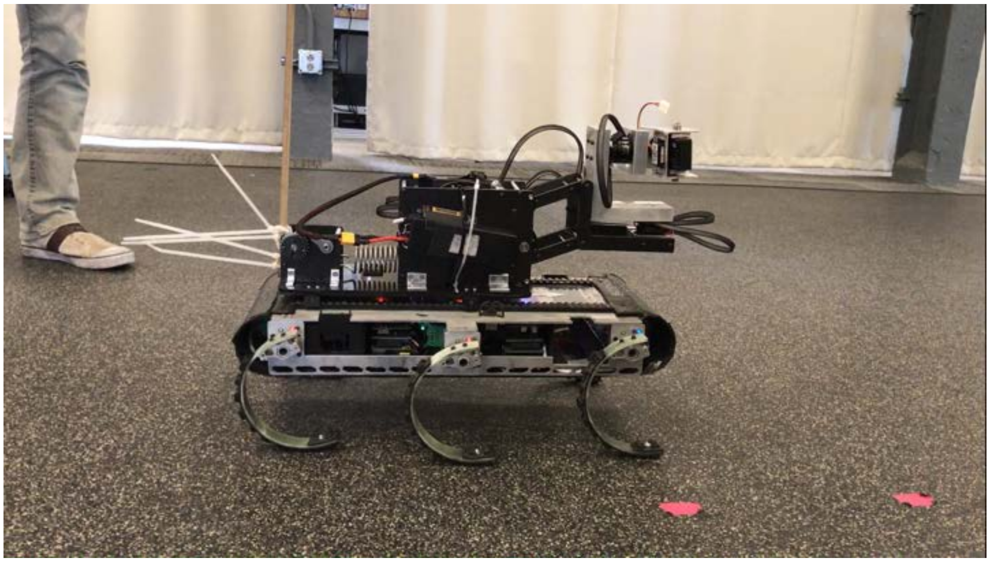


**Figure 6.5: Stabilize System Attached to RHex**

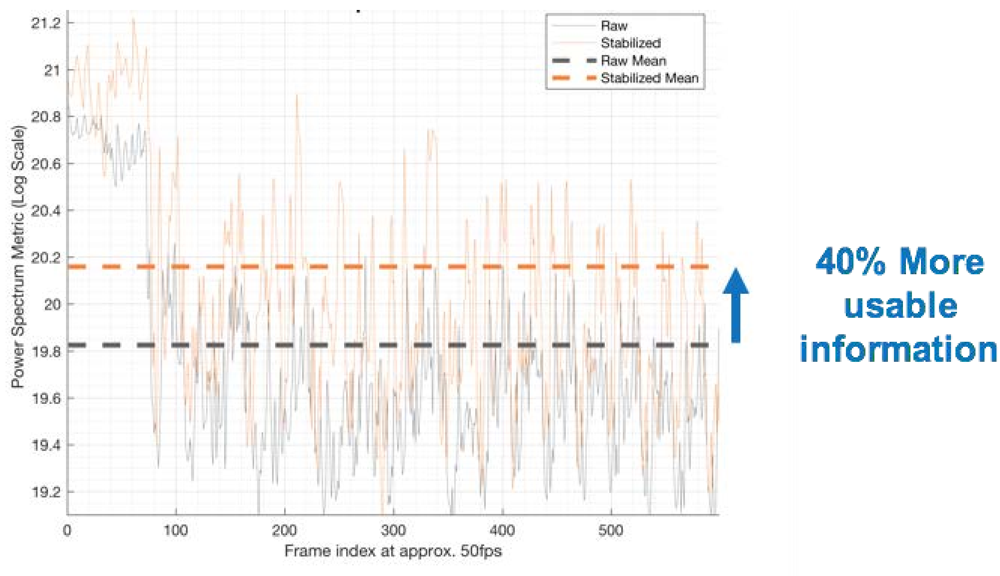



**Figure 6.6: Motion Blur Video Comparison Test on RHex**

## 6.3 FFT Analysis of Linear and Angular Velocities

In addition to evaluating metrics directly tied to computer vision, like frame loss and motion blur, we also analyzed how the angular and linear velocities varied between the body and camera for each degree of freedom. Using a fast fourier transform (FFT), we evaluated data collected during RHex's 1 Hz walking gait on an engineered surface. From IMU and motor feedback, the angular velocity in roll, pitch, and yaw as well as the linear velocity in z could be tracked over time. Plugging these velocities into MATLAB's FFT function, we created plots (shown below in Figures 6.7 to 6.10) that tracked both the body's and the camera's angular and linear velocity amplitudes at various frequencies. Given that frame loss, motion blur and any other stabilization metric is based on the angular and linear speed of the camera relative to its surroundings, these FFT plots can provide further and more detailed insight into how well the system stabilized at various frequencies and in various degrees of freedom. Overall, the observed average reductions in angular and linear velocity across all frequencies were: 47% reduction in roll, 41% reduction in pitch, 36% reduction in yaw, and 21% reduction in z.

Clearly, for all four degrees of freedom, the largest amplitude vibrations (for both the body and camera) occur at lower frequencies, as is typical for legged platforms. In addition, for all angular degrees of freedom (roll, pitch and yaw) we see a relatively uniform decrease in amplitude across all frequencies between the body and camera, indicating stabilization across nearly all frequencies. However, in z, while high frequencies (greater than 2 Hz) are substantially reduced, low frequencies (below 2 Hz) are not reduced and in fact are slightly increased (figure XXX). As mentioned earlier, the z system uses a combination of active and passive stabilization; the passive spring damps high frequency vibrations and the active motors damps low frequency vibrations. Thus, these results indicate an overall success of the isoelastic spring system, which stabilizes frequencies over about 2 Hz, and a failure of the z axis motor, which stabilizes frequencies below 2 Hz. This failure, as well as how it could be corrected, is discussed further in section 7 and likely explains why the average reduction in velocity in z is substantially lower than in the other 3 degrees of freedom.

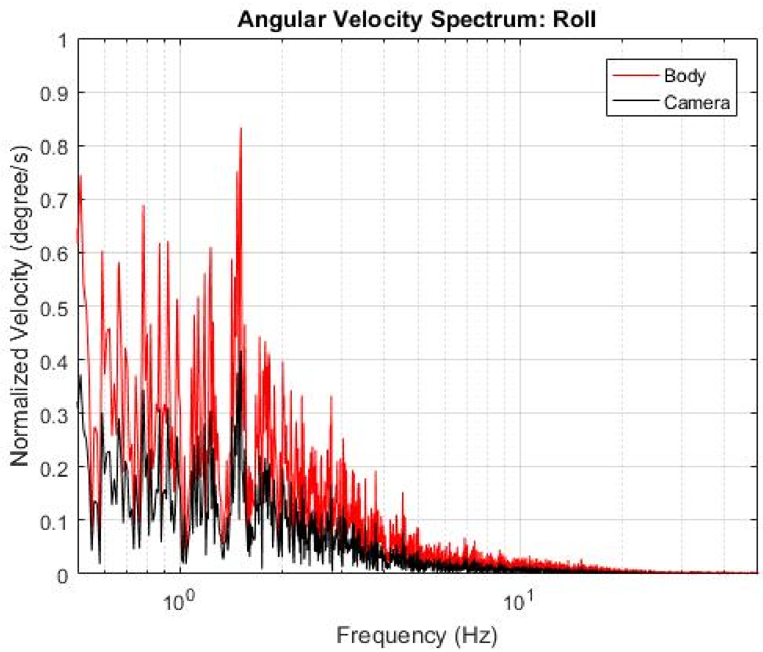



**Figure 6.7: Camera vs. Body Roll Response**

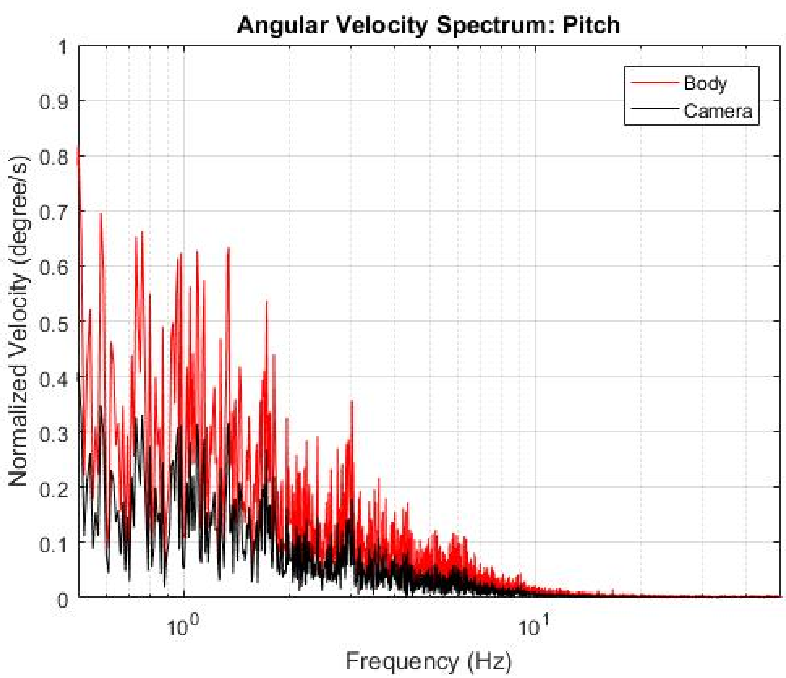



**Figure 6.8: Camera vs. Body Pitch Response**

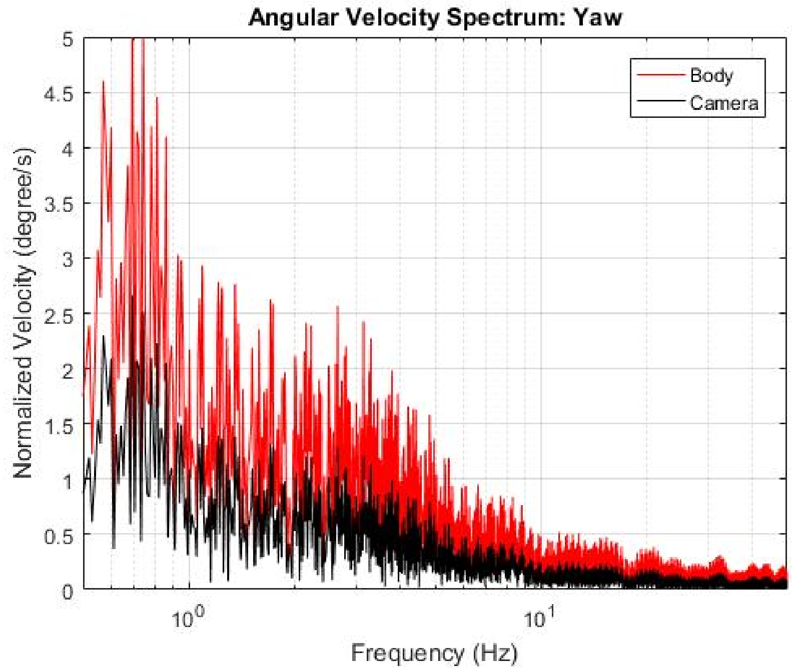



**Figure 6.9: Camera vs. Body Yaw Response**

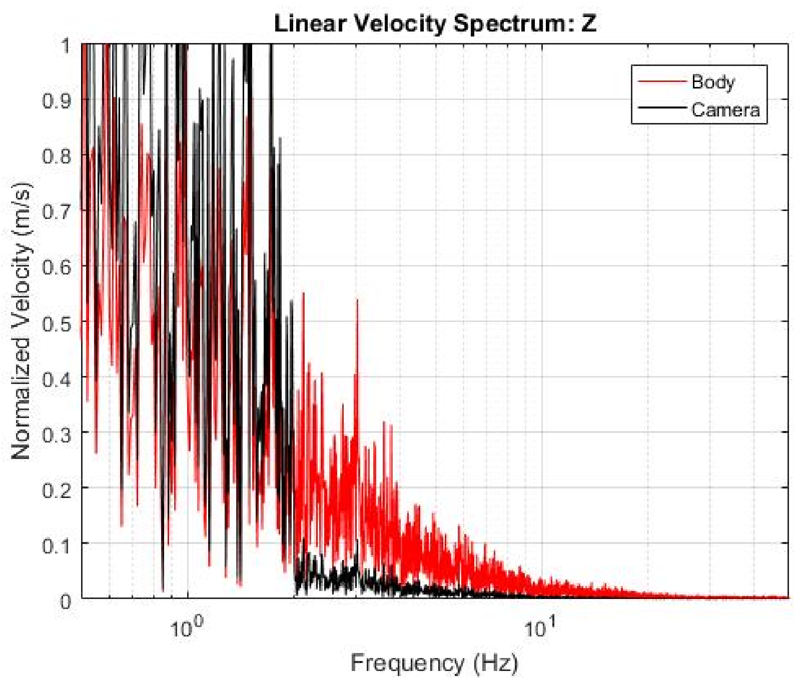



**Figure 6.10: Camera vs. Body Z Response**

# 7. Discussion

## 7.1 Target Versus Accomplished Performance [10]

| **Objective** | **Metric** | **Accomplished Performance** |
|---|---|---|
| Form Factor | Should Fit within the Rhex platform *(540mm x 390mm x 127mm)* | We were able to maintain the length and width requirements, however our system broke the height requirement. |
| Camera Support | Should support camera's of GoPro form factor *(Camera that Professor Aaron Johnson is using in his application)* | We were able to create an adjustable camera holder that would fit any camera around the GoPro form factor. |
| Response Directions | Should correct for rotations in Roll, Pitch and Yaw and translations in the Z direction | Our system was actuated in all 4 degrees of freedom required. |
| Frequency Response | Should be able to correct for 0.1 - 10 Hz vibrations in Roll, Pitch, Yaw and Z direction *(Based on vibrations experienced by RHex platform)* | Based on FFT analysis of initial testing on RHex from Figure 6.7-6.10, the Stabilize system reduced angular velocities across all frequencies in roll, pitch and yaw and reduced linear velocities in z for frequencies above 2 Hz. However, for frequencies below 2 Hz in z, no reduction in linear velocity was achieved. |
| Communications | Should be able to communicate to RHex platform over serial | We are able to send packets of information via serial, however we did not have time to fully integrate with the RHex platform. |
| Mounting | Should be able to integrate with picatinny rails on RHex platform | Our system was designed utilized picatinny rail mounts to integrate with both the RHex and Minitaur platform successfully. |
| Weight | Should be under the RHex platforms max payload of 8kg | Our system, including an extra onboard battery, weighed ~1.8 kg |
| Miscellaneous | Should have rugged design to mimic RHex platform | We successfully built a rugged fully enclosed system utilizing strong aluminum alloys, steel and carbon fiber. |

| | | |
|---|---|---|
| Motion Blur | 50% Reductions from initial testing on RHex platform *(Educated guess to perform application by Professor Aaron Johnson)* | From initial testing we were able to reduce the average frame loss by 40% |
| Average Frame Loss | 95% Reductions from initial testing on RHex platform *(Educated guess to perform application by Professor Aaron Johnson)* | From initial testing we were able to reduce the average motion blur by 85% |

## 7.2 Recommendations

There are several recommendations we suggest that would improve our system. First, we made a mistake in our Z axis controller, which was then evident in the results of our performance in this direction.  Our Z axis was controlled using a position controller, however should have used the Torque control in the IQinetics motor controllers.  This way acceleration does not need to be integrated to estimate position of the arm and can be directly translated into a correction by the motor. Furthermore, more tuning of our gains through experimentation will further improve performance our system. Our system had one point of structural failure in the motor mounts that came with the IQinetics motor modules, so manufacturing a more rigid mount would maintain the ruggedness of the system. Given more time we would have implemented our system with the RHex robot which could allow for more testing on the vision side to validate that the effective information was improving robot performance.  Following the previous point, it would be ideal to send the system to Dr. Aaron Johnson so that the system can be tested on RHex while it completes computer vision tasks. This should help us to further validate how accurate Dr. Johnson's initial estimates for frame loss and motion blur are correlated to successful autonomy on legged robots.

# 8. Budget

## 8.1 Monetary Budget Distribution

Our budget consisted of $2,520 given us by the Mechanical Engineering and Applied Mechanics department at the University of Pennsylvania. The table below describes the distribution of those funds into different component categories.

**Table 8.1:** Spending within each category type

| **Category** | **Cost** |
|---|---|
| Electronics | $191 |
| IMUs | $330 |
| Hardware | $580 |
| Motors | $354 |
| Carbon Fiber | $383 |
| Aluminum Stock | $387 |
| Shipping | $249 |
| **Total** | **$2,474** |

## 8.2 Other Resources

**Within MEAM Department of the University of Pennsylvania:**
Additive Manufacturing Laboratory (high precision 3D printing), GM Laboratory (general assembly), Precision Machining Laboratory (subtractive metal manufacturing), and Rapid Prototyping Laboratory (laser cutting and FDM 3D printing).

**Within GRASP Laboratory of the University of Pennsylvania:**
Kodlab (Use of RHex robot and lab space).

**External Resources:**
Ghost Robotics (Use of Minitaur robot) and IQinetics (Anti-cogging motor controllers/software).

# 9. References

[1] U. Saranli, M. Buehler and D. Koditschek. RHex: A simple and highly mobile hexapod robot. The International Journal of Robotics Research 20(7), pp. 616-631. 2001. Available: http://search.proquest.com/docview/230033338. DOI: 10.1177/02783640122067570.

[2] "Roll, Pitch, And Yaw | How Things Fly." Howthingsfly.si.edu. N.p., 2017. Web. 30 Apr. 2017.

[3] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Modeling and Control. Hoboken, NJ: Wiley, 2006.

[4] Jianbo Shi, "Interview with Jianbo Shi," October 17th, 2016.

[5] Rafael Gonzalez, Richard Woods, and Steven Eddins. Digital Image Processing Using Matlab. Prentice Hall, 2003.

[6] C. Taylor, "Interview with CJ Taylor," October 20th, 2016.

[7] E. Mavridaki and V. Mezaris. No-reference blur assessment in natural images using fourier transform and spatial pyramids. 2014 IEEE International Conference 2014.

[8] Properties of Fourier Transforms. http://www.mif.vu.lt/atpazinimas/dip/FIP/fip-Properti-2.html

[9] Ghost robotics, llc; ghost robotics launches ghost minitaur direct-drive legged robot platform for autonomous vehicle applications. Robotics & Machine Learning pp. 6. 2016. Available: http://search.proquest.com/docview/1822134024.

[10] Aaron Johnson, "Conversation with Dr. Aaron Johnson," October 26th, 2016.

[11] “GoPro HERO5 Black vs HERO 5 Session.” havecamerawilltravel.com. N.p., 2017. Web. November 15th, 2016.

[12] Gavin Kenneally, "Conversation with PhD candidate Gavin Kenneally," October 30th, 2016.

[13] Matthew Piccoli, "Conversation with Dr. Matthew Piccoli," November 5th, 2016.

[14] Rekcut, Teensy 3.0 Watchdog Timer. PJRC. February 13, 2017.

[15] The Aluminum Association, Inc, Aluminum Standards and Data 2000. February 04, 2017.

# 10. APPENDICES

# APPENDIX A:
# BILLS OF MATERIALS

## A.1 Gimbal Bill of Materials

| Purchased Components – All from McMaster Carr | | | |
|---|---|---|---|
| **Part** | **Part Number** | **Cost Per** | **Quantity** |
| 7075 Aluminum Block (2.5*6*6) | 9037K57 | $167.50 | 1 |
| 7075 Aluminum Plate (.065*6*6) | 8885K841 | $8.47 | 1 |
| M3 Flathead Screws | 92290A111 | $0.19 | 28 |
| M2 Flathead Screws | 92125A052 | $0.22 | 24 |
| Half inch end mill | 8918A33 | $33.27 | 1 |
| Quarter inch end mill | 8923A38 | $36.03 | 1 |
| M2 tap | 8305A78 | $15.76 | 1 |
| Epoxy | 75445A44 | $11.73 | 1 |
| Rubber | 1310N31 | $12.91 | 1 |
| 10-32 thumb screws | 91746A700 | $3.04 | 2 |
| Dowel rods | 98381A505 | $0.15 | 4 |

| Manufactured Components | | |
|---|---|---|
| **Part** | **Material** | **Quantity** |
| Holder A | 7075 Aluminum | 1 |
| Holder B | 7075 Aluminum | 1 |
| Holder C | 7075 Aluminum | 1 |
| Holder B Cover | 7075 Aluminum | 1 |
| Holder C Cover | 7075 Aluminum | 1 |
| Half Top Grip Board | 6061 Aluminum | 4 |
| Grip Board Bottom | 6061 Aluminum | 2 |
| 10-32 Thumb Vise Screw | 10-32 Thumbscrew | 2 |
| Cover Fixture | 6061 Aluminum | 1 |
| Ring Fixture | 6061 Aluminum | 1 |
| Ring | 7075 Aluminum | 2 |

## A.2 Z-Actuator Bill of Materials

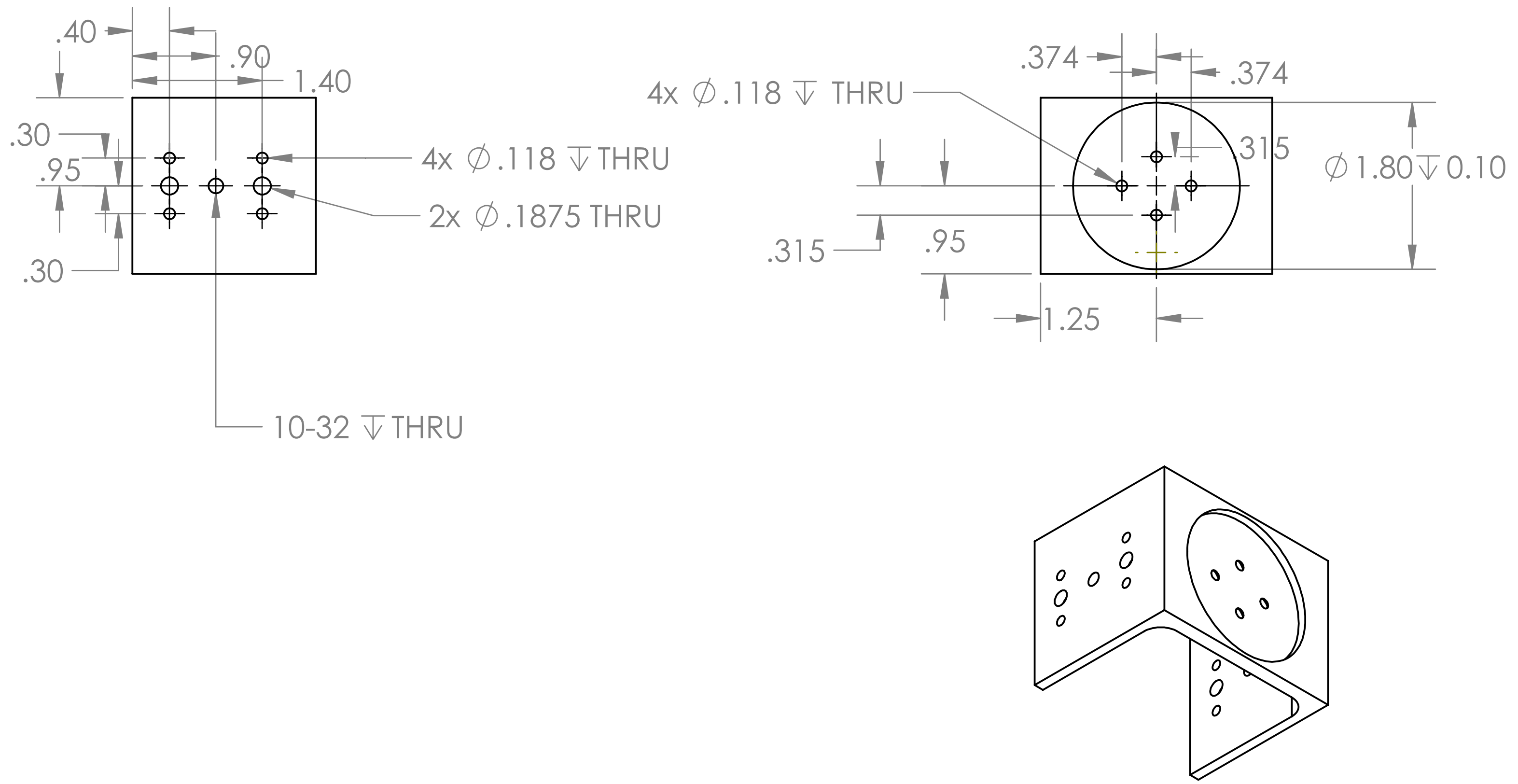
### Purchased Components

| Part | Part Name | Cost Per | Quantity | Order # | Link |
|---|---|---|---|---|---|
| 1/4" Locking Collar | Clamping Shaft Collar for 1/4" Diameter, 303 Stainless Steel | $3.94 | 4 | 6435K32 | https://www.mcmaster.com/#6435k32/=16a46jd |
| 6-32 Flathead | Passivated 18-8 Stainless Steel Hex Drive Flat Head Screw 6-32 Thread Size, 1/2" Long (100 pack) | $4.21 | 1 | 92210A148 | https://www.mcmaster.com/#92210a148/=16fb7t4 |
| 6-32 Panhead | 18-8 Stainless Steel Hex Drive Rounded Head Screws Black-Oxide, 6-32 Thread Size, 1/2" Long (50 pack) | $3.31 | 1 | 97763A143 | https://www.mcmaster.com/#97763a143/=16jiqoi |
| m3 Panhead | 18-8 Stainless Steel Hex Drive Rounded Head Screw M3 x 0.5 mm Thread, 8 mm Long | $9.37 | 1 | 92095A181 | https://www.mcmaster.com/#92095a181/=16geyf8 |
| 1/4" Bearing | Stainless Steel Ball Bearing Flanged Double Shielded, for 1/4" Shaft Diameter, 1/2" OD | $5.70 | 4 | 57155K323 | https://www.mcmaster.com/#57155k323/=16s64l8 |
| 3/16" Bearing | 3/16" Stainless Steel Ball Bearing Flanged Double Shielded with Ring, Trade No. R156-2Z | $6.93 | 8 | 57155K334 | https://www.mcmaster.com/#57155k334/=16rmlx6 |
| m3 Standoff | Male-Female Threaded Hex Standoff 18-8 Stainless Steel, 4.5 mm Hex Size, 12 mm Length, M3 Thread Size | $3.30 | 4 | 93655A099 | https://www.mcmaster.com/#93655a099/=16gevx4 |
| 1/4" Shoulder Screw | 18-8 Stainless Steel Thread-Locking Shoulder Screw 1/4" Diameter x 3/8" Long Shoulder, 10-24 Thread Size | $2.25 | 1 | 91327A107 | https://www.mcmaster.com/#91327a107/=16rozj3 |
| 1/4" Flanged Bushing | Oil-Embedded Flanged Sleeve Bearings with PTFE Ultra-Low-Friction, for 1/4" Shaft Diameter, 3/8" Length | $1.00 | 2 | 1677K2 | https://www.mcmaster.com/#1677k2/=16rox1j |
| 3/8" Flanged Bushing | Oil-Embedded Flanged Sleeve Bearings with PTFE Ultra-Low-Friction, for 3/8" Shaft Diameter, 1/4" Length | $1.05 | 3 | 1677K4 | https://www.mcmaster.com/#1677k4/=16romjj |
| 3/8" Locking Collar | Extra-Grip Clamping Shaft Collar for 3/8" Diameter, Black-Oxide 1026 Carbon Steel | $8.16 | 1 | 9951K33 | https://www.mcmaster.com/#9951k33/=16rogk0 |
| Ratchet | Ratcheting Gear 24 Teeth | $37.11 | 1 | 6283K24 | https://www.mcmaster.com/#6283k24/=16ro0tv |

Manufactured Components

| Part | Material | Quantity |
|---|---|---|
| 01_Plate_A | 1/4" Carbon Fiber Laminate | 1 |
| 02_Plate_B | 1/4" Carbon Fiber Laminate | 1 |
| 03_Arm_A | 1/4" Carbon Fiber Laminate | 2 |
| 04_Arm_B | 1/4" Carbon Fiber Laminate | 1 |
| 05_Arm_C | 1/4" Carbon Fiber Laminate | 1 |
| 06_Plate_Mount | 6061 Aluminum | 4 |
| 07_L-Bracket_A | 6061 Aluminum | 4 |
| 08_L-Bracket_B | 6061 Aluminum | 4 |
| 09_Idler_Pulley | 6061 Aluminum | 2 |
| 10_Motor_Plate | 1/4" Carbon Fiber Laminate | 1 |
| 11_Bottom_Plate | 1/4" Carbon Fiber Laminate | 1 |
| 12_Spring_Plate_A | 1/4" Carbon Fiber Laminate | 1 |
| 13_Spring_Plate_B | 1/4" Carbon Fiber Laminate | 1 |
| 14_Core_A | 303 Stainless Steel | 1 |
| 15_Flange_A | 1/4" Carbon Fiber Laminate | 2 |
| 16_Core_B | Acrylic Plastic | 2 |
| 17_Flange_B | 1/4" Carbon Fiber Laminate | 4 |
| 18_Support_Rod_A | 6061 Aluminum | 4 |
| 19_Support_Rod_B | 303 Stainless Steel | 4 |
| 20_Support_Rod_C | 6061 Aluminum | 3 |
| 21_Support_Rod_D | 303 Stainless Steel | 4 |
| 22_Pivot_Rod | 303 Stainless Steel | 5 |
| 23_Idler_Shaft | 303 Stainless Steel | 1 |
| 24_Ratchet_Rod | 303 Stainless Steel | 1 |
| 25_Axel | 303 Stainless Steel | 2 |
| 26_Interface_Plate | 6061 Aluminum | 1 |
| 27_Idler_Core | 6061 Aluminum | 1 |
| 28_Idler_Flange | 1/4" Carbon Fiber Laminate | 2 |
| 29_Pawl | 303 Stainless Steel | 1 |
| 30_L_Arm | 1/4" Carbon Fiber Laminate | 2 |
| 31_Ring | 6061 Aluminum | 1 |

# APPENDIX B:
# GIMBAL ENGINEERING DRAWINGS




| | |
|---|---|
| NEXT ASSY | USED ON |
| APPLICATION | |

**PROPRIETARY AND CONFIDENTIAL**



UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES

TWO PLACE DECIMAL ±0.005
THREE PLACE DECIMAL ±0.001

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL 7075 Aluminum

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | it | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS: Post CNC Machining

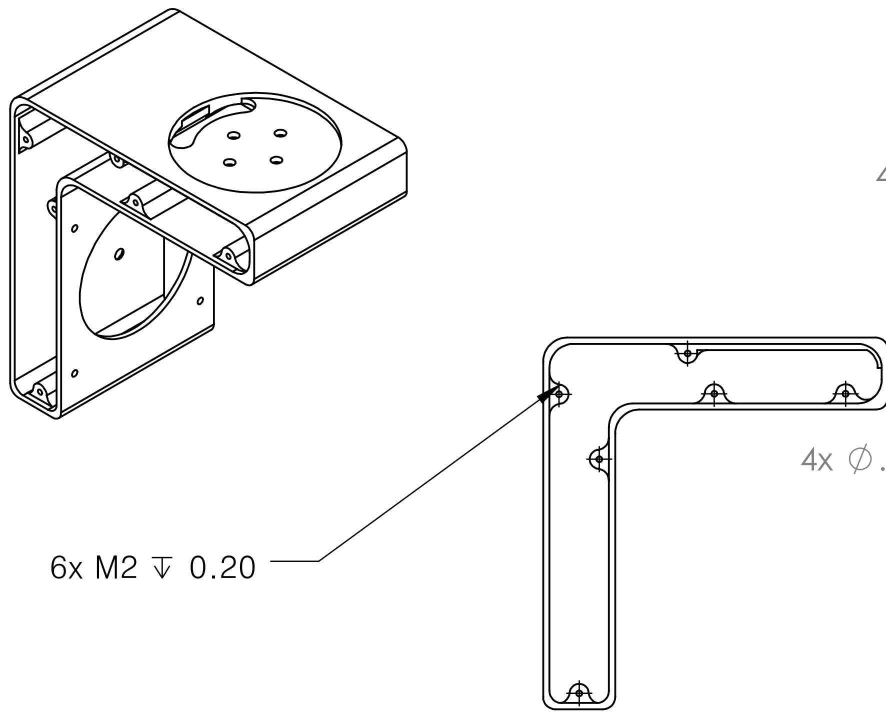
TITLE:

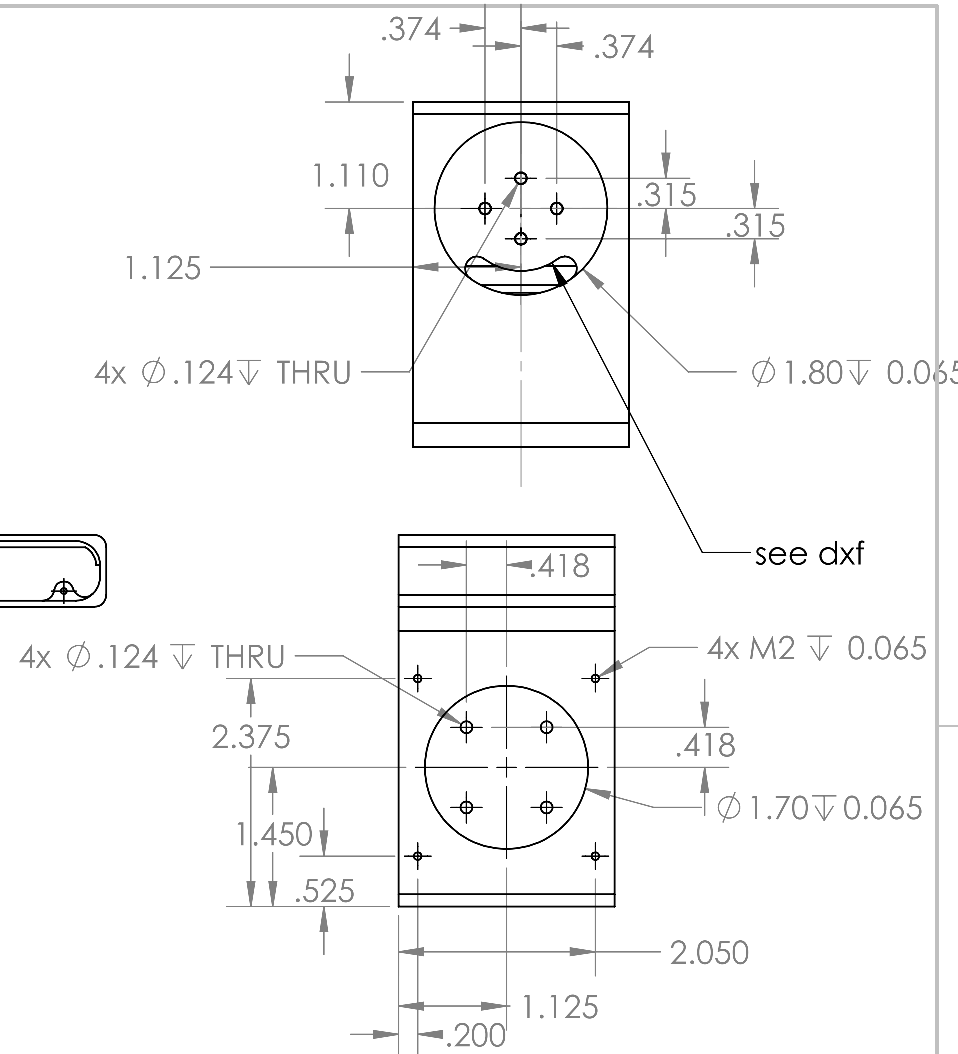
# Holder A

| SIZE | DWG. NO. | REV |
|---|---|---|
| A | | |

SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1

For Aid in CNC Machining and Manual Post-Processing




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY | USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
TWO PLACE DECIMAL ±0.005
THREE PLACE DECIMAL ±0.001

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL
7075 Aluminum

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | it | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS: use dxfs for cutting geometry after CNC machining

TITLE:

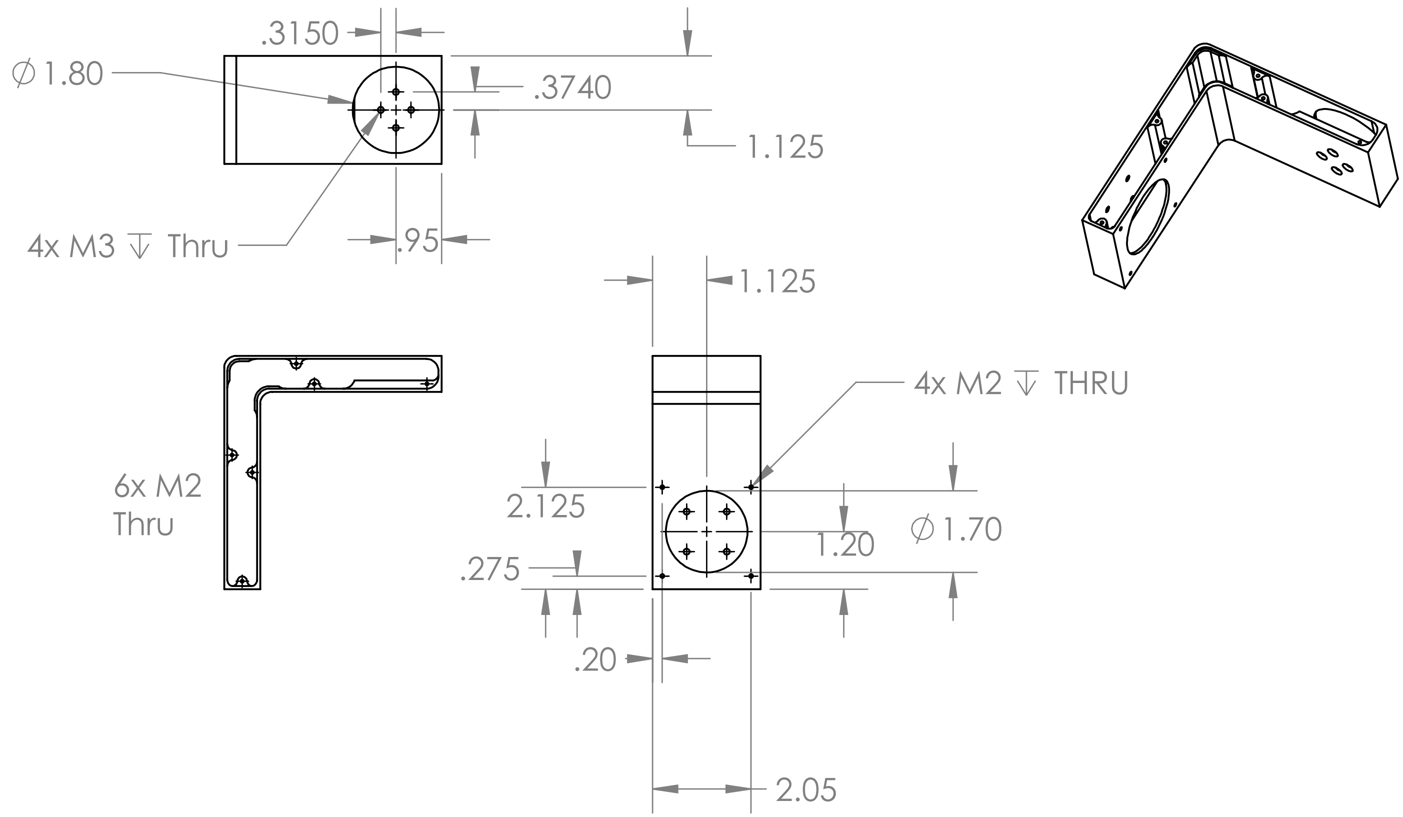
# Holder B

SIZE A | DWG. NO. | REV

SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1

2 1

B B




A A

| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE |
|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL .003<br>THREE PLACE ±<br>DECIMAL .001 | DRAWN | it | |
| | | | CHECKED | | |
| | | | ENG APPR. | | |
| | | | MFG APPR. | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | |
| | | MATERIAL 7075 aluminum | COMMENTS For Aid in CNC Machining and post processing: | | |
| NEXT ASSY | USED ON | FINISH | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | |

**PROPRIETARY AND CONFIDENTIAL**



TITLE:

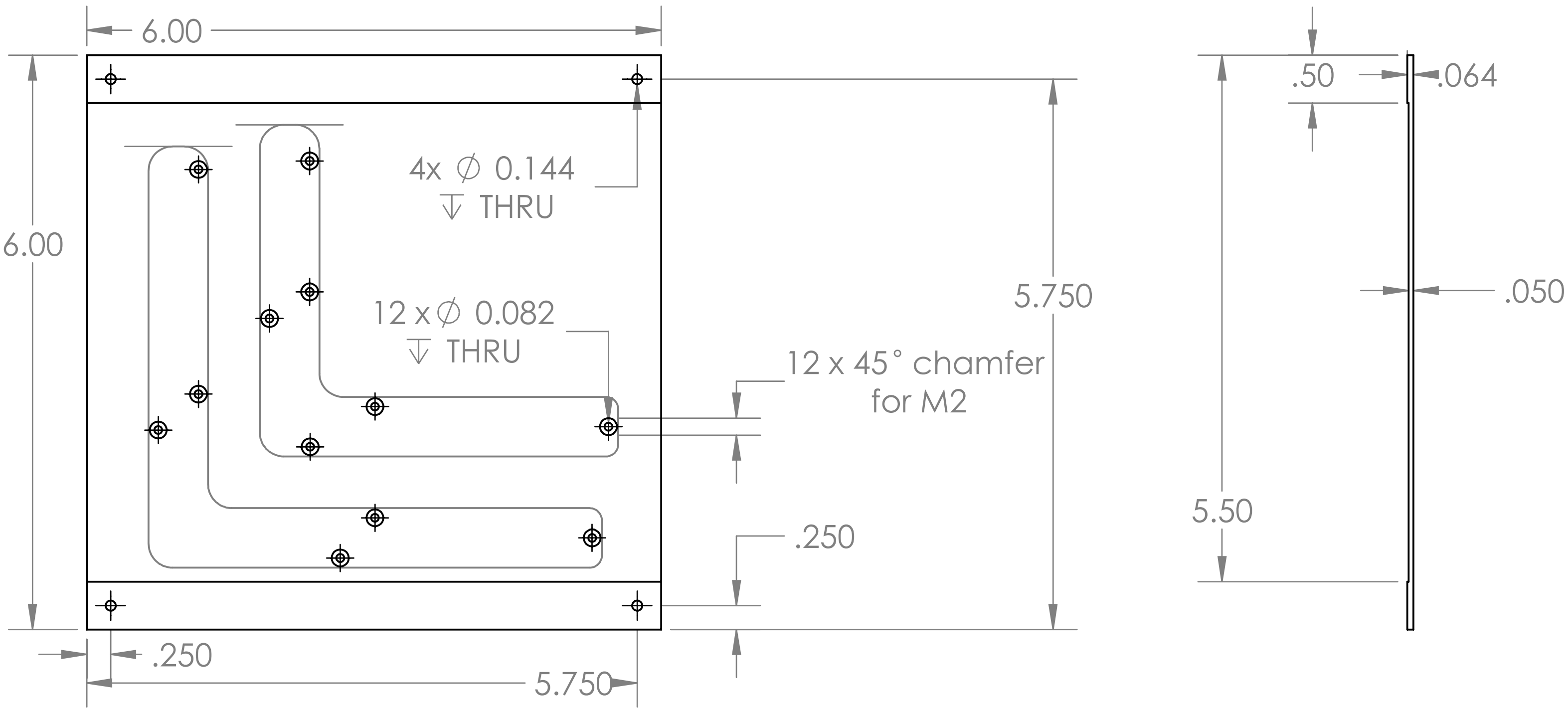
# HolderC

| SIZE | DWG. NO. | REV |
|---|---|---|
| **A** | | |

SCALE: 1:4 WEIGHT: SHEET 1 OF 1



2 1




| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE |
|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±.01<br>THREE PLACE DECIMAL ±.001 | DRAWN | | |
| | | | CHECKED | | |
| | | | ENG APPR. | | |
| | | | MFG APPR. | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | |
| | | MATERIAL | COMMENTS: holes done in prototrak chamfer until m2 fits flush against plate | | |
| NEXT ASSY | USED ON | FINISH | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | |

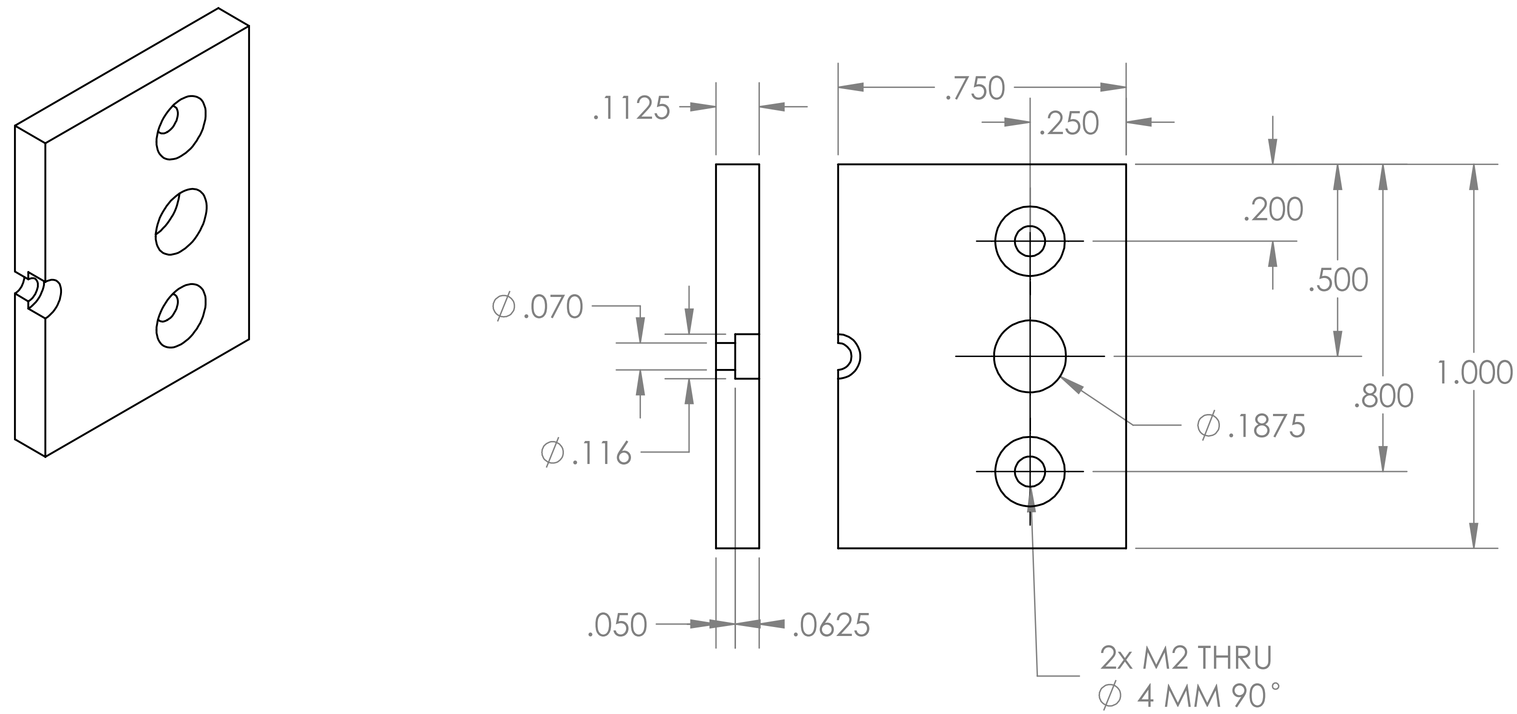
TITLE:

SIZE A | DWG. NO. CoverPlate | REV

SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1

2 1

B




B

A

| | | |
|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>THREE PLACE DECIMAL ±.001<br>FOUR PLACE DECIMAL ±.0005 |
| | | INTERPRET GEOMETRIC<br>TOLERANCING PER: |
| | | MATERIAL<br>ALuminum |
| NEXT ASSY | USED ON | FINISH |
| APPLICATION | | DO NOT SCALE DRAWING |

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS: 0.1875 is for a pressfit for dowel rods

TITLE: half top grip

SIZE A | DWG. NO. | REV

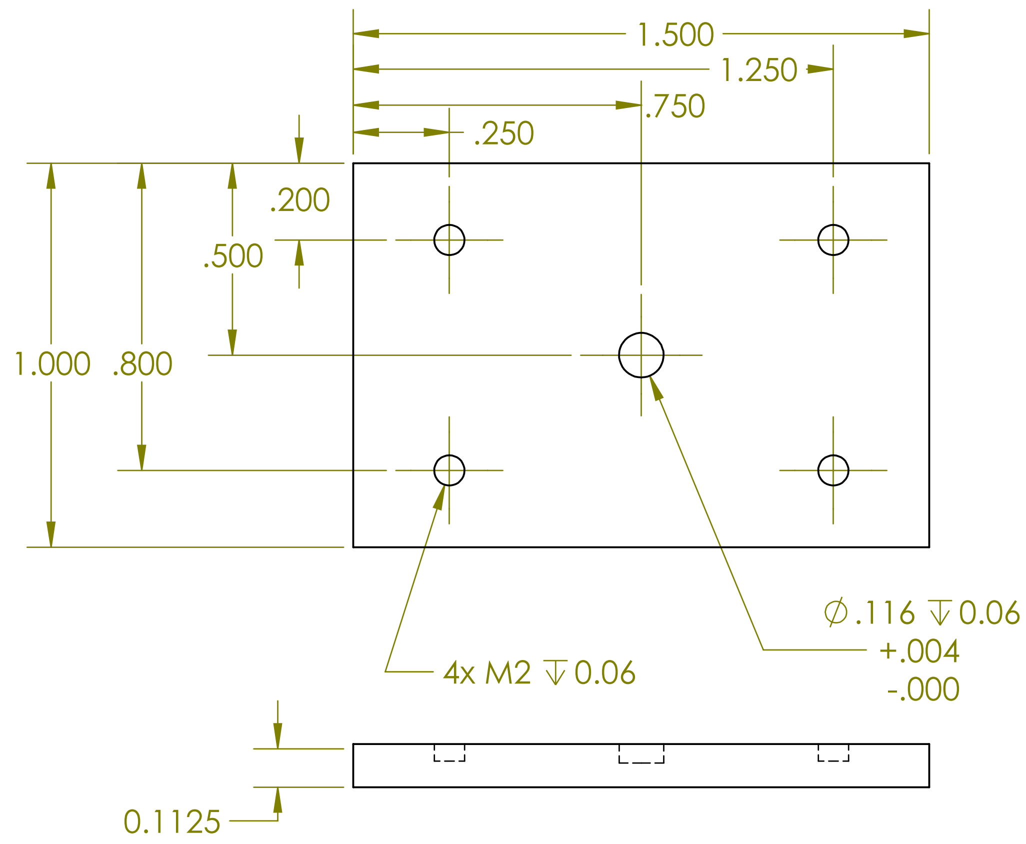
SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1



A



2 1



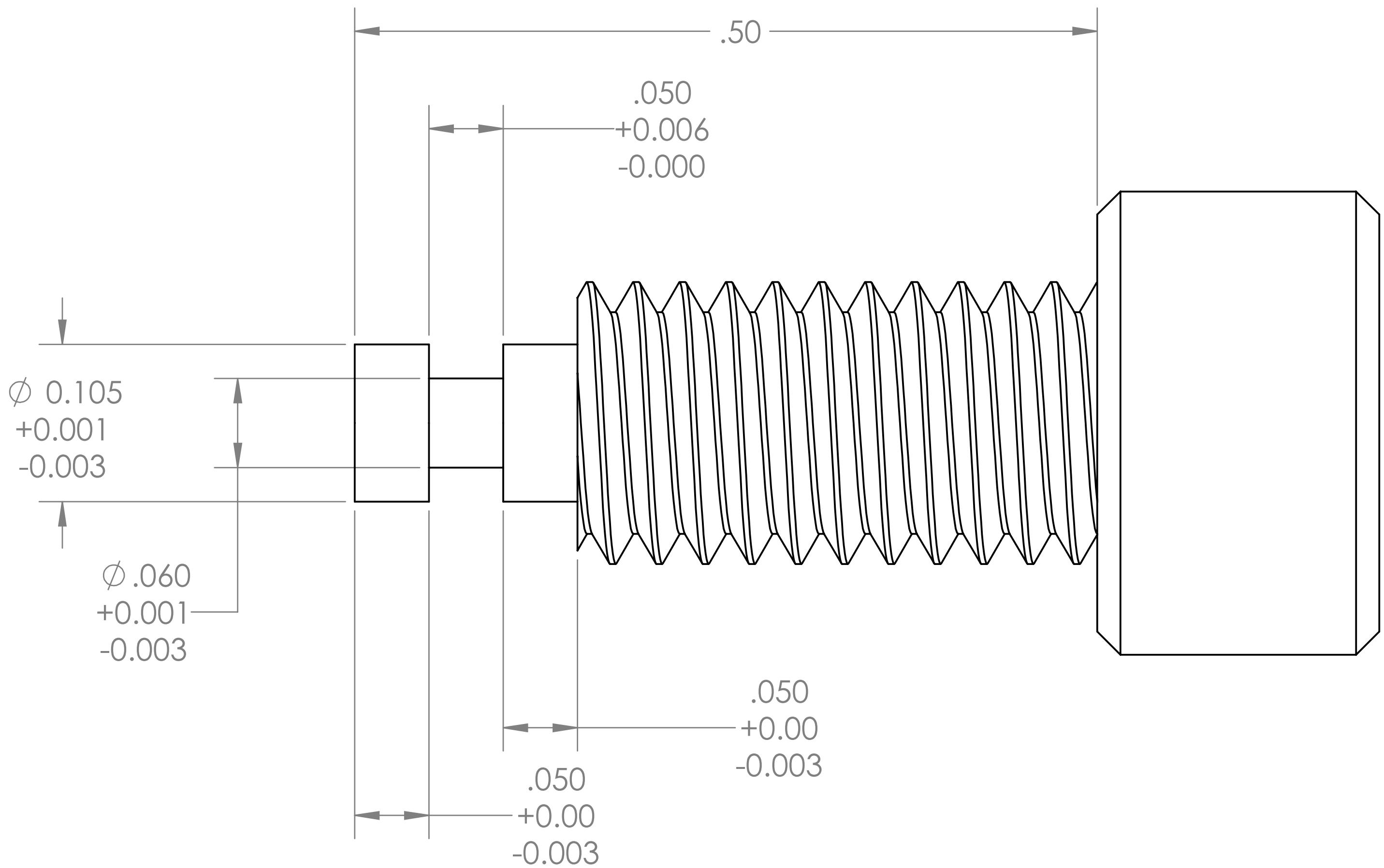

| | | | | | |
|---|---|---|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ±.0005 | DRAWN | | |
| | | | CHECKED | | |
| | | | ENG APPR. | | |
| | | | MFG APPR. | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | |
| | | MATERIAL<br>Aluminum 7075 | COMMENTS: center hole should slip- fit with turned screw | | |
| NEXT ASSY | USED ON | FINISH | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | |

TITLE:

# Grip_Board

| SIZE | DWG. NO. | REV |
|---|---|---|
| A | | |

SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES (unspecified):

TWO PLACE DECIMAL ±.005

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | it | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

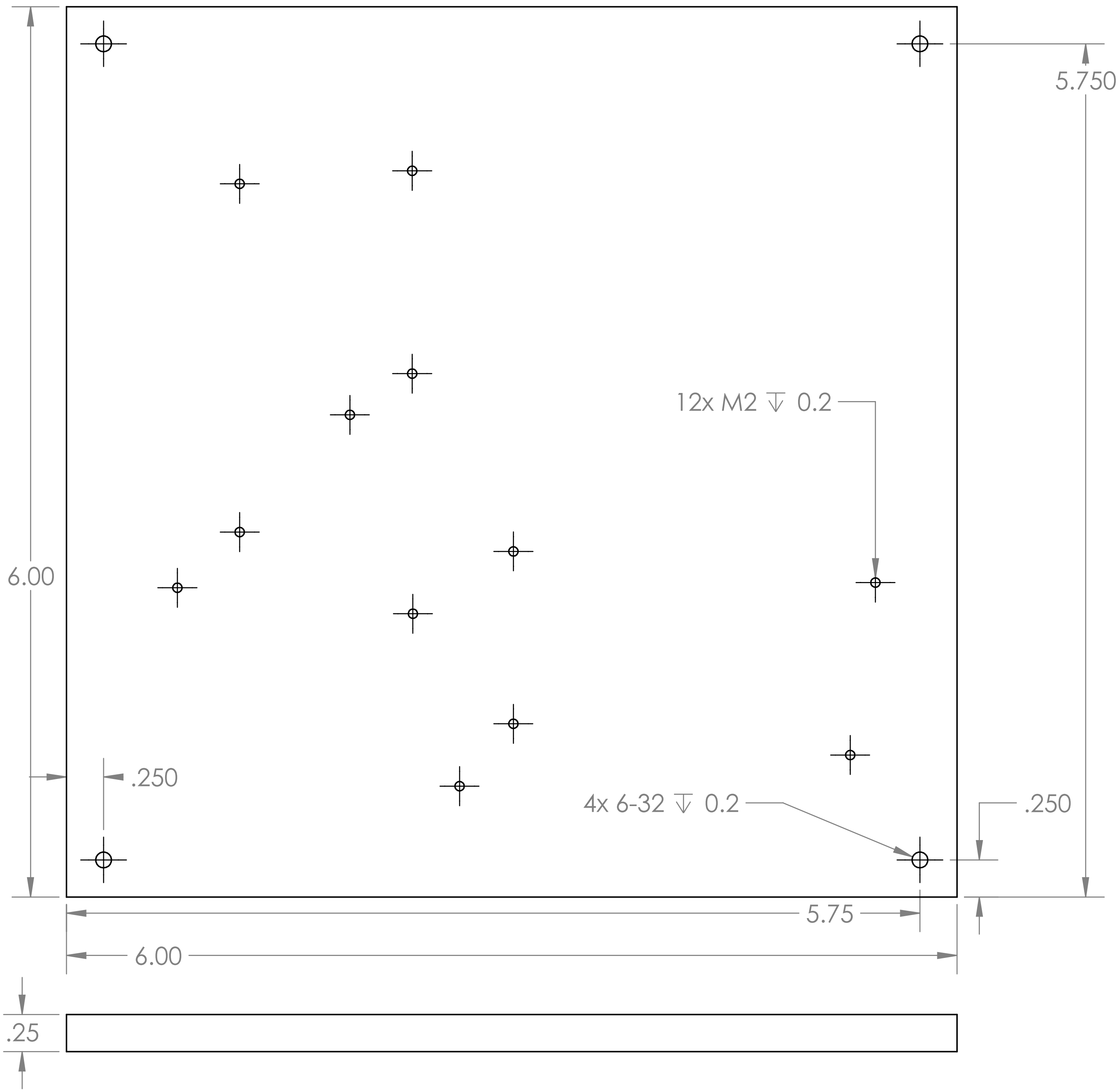
COMMENTS: Machine the end of a 1/2 inch 10-32 screw, preferably a low profile thumb screw

TITLE:

# 10-32 Thumb Vise Screw

SIZE **A** DWG. NO. REV

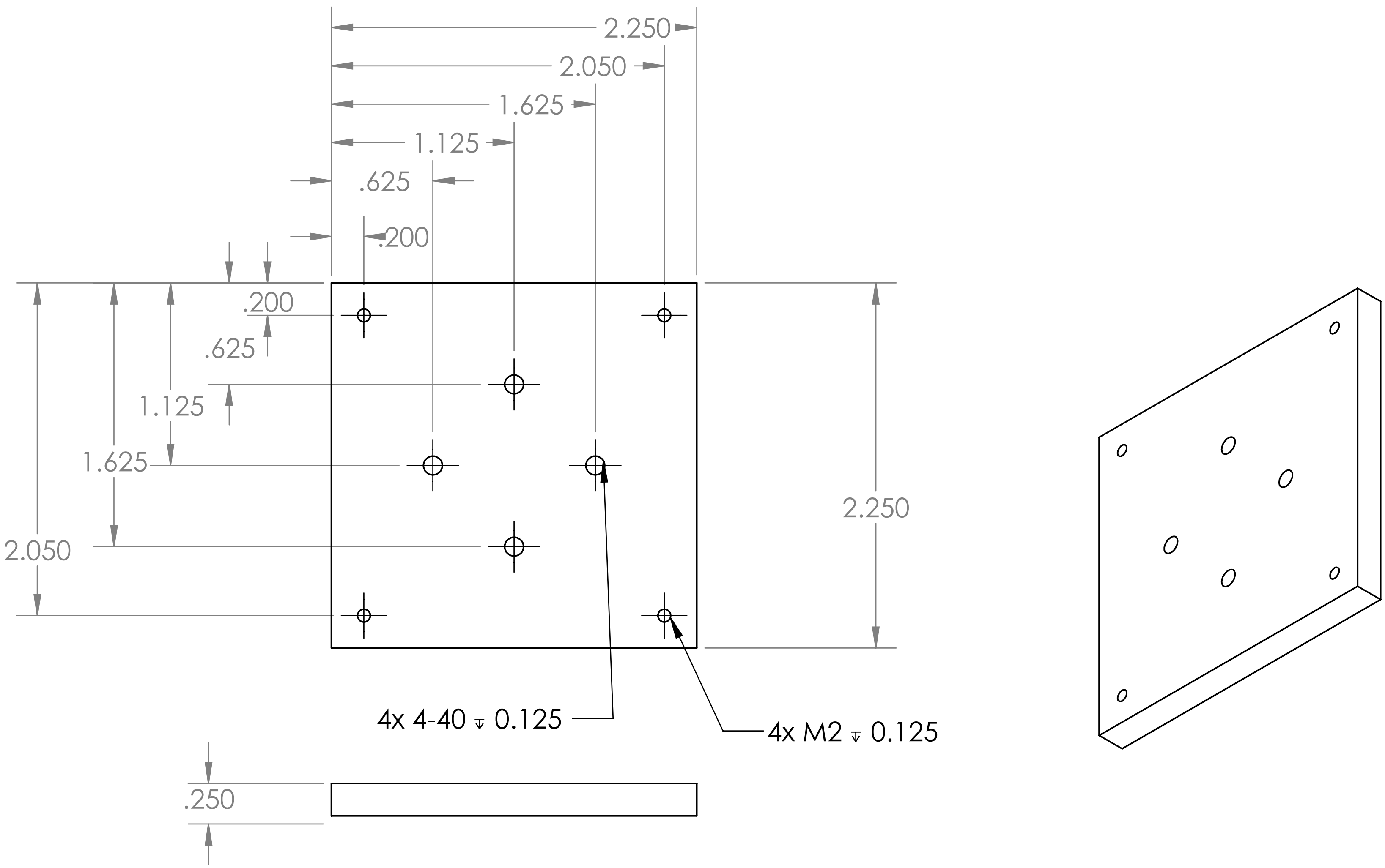
SCALE: 4:1 WEIGHT: SHEET 1 OF 1




DIMENSIONS ARE IN INCHES
TOLERANCES:
TWO PLACE DECIMAL ±.01
THREE PLACE DECIMAL ±.001
Holes placed by prototrak program
thickness of plate is arbitrary, can be 0.2
scrap aluminum is perfect for this

CoverFixture

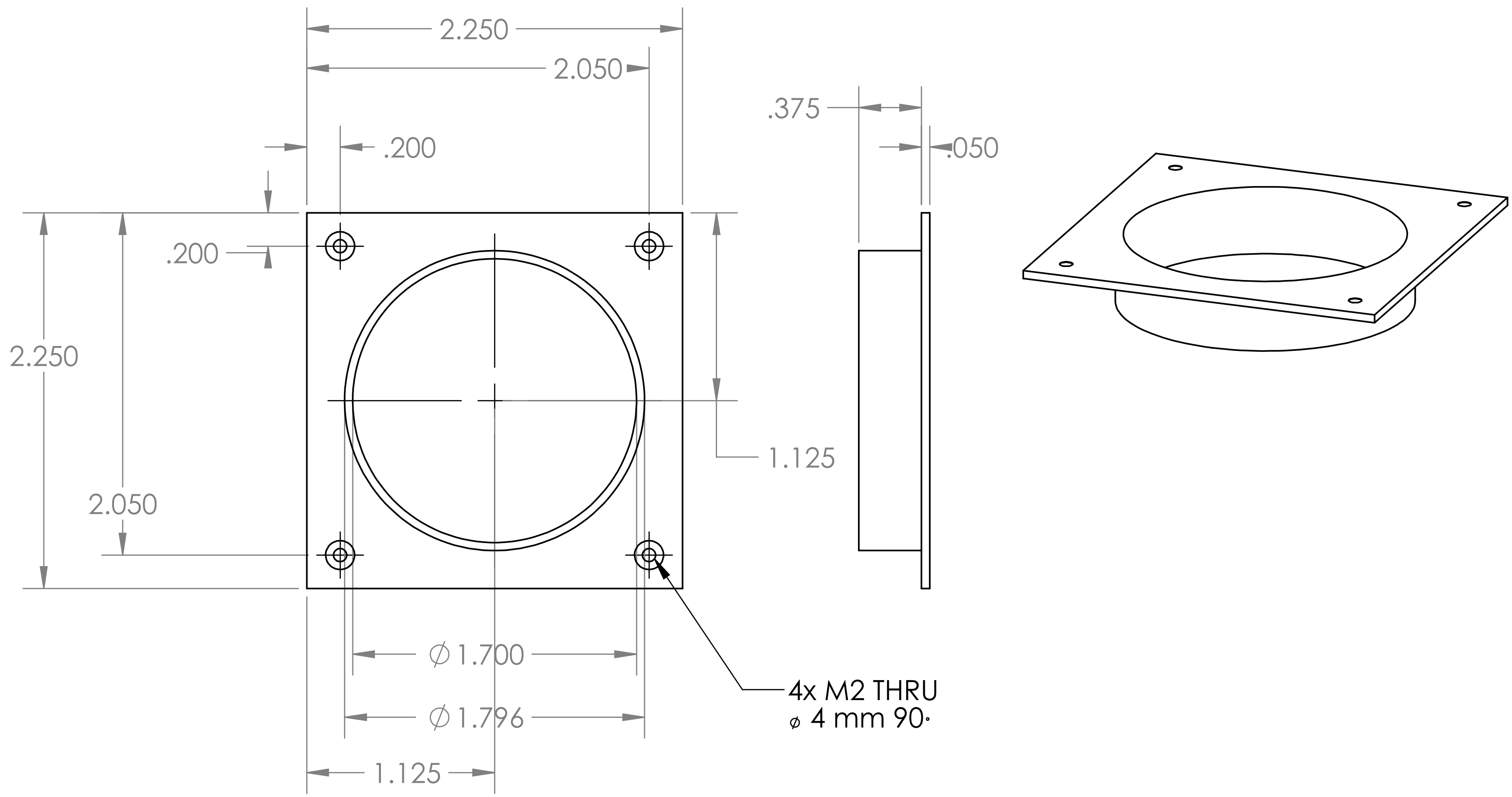
SCALE: 1:1




| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
|---|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ±.0005 | DRAWN | IT | | TITLE: |
| | | | CHECKED | | | |
| | | | ENG APPR. | | | |
| | | | MFG APPR. | | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| | | MATERIAL | COMMENTS: | | | SIZE A / DWG. NO. ring fixture plate / REV |
| NEXT ASSY | USED ON | FINISH | | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | | SCALE: 1:1 / WEIGHT: / SHEET 1 OF 1 |

2 1

B B




A A

| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
|---|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± .0005 | DRAWN | | | |
| | | | CHECKED | IT | | TITLE: |
| | | | ENG APPR. | | | |
| | | | MFG APPR. | | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| | | MATERIAL | COMMENTS: can use a 45 american drill for 2.1 mm | | | SIZE A DWG. NO. ring REV |
| NEXT ASSY | USED ON | FINISH | | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | | SCALE: 1:1 WEIGHT: SHEET 1 OF 1 |

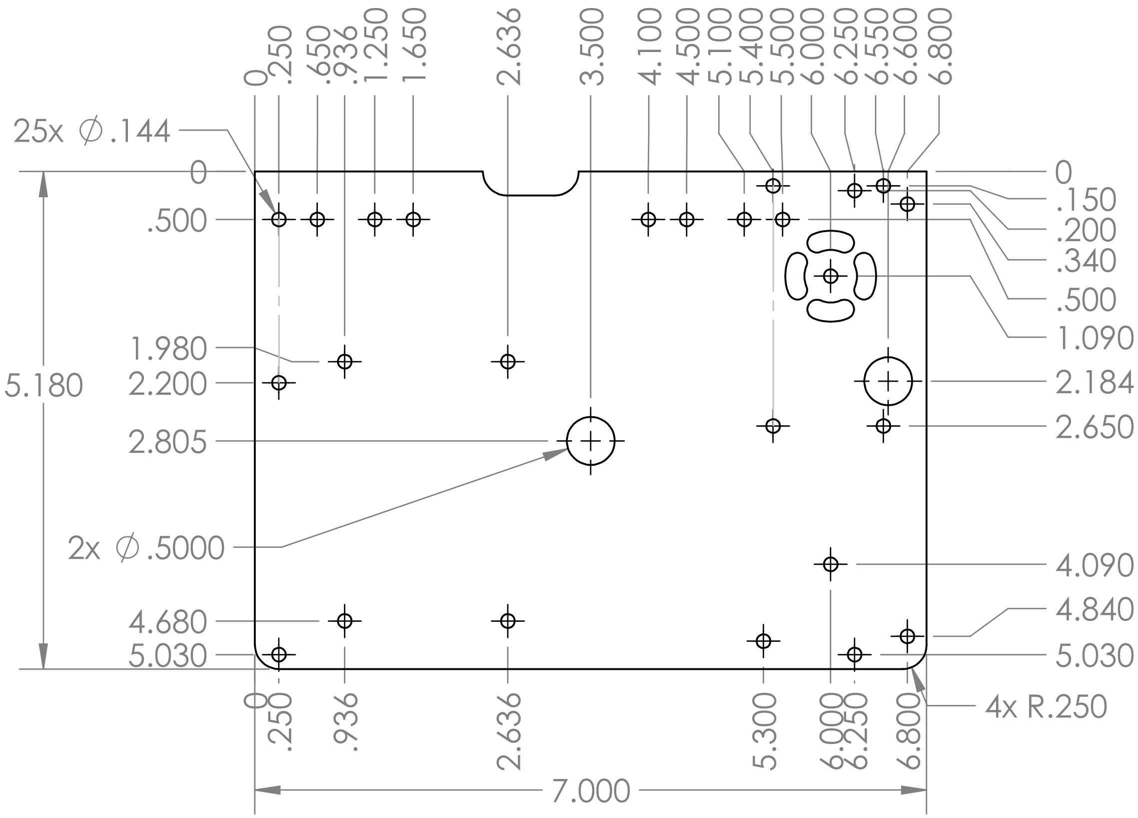




2 1

# APPENDIX C:
# Z-ACTUATOR ENGINEERING DRAWINGS

For Aid in CNC Machining




UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

1/4" Carbon Fiber Laminate

TITLE:

# 01_Plate_A

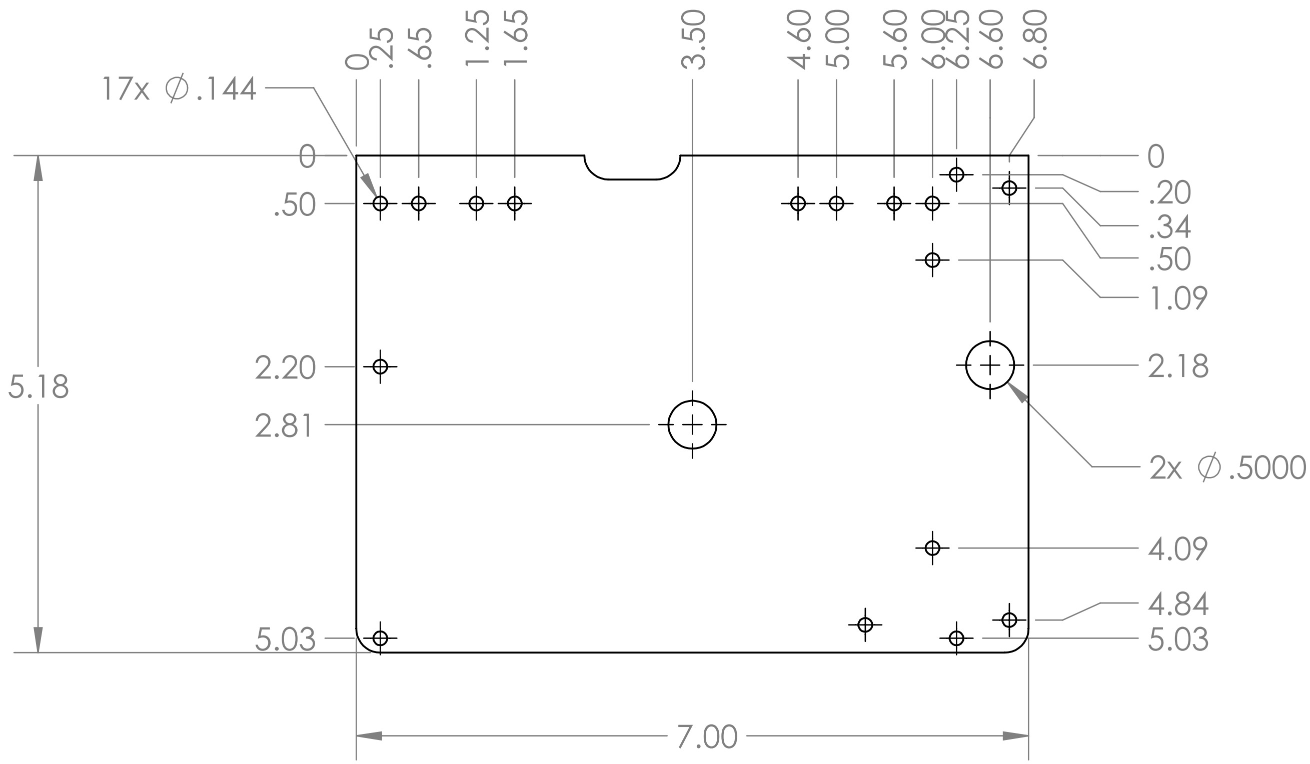
SIZE A | DWG. NO. | REV

SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1

NEXT ASSY | USED ON

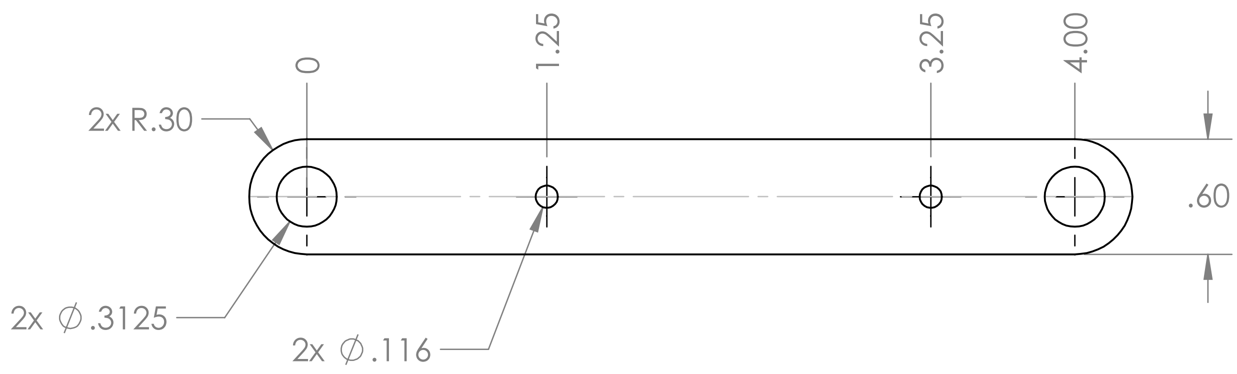
APPLICATION

**PROPRIETARY AND CONFIDENTIAL**




| | | | | |
|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
| DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | | TITLE:<br>02_Plate_B |
| | CHECKED | | | |
| | ENG APPR. | | | |
| | MFG APPR. | | | |
| INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| MATERIAL | COMMENTS:<br>1/4" Carbon Fiber Laminate | | | SIZE A | DWG. NO. | REV |
| FINISH | | | | |
| DO NOT SCALE DRAWING | | | | SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1 |

| | |
|---|---|
| NEXT ASSY | USED ON |
| APPLICATION | |




PROPRIETARY AND CONFIDENTIAL



| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

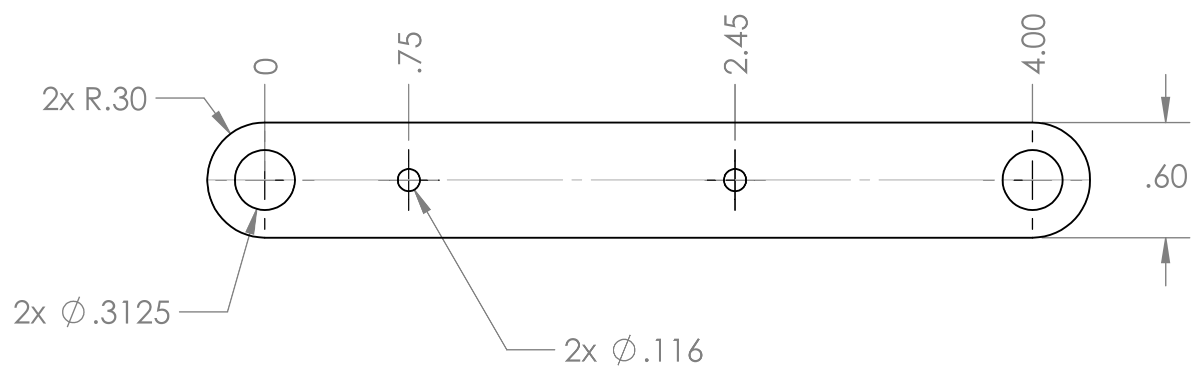
1/4" Carbon Fiber Laminate

TITLE:

# 03_Arm_A

SIZE A | DWG. NO. | REV

SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1




| | | | | |
|---|---|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN<br>CHECKED<br>ENG APPR.<br>MFG APPR.<br>Q.A. | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | COMMENTS: | | |
| | | MATERIAL | 1/4" Carbon Fiber Laminate | | |
| NEXT ASSY | USED ON | FINISH | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | |

**PROPRIETARY AND CONFIDENTIAL**

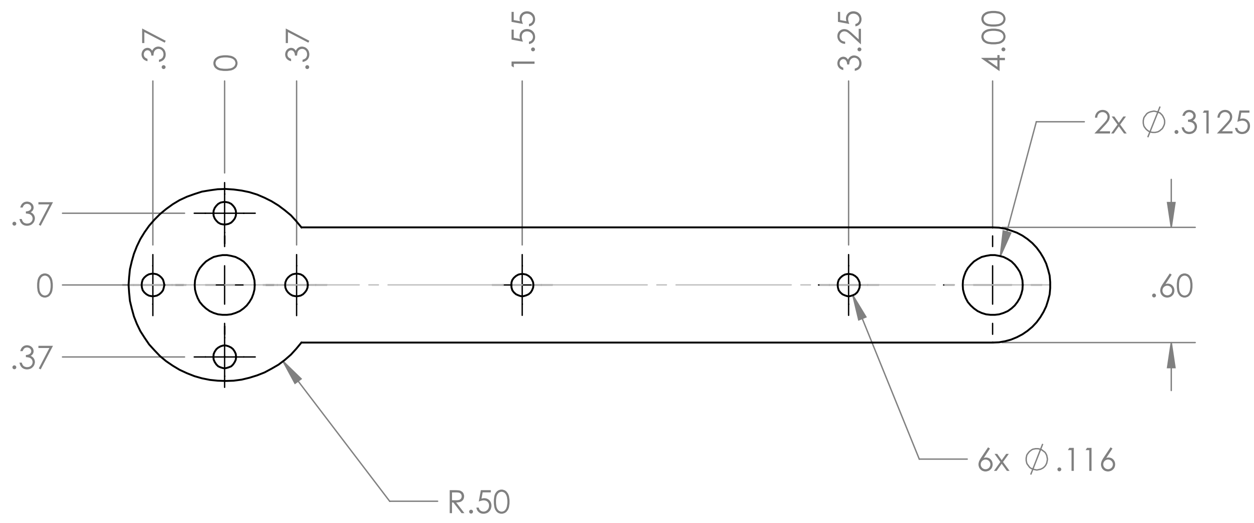
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

TITLE:

# 04_Arm_B

SIZE **A** DWG. NO. REV

SCALE: 1:1 WEIGHT: SHEET 1 OF 1




**PROPRIETARY AND CONFIDENTIAL**



NEXT ASSY | USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

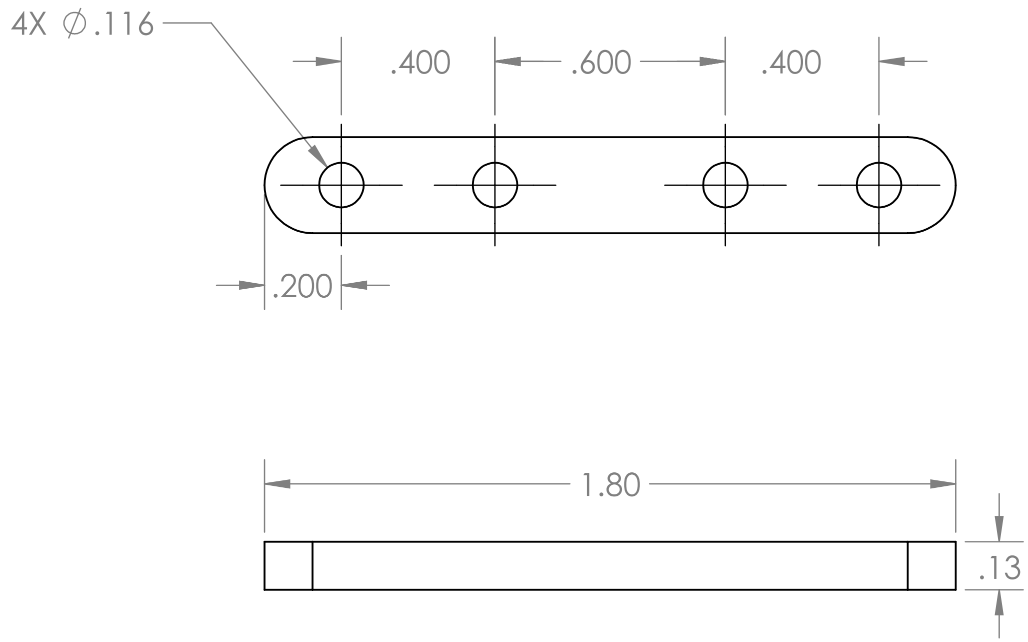
1/4" Carbon Fiber Laminate

TITLE:

# 05_Arm_C

SIZE **A** | DWG. NO. | REV

SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1




| | | |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| NEXT ASSY | USED ON | |
| APPLICATION | | DO NOT SCALE DRAWING |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

| | NAME | DATE |
|---|---|---|
| DRAWN | LM | 3/19 |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

4x 6061 Aluminum

**PROPRIETARY AND CONFIDENTIAL**

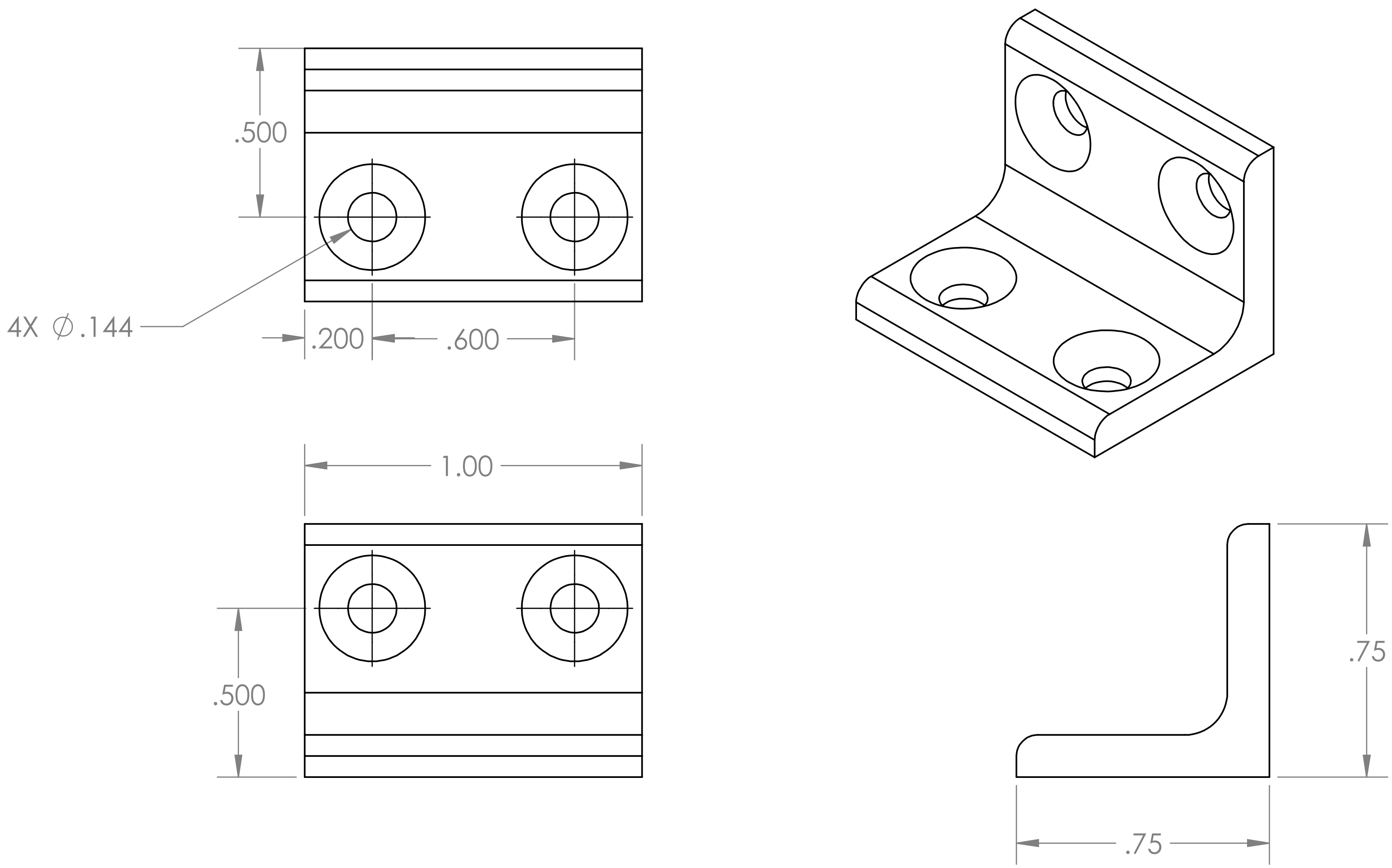


TITLE:

# 06_Plate_Mount

| SIZE | DWG. NO. | REV |
|---|---|---|
| **A** | | |

| SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1 |
|---|---|---|




PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

NEXT ASSY | USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | LM | 3/19 |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

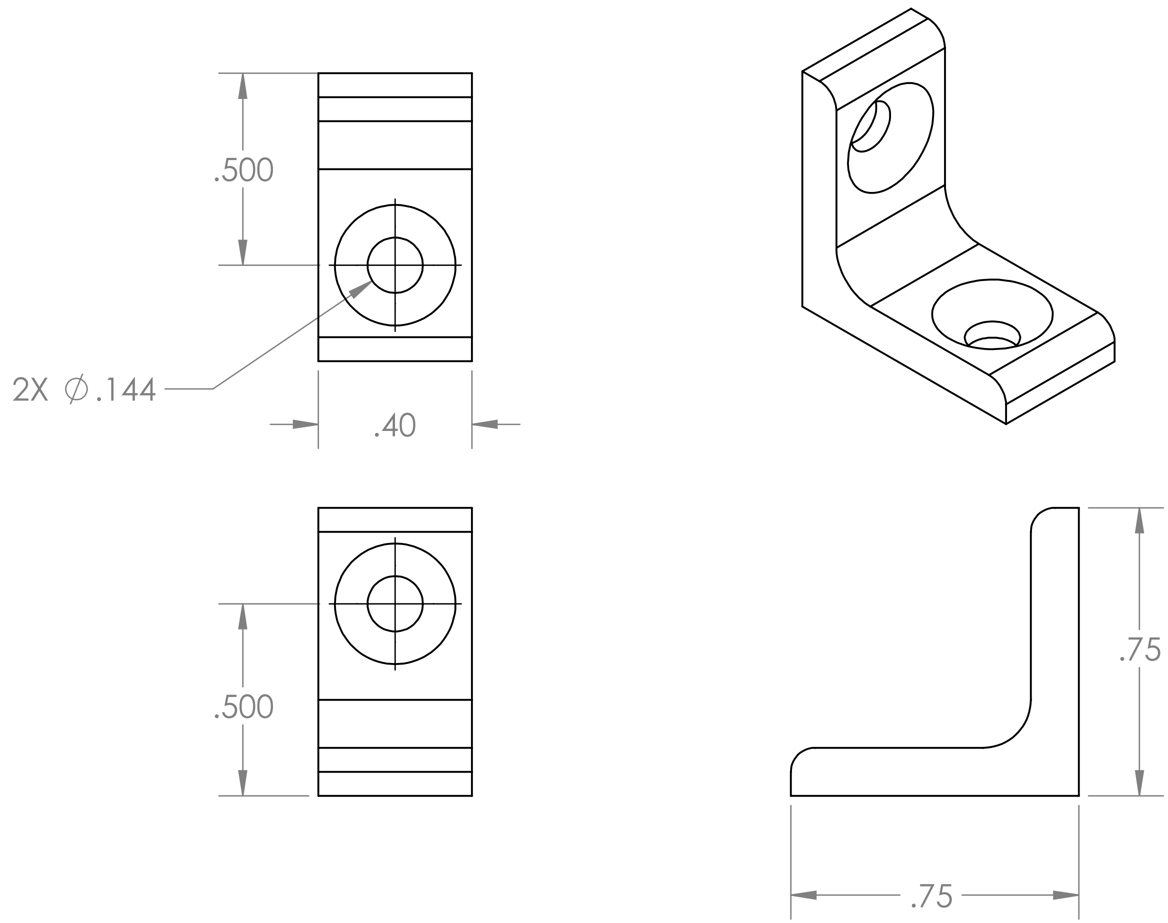
4X 6061 Aluminum

TITLE:

07_L-Bracket_A

SIZE A | DWG. NO. | REV

SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| | |
|---|---|
| NEXT ASSY | USED ON |
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | LM | 3/19 |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

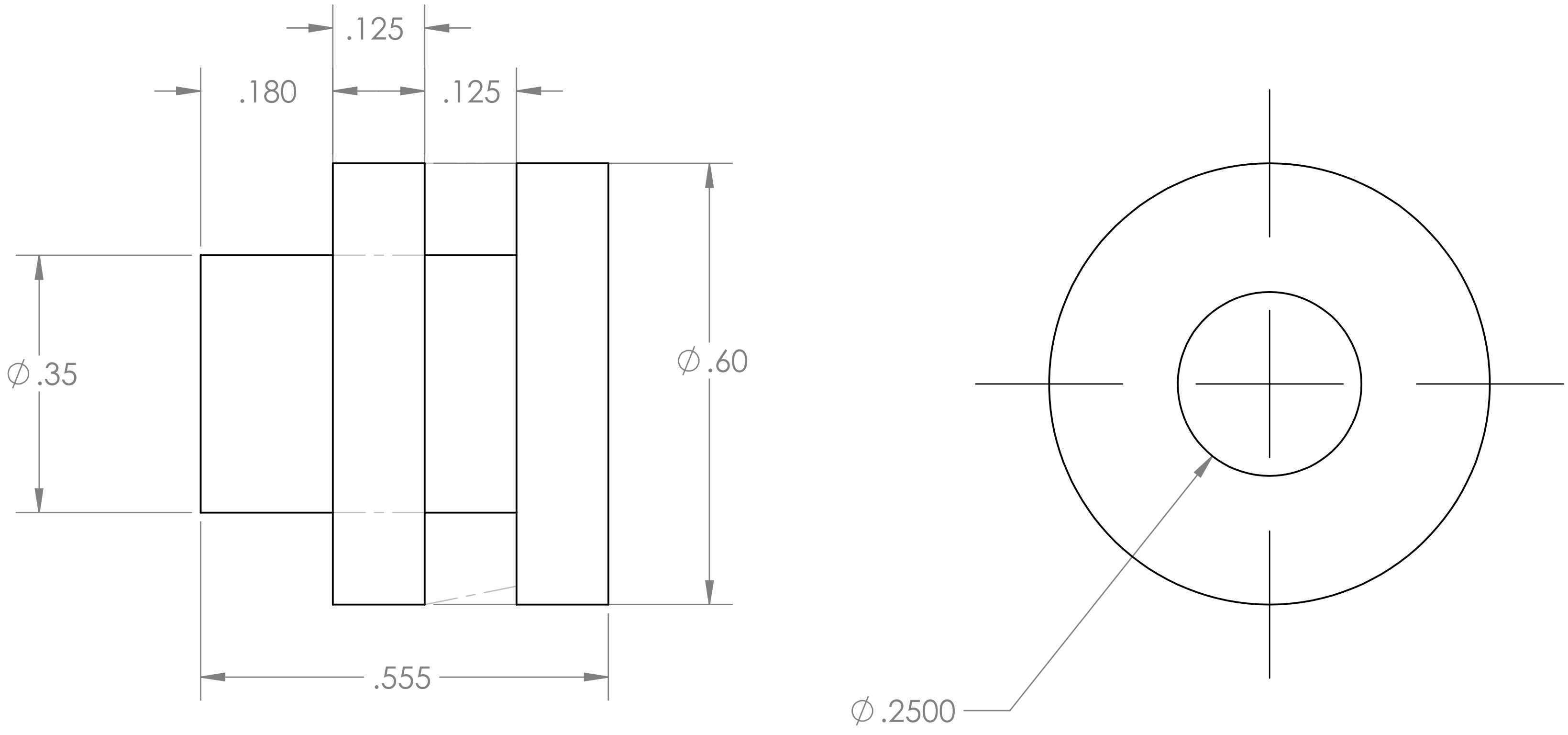
4X 6061 Aluminum

TITLE:

# 08_L-Bracket_B

SIZE **A** DWG. NO. REV

SCALE: 2:1 WEIGHT: SHEET 1 OF 1

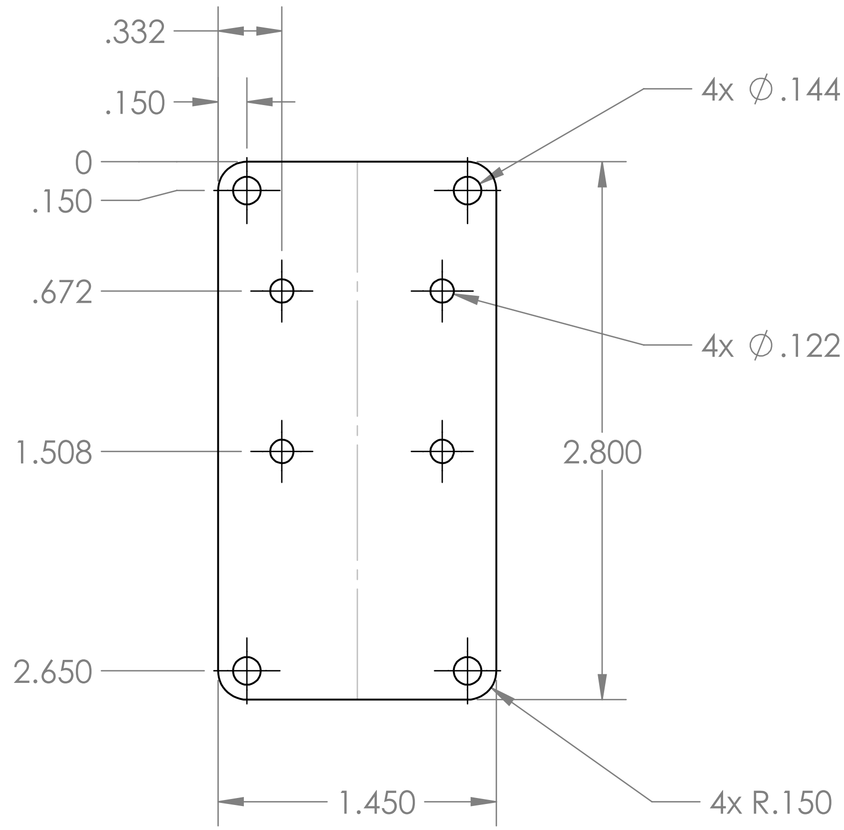



| | | | | | |
|---|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | | |
| DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | LM | 3/19 | TITLE: | |
| | CHECKED | | | 09_Idler_Pulley | |
| | ENG APPR. | | | | |
| | MFG APPR. | | | | |
| INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | | |
| MATERIAL | COMMENTS: | | | SIZE A | DWG. NO. | 
| FINISH | 2X 6061 Aluminum | | | | REV |
| DO NOT SCALE DRAWING | | | | SCALE: 4:1 | WEIGHT: SHEET 1 OF 1 |

| | |
|---|---|
| NEXT ASSY | USED ON |
| APPLICATION | |

**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

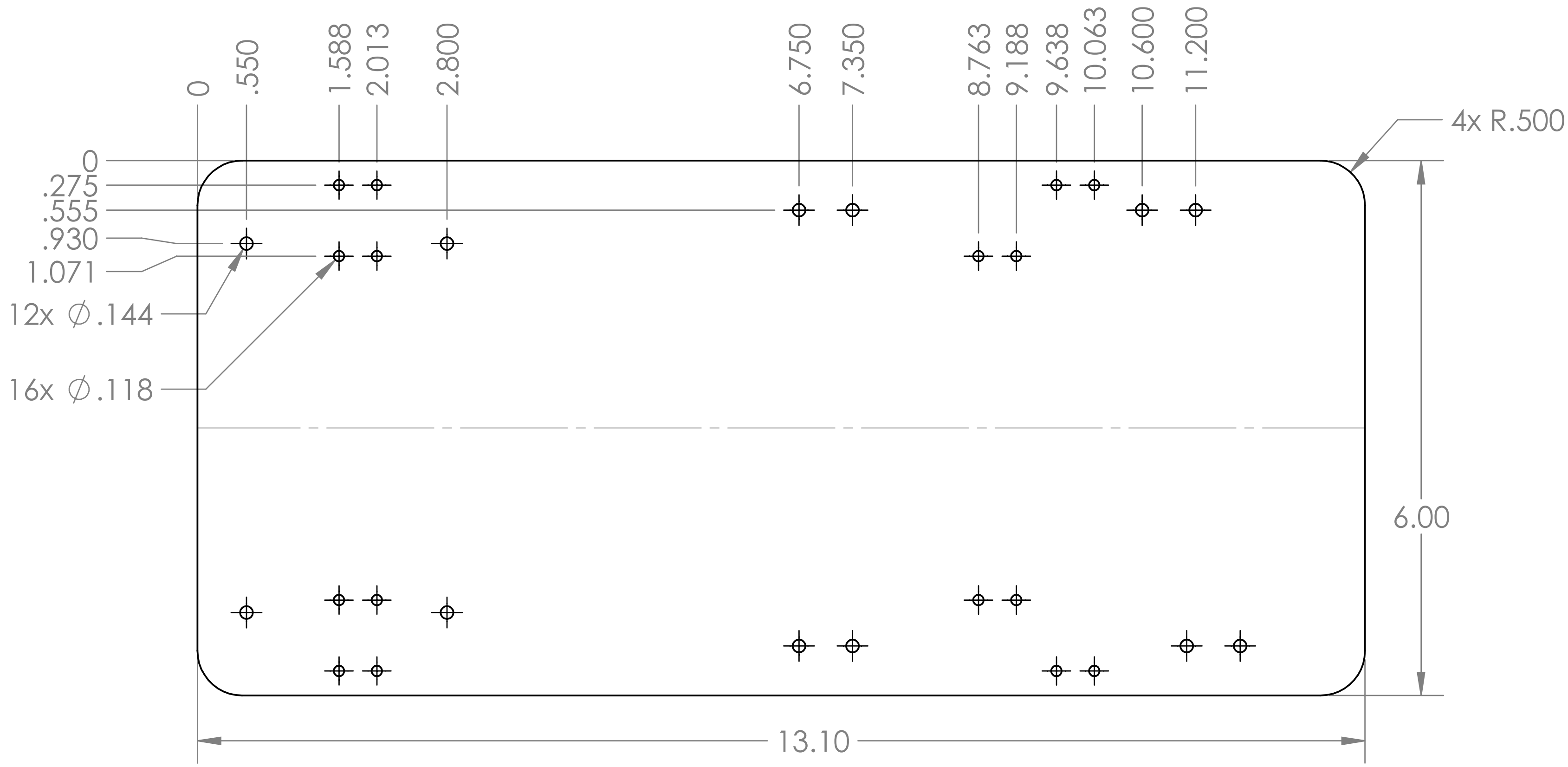
1/4" Carbon Fiber Laminate

TITLE:

# 10_Motor_Plate

| SIZE | DWG. NO. | REV |
|---|---|---|
| A | | |

SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1




PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

NEXT ASSY | USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

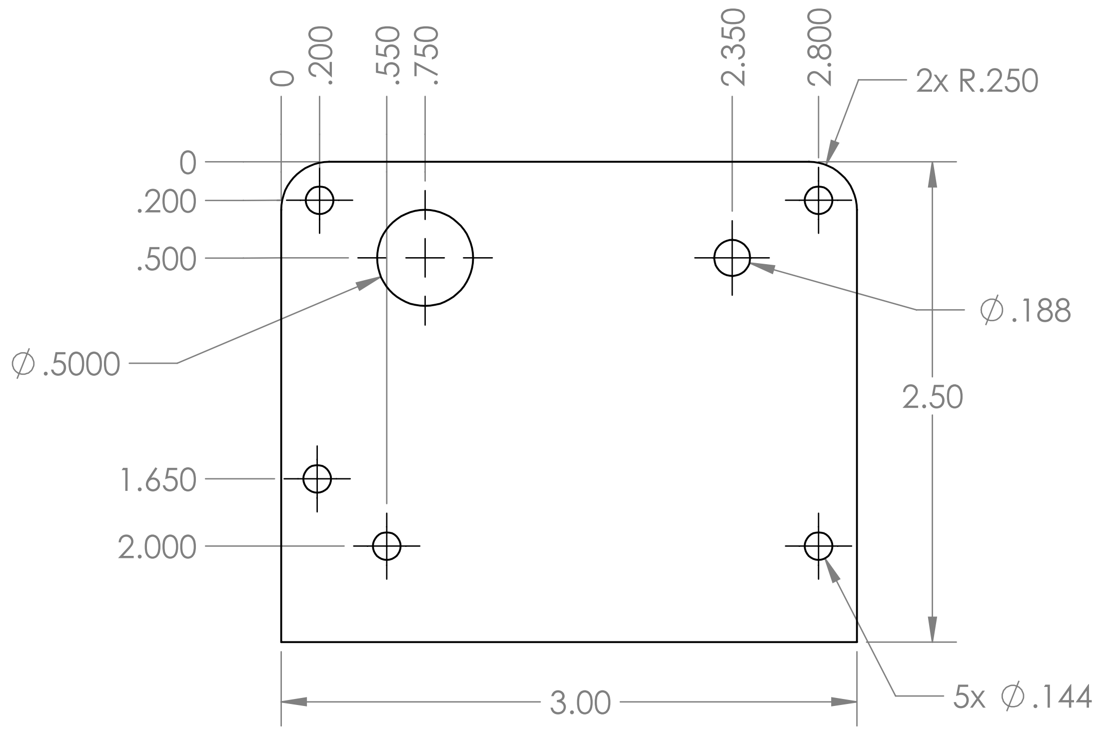
1/4" Carbon Fiber Laminate

TITLE:

# 11_Bottom_Plate

SIZE A | DWG. NO. | REV

SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

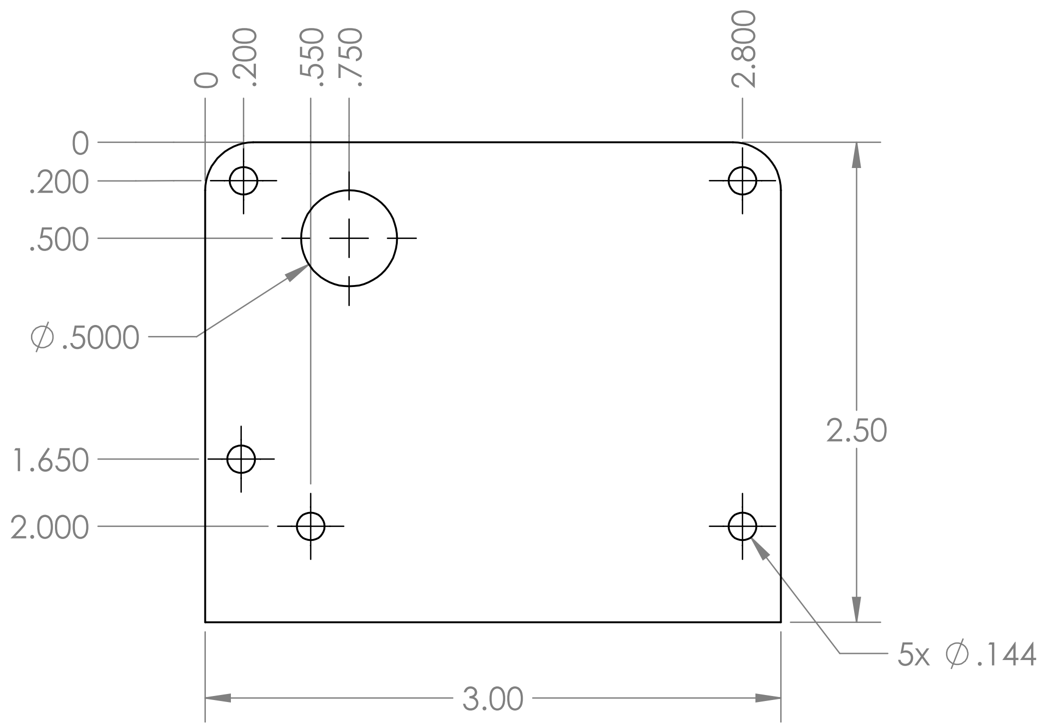
1/4" Carbon Fiber Laminate

TITLE:

12_Spring_Plate_A

SIZE A | DWG. NO. | REV

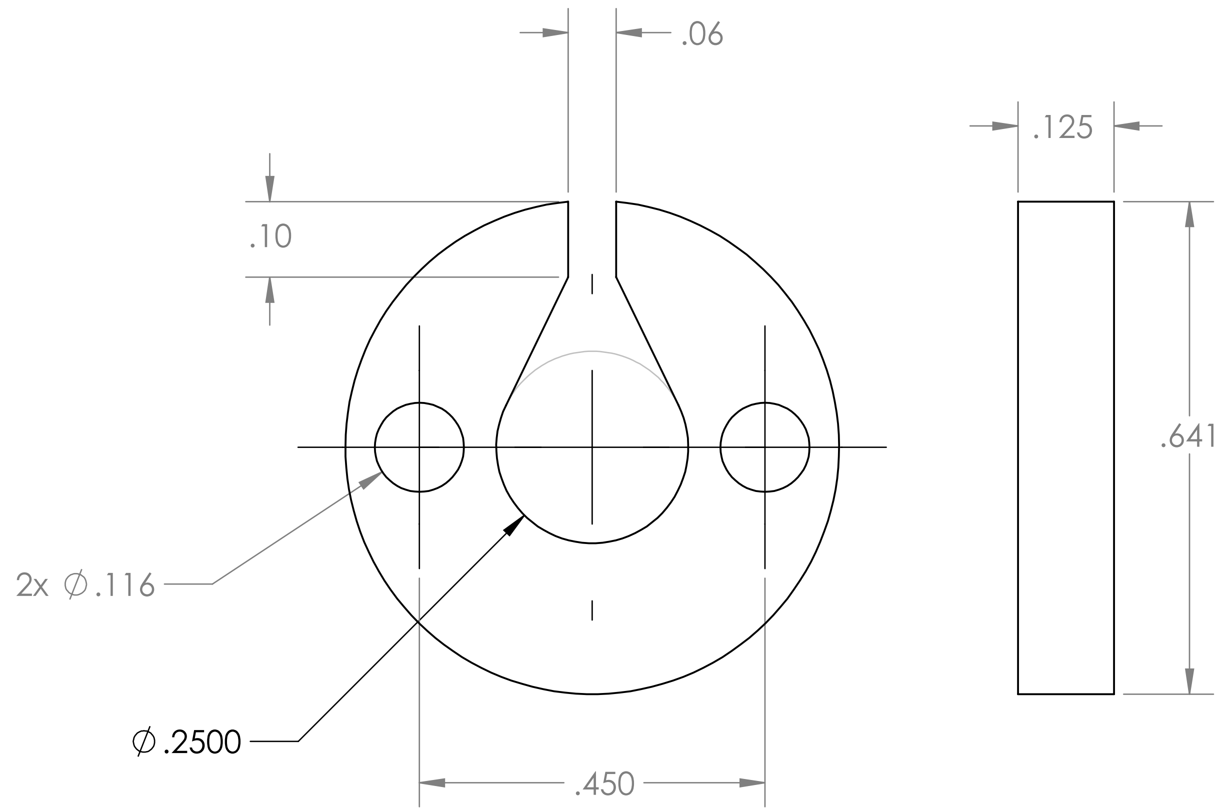
SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1




| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
|---|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | | |
| | | | CHECKED | | | TITLE: |
| | | | ENG APPR. | | | 13_Spring_Plate_B |
| | | | MFG APPR. | | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| | | MATERIAL | COMMENTS:<br>1/4" Carbon Fiber Laminate | | | SIZE A · DWG. NO. · REV |
| NEXT ASSY | USED ON | FINISH | | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | | SCALE: 1:1 · WEIGHT: · SHEET 1 OF 1 |

**PROPRIETARY AND CONFIDENTIAL**

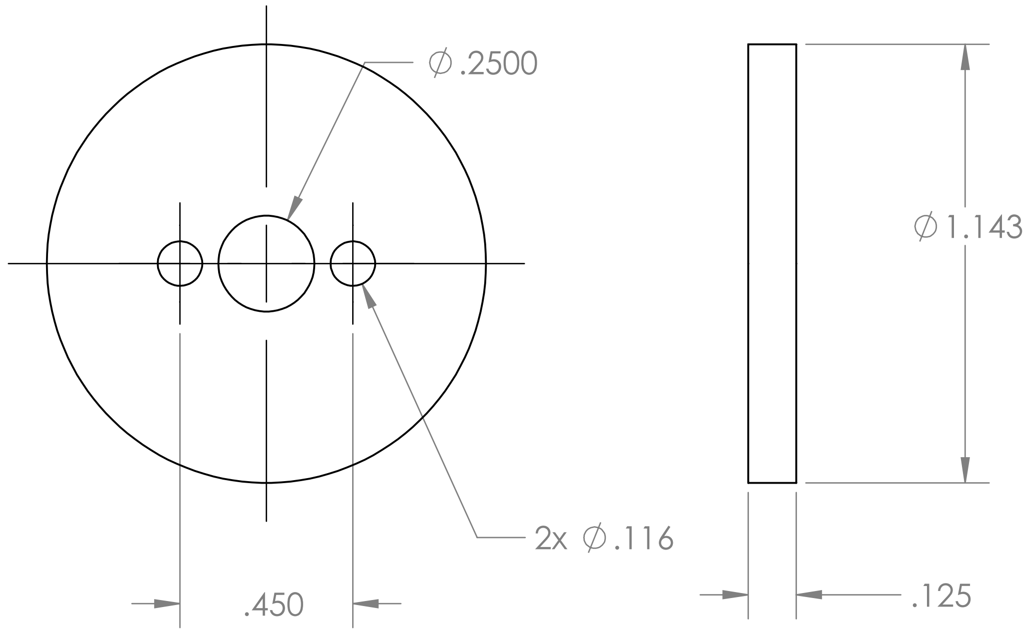
FOR AID IN CNC MACHINING




| | | | | | | |
|---|---|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | | | |
| DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | LM | 3/20 | TITLE: 14_Core_A | | |
| | CHECKED | | | | | |
| | ENG APPR. | | | | | |
| | MFG APPR. | | | | | |
| INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | | | |
| MATERIAL | COMMENTS: 303 Stainless Steel | | | SIZE A | DWG. NO. | REV |
| FINISH | | | | SCALE: 4:1 | WEIGHT: | SHEET 1 OF 1 |
| DO NOT SCALE DRAWING | | | | | | |

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

**PROPRIETARY AND CONFIDENTIAL**




**PROPRIETARY AND CONFIDENTIAL**



| | |
|---|---|
| NEXT ASSY | USED ON |
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

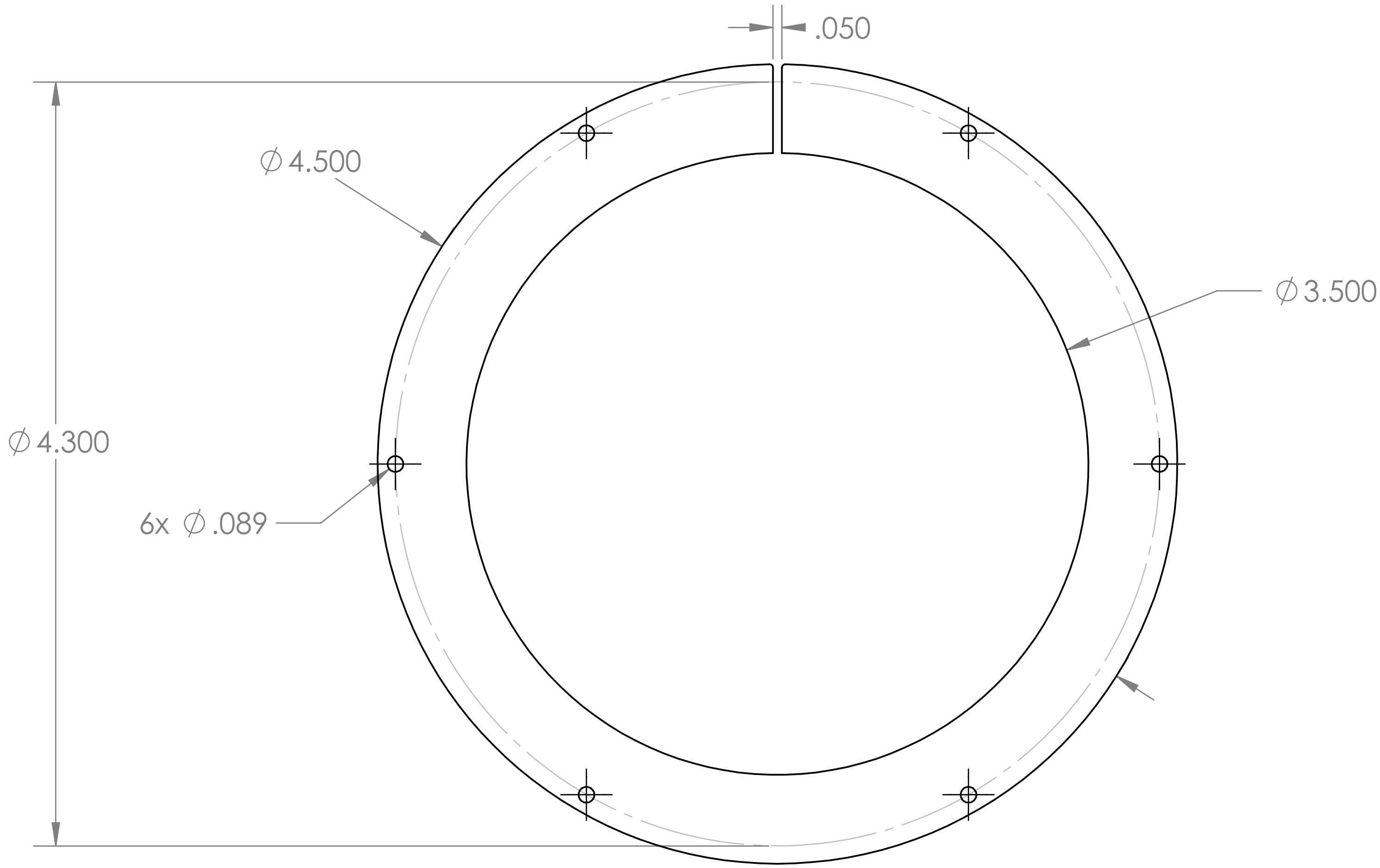
2x 1/8" Carbon Fiber

TITLE:

# 15_Flange_A

| SIZE | DWG. NO. | REV |
|---|---|---|
| **A** | | |

SCALE: 2:1 WEIGHT: SHEET 1 OF 1




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

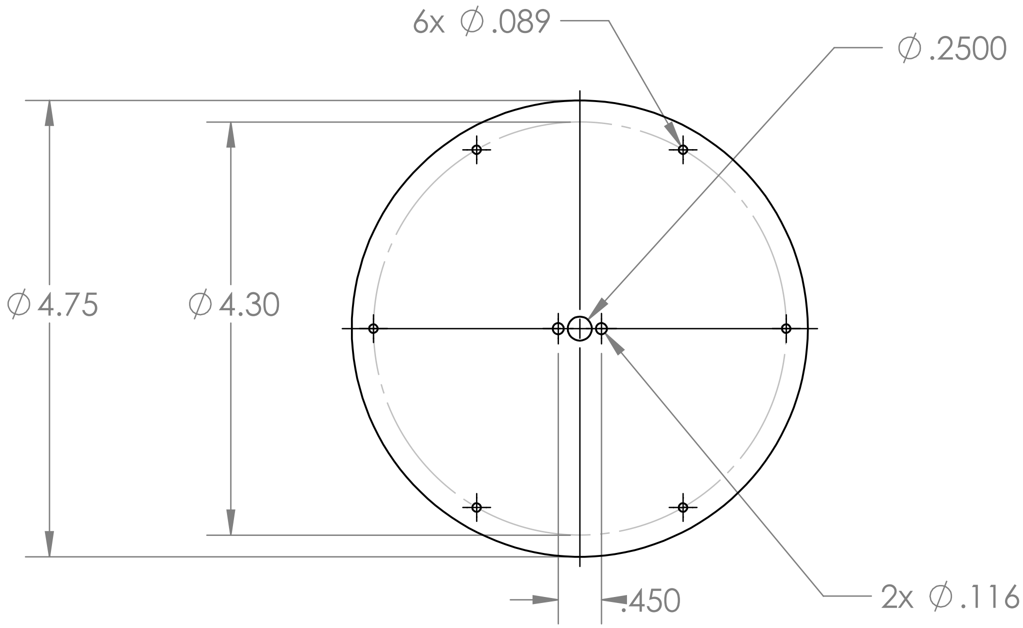
2x 1/8'' Acrylic Plastic

TITLE:

# 16_Core_B

SIZE **A** DWG. NO. REV

SCALE: 1:1 WEIGHT: SHEET 1 OF 1




| | | | | | |
|---|---|---|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | |
| | | | CHECKED | | |
| | | | ENG APPR. | | |
| | | | MFG APPR. | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | |
| | | MATERIAL | COMMENTS:<br>4x 1/8" Carbon Fiber | | |
| NEXT ASSY | USED ON | FINISH | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | |

**PROPRIETARY AND CONFIDENTIAL**

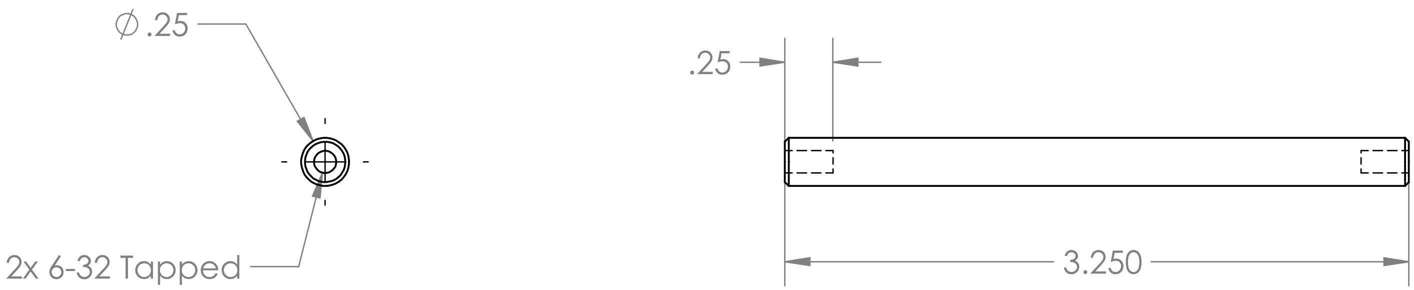
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

TITLE:

# 17_Flange_B

| SIZE | DWG. NO. | REV |
|---|---|---|
| **A** | | |

SCALE: 1:2 WEIGHT: SHEET 1 OF 1




# 18_Support_Rod_A

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

4x 6061 Aluminum

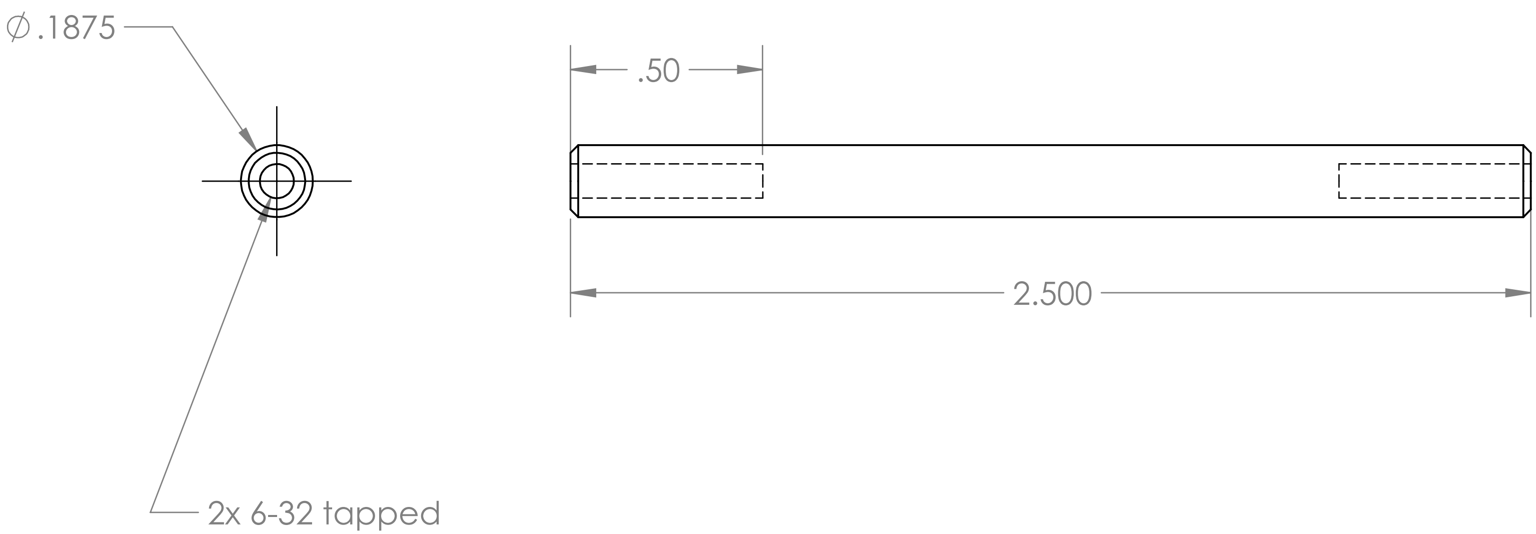
| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |



TITLE: 18_Support_Rod_A

SIZE A | DWG. NO. | REV

SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1




| | | | | | |
|---|---|---|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | |
| | | | CHECKED | | |
| | | | ENG APPR. | | |
| | | | MFG APPR. | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | |
| | | MATERIAL | COMMENTS: 4x 303 Stainless Steel | | |
| NEXT ASSY | USED ON | FINISH | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | |

**PROPRIETARY AND CONFIDENTIAL**

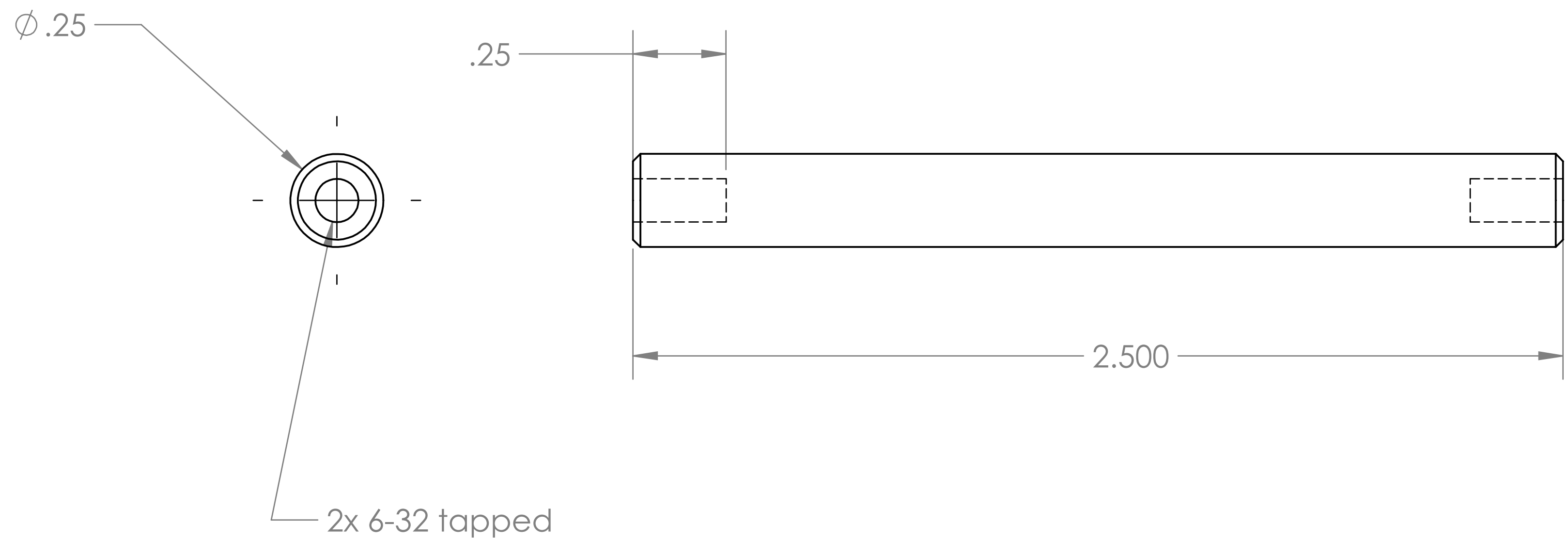
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

TITLE:

# 19_Support_Rod_B

| SIZE | DWG. NO. | REV |
|---|---|---|
| **A** | | |

| | | |
|---|---|---|
| SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1 |




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

6061 Aluminum

TITLE:

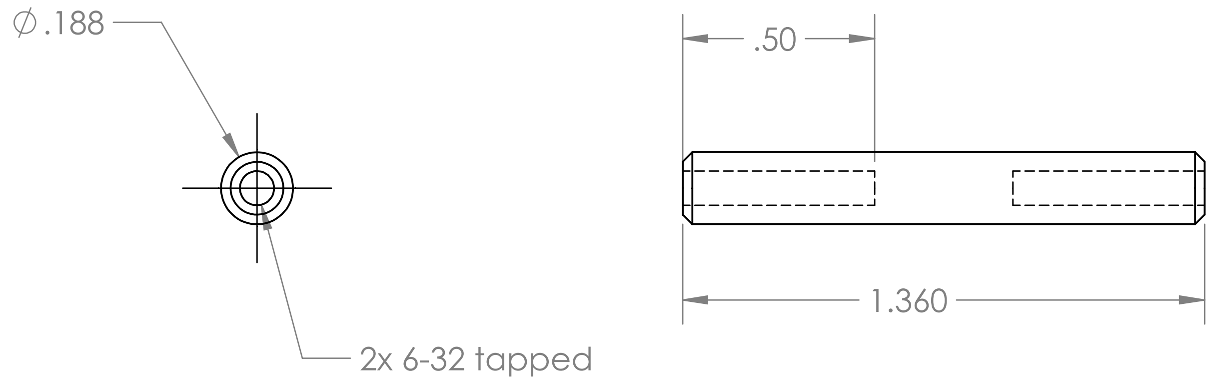
20_Support_Rod_C

SIZE **A** DWG. NO. REV

SCALE: 2:1 WEIGHT: SHEET 1 OF 1

2 1

B B



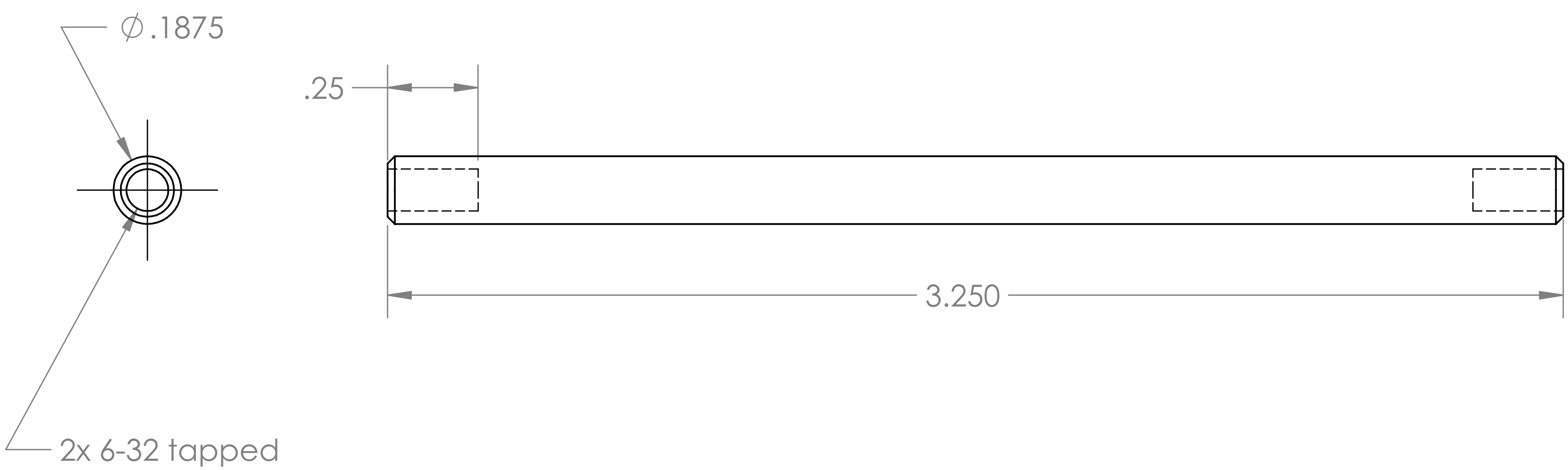

A A

| | | | | | | |
|---|---|---|---|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | | TITLE: |
| | | | CHECKED | | | 21_Support_Rod_D |
| | | | ENG APPR. | | | |
| | | | MFG APPR. | | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| | | MATERIAL | COMMENTS:<br>4x 303 Stainless Steel | | | SIZE A / DWG. NO. / REV |
| NEXT ASSY | USED ON | FINISH | | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | | SCALE: 2:1 / WEIGHT: / SHEET 1 OF 1 |





2 1




**PROPRIETARY AND CONFIDENTIAL**



NEXT ASSY | USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

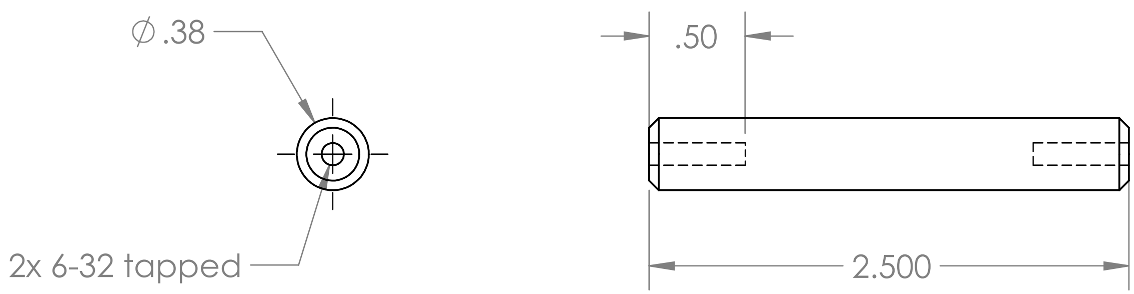
5x 303 Stainless Steel

TITLE:

# 22_Pivot_Rod

SIZE **A** | DWG. NO. | REV

SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1



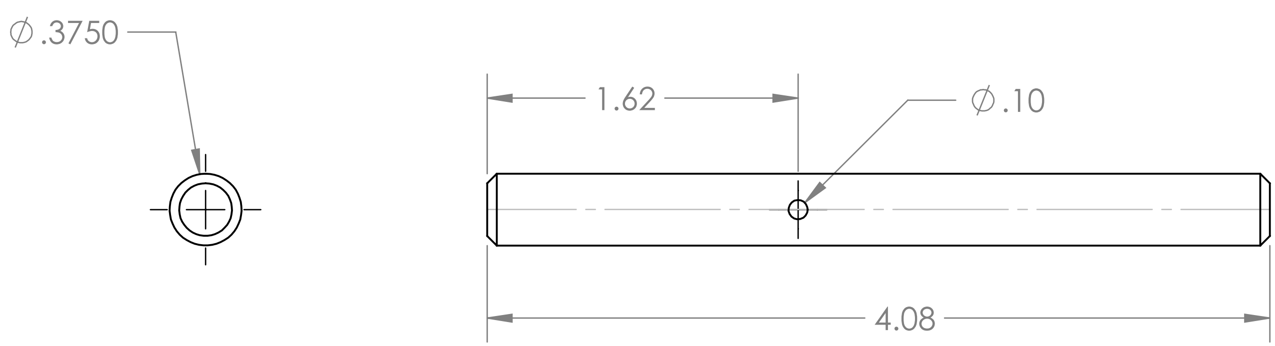

| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
|---|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | | TITLE:<br>23_Idler_Shaft |
| | | | CHECKED | | | |
| | | | ENG APPR. | | | |
| | | | MFG APPR. | | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| | | MATERIAL | COMMENTS:<br>303 Stainless Steel | | | SIZE A · DWG. NO. · REV |
| NEXT ASSY | USED ON | FINISH | | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | | SCALE: 1:1 · WEIGHT: · SHEET 1 OF 1 |

**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

2 1

B B




A A

| | | |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| NEXT ASSY | USED ON | |
| APPLICATION | | DO NOT SCALE DRAWING |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

303 Stainless Steel

TITLE:

# 24_Ratchet_Rod

SIZE **A** | DWG. NO. | REV

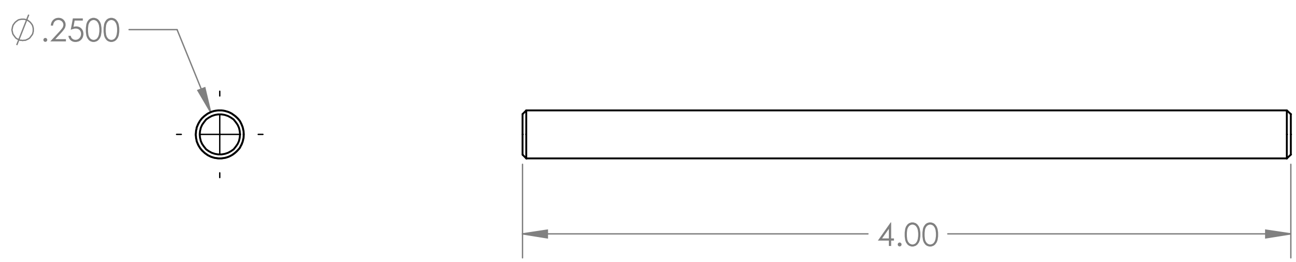
SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1





2 1

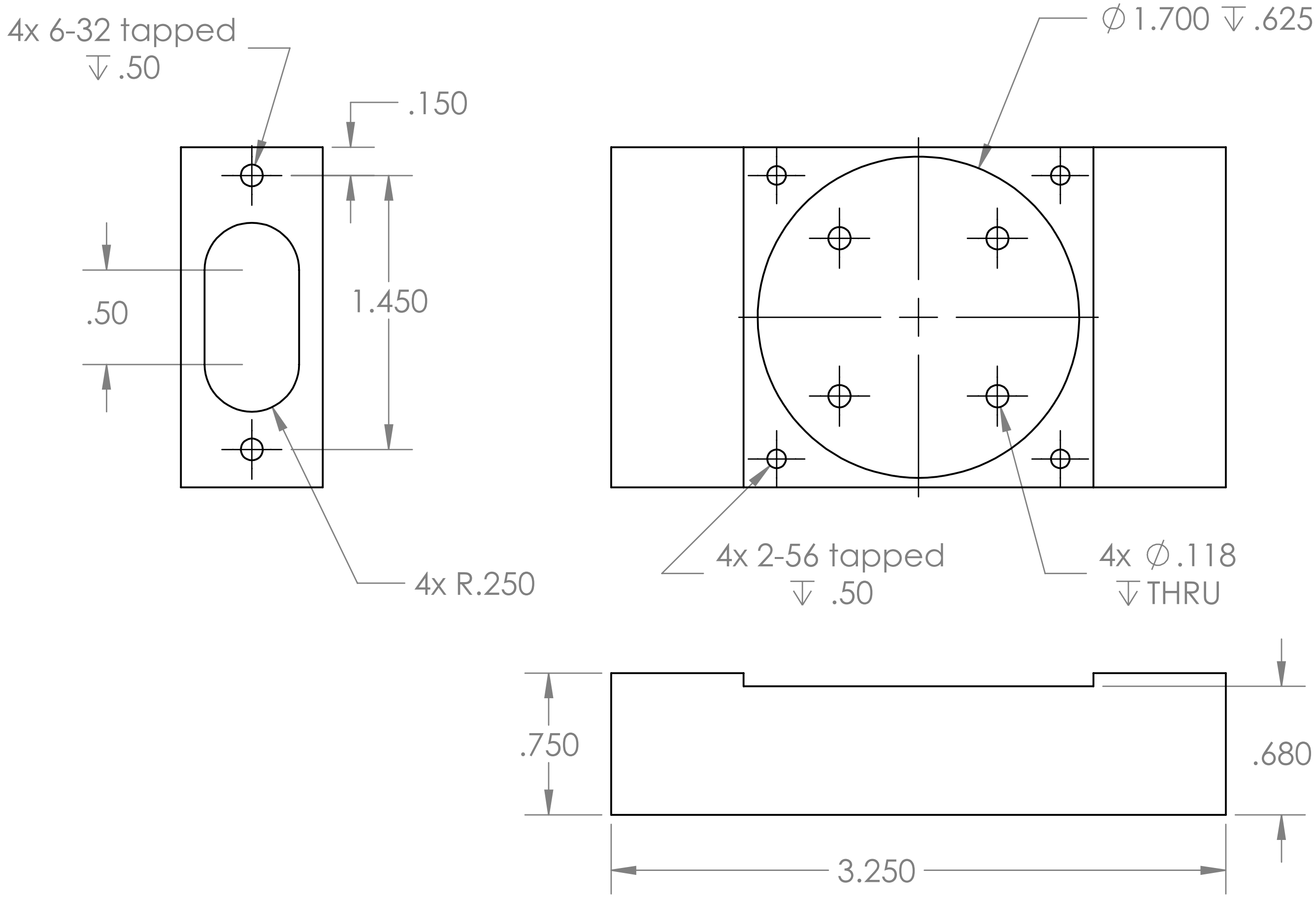
| 2 | 1 |
|---|---|




| | | UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | |
|---|---|---|---|---|---|---|
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | | TITLE: 25_Axel |
| | | | CHECKED | | | |
| | | | ENG APPR. | | | |
| | | | MFG APPR. | | | |
| | | INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | |
| | | MATERIAL | COMMENTS:<br>2x 303 Stainless Steel | | | SIZE A \| DWG. NO. \| REV |
| NEXT ASSY | USED ON | FINISH | | | | |
| APPLICATION | | DO NOT SCALE DRAWING | | | | SCALE: 1:1 \| WEIGHT: \| SHEET 1 OF 1 |

**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

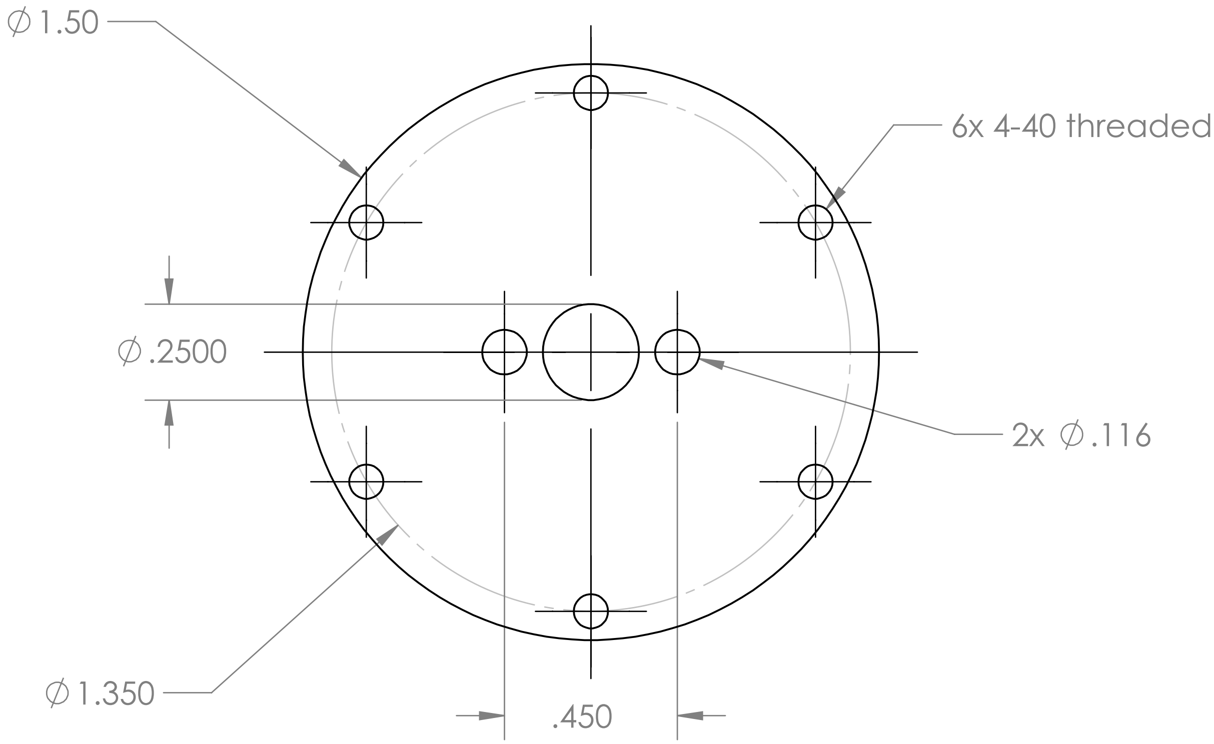
| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:
6061 Aluminum

TITLE:
# 26_Interface_Plate

SIZE **A** DWG. NO. REV

SCALE: 1:1 WEIGHT: SHEET 1 OF 1

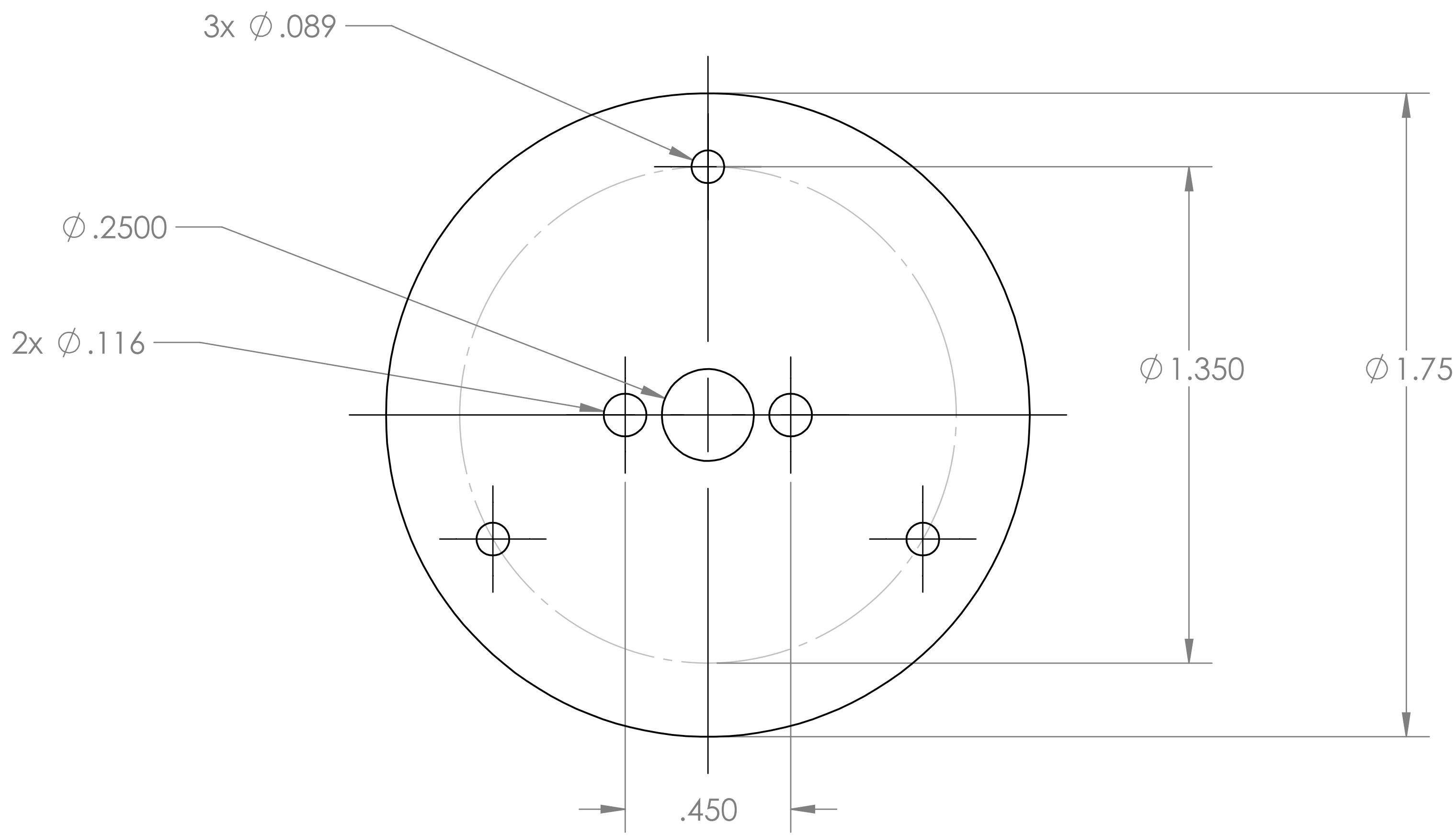



| | | | | | |
|---|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED: | | NAME | DATE | | |
| DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± | DRAWN | | | TITLE: | |
| | CHECKED | | | 27_Idler_Core | |
| | ENG APPR. | | | | |
| | MFG APPR. | | | | |
| INTERPRET GEOMETRIC TOLERANCING PER: | Q.A. | | | | |
| MATERIAL | COMMENTS:<br>1/8" 6061 Aluminum | | | SIZE A | DWG. NO. / REV |
| FINISH | | | | | |
| DO NOT SCALE DRAWING | | | | SCALE: 2:1 | WEIGHT: / SHEET 1 OF 1 |

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED




| | | |
|---|---|---|
| | | UNLESS OTHERWISE SPECIFIED: |
| | | DIMENSIONS ARE IN INCHES<br>TOLERANCES:<br>FRACTIONAL ±<br>ANGULAR: MACH ± BEND ±<br>TWO PLACE DECIMAL ±<br>THREE PLACE DECIMAL ± |
| | | INTERPRET GEOMETRIC TOLERANCING PER: |
| | | MATERIAL |
| NEXT ASSY | USED ON | FINISH |
| APPLICATION | | DO NOT SCALE DRAWING |

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

2x 1/8" Carbon Fiber

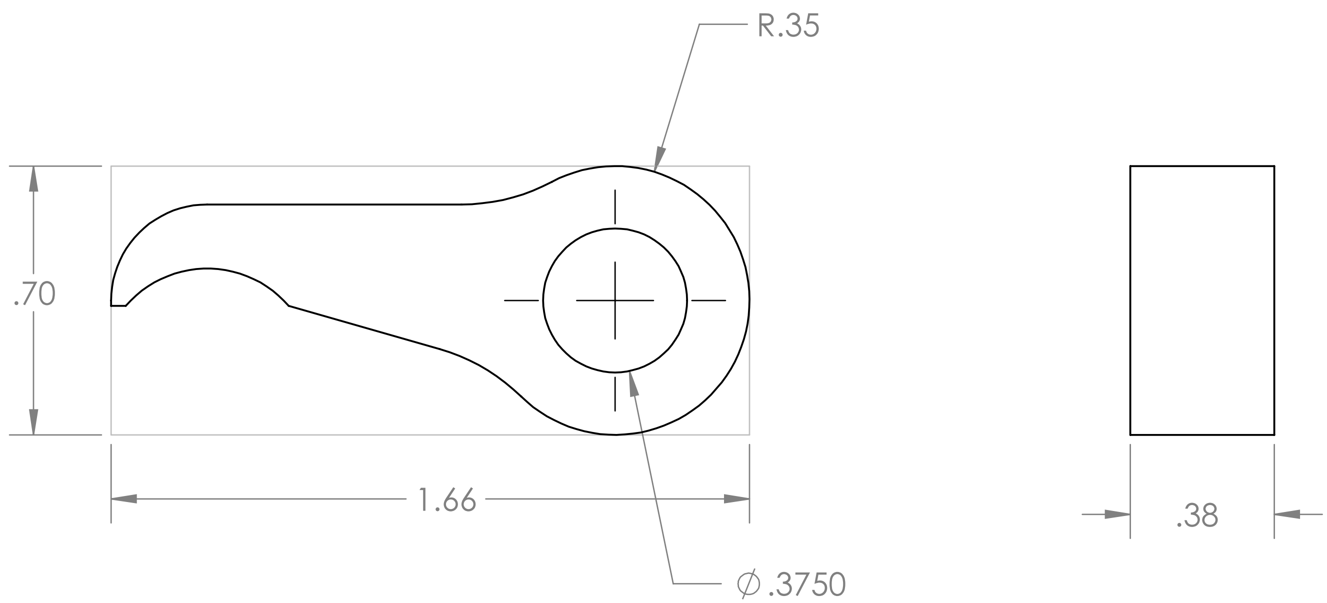
TITLE:

# 28_Idler_Flange

| SIZE | DWG. NO. | REV |
|---|---|---|
| A | | |

SCALE: 2:1 | WEIGHT: | SHEET 1 OF 1

For Aid in CNC Machining




**PROPRIETARY AND CONFIDENTIAL**

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF STABILIZE. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF STABILIZE IS PROHIBITED

| | |
|---|---|
| NEXT ASSY | USED ON |
| APPLICATION | |

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

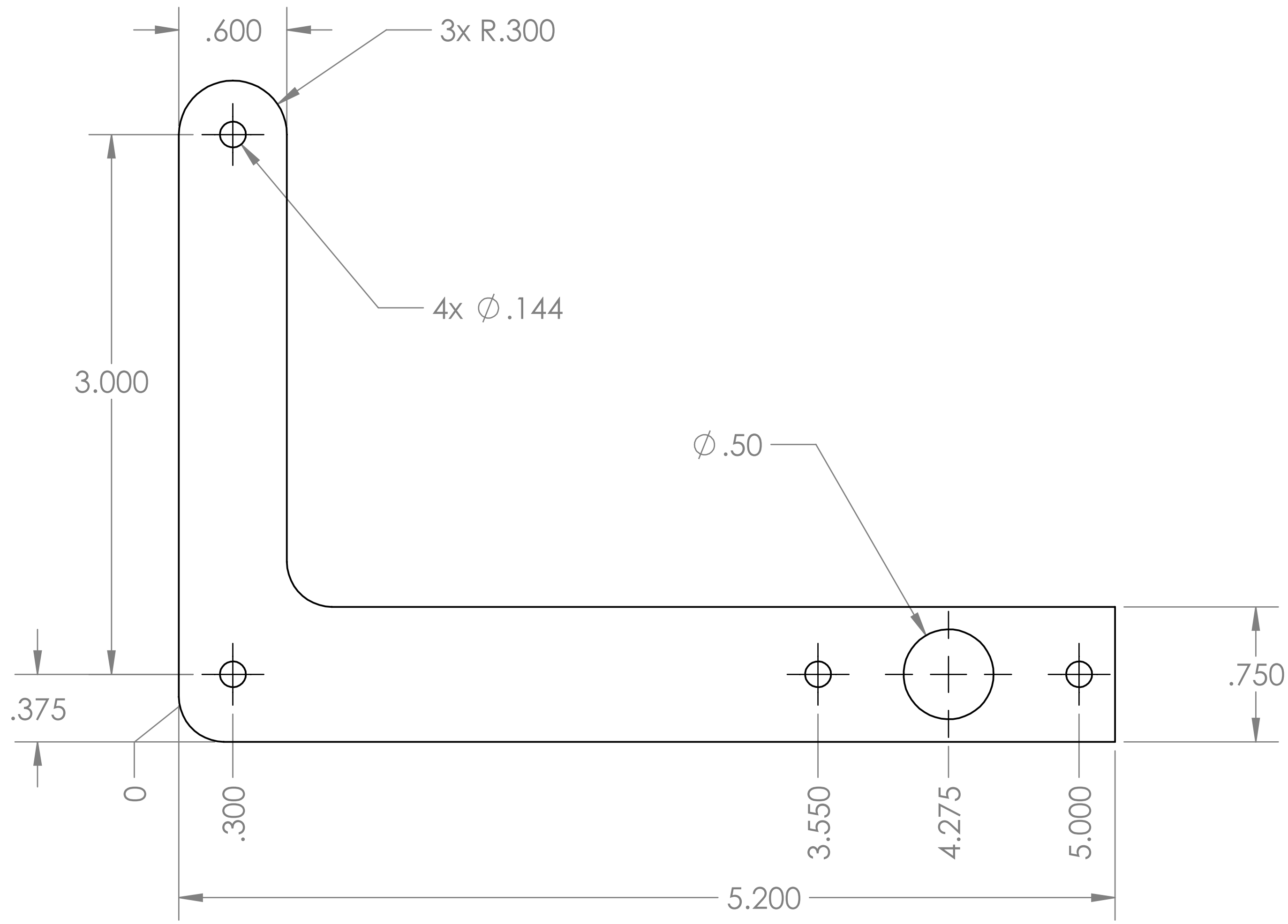
303 Stainless Steel

TITLE:

# 29_Pawl

| SIZE | DWG. NO. | REV |
|---|---|---|
| A | | |

SCALE: 2:1 WEIGHT: SHEET 1 OF 1




UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:

2x 1/4" Carbon Fiber Laminate

TITLE:

# 30_L_Arm

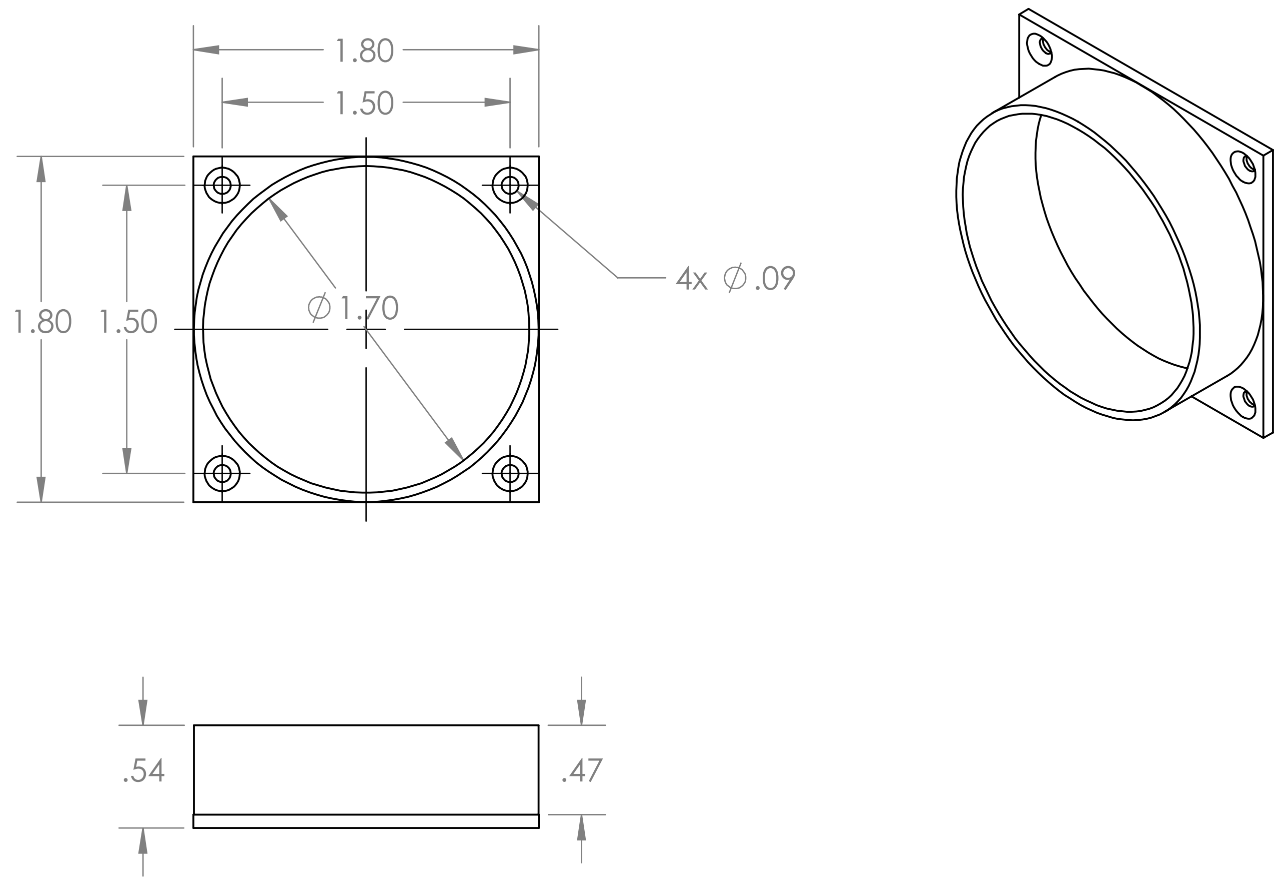
SIZE A DWG. NO. REV

SCALE: 1:1 WEIGHT: SHEET 1 OF 1

| NEXT ASSY | USED ON |
|---|---|
| APPLICATION | |

2 1

B B




A A

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

NEXT ASSY | USED ON
APPLICATION

| | NAME | DATE |
|---|---|---|
| DRAWN | | |
| CHECKED | | |
| ENG APPR. | | |
| MFG APPR. | | |
| Q.A. | | |

COMMENTS:
6061 Aluminum

TITLE:
# 31_Ring

SIZE **A** DWG. NO. REV
SCALE: 1:1 WEIGHT: SHEET 1 OF 1

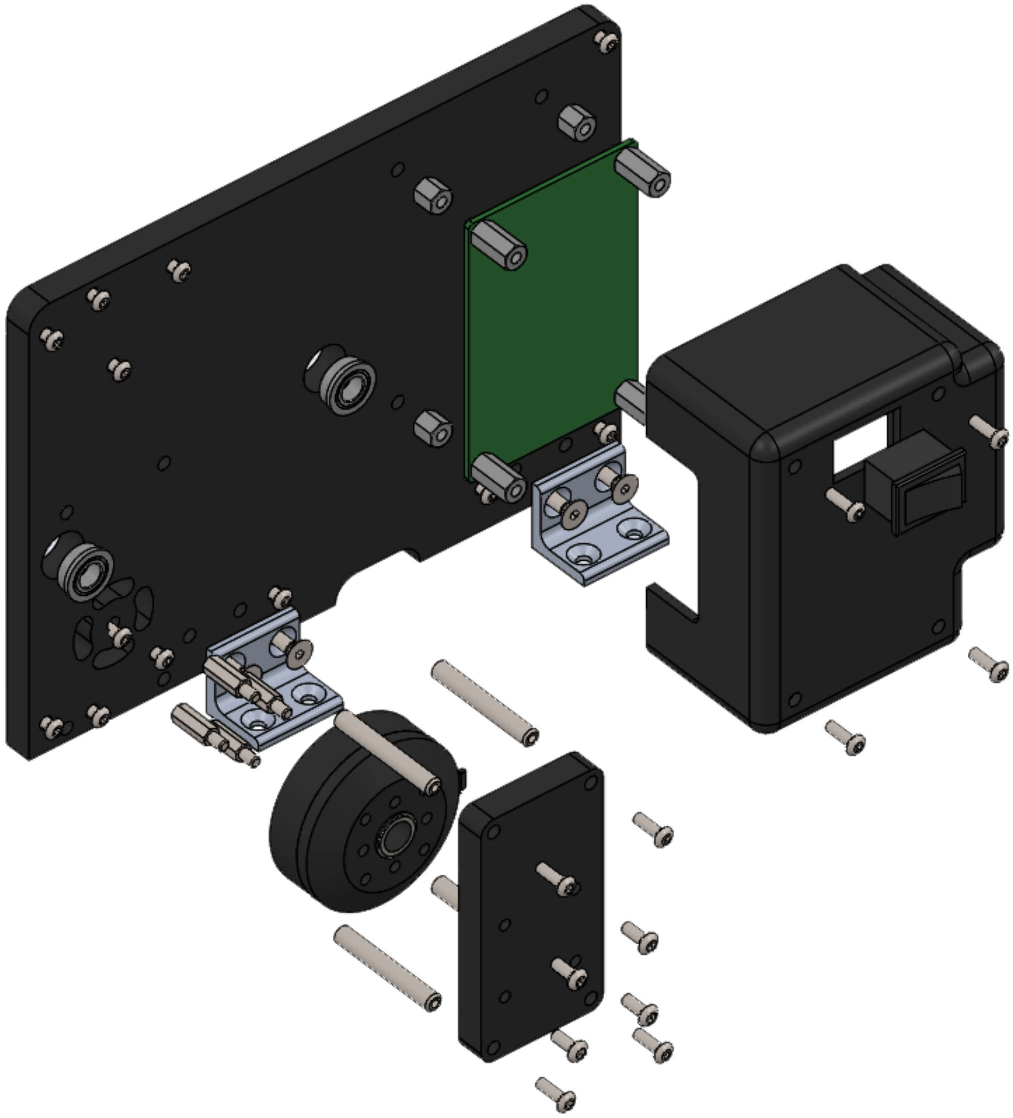




2 1

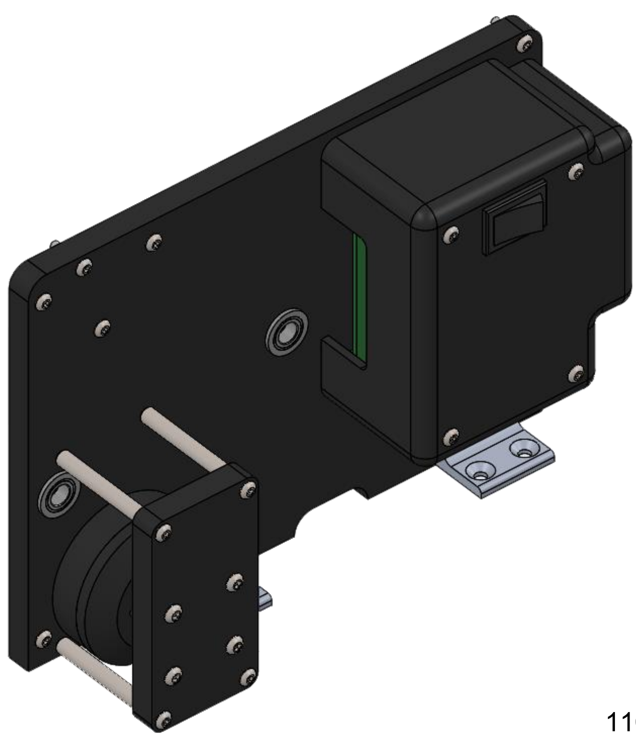
# APPENDIX D:
# Z-ACTUATOR ASSEMBLY INSTRUCTIONS

# 1

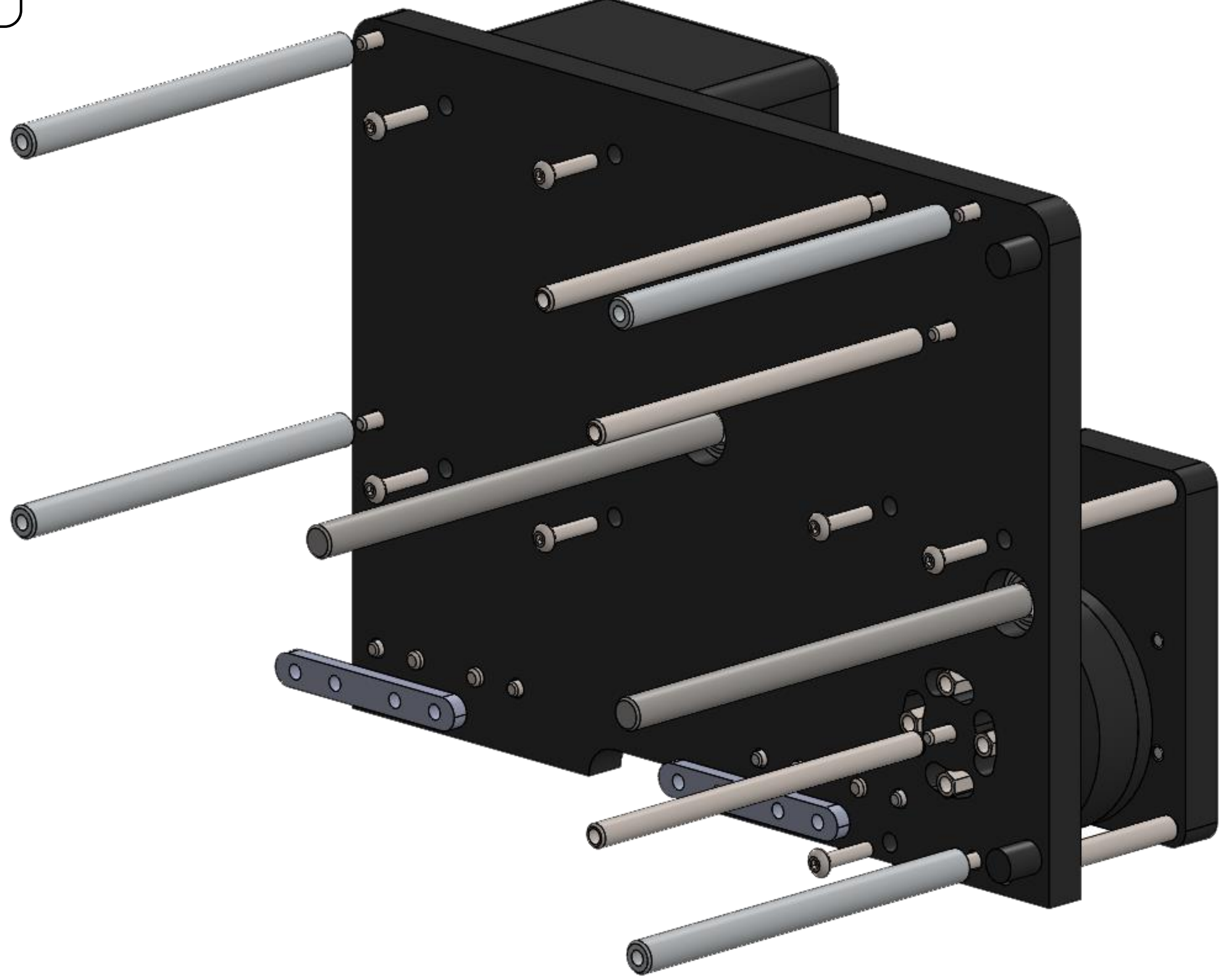


**COMPONENTS:**

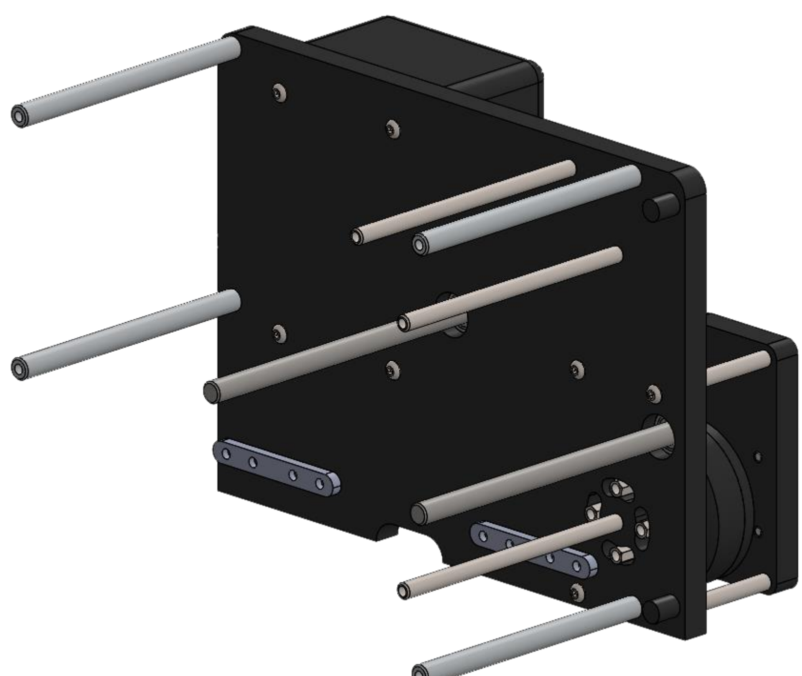
- 01_Plate_A – 1x
- 07_L-Bracket_A – 2x
- 10_Motor_Plate – 1x
- 21_Support_Rod_D – 4x
- ¼” 6-32 Standoff – 4x
- ¾” 6-32 Standoff – 4x
- 10mm m3 Standoff – 4x
- ¼” Ball Bearing – 2x
- 6-32 Flathead – 4x
- 6-32 Panhead – 20x
- m3 Panhead – 4x
- Switch – 1x
- Electronics Enclosure – 1x
- Motor – 1x



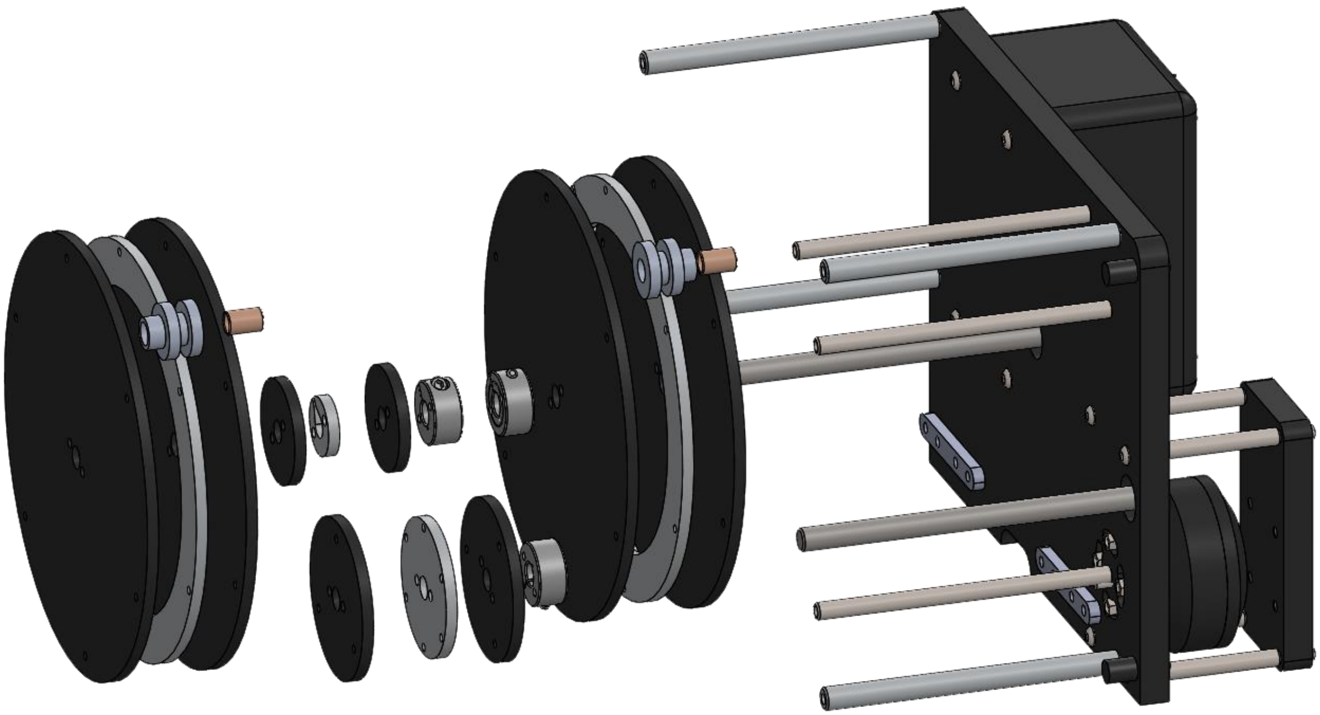
2



**COMPONENTS:**
- 18_Support_Rod_A – 4x
- 25_Axel – 2x
- 22_Pivot_Rod – 3x
- 06_Plate_Mount – 2x
- 6-32 Panhead – 7x
- Bumper – 2x

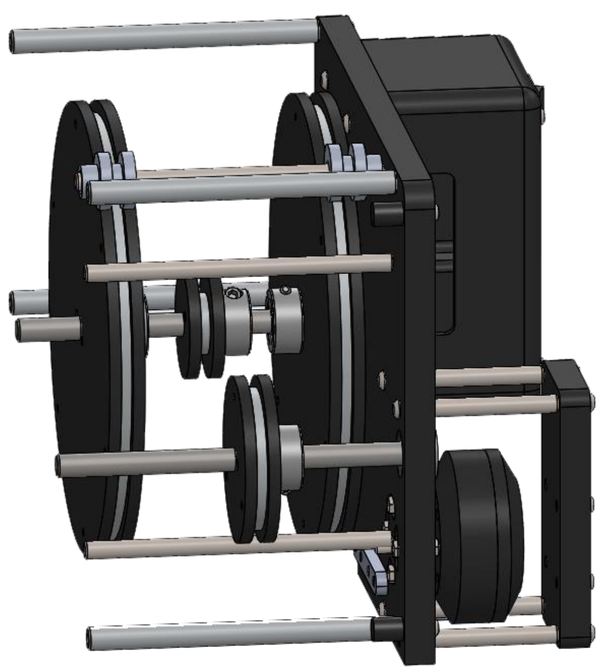


3

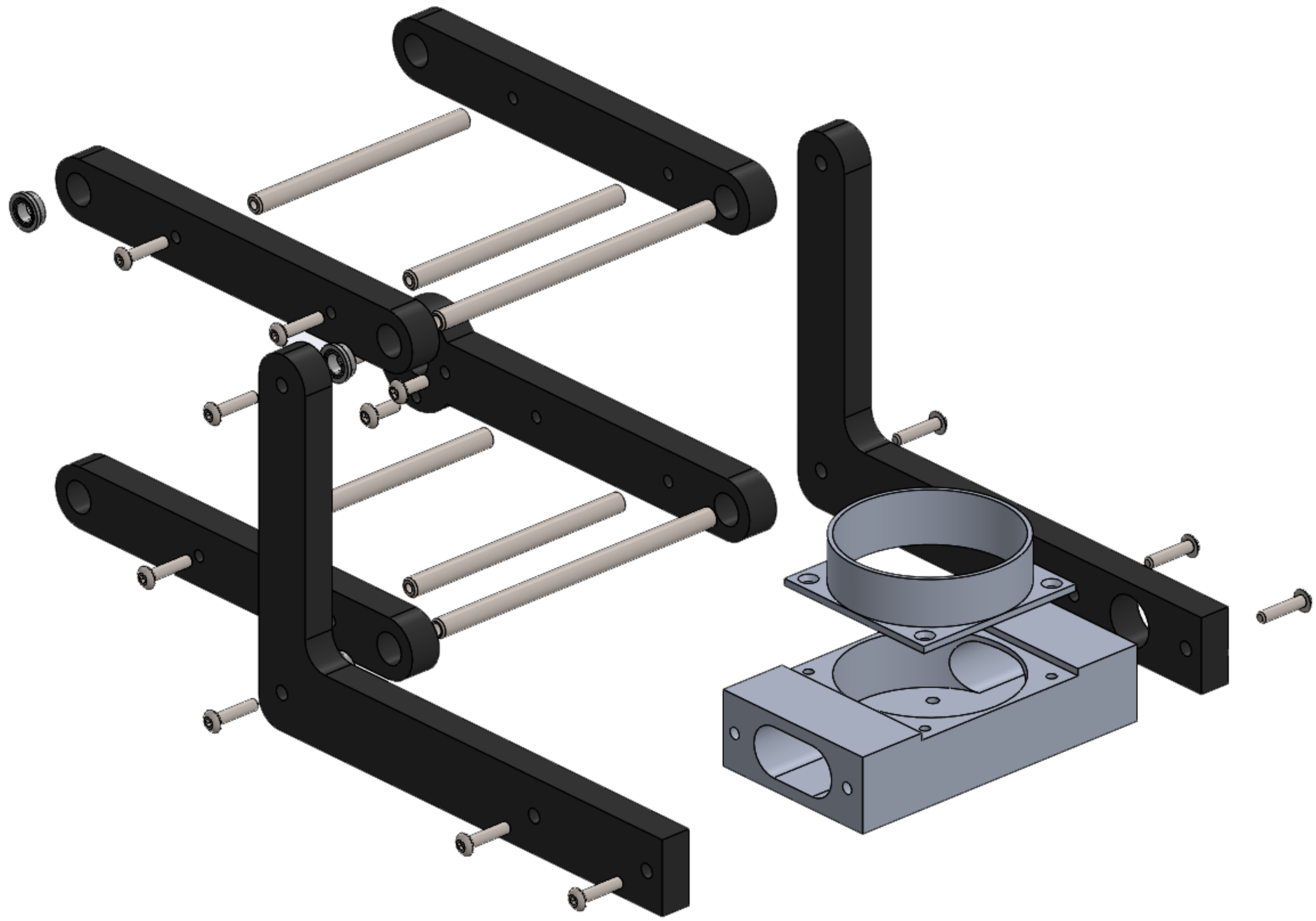


**COMPONENTS:**

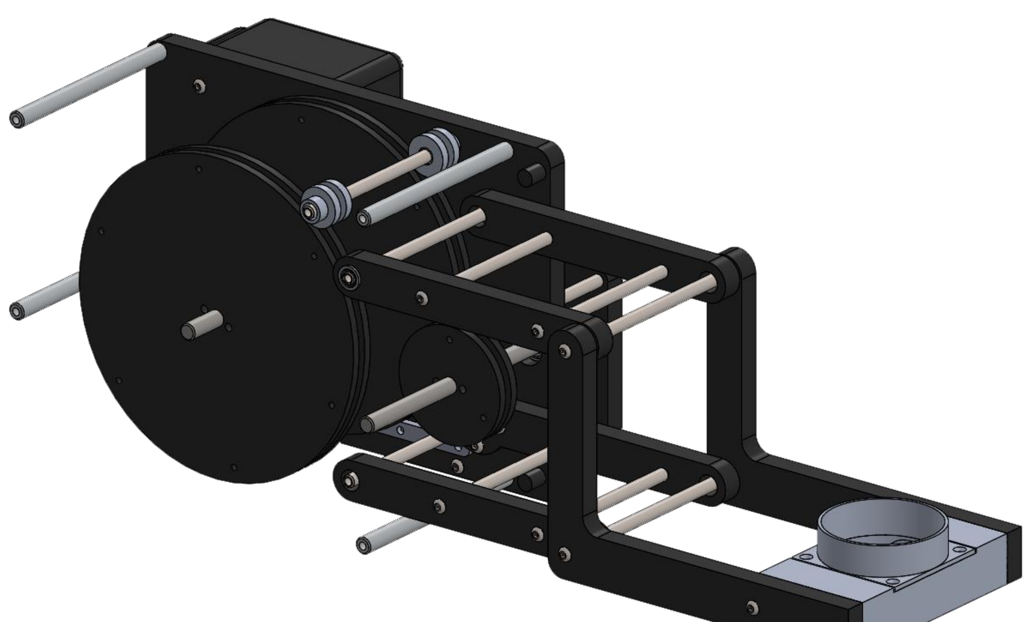
- 09_Idler_Pulley – 2x
- 14_Core_A – 1x
- 15_Flange_A – 2x
- 16_Core_B – 2x
- 17_Flange_B – 4x
- 27_Idler_Core – 1x
- 28_Idler_Flange – 2x
- ¼" Locking Collar – 4x
- 1" 4-40 Panhead – 8x
- ½" 2-56 Flathead – 12x
- 2-56 Nut – 12x
- ¼" 2-56 Flathead – 6x
- 3/16" Bushing – 2x



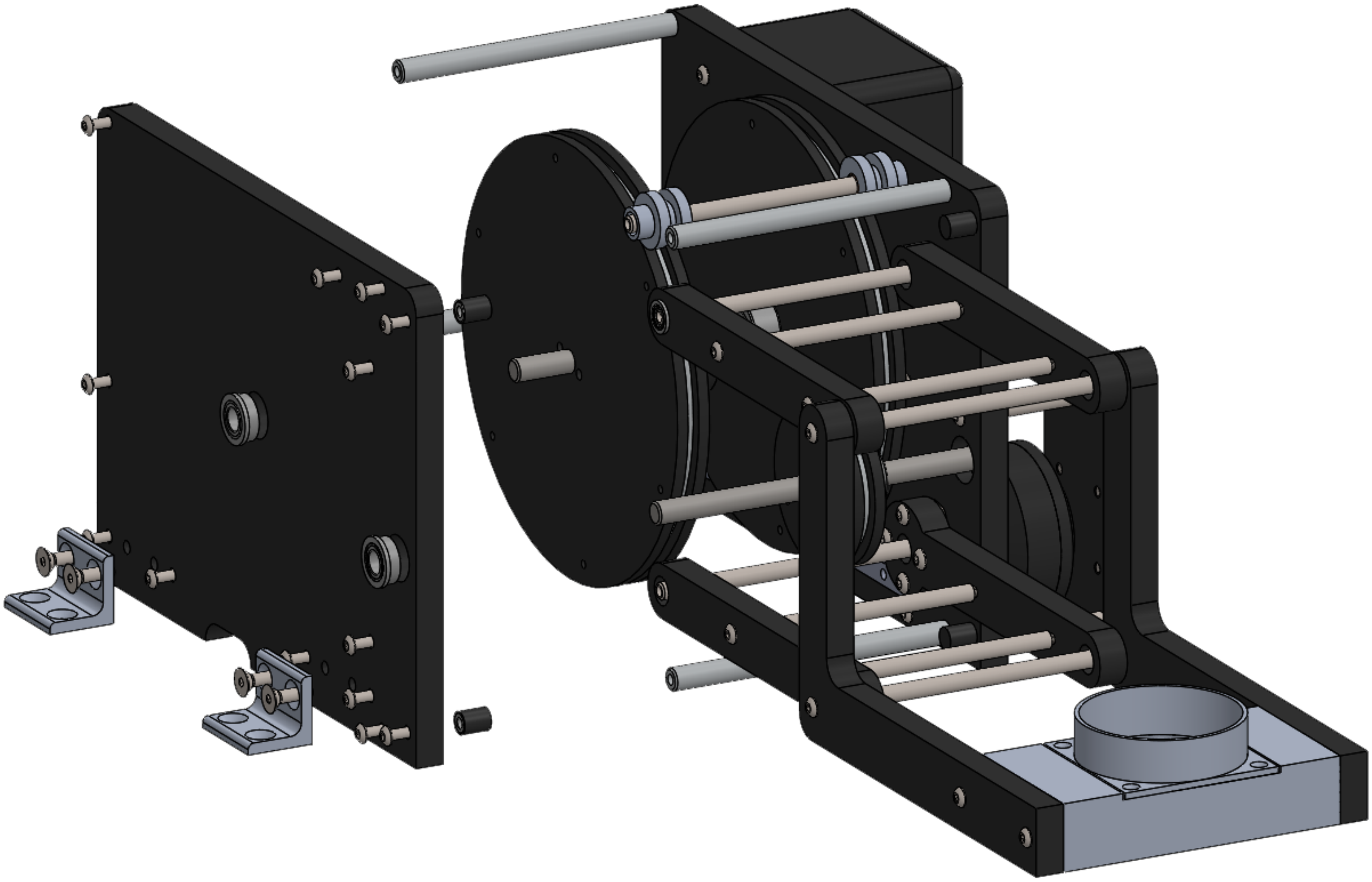
4



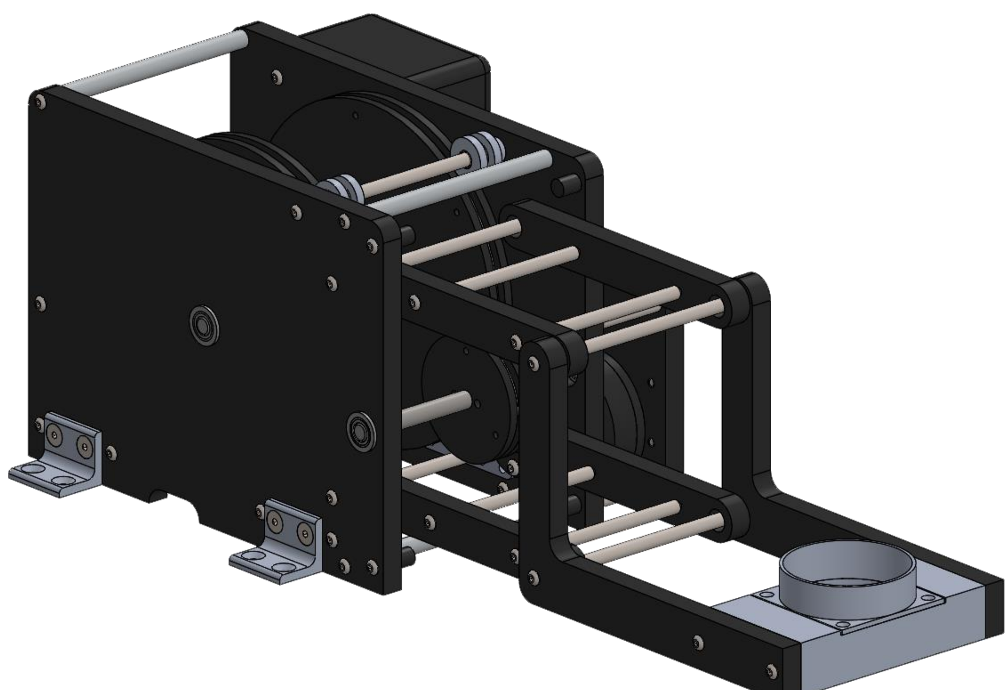
**COMPONENTS:**
- 03_Arm_A – 2x
- 04_Arm_B – 1x
- 05_Arm_C – 1x
- 19_Support_Rod_B – 4x
- 22_Pivot_Rod – 2x
- 26_Interface_Plate – 1x
- L_Arm – 2x
- Collar – 1x
- 3/16” bearing – 8x
- 6-32 Panhead – 14x
- m3 Panhead – 4x



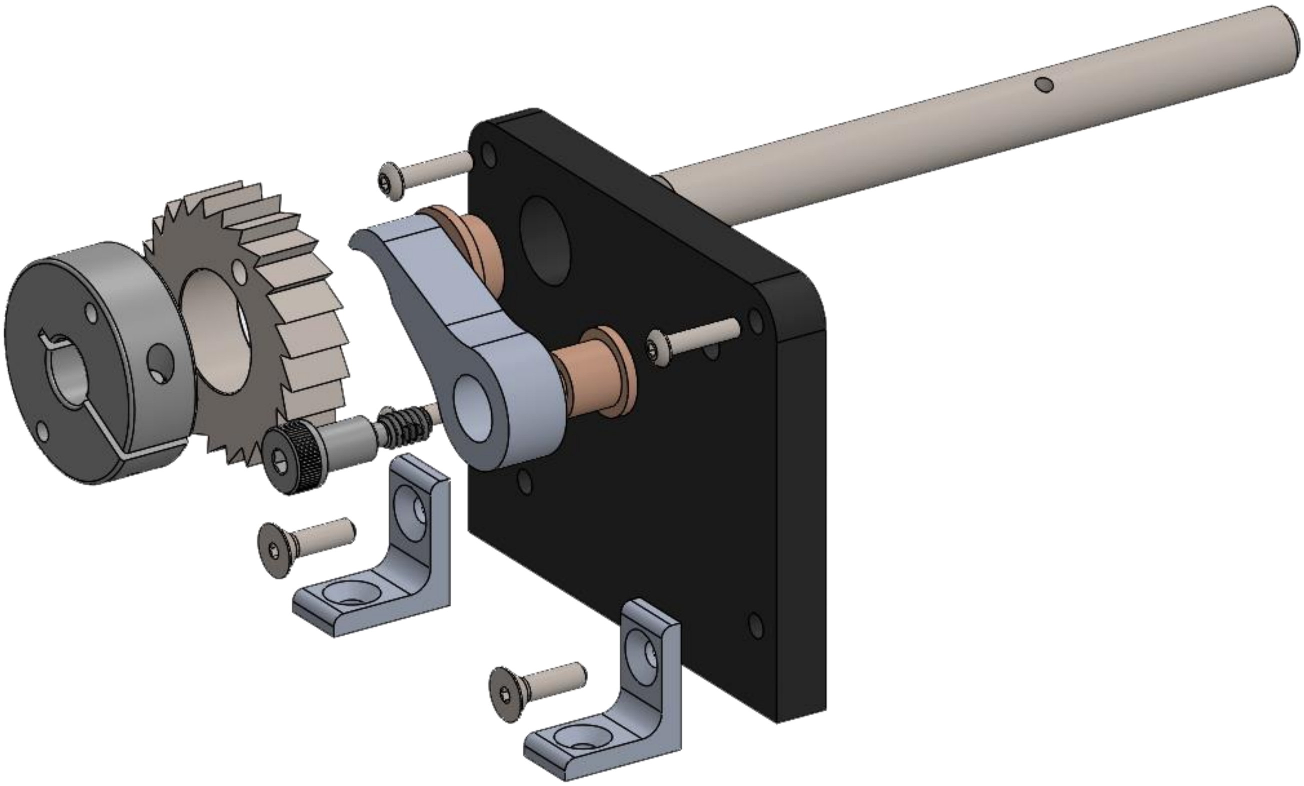
5



**COMPONENTS:**
- 02_Plate_B – 1x
- 06_Plate_Mount – 2x
- 07_L-Bracket_A – 2x
- Bumper – 2x
- ¼” Ball Bearing – 2x
- 6-32 Flathead – 4x
- 6-32 Panhead – 13x

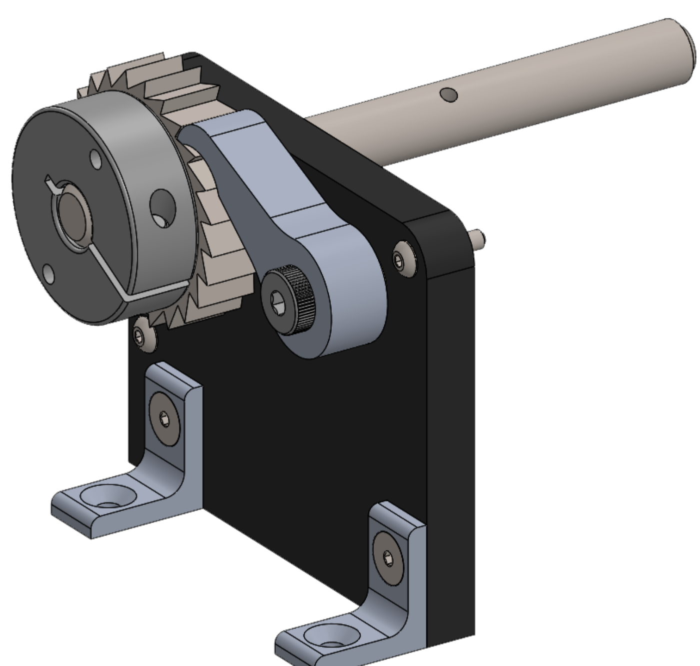


6

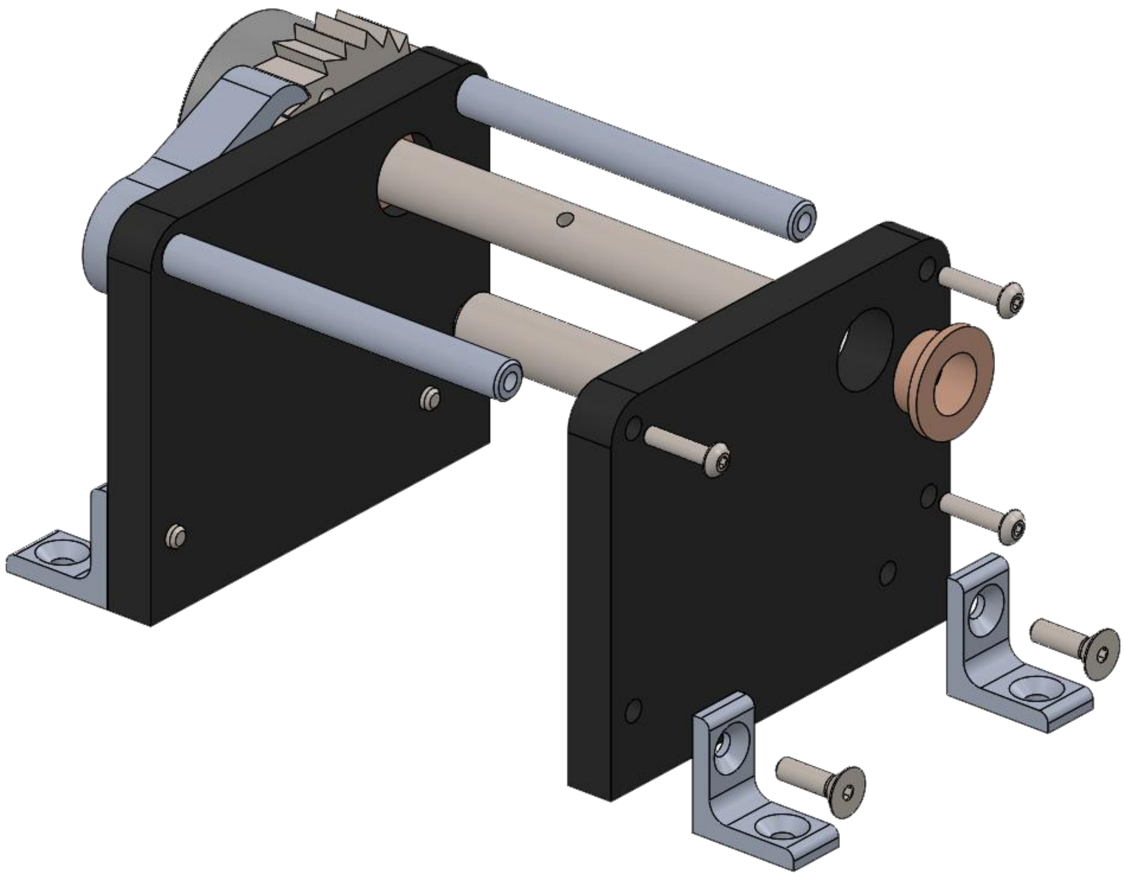


**COMPONENTS:**

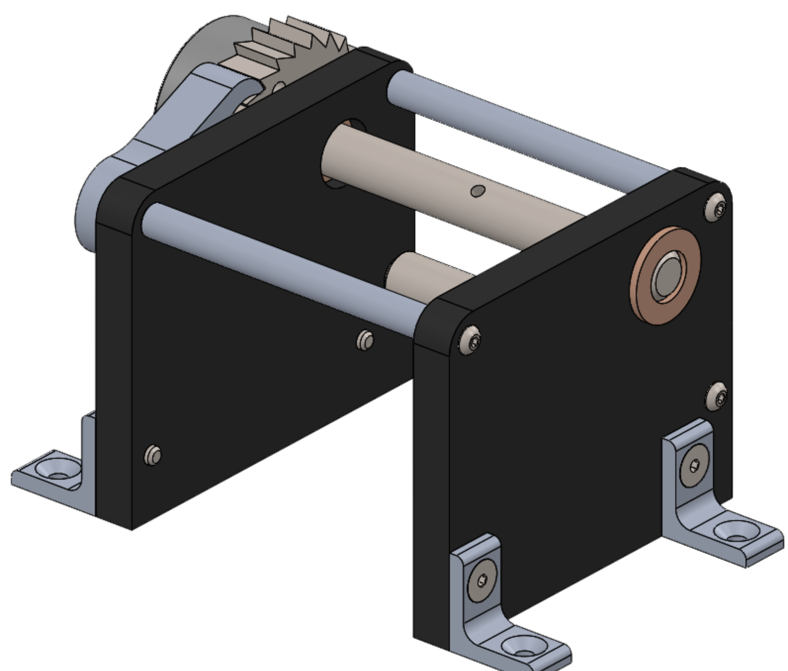
- 08_L-Bracket_B – 2x
- 12_Spring_Plate_A – 1x
- 24_Ratchet_Rod – 1x
- 29_Pawl – 1x
- ¼” Shoulder Screw – 1x
- ¼” Flanged Bushing – 1x
- 3/8” Flanged Bushing – 1x
- Ratchet – 1x
- 3/16” Locking Collar – 1x
- 6-32 Flathead – 2x
- 6-32 Panhead – 3x



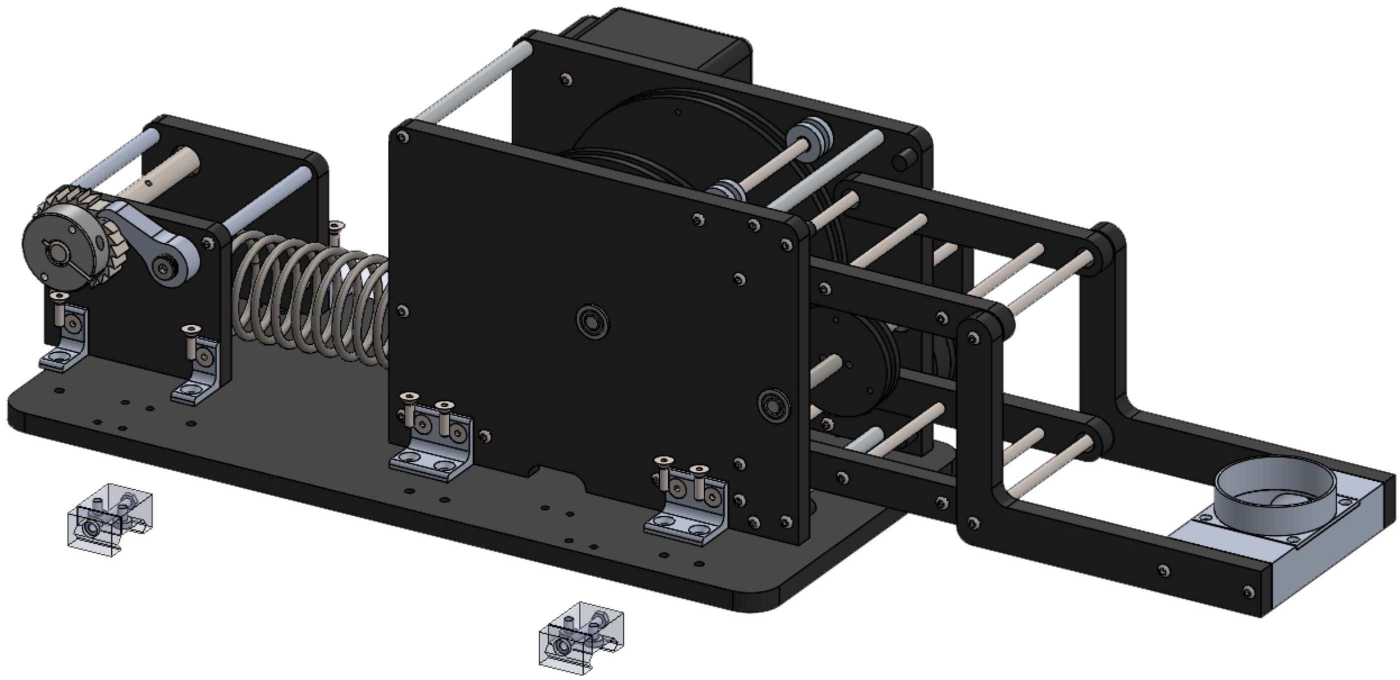
7



**COMPONENTS:**
- 08_L-Bracket_B – 2x
- 13_Spring_Plate_B – 1x
- 3/8” Flanged Bushing – 1x
- 6-32 Flathead – 2x
- 6-32 Panhead – 3x
- Spring

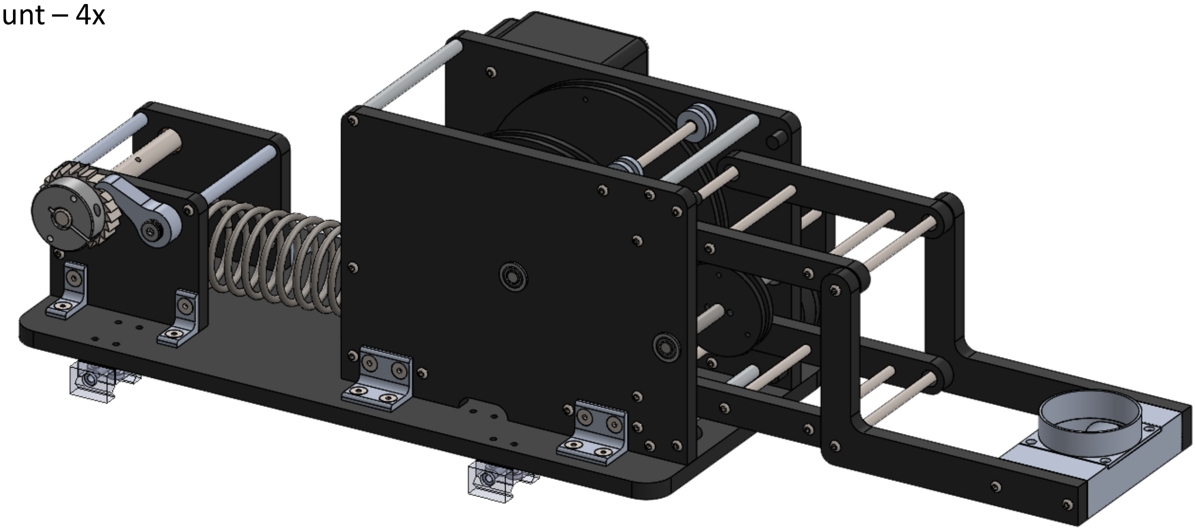


8



**COMPONENTS:**

- 6-32 Flathead – 12x
- m3 Panhead – 8x
- Picatinny Mount – 4x



# APPENDIX E:
# CODE BASE

## E.1 Stabilize Source Code

```
//motor value codes
#define ANGLE 1
#define ABS_ANGLE 11
#define VELOCITY 2

#include <MemoryFree.h>

#define ARM_LENGTH 0.1016 //meters

#define LOGGING 0
//0: no logging
//1: serial logging,
//2: SD card logging

#define LOOP_LOGGING 0
//1: enable logging in loop
//0: disable logging in loop

#define MAX_FREQUENCY 50
const long MIN_LOOP_TIME = 1000/MAX_FREQUENCY;


//Washout Settings
float rollFrequencyCutoff = 0.01;//0.25;
float pitchFrequencyCutoff = 0.01;//0.25;
float yawFrequencyCutoff = 0.01;//0.25;
float zFrequencyCutoff = 0.01;//0.25;
float vzFrequencyCutoff = 0.01;//0.01;

//Motor Calibration
const float roll_zero = 1.7825;
const float pitch_zero = -1.0861;
const float yaw_zero = 0.8115;
const float z_zero = 0;


//led output pin
const int kLedPin =  13;

//
///
//////////////////////////////////////////////
//LOGGING/////////////////////////////////////
//use this function to log a runtime message
void logln(char* msg);
//use this function to log a runtime message even if logging is disabled
void stronglogln(char* msg);
```

```
//Setup for data logging to SD card
void dataLoggingSetup();
//use this function to log data to the SD card
void dataln(char* dat);
//use this to set message logging permission
void setLogPermission(bool loggingP);
char logs[500];

///////////////////////////////////////////////
//IMU//////////////////////////////////////////
#include <UM7.h>
UM7 imu;
//IMU setup
void imuSetup();
//IMU update
bool updateImu();
///////////////////////////////////////////////
//MOT//////////////////////////////////////////
//Call this first
void motControlSetup();
//Call this for each motor
void motSetup(int motNum, float Kp, float Ki, float Kd, uint32_t fc);
//Set a value
void setVal(int motNum, int val, float pos);
//Get a value
float getVal(int motNum, int val);
//Call this every loop
void syncMotor(int motNum);
///////////////////////////////////////////////
///
//

float degree2rad(float degree) {
 return degree*2.0*PI/360.0;
}

float rad2degree(float rad) {
 return rad * 360.0/(2.0*PI/360);
}

float rollover(float* prev, float* mod, float value) {
 float margin = 50;

 //rollover 0
 if(*prev - margin < 0 && value + margin > 360.0){
   *mod = *mod - 360.0;
 }

 //rollover 360
 if(*prev + margin > 360.0 && value - margin < 0.0) {
```

```
  *mod = *mod + 360.0;
 }
 *prev = value;
 return value + *mod;
}

float alpha(long loop_time, float frequency_cutoff) {
 return 1.0/(2*PI*(loop_time*0.001)*frequency_cutoff + 1.0);
}

void setup() {
 pinMode(kLedPin, OUTPUT);
 dataLoggingSetup();
 imuSetup();
 motControlSetup();
 motSetup(1, 20,0,0.3,100); //pitch
 motSetup(2, 50,0,1.5,100); //roll (p:d ~50 is good)
 motSetup(3, 25, 0,3,100); //yaw (p:d ~10 is good for system)
 motSetup(4, 150,0.05,0.01,100); //z

 if(!LOOP_LOGGING) {
  setLogPermission(false);
 }
}

void loop() {
 //filter setup
 float rollHighpass = 0;
 float pitchHighpass = 0;
 float yawHighpass = 0;
 float zHighpass = 0;
 float vzHighpass = 0;

 //rollover storage//////////

 //modifiers
 float roll_mod = 0.0;
 float pitch_mod = 0.0;
 float yaw_mod = 0.0;

 //previous imu values
 float roll_prev = 0.0;
 float pitch_prev = 0.0;
 float yaw_prev = 0.0;

 //rolled over values
 float roll_rollover = 0.0;
 float pitch_rollover = 0.0;
 float yaw_rollover = 0.0;
```

```
//previous rollod over values
float roll_prev_r = 0.0;
float pitch_prev_r = 0.0;
float yaw_prev_r = 0.0;

///////////////////////////

//z position values
float z_position = 0;
float z_position_prev = 0;

float z_velocity = 0;
float z_velocity_prev = 0;

long start_time = millis();

while(1) {

  digitalWrite(kLedPin, LOW);
  //limit cycle frequency to MAX_FREQUENCY
  static long time_last = 0;
  while(millis()-time_last < MIN_LOOP_TIME);
  long loop_time = millis() - time_last;
  time_last += loop_time;
  digitalWrite(kLedPin, HIGH);

  //failover tasks/////

  //imu
  if(!updateImu()) {
    stronglogln("Something went wrong with the IMU.");
    continue; //skip the cycle if the imu doesn't update correctly
  }


  //wait for highpass to steady out
  if(millis() - start_time < 2000) {
    continue;
  }


  /////////////////////


  //update rollover
  roll_rollover = rollover(&roll_prev, &roll_mod, imu.roll);
  pitch_rollover = rollover(&pitch_prev, &pitch_mod, imu.pitch);
  yaw_rollover = rollover(&yaw_prev, &yaw_mod, imu.yaw);

  //update high pass filters
```

```
float rollAlpha = alpha(loop_time, rollFrequencyCutoff);
rollHighpass = rollAlpha*rollHighpass + rollAlpha*(roll_rollover - roll_prev_r);
roll_prev_r = roll_rollover;

float pitchAlpha = alpha(loop_time,pitchFrequencyCutoff);
pitchHighpass = pitchAlpha*pitchHighpass + pitchAlpha*(pitch_rollover - pitch_prev_r);
pitch_prev_r = pitch_rollover;

float yawAlpha = alpha(loop_time, yawFrequencyCutoff);
yawHighpass = yawAlpha*yawHighpass + yawAlpha*(yaw_rollover - yaw_prev_r);
yaw_prev_r = yaw_rollover;

//update z_position and filter
z_velocity_prev = z_velocity;
z_velocity = z_velocity - (imu.az+1.0) * (loop_time*0.001);
float vzAlpha = alpha(loop_time, vzFrequencyCutoff);
vzHighpass = vzAlpha*vzHighpass + vzAlpha*(z_velocity - z_velocity_prev);

float zAlpha = alpha(loop_time, zFrequencyCutoff);
zHighpass = zAlpha*zHighpass + zAlpha*(z_position - z_position_prev);
z_position_prev = z_position;




//set motor commands
setVal(1, ANGLE, degree2rad(-pitchHighpass) + pitch_zero); //pitch
setVal(2, ANGLE, degree2rad(-rollHighpass) + roll_zero); //roll
setVal(3, ANGLE, degree2rad(yawHighpass) + yaw_zero);  //yaw
setVal(4, ANGLE, -zHighpass/ARM_LENGTH + z_zero);



//sync motor info
syncMotor(1); //pitch
syncMotor(2); //roll
syncMotor(3); //yaw
syncMotor(4); //z



logln("Compiling output...");

static long ct = 0;
ct++;

static char string[300];

//compile output
```

```
  sprintf(string,
"%d,%d,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f",ct,loop_time,imu.roll,imu.pitch,imu.yaw,imu.ax,imu.ay,
imu.az,getVal(1, ANGLE) - pitch_zero,getVal(2, ANGLE) - roll_zero,getVal(3, ANGLE) -
yaw_zero, getVal(4, ANGLE) - z_zero);

  //output
  dataln(string);
  stronglogln(string);
 }
}

#define IMUSerial Serial6

void imuSetup() {
 IMUSerial.begin(115200);
}

bool updateImu() {
 logln("Updating IMU...");
 bool worked = false;
 if(IMUSerial.available()) {
  while (IMUSerial.available()){  // Reads byte from buffer.  Valid packet returns true.
    imu.encode(IMUSerial.read());
    worked = true;
  }
 }
 return worked;
}

#include <SD.h>
#include <SPI.h>

//Filenames for logging
#define LOGFILE "slog.txt"
#define NAMEFILE "slast.txt"
#define DATAFILESTEM "sdat"//##.txt
char fileName[20] = DATAFILESTEM;
int fileNumber = 0;
const int chipSelect = BUILTIN_SDCARD;

bool sdActive = false;

//logging permission control
bool logging = true;
void setLogPermission(bool loggingP) {
 logging = loggingP;
}

//use this function to log a runtime message
void logln(char * msg) {
```

```
 if(!logging) {
   return;
 }
 if(LOGGING == 1) {
   Serial.println(msg);
 } else if(LOGGING == 2 && sdActive) {
   File logfile = SD.open(LOGFILE, FILE_WRITE);
   if(logfile){
     char msgcat[300];
     sprintf(msgcat, "%d|%s", fileNumber, msg);
     logfile.println(msgcat);
     logfile.close();
   } else {
     return;
   }
 }
}

void stronglogln(char* msg) {
 bool prevPerm = logging;
 setLogPermission(true);
 logln(msg);
 setLogPermission(prevPerm);
}


//Setup for data logging to SD card
void dataLoggingSetup() {
 if(LOGGING == 1) {
   //wait for USB
   Serial.begin(115200);
   while (!Serial);
 }

 //activate SD card
 logln("Initializing SD card...");
 if(!SD.begin(chipSelect)) {
   logln("Card failed, or not present");
   return;
 }
 sdActive = true;

 sprintf(logs,"Looking for NAMEFILE");
 logln(logs);

 if(SD.exists(NAMEFILE)) {
   logln("Previous namefile exists");
```

```
  //read existing namefile and parse new file number
  File namefile = SD.open(NAMEFILE, FILE_READ);
  String fileNumberS = "";
  while(namefile.available()){
    char next = namefile.read();
    fileNumberS += next;
  }

  fileNumber = fileNumberS.toInt() + 1;

  //log last file number
  sprintf(logs, "Previous file number:%d", fileNumber-1);
  logln(logs);


  //erase previous namefile
  namefile.close();
  SD.remove(NAMEFILE);

  //write a new namefile
  namefile = SD.open(NAMEFILE, FILE_WRITE);
  if(namefile) {
    sprintf(fileName, "%s%d.txt", DATAFILESTEM, fileNumber);

    char fileNumberS[20];
    sprintf(fileNumberS,"%d",fileNumber);
    namefile.print(fileNumberS);
    namefile.close();
  } else {

    sprintf(logs, "Error opening %s!", NAMEFILE);
    logln(logs);
    return;
  }


} else {
  logln("No previous namefile exists. Creating new namefile.");

  sprintf(fileName, "%s0.txt", DATAFILESTEM);

  //make a namefile
  File namefile = SD.open(NAMEFILE, FILE_WRITE);
  if(namefile) {
    namefile.print("0");
    namefile.close();
  } else {
    sprintf(logs, "Error opening %s!", NAMEFILE);
    logln(logs);
    return;
```

```
  }

 }

 //Done!

 sprintf(logs, "Initialized SD card logging. \nFile Name: %s \nFile Number: %d", fileName,
fileNumber);
 logln(logs);
}

//use this function to log data to the SD card
void dataln(char* dat) {
 File datafile = SD.open(fileName, FILE_WRITE);
 if(datafile) {
   datafile.println(dat);
   datafile.close();
 } else {
   sprintf(logs, "Failed to open %s for fata logging. data: %s", fileName, dat);
   stronglogln(logs);
 }

}


#include <bipbuffer.h>
#include <byte_queue.h>
#include <communication_interface.h>
#include <crc_helper.h>
#include <packet_finder.h>

// Includes required for communication
// Message forming interface
#include <generic_interface.hpp>
// Client that speaks to complex motor controllers
#include <complex_motor_control_client.hpp>




// This buffer is for passing around messages.
// We use one buffer here to save space.
uint8_t communication_buffer[256];
// Stores length of message to send or receive
uint8_t communication_length;

ComplexMotorControlClient mot_client1(0);
GenericInterface com1;
```

```
ComplexMotorControlClient mot_client2(0);
GenericInterface com2;

ComplexMotorControlClient mot_client3(0);
GenericInterface com3;

ComplexMotorControlClient mot_client4(0);
GenericInterface com4;


void motControlSetup() {
 Serial1.begin(115200);
 Serial2.begin(115200);
 Serial3.begin(115200);
 Serial4.begin(115200);
 sprintf(logs, "Started serial for all motors.");
 logln(logs);
}

void motSetup(int motNum, float Kp, float Ki, float Kd, uint32_t fc) {
 ComplexMotorControlClient *mot_client;
 GenericInterface *com;

 switch(motNum) {
   case 1:
     mot_client = &mot_client1;
     com = &com1;
     break;
   case 2:
     mot_client = &mot_client2;
     com = &com2;
     break;
   case 3:
     mot_client = &mot_client3;
     com = &com3;
     break;
   case 4:
     mot_client = &mot_client4;
     com = &com4;
     break;
 }


 mot_client->AngleKp.Set(*com,Kp);
 mot_client->AngleKi.Set(*com,Ki);
 mot_client->AngleKd.Set(*com,Kd);

 mot_client->VelocityFilterFc.Set(*com, fc);
 sprintf(logs, "Motor Setup for %d done.", motNum);
 logln(logs);
```

```
}

void setVal(int motNum, int val, float pos) {

 ComplexMotorControlClient *mot_client;
 GenericInterface *com;

 switch(motNum) {
  case 1:
   mot_client = &mot_client1;
   com = &com1;
   break;
  case 2:
   mot_client = &mot_client2;
   com = &com2;
   break;
  case 3:
   mot_client = &mot_client3;
   com = &com3;
   break;
  case 4:
   mot_client = &mot_client4;
   com = &com4;
   break;
  default:
   return;
 }

 switch(val) {
  case ANGLE:
   mot_client->CmdAngle.Set(*com, pos);
   break;
  default:
   return;
 }

}

float getVal(int motNum, int val) {

 ComplexMotorControlClient *mot_client;

 switch(motNum) {
  case 1:
   mot_client = &mot_client1;
   break;
  case 2:
   mot_client = &mot_client2;
   break;
  case 3:
```

```
    mot_client = &mot_client3;
    break;
  case 4:
    mot_client = &mot_client4;
    break;
  default:
    return 0;
 }

 switch(val) {
  case ANGLE:
    return mot_client->ObsAngle.GetReply();
  case ABS_ANGLE:
    return mot_client->ObsAbsoluteAngle.GetReply();
  case VELOCITY:
    return mot_client->ObsVelocity.GetReply();
  default:
    return 0;
 }
}

void syncMotor(int motNum) {
 ComplexMotorControlClient *mot_client;
 GenericInterface *com;

 switch(motNum) {
  case 1:
    mot_client = &mot_client1;
    com = &com1;
    break;
  case 2:
    mot_client = &mot_client2;
    com = &com2;
    break;
  case 3:
    mot_client = &mot_client3;
    com = &com3;
    break;
  case 4:
    mot_client = &mot_client4;
    com = &com4;
    break;
  default:
    return;
 }

 logln("Making 'get' commands...");

 mot_client->ObsAngle.Get(*com);
 mot_client->ObsAbsoluteAngle.Get(*com);
```

```
mot_client->ObsVelocity.Get(*com);



logln("Recieving Updates...");

if(com->GetTxBytes(communication_buffer,communication_length)) {
  switch(motNum) {
    case 1:
      Serial1.write(communication_buffer,communication_length);
      communication_length = Serial1.readBytes(communication_buffer, Serial1.available());
      break;
    case 2:
      Serial2.write(communication_buffer,communication_length);
      communication_length = Serial2.readBytes(communication_buffer, Serial2.available());
      break;
    case 3:
      Serial3.write(communication_buffer,communication_length);
      communication_length = Serial3.readBytes(communication_buffer, Serial3.available());
      break;
    case 4:
      Serial4.write(communication_buffer,communication_length);
      communication_length = Serial4.readBytes(communication_buffer, Serial4.available());
      break;
    default:
      return;
  }
}


logln("Serial operations complete...");

com->SetRxBytes(communication_buffer,communication_length);

logln("Transmitting commands...");




uint8_t *rx_data;   // temporary pointer to received type+data bytes
uint8_t rx_length;  // number of received type+data bytes
// while we have message packets to parse
while(com->PeekPacket(&rx_data,&rx_length))
{

  sprintf(logs, "Length in the thingy %d", rx_length);
  stronglogln(logs);

  // Share that packet with all client objects
  mot_client->ReadMsg(*com,rx_data,rx_length);
```

```
   // Once we're done with the message packet, drop it
   com->DropPacket();
  }

 }
```