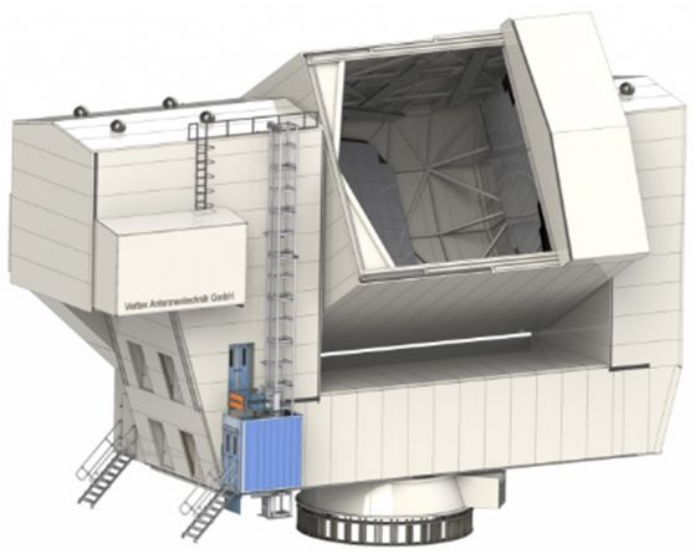
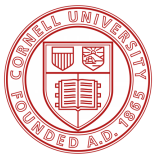


CCAT-prime Wall-Climbing Robot

Ruohan Gao and Seth McCall

Cornell University
Masters of Engineering (M.Eng.)
Department of Mechanical and Aerospace Engineering

May 20, 2019



Abstract

This is a final report from the MAE 6900 independent research project of Ryan Gao and Seth McCall. Our project was to design a wall-climbing robot capable of performing metrology on the CCAT-prime telescope in Cerro Chajnantor, Chile. We worked on the Controls Team, focusing in the areas of Estimation via Kalman Filter and I.M.U., Control Scheme, Edge Sensing and Coordinate System Transformation. This paper delineates the motivations, materials, testing procedures, and results for each of these aspects. It also summarizes the future efforts that must take place to ensure successful project completion. We worked alongside other Cornell University students on the Controls Team, as well as students on the Mechanical Team. Overseeing the project was our MAE faculty advisor, Professor Dmitry Savaransky as well as Stephen Parshley, the Project Engineer and Terry Herter, the Project Director and a faculty member of the Cornell University Department of Astronomy.

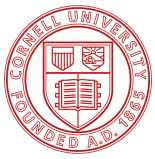


Table of Contents

Abstract 1

Table of Contents 3

List of Terms 5

Acknowledgments 5

Introduction 6

Background 6

- Measurement Efforts 7
- Prototype and Previous Work 7

Literature Review 7

Theory 8

Decision and Design Criteria 9

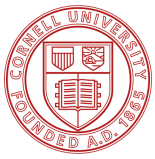
- Design Criteria Overview 9
- CCAT-p Environmental Conditions 9
- Measurement Program Requirements 10
- Safety and Operability Requirements 10

Materials and Apparatus 11

- Kalman Filter Materials and Apparatus 11
- Edge Sensor Materials and Apparatus 12
  - Edge Detection Test Materials and Apparatus 12
  - Gap Detection Test Materials and Apparatus 13
  - Corner Orientation Test Materials and Apparatus 14
- Coordinate System Transformation Materials and Apparatus 14

Procedure 14

- Kalman Filter Motivation and Procedure 14
- Controls Architecture Procedure 17
- Edge Sensor Motivation and Procedure 20
  - Edge Sensor Motivation 20
  - Edge Detection Test Motivation and Procedure 20
  - Gap Detection Test Motivation and Procedure 20
- Coordinate System Transformation Motivation and Procedure 21
  - Coordinate System Transformation Motivation 21



Coordinate System Transformation Procedure 23

Discussion and Results 25

Kalman Filter Results 25

Control Scheme Results 26

Edge Sensor Results 26

Edge Detection Results 26

Gap Detection Results 27

Corner Orientation Results 29

Coordinate System Transformation Results 30

Recommendations (Next Steps) 34

Tilt 34

Edge Sensing Code 35

Encoder Integration 35

Complete Simulation Testing 35

Physical System Integration 35

Physical Testing 36

10. Conclusion 36

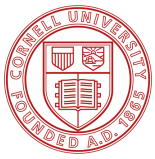
11. References 37

List of Terms

- **Mirror panels/panels:** individual aluminum panels, ~0.6mx~0.7m, which comprise the entire telescope mirror
- **Mirror:** the entire structure of each mirror, all the panels combined
- **Mirror 1:** Primary mirror. 77 panels.
- **Mirror 2:** Secondary mirror. 69 panels.
- **Panel Gap/Gap:** small (0.5-2mm) distance between each mirror panel.
- **Mirror Edge/Edge:** edge of entire mirror, not just edge of mirror panel
- **Measurement Node/Node:** measurement location for robot. Nine nodes per panel.
- **Waypoints:** Another way of saying ‘measurement node’, mainly used in reference to robotic path planning and estimation.
- **Horizon/World Frame:** synonymous, the three-dimensional frame used to characterize the observatory’s features and pointing.
- **Boresight Axis:** The z-axis equivalent in the telescope World Frame.
- **Elevation Axis:** The y-axis equivalent in the telescope World Frame.
- **IMU:** Inertial Measurement Unit
- **Prediction step:** First step of Kalman filter, update the state variables through dynamics
- **Update step:** Second step of Kalman filter, update the state variables from the known measurements
- **PWM:** Pulse width Modulation
- **Duty cycle:** The proportion of time the signal is stayed at max during each period
- **A2D converter:** Analog to digital converter
- **I2C protocol:** A serial protocol for two-wired interface to connect low-speed device.

Acknowledgments

We would like to acknowledge the support from Prof. Savaransky who served as our project advisor for providing us with many resources and technical guidance, as well as Stephen and Terry, the Project Engineer and the Project Director, for giving us this opportunity to work on this great project. We would also like to thank the other Cornell University students alongside whom we worked.



1. Introduction

Over the summer of 2018, two M.Eng. students from Cornell University developed a prototype autonomous mobile robot to assist in the process of deformation analysis on the primary and secondary mirrors of the CCAT-p observatory in Cerro Chajnantor, Chile. This robot proved to be an adequate solution to the problem of positioning a retro-reflective puck at specific measurement nodes across the mirror panels. Used in tandem with a laser measurement system from Etalon AG, a complete deformation analysis of the mirrors will result.

Upon approval of the robot's capabilities, a project group was formed to design, assemble and test a robot capable of meeting a stringent set of requirements. These requirements are derived from the harsh environmental conditions of the observatory, the necessity of precision measurement and the damage risks associated with an autonomous wall-climbing robot. This group is split into two subgroups of Cornell University students, Mechanical and Controls, and is overseen by Professor Dmitry Savransky. At the beginning of the 2018-2019 school year, the project was in its initial stages. The major deadline for this project is August 2019, when a second prototype will need to be completed in order to conduct tests in Germany in a simulated operational environment.

2. Background

In Cerro Chajnantor, Chile, the CCAT-p telescope is currently being constructed by Vertex Antennentechnik GmbH. This telescope is designed to operate at submillimeter to millimeter wavelengths, 200 - 3100 μm, and will feature a 'crossed-dragone' optical design with a 6-meter aperture. To accomplish this, two mirrors will be used, a primary and secondary mirror, each of which is comprised of individual aluminum mirror panels, 77 and 69 panels respectively, which are each roughly 700x600mm in area. Behind each of these panels are four actuators which will be used to precisely adjust the mirror panels' orientations to achieve the desired overall mirror shape. Due to the necessity of precision in the mirror actuation and positioning, these orientations will need to be confirmed via deformation analysis, and this analysis is the major motivation for the CCAT-p Wall Climbing Robot. This robot will be used to traverse the two mirrors while carrying a reflective puck to specified measurement locations. Once at the locations, the laser measurement system from ETALON AG, a company which specializes in metrology, will 'hit' the puck with lasers and provide distance measurements which will then be used to determine the exact panel locations and orientations.

Although not entirely clear the frequency with which the orientation measurement will take place, it seems that Etalon would like to measure the mirrors after every change in telescope orientation. The telescope is able to rotate 360° in Azimuth and 180° in elevation (altitude) to achieve full sky coverage, so whenever the observatory changes its pointing angle, the mirror panels' orientations will need to be remeasured (Horizontal Coordinate System). A figure showing the details of the Horizon Frame is below.

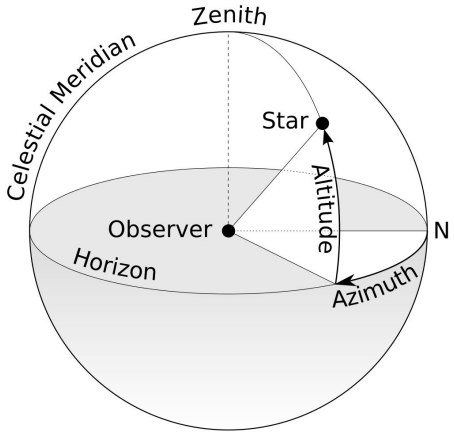


Figure 1: Horizon Frame Schematic

2.1. Measurement Efforts

When deciding how to conduct this deformation analysis, many methods were considered, with one particular method including a large robotic arm to hold and position the reflective puck at each of the measurement nodes. The idea which was decided upon was the CCAT-p Wall Climbing Robot.

2.2. Prototype and Previous Work

Before joining this project, previous work was done. Todd Krichoff and Hyunji Kim began work in the summer of 2018 to develop a prototype robot to demonstrate the feasibility of this solution. To do this, they ordered a wall-climbing robot from a small business in Florida which used a fan to suction itself to a wall, ceiling, etc. and had the ability to traverse in any direction while remaining suctioned. Once purchased, Todd and Hyunji altered the robot by adding a tower designed to hold the retro-reflective puck. After doing this, they demonstrated how this design could hold the puck and traverse the mirrors successfully which thus solidified Etalon's decision to pursue this project as the solution to the deformation analysis problem. Though this purchased bot was able to carry the puck, its design was not optimal to meet all the project requirements, so the initial design of a brand new robot began with using the purchased bot as a benchmark. This began right at the beginning of the 2018-2019 school year

3. Literature Review

This section of literature review mainly focuses on the localization of our vehicle. Localization is needed in our application because in the position control loop, the amount of voltage we apply to motors is a function of the distance between the robot and its current destination. There are many literatures online that introduce the implementation of localization. Most of them involves an I.M.U. (Inertial Measurement Unit). An I.M.U. consists of an accelerator, a gyroscope, and a magnetometer. The I.M.U. detects the acceleration of the

sensor in x, y, z direction and the angular acceleration around the three axis. By integrating these acceleration, we are able to get the distance it traveled in this direction. This distance is relative to the sensor's starting point. In the paper 'IMU-Based Indoor Localization for Construction Applications' by researchers at Concordia University in Canada, an I.M.U is used for tracking the activity of indoor construction workers due to the lack of signal coverage from the GPS in an indoor environment (Ibrahim and Moselhi, 2016) They proposed the idea of using a Kalman filter to obtain the position information from an IMU. The advantage of using a Kalman filter in this application is that it is robust in the sense that it assumes a prior gaussian distribution of the robot's position with a covariance. The output of the algorithm combines the knowledge of the dynamics based on the current location and the actual measurement. Therefore, even if our measurement data is noisy, we are still able to get a reasonable estimation based on the dynamics.

4. Theory

Kalman filter was first introduced in 1960 by Rudolf Kalman. It was presented as a solution to discrete data linear filtering problem(Ibrahim and Moselhi, 2016). The Kalman filter is divided into two sections, prediction and correction. As introduced in the previous section, the kalman filter assumes a gaussian distributed prior of the state variables and linear dynamic and linear measurements. In the prediction step, the state variable x is first fed through the dynamics A and inputs B. The covariance of the state variable is also updated in this step adding the process noise Q which is the uncertainty of the dynamics. In the correction stage, the actual measurement is used to correct the result of the dynamics. Kalman gain K is first calculated with the measurement noise R which is the uncertainty in the measurements. After that, the state variable X and its covariance is updated with the Kalman gain and the transition matrix H which maps the state variables to variables that we are measuring.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = AP_{k-1}A^T + Q$$

Figure 2: Prediction

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H)P_k^-$$

Figure 3: Correction

5. Decision and Design Criteria

5.1. Design Criteria Overview

For this project, there are many requirements that drive the design and decision making of the robot. These requirements can be allocated into three sections: Environmental Conditions, Measurement Program and Safety & Operability.

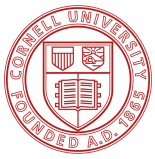
Environmental conditions are important for this project because the bot will have to live and operate in many extreme ambient conditions due to its high elevation and far-southern latitude. At 5000 meters of altitude, the bot will have to withstand extremely low temperatures and atmospheric pressure which is roughly half of the pressure at sea level. Temperature, while not relevant for many of the mechanical aspects of the bot, proves to be troublesome for some of the electronics on board. One example of this is the Eddy Current Sensor Conditioner, which is an electronics box used to condition the Eddy Current signal. Its operating temperature is higher than the required temperature in which the bot will be expected to operate. Pressure is another large factor. Debatably the greatest example of this are the fans, which will have to work significantly harder to provide the necessary suction force. A full list of environmental conditions/requirements is below.

The next section involves the measurement program. These requirements were mostly defined by Etalon, the company in charge of the measurement hardware and procedure. These requirements, like the environmental conditions, affect both the controls and mechanical aspects of the robot design. To achieve these stringent metrology requirements, some resultant efforts included thermal deformation testing on the puck tower, minimizing measurement time through optimal path planning and research into effective estimation and localization navigation techniques.

The final section of requirements involve the safety of the robot and, more importantly, the observatory. These requirements again impact the efforts of both the mechanical and controls teams. If all of these requirements were to be summarized in one line, it would simply read, “The robot shall not damage the observatory in any way”. This includes: incorporating safe mode in case of loss of observatory power, ensuring the robot does not drive off the edge of the mirror, having a method of ‘catching’ the robot in case it detaches from the mirrors, and other factors of consideration.

5.2. CCAT-p Environmental Conditions

5.2.1. Operating temperature: -21° to +9° C
5.2.2. Survival air temperature: -30° to +25° C
5.2.3. Air pressure: 50 to 53 kPa
5.2.4. Relative humidity: 0 to 90%



5.3. Measurement Program Requirements

5.3.1. Retro-reflector must be placed at a minimum of 5 points per panel (over Z adjusters, with a goal of 9 points (additional 4 points between outer adjusters).

5.3.2. Placement repeatability is 1 cm absolute.

5.3.3. Retro-reflector z-axis offset from mirror surface measurement or repeatability is 0.5 micrometers RMS.

5.3.4. If z-axis offset is being established via measurement, z-offset measurements must be synchronized with Etalon measurements to within 1 ms, not accounting network latency. (NB: Etalon/Project should weigh in on synchronization).

5.3.5. Total mirror measurement time should not exceed 1 hour per mirror.

5.3.6. The robot must navigate in a pre-planned path across each mirror surface, stopping for each measurement.

5.3.7. At each measurement position the robot control system must broadcast a start single measurement request to the Etalon multiline server (Ref. doc: Annex-MultiLine as server-Online-EN-20.pdf).

5.3.8. The Retro-reflector must be unobscured to the laser measurement system (60 degree of clearance around).

5.3.9. The robot cannot inject greater than 1 micrometer RMS of un-filterable vibration into the mirror surface during measurements. (Ref. doc: Summary report_ current state of CCAT at Etalon_20180221.pdf)

5.3.10. Measure multiple elevation angles in one night without human interaction.

5.4. Safety and Operability Requirements

5.4.1. The robot cannot become detached from the mirror surface, or if detachment occurs cannot impact any mirror or observatory surface or equipment.

5.4.2. The robot must be capable of completing a full measurement cycle of one mirror without interruption (i.e., the robot must be continuously operable for the duration of one mirror measurement cycle).

5.4.3. The robot cannot drive off of the edge of the mirror.

5.4.4. In the event of any operational anomaly, the robot must be capable of placing itself in a safe mode.

5.4.5. Safe mode is defined as the robot meeting all safety requirements in a full power-off state.

5.4.6. An operational anomaly is defined as a violation or potential imminent (within 1 s) violation of any safety or operability requirement.

5.4.7. The robot must be capable of traversing the mirror surfaces, including any surface gaps or defects.

5.4.8. The robot cannot be capable of scratching, scuffing or in any other way damaging or affecting the performance of any mirror surface.

5.4.9. The robot must be capable of carrying out the measurement program (Sec. 2) in the full range of environmental conditions (Sec. 1).

5.4.10. The robot must survive and be capable of placing itself into a safe mode in the event of total loss of observatory power.

5.4.11. The robot must reply to Observatory Control System alarms/alerts (interface must be provided by project).

5.4.12. The robot must operate safely in the event of an earthquake, up to accelerations of 1 g.

6. Materials and Apparatus

6.1. Kalman Filter Materials and Apparatus

The microcontroller we are using to collect data from the I.M.U. is an Arduino MKR 1000. This specific type was chosen because when we originally needed its wifi capability to communicate with the off board computer when Raspberry Pi was not considered. Now that we are using a Raspberry Pi, we could potentially switch to a higher performance microcontroller without a built in wifi. The I.M.U. sensor we are using is an sparkfun MPU 6050 show in the left of Figure 4.

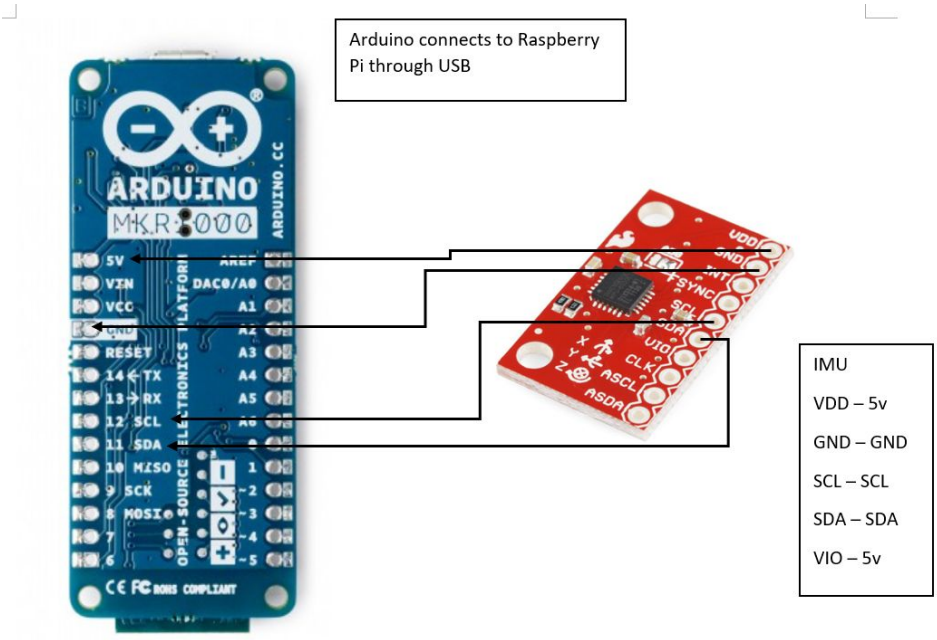


Figure 4: Hardware setting of the I.M.U.

Materials:

- Arduino MKR 1000
- Spark fun MPU 6050

- A micro-USB to USB cable
- A breadboard
- A computer with Arduino IDE
- The driver program written in Python 2.7 that can be found in the google doc under the controls folder in the Kalman filter code folder.

6.2. Edge Sensor Materials and Apparatus

Three distinct types of testing were done using the edge sensors. The first, more preliminary test will be referred to as the ‘Edge Detection Test’. The purpose of this test was to confirm that the QRE1113 Line Sensors are capable of sensing the presence of an edge of a surface. The second test, called the ‘Gap Detection Test’, was meant to test if the sensors could detect a small gap between two panels. The final test, which is in the process of taking place, is the ‘Corner Orientation Test’, which was a test to verify the robot’s ability to reorient itself upon switching to a new mirror panel. Each of these three tests uses slightly different materials and apparatus setups.

6.2.1. Edge Detection Test Materials and Apparatus

Materials:

- 1 QRE1113 Line Sensor (Digital)
- 1 line sensor holder
- 1 Arduino MKR1000 Microcontroller
- 1 Micro-USB to USB cable
- 1 Computer w/ Arduino software
- 3 Female to male pin wires
- 1 small breadboard

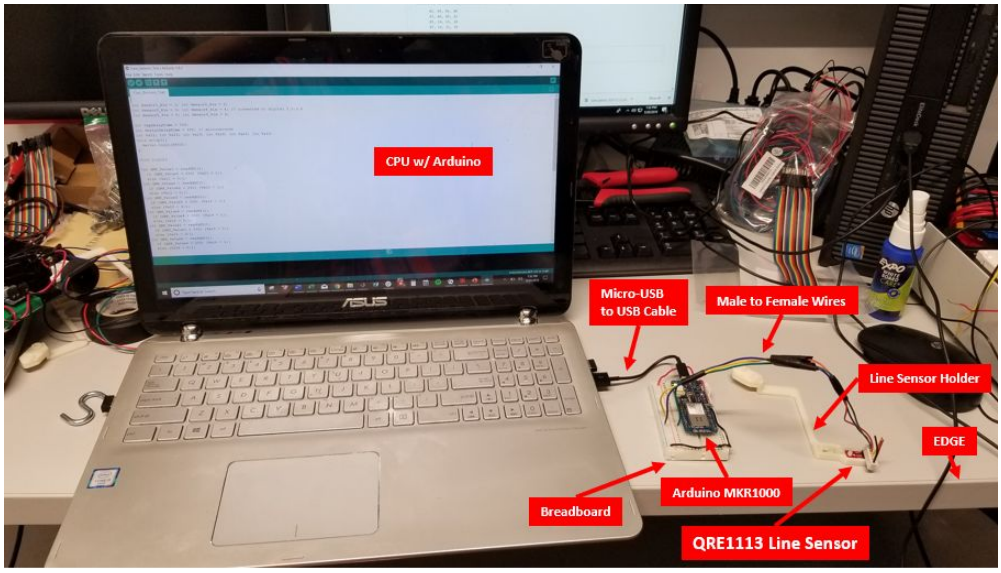


Figure 5: Edge Detection Test Apparatus

6.2.2. Gap Detection Test Materials and Apparatus

Materials:

- 1 QRE1113 Line Sensor (Digital)
- 1 line sensor holder
- 2 steel panels
- 1 Arduino MKR1000 Microcontroller
- 1 Micro-USB to USB cable
- 1 Computer w/ Arduino software
- 1 dial caliper
- 1 black Sharpie

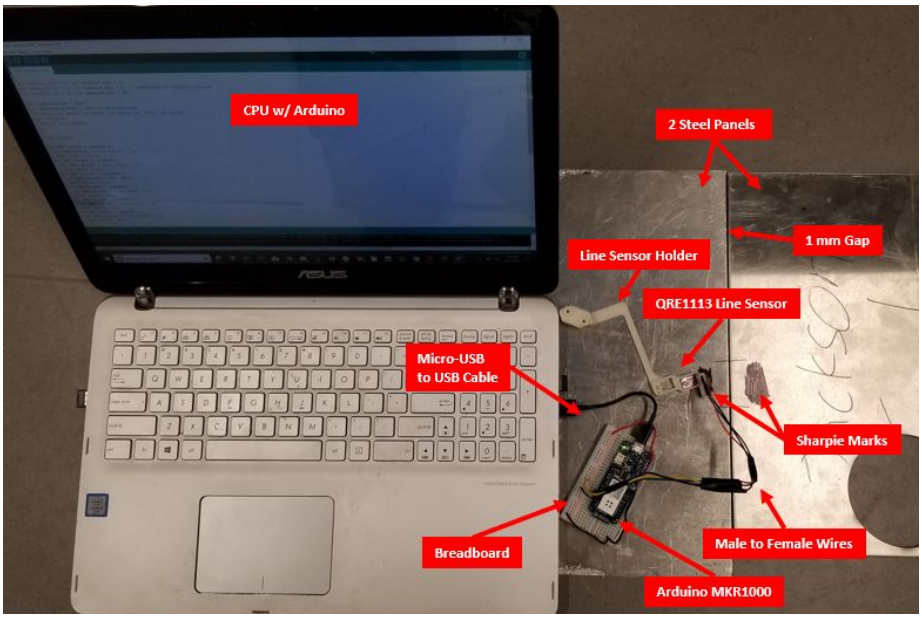


Figure 6: Gap Detection Test Apparatus

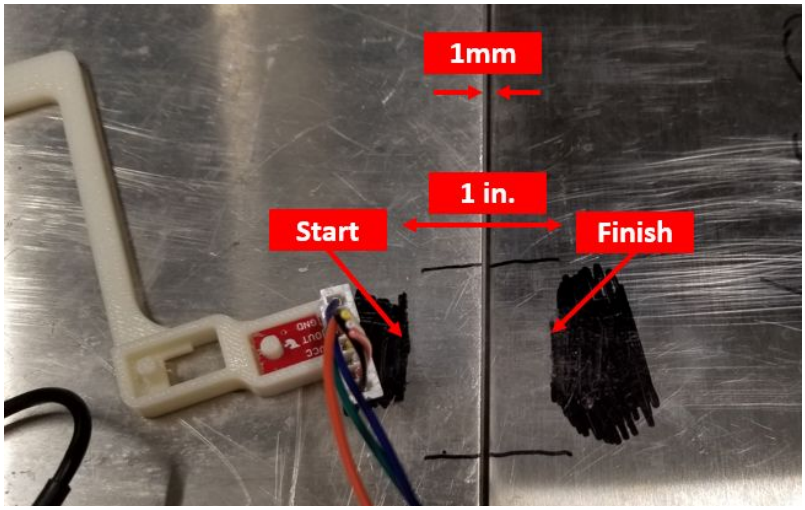
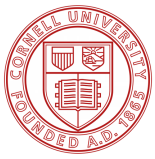


Figure 7: Zoomed in View of Gap Detection Test Apparatus



In Figure 7 a close-up image of the Sharpie marks can be seen. These marks were drawn so that when the steel panels had a 1mm gap, the closest points of the marks would be 1 inch apart. These marks were necessary because they allowed for the speed at which the sensor crossed the gap to be estimated. How this was accomplished is explained in the Procedures section.

6.2.3. Corner Orientation Test Materials and Apparatus

Materials:

- 6 QRE1113 Line Sensors (Digital)
- 3 line sensor holding ‘arms’
- 4 steel panels
- 1 Arduino MKR1000 MIcrocontroller
- 1 Raspberry Pi 2 Model B
- 1 Micro-USB to USB cable
- 1 Voltage Supply
- 2 Adafruit DRV8871 DC Motor Driver Breakout Boards
- 1 wall-climbing robot chassis
- 2 small breadboards

This test will take place in the near future. Thus, there is no image of the testing apparatus.

6.3. Coordinate System Transformation Materials and Apparatus

Materials:

- Computer
- MATLAB

For the testing of the coordinate system transformation, no physical apparatus was needed as all of the testing was done within MATLAB.

7. Procedure

7.1. Kalman Filter Motivation and Procedure

The purpose of using a Kalman filter is to improve the accuracy of the localization program. Our method of obtaining the position from acceleration is through integration. Integration is a noisy process because the error in each time step is accumulated through time. Furthermore, the I.M.U. sensor has a noisy output, merely integrating over the raw acceleration from the I.M.U. is not feasible. Figure 8 shows the comparison between acceleration, velocity,

and position integrating over the raw acceleration from the IMU. As we can see even when the I.M.U. is stationary, the outputs have an offset of roughly 0.1 m/s^2 and have a relatively large variance. This noise in the raw data results in a large discrepancy of the integration results. As we can see, the position quickly moved from 0 meter to 6 meters even when the I.M.U. sensor is not moving at all. This motivated us to use a Kalman filter to deal with this noise.

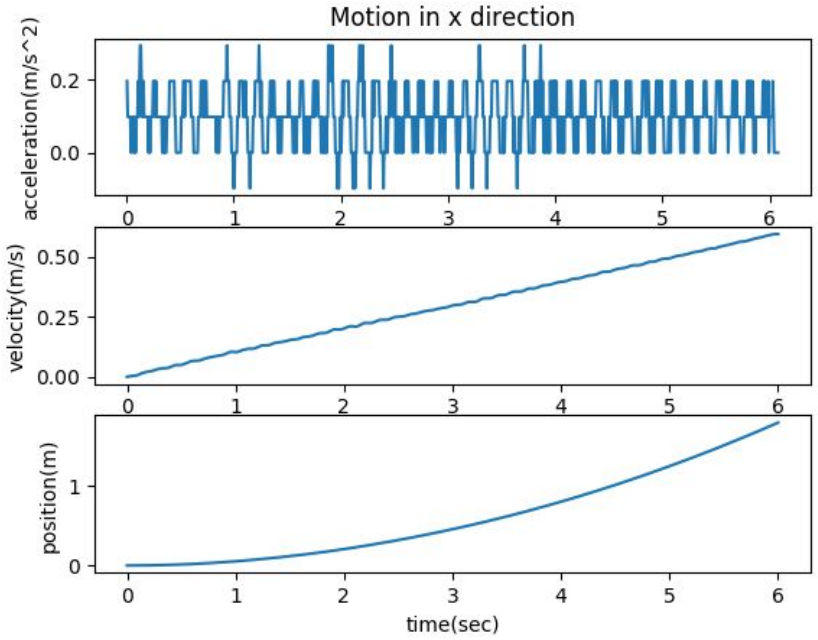


Figure 8: The result from integrating the raw I.M.U. data

To implement a Kalman filter, state variables first need to be introduced. In our case, the state variables are the position, velocity and acceleration in the x, y, and z direction. Input is assumed to be 0 for this version of the program. The integration process is included in the dynamics, for example:

$$X\boldsymbol{t} = X\boldsymbol{t}\text{-}1 + \Delta T + \Delta T\text{^}2/2$$

Hence the first row of the A matrix. The rest of the A matrix can be derived in a similar fashion. C is a 3 by 9 matrix because we only have acceleration in three directions for our measurements. The state variables multiplied by the C matrix will output the acceleration terms of the state variables. Q is a 3 by 3 matrix represents the measurement noise of our sensor. R is a 9 by 9 matrix represents the process noise in our dynamics.

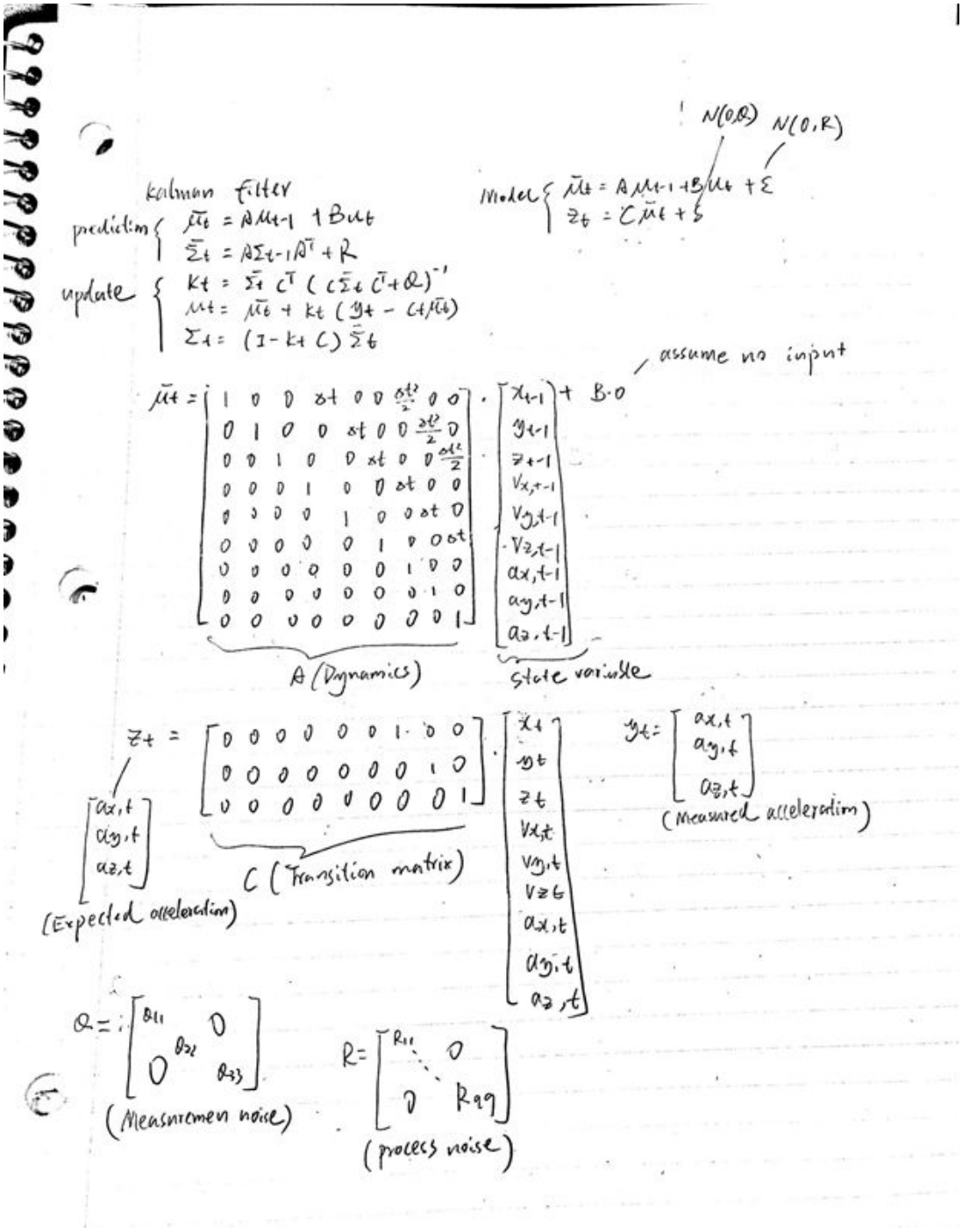


Figure 9: Equations for the Kalman filter

Figure 10 shows the simulation result of this Kalman filter in Matlab. The graph shows the comparison between the filtered and unfiltered acceleration in the x direction with the prior knowledge of sensor being in a stationary position. The process covariance is a 9 by 9 matrix with diagonal elements being 0.1. The measurement covariance is a 3 by 3 matrix with diagonal elements being 0.05. We can see from the graph that the blue line representing the filtered

measurements are much more smoother than the unfiltered red curve. Integrating over this filtered curve will generate a much better results than integrating over the unfiltered data. The entire data was taken in 20 seconds with a sampling frequency of 5 Hz.

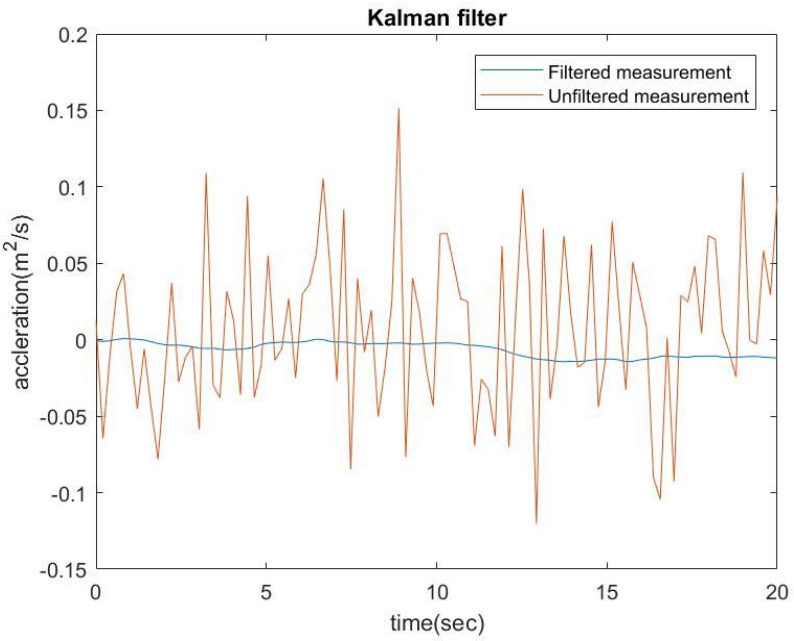


Figure 10: The comparison between filtered and unfiltered acceleration in x

After the simulation, this program was tested in the lab using the parameters that reflects the property of the physical sensor. Note that the parameter reflected in figure 10 is acceleration, to get the position, this value needs to be integration twice. Due to the nature of integration, the error in each time step accumulate. This means the error from the true pose will grow as the duration of the filtering increases. One way to mitigate this effect and keep the localization close to true pose is to update the global location every time we reach a measurement point. It is pointed by our client that the ETALON system is able to give us the global position of the puck everytime it point its laser to the puck. Therefore, we can rely on this location once we reach a measurement and we can start next step Kalman filter with current location provided by the ETALON system. The only parameter we need to consider is the travelling time between one measurement point to another. As specified before, the localization error grows with the time of integration. To get a precise localization, a shorter time period is preferred.

7.2. Controls Architecture Procedure

The controls architecture consists of 3 main components. An off board computer that performs all the computations and communicate with the multiline server from Edlon, a raspberry pi that executes commands from the off board computer, and a microcontroller that collects all the sensor information and send them to the Raspberry pi. Figure 11 shows the

interface between the different modules. The software we are using is Matlab, and its Raspberry Pi toolbox. This package allows us to establish communication between our off board computer and Raspberry Pi through SSH. The main purpose of this communication is to send control commands and collect sensor information from the Pi. For example, a constantly running while loop in the off board computer reads all the sensor information and based on these data runs Kalman filter to calculate robot's current location. Once the current location is obtained, it calculates the voltage that needs to be applied to the motors and sends this command to the Pi. On the data acquisition side, we have a microcontroller that only does data collection. An Arduino UNO is used to read data from the IMU, edge sensors, wheel encoders and eddy current sensor. It reads data from all these components for each time step and output it to the serial port on the Raspberry Pi.

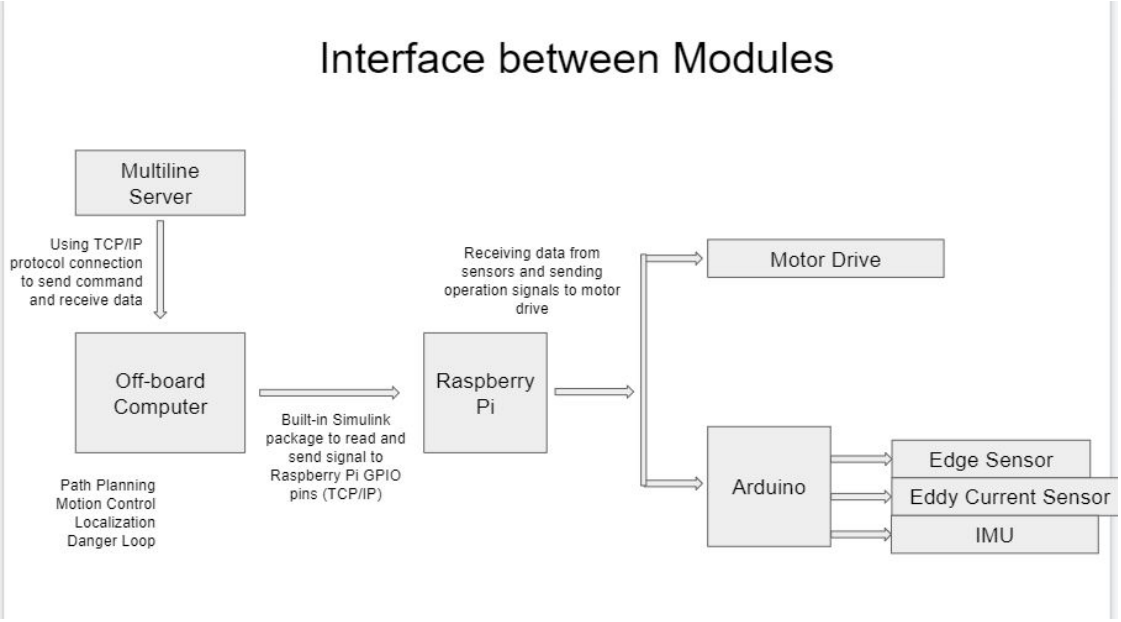


Figure 11: The interface structure between different modules

The advantage of using this structure is that we are allocating all the computations to the off board computer which has a much larger computing capability than the Raspberry Pi. The computational delay therefore is minimized. However, the communication between the off board computer and the Raspberry Pi might introduce delay when sending commands to the Raspberry Pi. This is something that needs to be tested in the future

Besides from collecting sensor data, another important job of the Pi is motor control. The PWM pins on the Raspberry Pi does exactly this job. By varying the width, in other words the duty cycle of a period, we are able to change the voltage that is applied through the PWM pin to the motor drive. By setting the duty cycle to 100% we are supplying the motor with the max voltage and by setting the duty cycle to 0%, we are essentially grounding the signal. Figure 12 shows the pin layout of the motor controls. To control the brushless motor that drives the fan, the setup is similar except a ESC (Electronic Speed Control) is included. In this setting instead of feeding to pwm signal to a motor control, the pwm signal goes directly to the ESC. The pin

layout can also be seen in Figure 12. There is one caveat with using PWM pins on the Raspberry Pi. The max voltage a PWM pin can output is 3.3 voltage. However, for brushless motors to work at its peak power, we need more than 3.3 V. In this case, the voltage from the pwm pins will need to be stepped up to match the requirement on the ESC.

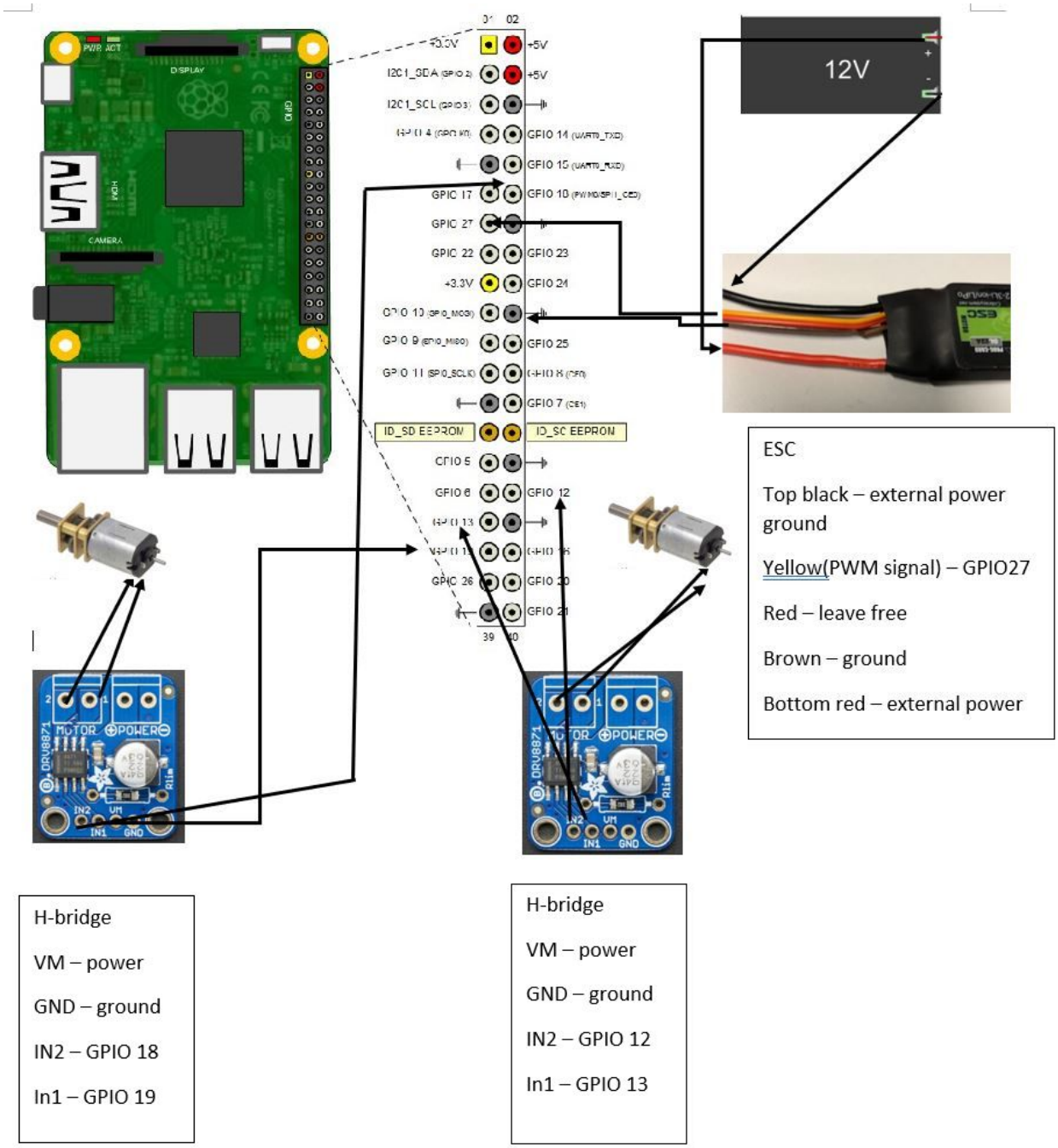


Figure 12: Pin layout of the motor controls

7.3. Edge Sensor Motivation and Procedure

7.3.1. Edge Sensor Motivation

Before describing the procedures of the edge sensor tests, the reasoning behind having edge sensors, as well as the features of the edge sensor, should be explained. The QRE1113 line sensor is a sensor most commonly used for line following robots. When placed very near to a surface - the optimal viewing distance being between two and three millimeters - the edge sensors begin to detect the presence of a surface. The QRE1113 has an IR emitting LED as well as an IR sensitive phototransistor which collectively send, receive and translate the IR signal into a numeric output value. The output value ranges from 0-3000, with numbers closer to zero signifying the sensor being closer to the surface. The actual numeric values are not necessarily important for this project; they are derived from charging a virtual capacitor within the program code and seeing how long it takes to discharge, which is directly related to the reflectivity of the surface (high reflectivity → shorter discharge time → lower output value). In its typical application of line-following, the low values would refer to the bot diverging from the black line, with the surrounding surface having a higher reflectivity than the black line. For this project only the *relative difference* of the output values between the surface-sensing case and edge-sensing case is important.

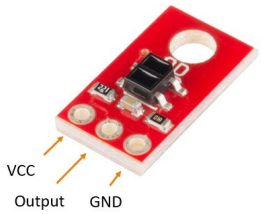


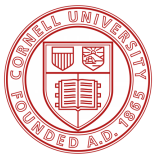
Figure 13: QRE1113 Line Sensor with pinout

7.3.2. Edge Detection Test Motivation and Procedure

The purpose of the edge detection test is to confirm that the QRE1113 sensor is able to clearly sense the difference between a surface and an edge. An edge, in the context of the telescope, would be when the bot is at the top, bottom, or either of the sides of the mirror. The edge is where the mirror ends. To simulate this, the edge sensor was placed in the edge sensor holder, which was 3-D printed in the Rapid Prototyping Lab specifically to house the edge sensors for testing. The holder was placed just off the edge of the table in the Upson 406 Lab and the program began acquiring data. The holder was repeatedly moved back and forth, on and off the edge of the table numerous times. Following the final iteration, the program was stopped and the output data was sent to an Excel document.

7.3.3. Gap Detection Test Motivation and Procedure

The motivation for the Gap Detection Test was to discover whether or not the edge sensor would be capable of detecting a small gap, on the order of 1mm, between surfaces. This



is necessary for this project for many reasons. First, the robot must know when it crosses on to a new mirror panel. According to Etalon, the gaps between the individual mirror panels are roughly 1mm in magnitude. The robot must know about its panel crossing in order to keep track of which mirror panel it is on at all times. Knowing the robot's current panel will simplify the measurement, navigation and danger loop processes. For the danger loop, knowing the current panel will allow the robot to know if it is near the edge of the mirror. For navigation, the robot must know how to navigate using estimation in its current 2-D panel frame.

For the Edge Detection Test, the two steel panels were carefully placed next two each other with a gap of 1mm. The steel panels were obtained from the basement of Upson, and are old, scuffed and different sizes. None of these are an issue, however; the test simply requires two surface similar to the mirror surface (aluminum), to receive comparable results. The gap was measured using dial calipers and, though likely not *exactly* 1mm, was close enough for the purposes of this test. Once positioned, the QRE1113 Line Sensor was placed in the 3-D printed holder and placed over the Sharpie mark on the panel. The reasoning for the Sharpie mark was to give the test a starting and finishing point with a known distance in between. While not necessary for the gap detection portion of the test, it was necessary for the speed estimation part of the test, which is another crucial part. Once over the Sharpie, the test began acquiring data and the holder was manually moved across the first panel, across the gap, and onto the Sharpie portion of the next panel. The data from this portion was then copied into an Excel document, which was then examined to determine the results of the test. This test was conducted multiple times with differing data acquisition frequencies as well as differing manual translational speeds.

To determine the *approximate* speed of the sensor's translation, the time stamps from the acquired serial data were used to find the difference between the time the sensor left the initial Sharpie mark to the time it arrived at the final Sharpie mark. The Sharpie locations are clearly defined by looking at the data, as the output at both ends of the test spiked to a much larger value to indicate the sensor was no longer on a reflective steel surface but rather on a Sharpie-marked part of the plate with less reflectivity. The Sharpie marks are shown in Figure 7.

7.4. Coordinate System Transformation Motivation and Procedure

7.4.1. Coordinate System Transformation Motivation

There are many sources of motivation for needing a way of transforming coordinate systems. One large issue in this project is knowing where the bot is always. Knowing the bot's exact location is useful in many ways: for safety purposes, to navigate back to the start in case of emergency, to have the correct tether lengths for the appropriate tension, to tell the Etalon measurement system where to shoot the lasers, etc. To know the bot's location on the mirror, it was decided that the measurement system would 'zero' itself upon the first measurement of the entire mirror. This measurement would take place on the very first panel and would become the datum [0,0,0] coordinate for the rest of the measurements. This way, both the robot and the

Etalon system would only need to worry about the bot's relative location rather than its absolute location in space.

While this is a simple solution, the problem becomes much harder when the layout of the mirror panels is considered. While the individual panels can be assumed to be two-dimensional surfaces, the overall shape of the mirror is three-dimensional. To achieve the crossed-dragone optical layout, the mirrors must be curved to fit a specified high order polynomial function which is given in the Optical Layout document (cite document). This higher order polynomial function is shown in Figure 14.

$$z(x,y) = \sum_{i=0}^{k} \sum_{j=0}^{k} a_{ij} \left(\frac{x}{R_N}\right)^i \left(\frac{y}{R_N}\right)^j$$

Figure 14: Higher Order Polynomial for z-coordinate of mirrors (The Optical Design)

The output of the above formula is the z-coordinate of the mirror, or the mirror 'sag'. The mirror sag is a function of x & y, the coordinates of the two-dimensional mirror surface, as well as R_N, the normalization radius. For both mirrors, $R_N = 3000\ mm$. These values are then summed up across the range of higher order polynomials, i, j to arrive at the correct z-value of mirror sag for every part of the mirror. In this case, $k = 6,\ k = 7$ for the primary and secondary mirrors, respectively. The last part of the formula is a_{ij}, which is a matrix of coefficients applied to every value of i, j throughout the summation. The a_{ij} matrices for both mirrors are shown in Figure 15.

```
% Matrices of 'a' coefficients from Optical Layout Doc.
%       j=0           j=1       j=2       j=3      j=4        j=5      j=6
a1Mat = [0             0      -57.74022  1.5363825  1.154294 -0.441762 0.0906601    % i=0
         0             0         0         0        0         0        0            % i=1
         -72.17349 1.8691899 2.8859421 -1.026471  0.2610568   0        0            % i=2
         0             0         0         0        0         0        0            % i=3
         1.8083973 -0.603195 0.2177414     0        0         0        0            % i=4
         0             0         0         0        0         0        0            % i=5
         0.0394559     0         0         0        0         0        0 ];         % i=6

 %       j=0           j=1       j=2       j=3      j=4        j=5      j=6       j=7
a2Mat = [0             0     103.90461 6.6513025  2.8405781 -0.7819705 -0.0400483 0.0896645    % i=0
         0             0         0         0        0         0         0         0            % i=1
         115.44758 7.3024355 5.7640389 -1.578144 -0.0354326 0.2781226     0         0            % i=2
         0             0         0         0        0         0         0         0            % i=3
         2.9130983 -0.8104051 -0.0185283 0.2626023     0         0         0         0            % i=4
         0             0         0         0        0         0         0         0            % i=5
         -0.0250794 0.0709672      0         0        0         0         0         0            % i=6
         0             0         0         0        0         0         0         0 ];         % i=7
```

Figure 15: [A] Matrix of coefficients for primary mirror polynomial

Because of the mirror curvature, knowing the exact location of the robot becomes far more difficult. The issue of autonomous navigation for the robot becomes far more difficult as

well. A robot cannot be expected to autonomously traverse a 3-D surface, especially a 3-D surface with such a complicated curvature. Thus, it was decided that the entire structure of the mirror cannot be used for navigational purposes. A different, simpler method must be chosen.

The decided upon solution was to use each individual panel as its own coordinate frame for navigation. Since each panel is only a 2-D surface, the problem of navigation and estimation was converted from a 3-D to a 2-D problem. Though the 2-D plane would still be in 3-D space – that is, that it would lie outside of the x-y, x-z or y-z planes in the World Frame – it would nonetheless still be two-dimensional and simple rotation matrices could be used to transform the I.M.U. 's 3D integrated positioning into the current 2-D frame.

Once this solution was in place, it became necessary to create the rotation matrices to rotate the 3-D World Frame I.M.U position vectors into 2-D Panel Vectors. This task is difficult, since the 3-D mirror curvature must be accomplished using the 2-D front surfaces of the mirror panels. To do this, MATLAB was used extensively.

7.4.2. Coordinate System Transformation Procedure

To begin, the two-dimensional layout for both mirrors was created using the mirror panel dimensions in Figure 16 and Figure 17 and the mirror panel layouts in figures 18 and 19.

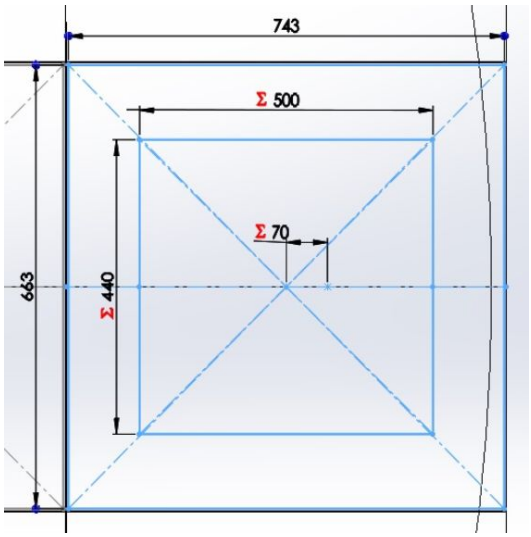


Figure 16: Mirror 1 panel dimensions

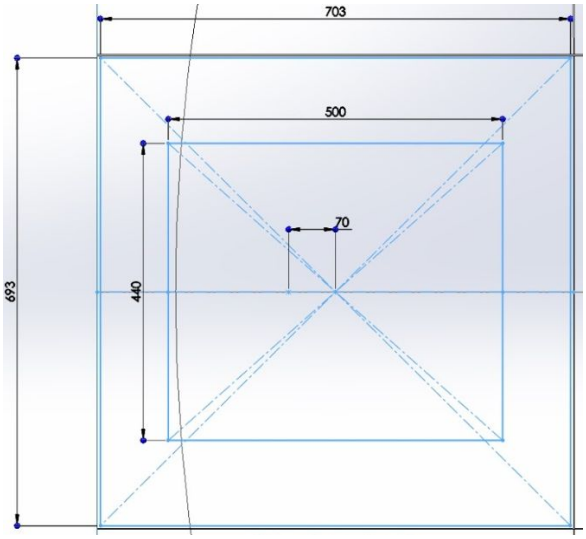


Figure 17: Mirror 2 panel dimensions

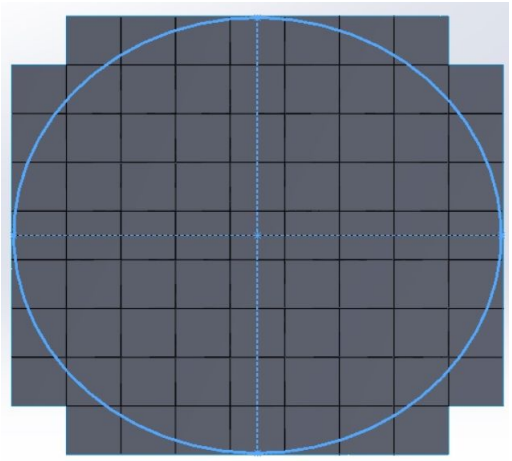


Figure 18: Mirror 1 panel layout

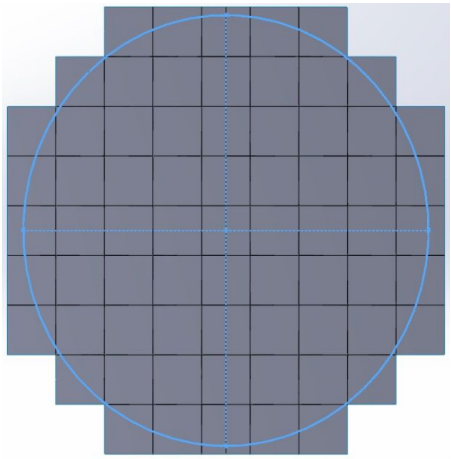


Figure 19: Mirror 2 panel layout

Once the two-dimensional mirror layouts were created, the higher order polynomial fit was applied to the matrices to create a three-dimensional layout of the mirrors. These 3-D mirrors were then rotated into their relative 3-D positions using their respective angles shown in the optical layout document. Once rotated, the mirrors were then translated from being centered at the [0,0,0] World Frame origin to their respective positions in the observatory. Mirror 1 was translated down the boresight axis, while Mirror 2 was translated down the elevation axis. The relative locations in the World Frame are shown in the figure below.

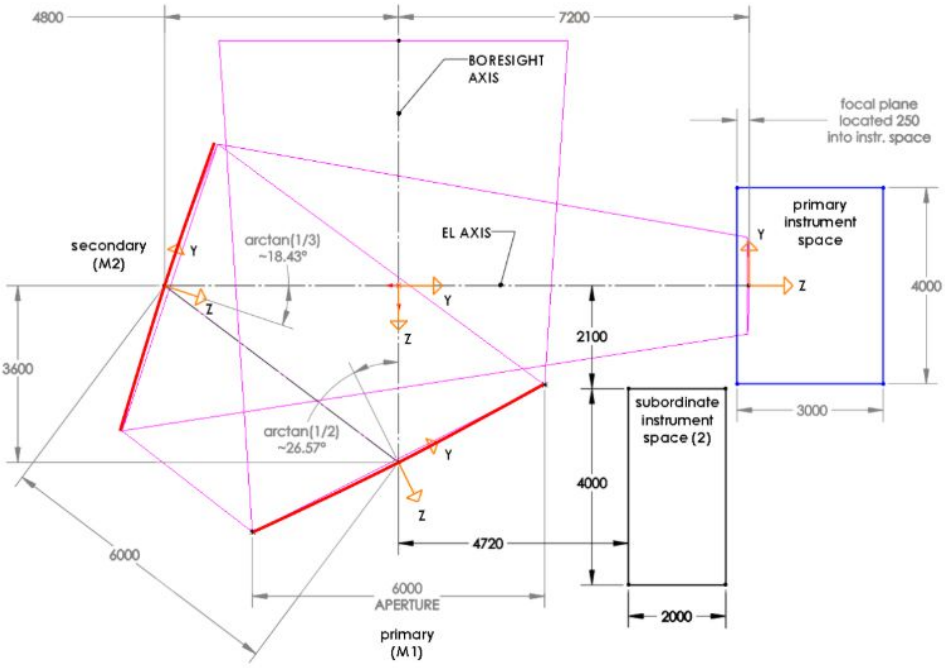


Figure 20: Optical Layout for production design (The Optical Design)

It is important to use the terms ‘boresight’ and ‘elevation’ when referring to the World Frame axes, because the telescope rotates about both of these axes, thus rotating the world frame as well.

8. Discussion and Results

8.1. Kalman Filter Results

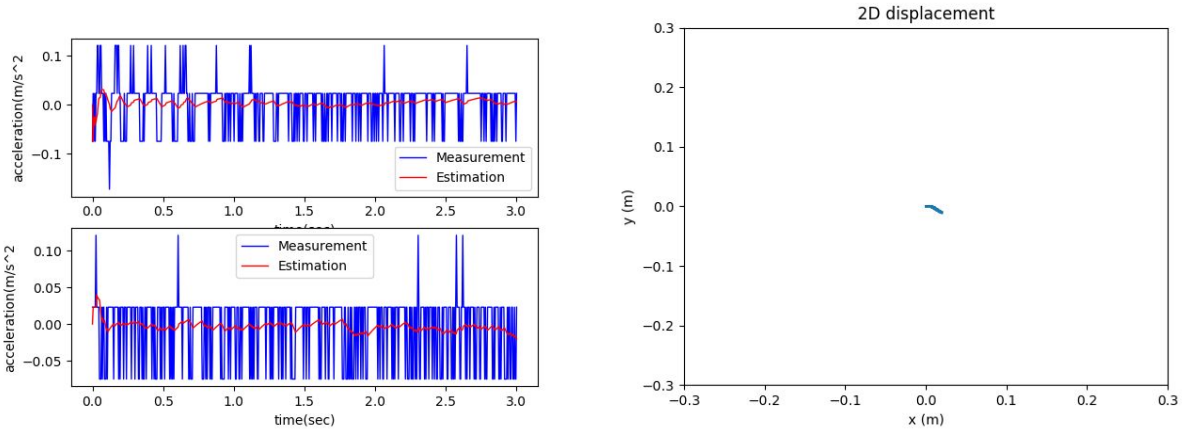


Figure 21: The testing results of Localization using Kalman filter

Figure 21 shows the localization testing results. The left graph shows the unfiltered and filtered acceleration in x and y direction. The blue curves are the raw acceleration from the IMU, and the red curve is the acceleration after passing the raw data through the Kalman filter. The right graph is the sensor coordinates after integration in a global frame. This test was conducted while keeping the I.M.U. sensor stationary. The total testing duration is 3.3 second with a frequency of 30 Hz. Repeating this test for 10 trials, the results were recorded in Table 1.

Table 1: Results of Kalman Filter Test

Trials	1	2	3	4	5	6	7	8	9	10	variance
X(cm)	-0.16	-1.33	2.21	-0.35	-1.13	-0.48	-0.33	-0.04	0.18	-1.18	1.016
Y(cm)	0.43	-0.25	-0.81	-0.73	1.77	0.66	-0.46	-2.07	1.73	-0.45	1.339

Based on these 10 trials, the variance in x direction is about 1.016 cm, and the variance in y is about 1.339 cm. This means when the sensor is stationary, the position from integrating the acceleration with in a 3 second time frame would be about (1.016, 1.339) cm.

8.2. Control Scheme Results

The testing of the control scheme requires the testing and integration of all computers and sensors. As a result, for the communication between the off board computer and the on board computer, Matlab is used in combination with a Raspberry Pi toolbox that can be found as an addon for Matlab. Edge sensor and I.M.U. can be directly wired to the arduino through I2C protocol. However, eddy current sensor can not be wired directly to arduino. The output of the eddy current sensor is an analog signal from 0 to 10V, but the maximum analog signal an arduino can read is about 5V or 3.3V. This requires us to have an a2d (analog to digital) converter between the arduino and eddy current sensor. The purpose of an a2d converter is to convert the signal from analog to digital so that the operating voltage is not bounded by the maximum analog voltage on an Arduino. As for motor control, four h bridges are used to control four motors. However, for brushless motor that drives the fan, an ESC(electronic speed controller) is used instead of a motor drive. As specified in the previous section, the maximum voltage a PWM pin can output on a raspberry pi is 3.3. Therefore, to get the max power from the brushless motor, an additional voltage step up needs to happen between the PWM pins and the ESC.

8.3. Edge Sensor Results

8.3.1. Edge Detection Results

The results of the Edge Detection Test are shown in Figure 11..

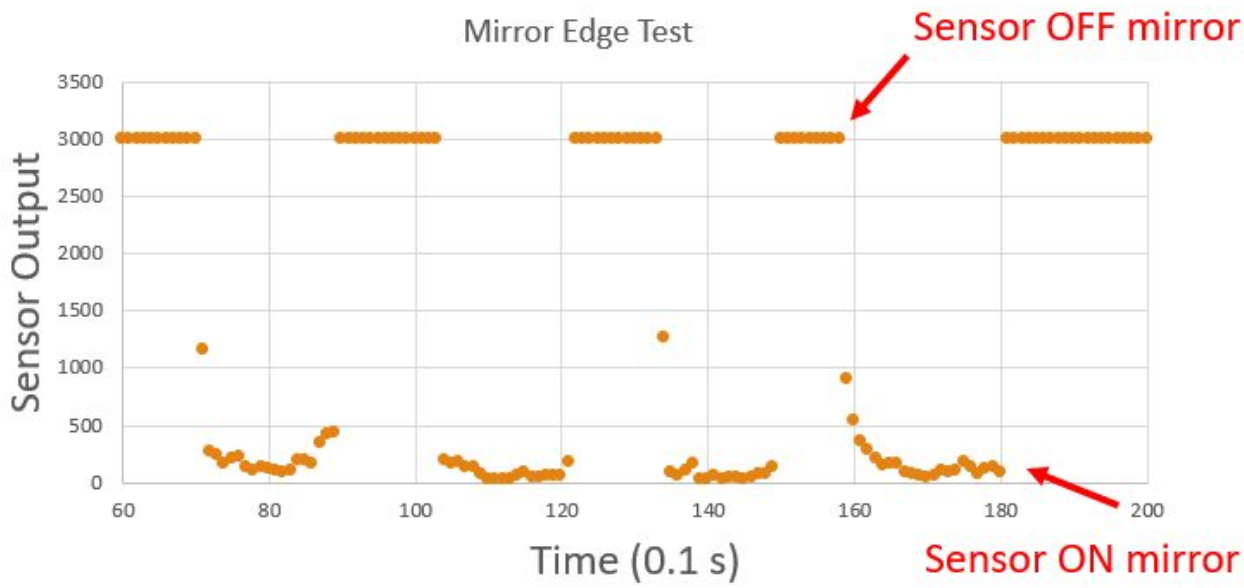


Figure 22: Edge Detection Test Results

The results of the Edge Detection Test are clear and straightforward. The sensor output reads high (3000) when the sensor was off of the edge and low (50-200) when the sensor was

above the table surface. The edge of the table was slightly rounded and the sensor was not manually translated exceptionally fast, which is a possible explanation for the curvature at the beginning of each of the low output sections.

8.3.2. Gap Detection Results

The first gap detection test resulted in the data shown below in Figure 23.

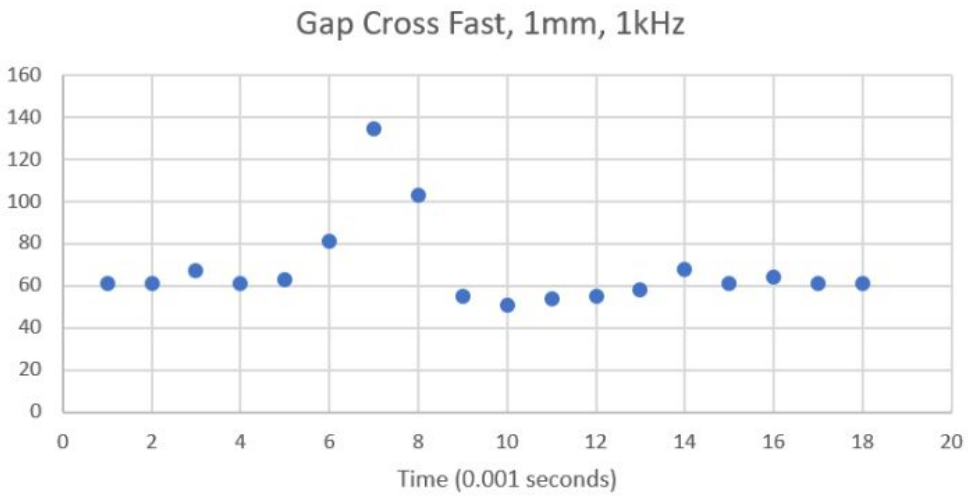


Figure 23: First Gap Detection Test Results

As seen in the figure, there is a clear spike in the output data between 0.006 and 0.008 seconds. This spike represents the gap between the mirrors when the sensor was no longer reading the reflective surface. For this test, the estimated speed the sensor traveled while crossing the gap was determined to be ~1270 mm/s. This value was determined using the method described in the Procedure section. This speed is a speed far faster than the bot will ever travel during the measurement sequence. Thus, the fact that a gap was detected at this speed was an initial promising result that the edge sensors could achieve what we needed of them. However, the data acquisition frequency was 1kHz, which only resulted in one notably different data point to signal the gap.

The second gap test was done at a much slower rate with the data acquisition frequency at the same 1kHz value. The results are shown in the figure below.

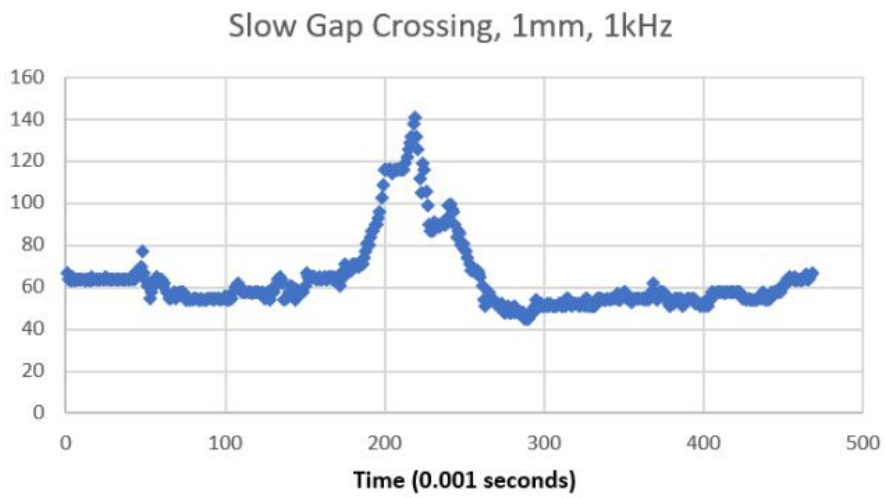


Figure 24: Second Gap Test Results

From this test, the gap is much more evident and many more data points signal the gap. This test was conducted at ~53 mm/s. This value is more in-line with the speed the bot will be traveling than in the first test.

From both the initial two tests, the gap is clearly detected. However, the first test shows that a faster data acquisition rate will be necessary to create more data points to signal the gap. With only one data point showing a gap, there is a much higher likelihood of an error occuring the robot's code which would result in not signaling to the robot that a gap has been crossed. The following tests were conducted with a higher data collection frequency.

The results from the third test are shown below.

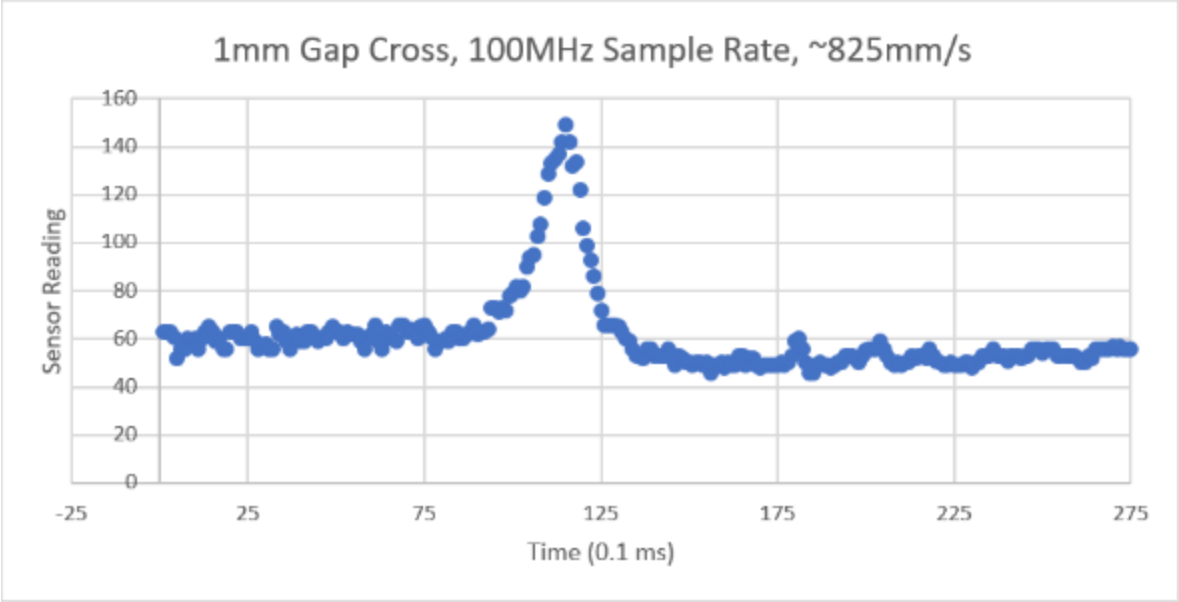


Figure 25: Third Gap Test Results

From Figure 25 there is again a clear gap indication via the spike in output. This test was conducted at ~825 mm/s, which is still a value higher than the robot will ever travel. At the frequency of data collection, there are many data points which make the presence of a panel gap indisputable. More specifically, the number of points above the output value of 100 was determined to be 14.

The results of the final two gap detection tests are shown in the two figures below.

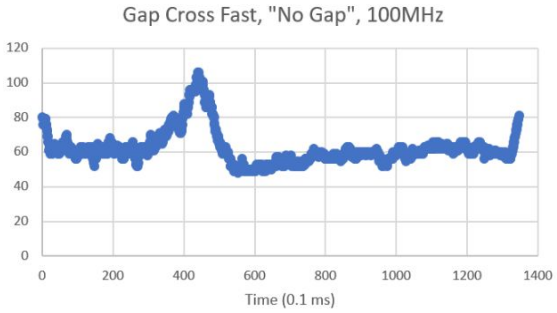


Figure 26: 'No Gap' Test #1

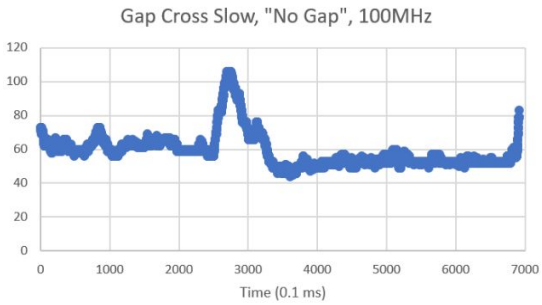


Figure 27: 'No Gap' Test #2

For these final two tests, the steel panels were pushed completely together, eliminating the 1 mm gap. The goal of these tests was to 'go overboard' by proving that even an unmeasurably small gap between panels will be detected by the edge sensors. For the first of these tests, the sensor had a speed of ~191 mm/s and in the second test the sensor was traveling ~37 mm/s. Both of these tests have results in full agreement with the previous tests and definitively confirm that the QRE1113 Line Sensor is an optimal choice for the CCAT-p Wall Climbing Robot.

One last thing to note from the final two figures is the presence of the Sharpie marks bookending the data points. In the first figure both the beginning and end contain the Sharpie mark data, and in the second figure only the end shows the effect of the Sharpie. This simply shows the effectiveness of using the Sharpie markings to discern how long it took for the sensor to travel the one inch between them.

8.3.3. Corner Orientation Results

The Corner Orientation Test is still in process, so there are no results from it yet. The end result will ideally be that the bot is able to use three pairs of sensors to sense the bottom and left/right edges of the panel to 'zero' itself at the bottom corner of the panel. After doing so, the bot will know exactly where it is on the panel, and will thereafter refer to this point as [0,0,0], or the origin of the current panel frame. All navigation and estimation functions will use this point as the datum and will use the 2D panel frame coordinates to traverse the current panel. As for this test, a successful result will be the bot traveling to the panel corner, sensing the respective mirror panel gaps, and 'shimmying' itself until said gaps are located between each pairing of the edge sensors. Figure 28 shows how the locations of the edge sensors are designed to allow for this reorientation.

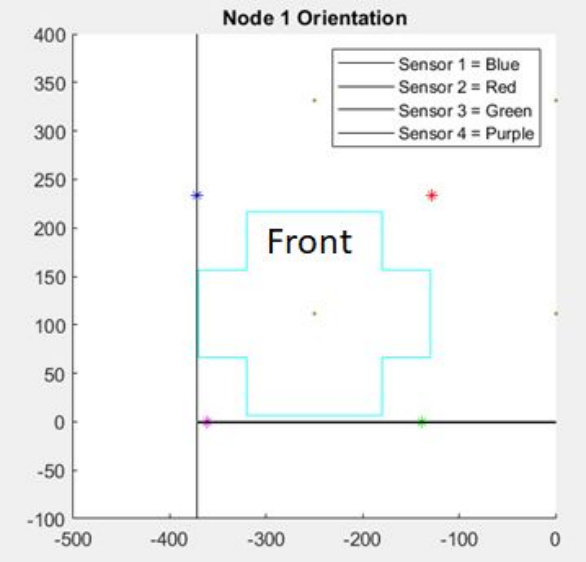


Figure 28: Diagram of robot upon completion of corner reorientation

The above figure shows the locations of the edge sensors and the chassis of the robot when it has reoriented on the bottom corner of a new panel. The blue outline is a scale representation of the bot's chassis and the stars are the edge sensor locations. Each of the stars represents a *pair* of sensors. Upon completion of the reorientation, three sensor pairs will be straddling gaps between panels. In the case shown above, with the robot translating up the panels, the first sensor pair (blue star) will be straddling the left panel gap (or mirror edge), while the third and fourth sensor pairs (purple and green stars, respectively) will be stradling the bottom panel gap. At this point, the first measurement for the panel will be taken, as the sensors were strategically placed for the bot to be centered about the first node from orientation. Node 1 in the figure above is the dot centered in the robot chassis.

8.4. Coordinate System Transformation Results

Following the sequence of events in the procedure, the first step in this process was to create the two-dimensional mirror layouts. This was done by taking the mirror panel dimensions and iteratively plotting the outlines of each of the panels. The result is shown in the figure below. On this figure is also the path of the robot, colored blue.

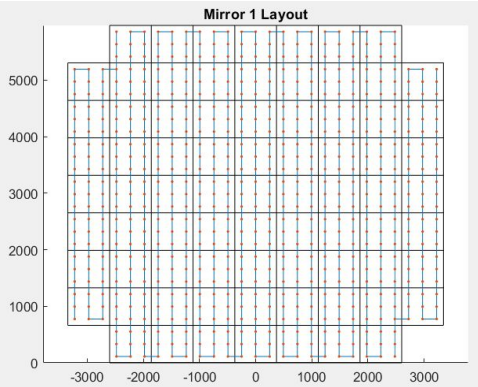


Figure 29: Mirror 1 panel layout

After the layouts were created, the mirror sag polynomial fitting function was incorporated to create the corresponding three-dimensional versions of the mirrors. The resultant mirrors are shown below.

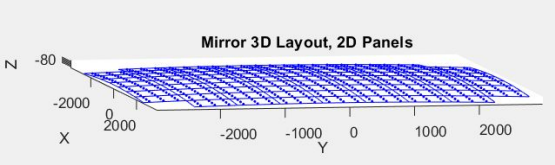


Figure 30: Mirror 1 with 3D mirror sag

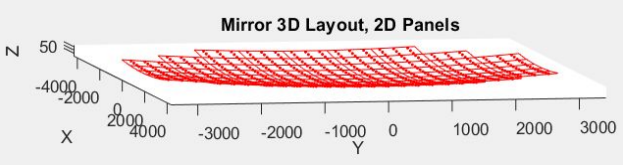


Figure 31: Mirror 2 with 3D mirror sag

Following the transformation from two to three dimensions, the mirrors were then rotated to the angles at which they reside inside the telescope. The respective angles are shown in Figure 20. Mirror 1 is rotated 26.57° CCW from the boresight axis while Mirror 2 is rotated 18.23° CW from the elevation axis. Both rotations occur within the boresight-elevation plane. The resultant mirrors are shown in Figure 32. Note, in the code the z-coordinates of Mirror 1 were multiplied by -1 to flip about the elevation axis before being rotated by 26.57°.

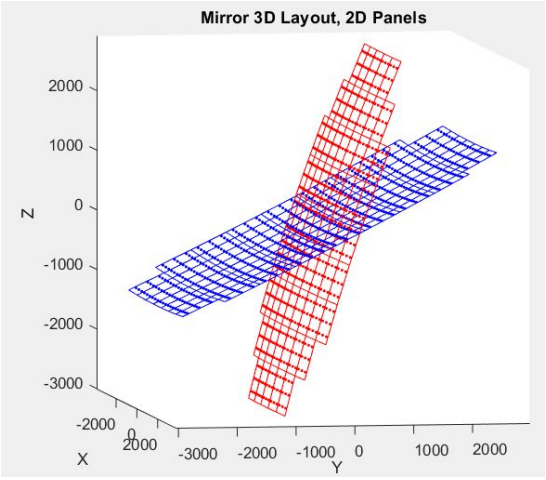


Figure 32: Rotated Mirrors

Following this, the mirrors were translated from their positions in Figure 32 into the relative positions to each other within the observatory. The result is shown in Figure 33. Mirror 1 was translated by 3600 mm while Mirror 2 was translated by 4800 mm.

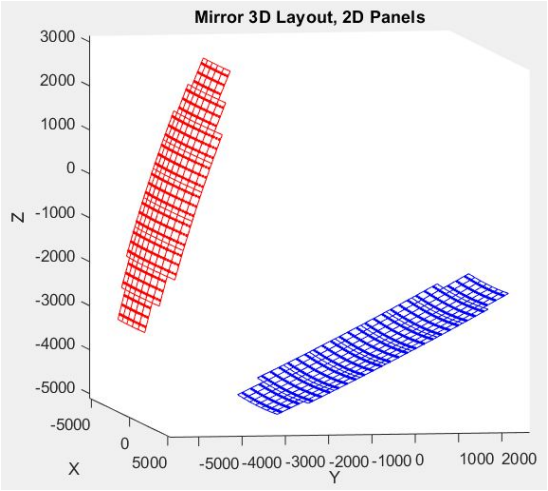


Figure 33: Rotated and translated mirrors

Figure 33 shows the mirrors in their final positions and orientations within the observatory and in the Horizon/World Frame. From here, individual Panel Frames were created for each of the panels on both mirrors. This was done by taking the bottom left, top left, and bottom right corner points of the panels and creating an x and y unit vector from them. Then, a z unit vector was created by taking the cross-product of the two vectors. This process was iterated throughout the 77 panels of Mirror 1 and the 69 panels of Mirror 2, resulting in two large matrices, 231x3 and 207x3 for Mirror 1 and Mirror 2, respectively. Every three rows of these matrices begins a new set of unit vectors for the Panel Frames represented in World Frame coordinates. Since the vectors are already of unit length, these matrices became the rotation matrices by which the World Frame unit vectors are multiplied to arrive at the Panel Frame. An example of a set of Panel Frame unit vectors is shown in Figure 34 and is the rotation matrix associated with the first panel of Mirror 1.

1	2	3
0.9990	0.0196	-0.0391
0	0.9044	0.4268
0.0438	-0.4264	0.9035

Figure 34: Mirror 1, Panel 1 unit vectors (World Frame Coordinates)

Lastly, the observatory elevation and azimuth angles, Φ and Ψ, were incorporated into the model. Originally this was seemingly not needed for this project, since the Etalon measurement system will be fixed in the observatory, thus it will rotate as the mirrors do resulting in unchanged 3-D

measurements regardless of orientation. However, the I.M.U. used to track the robot's position will be measuring in absolute World coordinates and cannot differentiate the observatory's different angles, so incorporating the angles when calculating the Panel Frame unit vectors is, in fact, necessary. Below shows the mirror layout when ϕ=0 and ψ=60°.

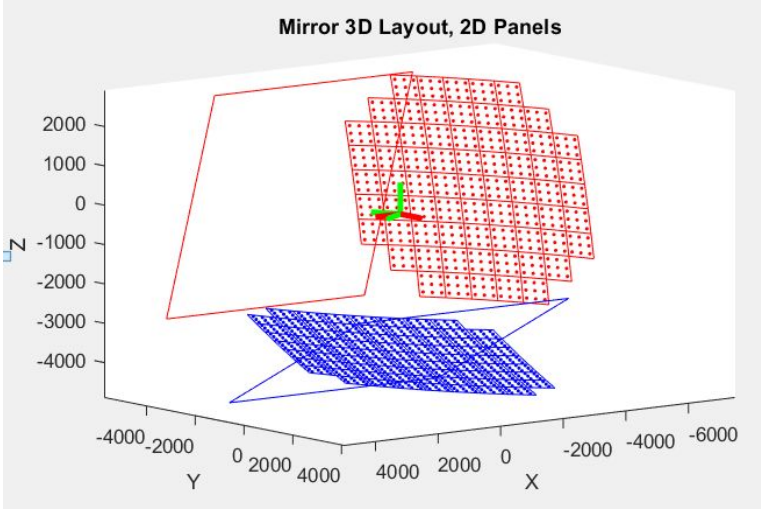


Figure 35: Mirror diagram, ϕ=0 and ψ=60°

The lines in the middle of the observatory are the unit vectors of the initial and final World Frames. The red lines show the WF unit vectors before rotation and green shows the resultant rotated WF axes. Also, the red and blue squares represent the original position of the mirrors. To now demonstrate the effect of ϕ on the mirrors, Figure 36 shows the mirrors from Figure 35 rotated 30° about the intermediate elevation axis.

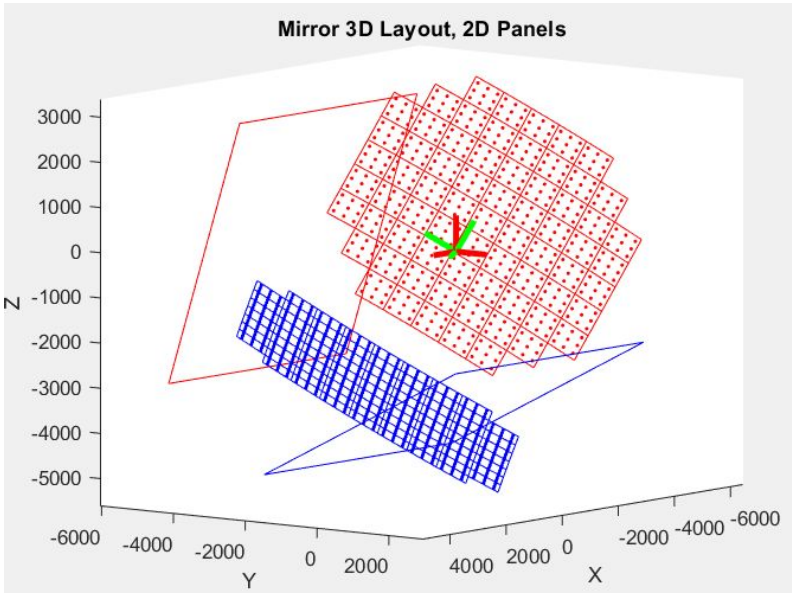


Figure 36: Mirror Diagram, ϕ=30° and ψ=60°

Thus, both the observatory angles are taken into account in Figure 36. To show the effect of the rotations on the Panel Frame unit vectors, Figure 37 shows the same Panel Frame coordinates as in Figure 34 after applying the rotation.

5	0.4258	-0.7480	0.5091
5	0.6576	0.6422	0.3938
7	-0.6215	0.1672	0.7653

Figure 37: Rotated Mirror 1, Panel 1 unit vectors (World Frame Coordinates)

After this process, the same concepts and much of the same code was transferred to the Simulink simulation and tested. The result of this consisted of the following input: phi and psi angles and a methodically created ‘mock’ Etalon measurement value in World Frame coordinate, and the following output: Etalon measurement value in final Panel Frame coordinates.

9. Recommendations (Next Steps)

Given the timeframe for this project, the current state is not near completion. There are many areas in which more efforts must be made in order to achieve the long list of requirements. In this section, some full-group efforts will be listed, but the tasks written about are mostly related to the controls team specifically.

9.1. Tilt

One necessary source of further work is the issue of robot tilt. Because of the offset from the reflective puck to the mirror surface, the Etalon measurement will need to take into account the height of the puck. This height will be assumed along the Panel Frame z-axis, meaning the puck tower is exactly perpendicular to the panel surface. However, if the chassis of the robot is tilted relative to the panel surface, the puck tower will no longer be along the z-axis and will give a slightly incorrect measurement reading. Etalon requires that the puck center be no more than 1µm offset from the original z-value in the Panel Frame. This is shown in the diagram below.

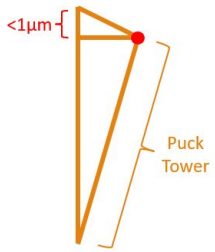


Figure 38: Puck Tower Tilt Diagram

With the current puck tower height, this tilt cannot exceed ±0.3 degrees. The source of this tilt could come from unequal wheel deformation, unequal mass loading on the bot and/or uneven fan suction force, to name a few. As of right now, ideas for how to either mitigate or precisely sense this tilt have been proposed, but no solution has resulted.

9.2. Edge Sensing Code

With the edge sensors proving their feasibility for this project, the next step is to write the code which will be used to sense the edge of the mirror. Also, this code must be integrated into the overall control scheme so the danger loop, estimation, coordinate transformation and other operations can work together.

9.3. Encoder Integration

To further improve the accuracy of localization and controls, an encoder is needed for each wheel to keep track of how far the motor has turned. Essentially, the encoder will provide two insights, linear displacement and angular displacement the robot has traveled since last time step. Unfortunately, to encoders we ordered this semester never arrived due to some complication in placing the order. Therefore, in the future encoders are still need to be integrated into the system.
Once the encoders arrived, we can replace the dynamics in the Kalman filter with a dead reckoning matrix knowing the linear displacement and angular displacement for each time step. The expected measurement can be derived by taking the second derivative of the position. This will improve the accuracy of the localization program drastically since we are no longer integrating over the acceleration.

9.4. Complete Simulation Testing

All the modules need to be tested in simulation before the field testing. So far a simulation is being built in simulink. The purpose of this simulation is to run all the modules the simulate the robot trajectories under the assumed condition. Currently the code I.M.U. program, frame transformation program and the motor drive program have been integrated in the simulation. We still need the eddy current sensor and the edge sensing code in the simulation.

9.5. Physical System Integration

The physical system still needs to be integrated. So far, each individual module has been tested. For example, we have written the arduino program and tested all the sensors individual on the microcontroller. However, we still need to assemble all the sensors on one single Arduino. The first step to achieve this is to obtain a better arduino microcontroller. So far the model we are using is MKR 1000. It has a built in wifi, however, the number of pins on it is

limited. Based on the number of sensors and the type of pins each sensor requires, we need to select a larger board that is more suited for our purpose.

9.6. Physical Testing

Once final assembly is complete, the bot will need to be sent to Germany to conduct tests with the Etalon measurement system. Before this, the bot will be tested in-house to ensure it is prepared to be tested in an operational environment.

10. Conclusion

In conclusion, significant strides were made in the areas of localization, control scheme, edge sensing and coordinate system transformation. A method of integrating the I.M.U. acceleration readings, layout of the hardware structure, preliminary testing of sufficient edge sensors and a method of rotating reference frames are a few of the results explained in this paper. These efforts have and will go a long way in the design and assembly of a robot capable of meeting the aforementioned requirements. These efforts will be continued and refined by a few students over the summer of 2019 and through the 2019-2020 academic year. All of the documentation, code, methods and results mentioned in this paper will be recorded, annotated and relayed to the future students of this project in order to assist in their understanding of this project.

11. References

Ccatobservatory.org. 2019. *CCAT*. http://www.ccatobservatory.org/

En.wikipedia.org. 2019. *Horizontal coordinate system*. https://en.wikipedia.org/wiki/Horizontal_coordinate_system [Accessed 21 May 2019].

Ibrahim, M. and Moselhi, O. (2016). Inertial measurement unit based indoor localization for construction applications. *Automation in Construction*, 71, pp.13-20. Summary report_ current state of CCAT at Etalon_20180221.pdf

Industries, A. 2019. *Adafruit DRV8871 DC Motor Driver Breakout Board - 3.6A Max*. Adafruit.com. https://www.adafruit.com/product/3190

Parshley, Stephen. *The Optical Design of the Six-Meter CCAT-Prime and Simons Observatory Telescopes*. 2018, *The Optical Design of the Six-Meter CCAT-Prime and Simons Observatory Telescopes*.

QRE1113, QRE1113GR Miniature Reflective Object Sensor. Fairchild Semiconductor Corporation, 2011, cdn.sparkfun.com/datasheets/Sensors/Proximity/QRE1113.pdf.

SparkFun Line Sensor Breakout - QRE1113 (DIgital) - ROB-09453 - SparkFun Electronics.Sparkfun.com. https://www.sparkfun.com/products/9453