

ROBOTIC LABORATORY FOR DISTANCE EDUCATION

By

RYAN KENNETH BERGSMA

---

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors degree  
With Honors in

Systems Engineering

THE UNIVERSITY OF ARIZONA

MAY 2016

Approved by:

---

Dr. Gerald Pine  
Department of Aerospace and Mechanical Engineering

## Abstract

Distance education is a growing trend in today's society. One component, however, that has not been able to be translated well to online learning is the physical experience of a laboratory experiment. Our particular project was to motorize a preexisting optical engineering laboratory experiment so that it could be performed remotely over the Internet anywhere in the world. We engineered this with the goal of providing as much of an in-lab experience as possible hopefully demonstrating the viability of this concept as an educational tool. Ultimately, the project does demonstrate that it is possible to translate physical laboratories to online experiences where the critical concepts of the lab are still effectively learned through the manipulation of a limited selection of the hardware components. However, as a supplement to distance students' restricted curriculum, this system can be an invaluable learning tool. In the future, this system could be scaled to incorporate more laboratory experiments with the ultimate goal of creating a fully online curriculum.

## **Statement of the Roles and Responsibilities of all Group Members:**

### **Ryan Bergsma (Systems Engineer):**

Ryan was the systems engineer for the project and served as the project team leader. Ryan was heavily involved in the preliminary system design, including the development of both functional and non-functional requirements, test cases, verification matrices, and tradeoff analyses. He served as the primary communication with the sponsor as well as with the mentor. Ryan was in charge of setting up meetings, taking meeting minutes, and delegating tasks. Furthermore, Ryan outlined most of the presentations and reports and filled in his respective sections. Ryan was also responsible for project management and recording test results.

### **Scott Appleby (Mechanical Engineer):**

Scott was the mechanical engineer for the project. He designed and manufactured all of the mechanical aspects of the project, including the motor mounts for the spatial filter, the camera mounts, and the projection screen mounts. Scott designed the mechanical components in CAD, and then purchased the materials needed to manufacture the mechanical components. He then cut, machined, welded, and painted the steel to produce the final product. He also served as the purchasing liaison for the group, ordering the parts we needed as we needed them.

### **Kade Gigliotti (Electrical and Computer Engineer):**

Kade was one of the two electrical and computer engineers for the project. Kade designed a majority of the electrical components of the system, including the connections between the Raspberry Pi, Arduino Uno, shield, motor drivers, and wiring to the motors. Kade also worked closely with Sarah Luciano to develop the software and firmware for the system. Kade selected and designed most of the parts for the electrical system.

### **Timmy Hefferan (Optical Science and Engineer):**

Timmy was the optical engineer for the project, and therefore the optical subject matter expert. Timmy was heavily involved in the selection of the optical experiment that was translated to the online laboratory. He determined what optical components were necessary to build the experiment for our project. Timmy also provided the knowledge and expertise on setting up the optical experiment, aligning the spatial filter, and providing the background information for the project. He wrote the instructions for use and set up of the system.

### **Kyle Walker (Electrical and Computer Engineer):**

Kyle was the other of the two electrical and computer engineers for the project. Kyle worked with Kade to develop the electrical subsystem. Furthermore, Kyle took on the task of integrating the video streaming component of the system, including streaming the video stream to the website and providing video feedback with minimal delay. Kyle also worked heavily with Sarah Luciano to develop the software and firmware necessary to pass commands and information to the server and ultimately to the motors.

**Sarah Luciano (Computer Science Graduate Student):**

Sarah was a graduate student who helped us with the project and will continue working on it after Design Day. She was responsible for the user interface and the website that it would be hosted on. She made the website open to further expansion as well as functional and attractive.



**Final Report  
Robotic Laboratory for Distance Education**

Ryan Bergsma  
Kade Gigliotti  
Scott Appleby  
Kyle Walker  
Timmy Hefferan

May 3, 2016



## **Table of Contents**

<u>Topic</u>	<u>Page</u>
Introduction .....	2
Scope of document .....	2
Changes made since Critical Design Review .....	2
Problem statement and background information .....	4
Scope of project: objectives and limitations .....	5
What the product is expected to do .....	6
Description of customer .....	6
System Requirements .....	7
Summary of requirements with functional decomposition .....	7
Requirements metrics and interface specifications .....	8
Brief Summary of PDR Results .....	9
Preferred concept and analysis behind decision .....	9
Top-level Design of Final Design Concept .....	9
Subsystem / Sub-assembly and Interface Design (Hardware) .....	11
Optical .....	11
Mechanical .....	15
Electrical .....	26
Algorithm Description and Interface Document (Software) .....	31
Analysis .....	34
Development Plan and Implementation .....	35
Development plan timeline .....	35
Final product implementation .....	35
Requirements Review / Acceptance Test Results / Performance .....	42
Review of product against requirements .....	42
Results of Acceptance Test .....	42
Future Improvements .....	42
Closure .....	42
Summary .....	42
Challenges .....	42
Future Improvements .....	43
References .....	44
Appendices .....	45
Engineering Drawings .....	45
Acceptance Test Results.....	55
Budget and Suppliers .....	64
Project Management .....	64
Requirements Matrix .....	65
Source Code .....	69

## **1.0 Introduction**

### **1.1 Scope of Document**

The purpose of this document is to provide an overview as well as an in-depth composition and functional analysis of the Robotic Laboratory for Distance Education (RLDE) system and its requisite subsystems. Next, the project's goals, scope, and viability will be discussed in order to clarify the best use and purpose for such a system. The entire design process is presented: the development of requirements, summary of the preliminary design, detailed analyses and diagrams of the final system, a development plan, and final testing, verification, and validation.

The requirements for each subsystem and their interactions, as well as the system as a whole will be outlined and specified in regards to the methods used for verification and testing to ensure all requirements set forth have been met adequately. Detailed documentation on all components necessary along with materials used to create this system, one should be able to fully understand and possibly recreate the system with this document and a basic understanding of what is feasibly reproducible in future derivatives of said system.

### **1.2 Changes since Critical Design Review**

#### **1.2.1 Mechanical Changes:**

The motor mounts were designed assuming industry-quality machining and welding. Thus, the motor mounts did not allow for much adjustability as everything would be where it needed to be. Not having to adjust each part would cut down on setup time when setting up the experiment. However, the parts were made by a student and did not obtain the same level of quality as expected. As such, the system was redesigned to include much more adjustability. This requires more setup time, but ensures that the system will be able to serve its purpose. Additionally, the X and Y micrometers translated more significantly than expected. As such, the motors coupled to these micrometers, which were initially stationary, now are allowed to translate with the micrometers they are coupled to. The motors are mounted between small, adjustable rollers that allow for translation, but prevent axial rotation. The Z-axis micrometer does not translate, however, so the mounting of that motor is still fixed. However, the mounting means has changed so as to allow a little greater adjustability and resemblance to the other mounting methods. Additionally, since the motors had to translate, the lengths of each motor mount part was adjusted slightly to accommodate this change. Additionally, a box housing the Arduino was added and mounted to the structure to make the

wiring more convenient. Finally, a second camera was added, so a new, adjustable, mounting means was implemented for that camera. Also, the mount for the first camera is being changed to a tripod as this allows for greater adjustability as well as is more efficient all around economically. Each of these changes will be detailed out in the discussion of the final design.

### 1.2.2 Electrical Changes:

With an increased focus on accessibility, and reproducibility the decision was made to transition from using an existing Windows computer, in the Optics laboratory, to using a Raspberry Pi 2 as the host computer. This allows for a large reduction in the initial setup cost of the electrical hardware, if the user requires the purchase of a host computer. The Raspberry Pi has also been implemented as a LAMP (Linux, Apache, MySQL, and PHP) web server to allow the laboratory website to be hosted locally.

### 1.2.3 Optical Changes:

Given the ability to position the viewing screen at a variable distance after the spatial filter, the collimation lens was no longer needed in our system. The viewing screen can be positioned such that the expanding output beam is at a size most suitable for the webcam to capture. So, the collimation lens is now listed as an optional part in the Optical Bill of Materials. It was not removed because it still may be of use in order to replicate the experiment given the constraints of a short optical table. If this was the case, the collimation lens could be used to keep the output beam at a fixed size and project it to the output screen located on a different table.

The removal of a collimation lens from the system left an extra optical post. This along with its rail carrier was then used as a second mounting point for the spatial filter platform. Prior to this, there had been trouble aligning the spatial filter platform in yaw. It had not only been difficult to align using back reflection from the multiple elements in the microscope objective, but was very sensitive to any accidental bumping. This second post constrained the yaw, forcing the spatial filter to be aligned parallel to the rail.

Lastly, one of the V-mounts for the laser was removed from the system (along with its 8" optical post). The benefit of using two V-mounts was for constraint from yaw, similar to the discussion of the spatial filter, as well as fine adjustment in pitch. However, it was found that there was slight misalignment in yaw using the pair of V-mounts. By using only one V-mount, we had rough

control over this yaw using the mount's quick-release knob. Though rough, this control proved to be enough to align the system without much hassle.

#### 1.2.4 Software Changes:

Originally the web server was going to interface directly with the Arduino to send serial communicated messages, which contained commands to control the motors. In the updated design, a text file is used as a command buffer and a python script was written to interpret the command and then serially connect to the Arduino to communicate.

The video streaming service chosen prior, Open Broadcasting Software, was determined to be incompatible with the Raspberry Pi webserver so a new Linux-compatible program called 'avconv' was selected. Using 'avconv' both video streams could be sent to their respective MonaServer instances in the same way OBS did (effective fully functional replacement).

### 1.3 Problem Statement and Background Information

The project began with the following need statement from the project sponsor, who is the main stakeholder:

“The proposed project involves the construction of a remote-controlled laboratory experiment that can be accessed by online students. The project addresses a need to provide a laboratory experience for students who are taking online courses. The experiment to be automated will be selected from among those currently carried out by students in the Optical Sciences undergraduate laboratories. The experiment hardware will be augmented by remote-controlled actuators, cameras, and other sensors so that an online student will be able to make adjustments to the hardware and observe the effects. The student team will be aided by a graduate student in computer science who will be doing her Master's thesis research in the area of online education.”

This project is beginning with no pre-existing system in place. There appears to be no existing products like this one on the market already, which allow a student to remotely participate in a lab session. The current technology used for distance learning is applied to lecture-style learning environments, where a lecture is recorded and often live-streamed to a network. Students can access the recordings real time or at a later date. These systems are instrumented with video recorders as well as microphones so that there is a visual and auditory aspect to learning. However, this leaves for little interaction between the distance students and the instructor and/or classroom. There are, however, pre-existing laboratory

experiments that will be adapted for remote learning. In addition, there are systems that allow for long-distance control of hardware such as in remotely controlling telescopes or doctors performing surgery remotely. This existing technology could be incorporated and expanded upon to allow for a more interactive environment so that a remote student can obtain a laboratory experience similar to one obtained in person.

Our goal for this project is to give the distance user as much of an in-lab experience as possible. We will instrument a spatial filter experiment with motors and cameras to allow the user to remotely control the hardware. The user will interact with a software on their computer, which will communicate with a server via Internet connection to the host computer in the Optics Laboratory at the U of A. The host computer will decode the messages, and provide the proper output to the microcontroller, which will in turn give a signal to the motors to move a specific amount in a given direction. The cameras will give visual feedback to the user, which allows them to see the experiment.

#### **1.4 Scope of Project: Objectives and Limitations**

The primary objective of the RLDE is to provide a laboratory experience for online students. From a broader perspective, the objective of this project is to provide a prototype system that serves as a proof of concept that physical laboratories can be translated to online learning. The reach of this project is yet to be determined, as it could be utilized for several labs for distance students at the University of Arizona, or as a tool to be used by handicapped students who do not have the fine motor skills needed to perform certain labs. The project also has secondary objectives, such as further developing the College of Optical Sciences' laboratory resources, providing the experience of aligning a spatial filter to students in Puerto Rico, and testing the viability of hardware manipulation via software as a viable online experimentation option.

The scope of the project extends to the physical hardware that is developed specifically for the spatial filter experiment as well as the accompanying software developed by Sarah Luciano. The software will undergo subsequent revisions in the future to scale the project as desired. As built, the system will allow for a single user to log on and perform the experiment. The RLDE was designed to perform a spatial filter experiment only. If other experiments are to be added in the future, the hardware and software will need to be further developed. The system has potential to be fully integrated into a curriculum; however, this project will only consider the single use of performing the spatial filter experiment.

## **1.5 Final Product**

The final product is a mixture of both hardware and software to create an end-to-end user experience. The hardware is comprised of optical hardware, custom-machined motor mounting brackets, and electrical hardware. The software includes a web page and the local software that decodes the messages and translates those messages to direct electrical commands.

The optical hardware purchased is all necessary hardware to perform a spatial filter experiment, including laser mounts, optical rail, objectives, spatial filter platform, and a pinhole. These components were augmented with mechanical hardware, which included three stepper motors that coupled to the micrometers, a custom fabricated tube steel motor mount assembly, and posts for the camera and screen mount. The electrical hardware connected the three stepper motors to the Raspberry Pi, which acted as a server for the website. The electrical components used were an Arduino, shield, motor drivers, and wires to connect all of the components.

The software was developed to complement the hardware. The website code and final user interface was developed by Sarah Luciano as a graduate project, and will therefore only be discussed vaguely in this report. The rest of the requisite functional code was developed to work with the website's output and translate commands to the actual movement of hardware in the lab. This software will be covered in more detail in Section 6, Algorithm Description and Interface Document.

## **1.6 Description of Customer**

The customer for the RLDE is Dr. Alan Kost, a professor in the College of Optical Sciences. Dr. Kost's primary interest is to translate the learning that takes place in labs to an online environment. Additionally, Dr. Kost is working in conjunction with Sarah Luciano to provide an extended learning environment for the University of Puerto Rico. While Dr. Kost is the primary stakeholder, the College of Optical sciences also benefits from the project, since it is expanding the reach of the undergraduate laboratories, run by Dr. Mike Nofziger. Finally, the University Information Technology Services (UITs) and the UA Online departments could potentially benefit from the RLDE, since the project will be implemented online through the University of Arizona.

## 2.0 System Requirements

### 2.1 Summary of Requirements

Below is a summary of the system requirements. The functional requirements detail what functions the system shall perform. The functional requirements also drive the non-functional requirements, which are functionally decomposed into subsections corresponding to different aspects of the project. All of the requirements listed below were approved by the project sponsor and drove the development of the RLDE system.

#### **Functional Requirements:**

- The system shall allow the user to remotely connect to the device online, through a user interface.
- The system shall allow the user to control the lab hardware remotely.
- The system shall provide real-time visual feedback of the experiment, as it is being performed.
- The system shall be disassemblable.
- The experiment instructions shall be posted online.
- The system shall operate in real time.
- Sarah Luciano shall assist in providing the computer programming for the online interface.
- The system shall have safety precautionary measures in place to prevent physical harm to humans in the lab or the equipment.
- The system shall have a controller that operates all of the lab hardware and equipment.
- The system shall give feedback messages to provide the user with a more complete understanding of the current system configuration.
- The system shall be operable 24 hours per day.

#### **Non-Functional Requirements (Functional Decomposition):**

Mechanical:

1. Mechanical setup shall fit within an area of 3 feet by 6 feet.
2. System shall provide at least 2 degrees of freedom for the operator.
3. System shall have end stops that will prevent any components from moving too far as well as prevent any permanent deformation to the other parts.
4. Laser shall stay fixed within  $\pm 0.5$  degree.
5. The system shall be mounted using a standard optical rail.
6. The system shall be able to mount to an optics table with 1/4-20 holes spaced on a 1-inch grid.

Software:

1. The online interface shall have several windows showing the different camera views.
2. The software shall be accessible from Internet Explorer, Safari, or Google Chrome.
3. The software shall run as an Internet application.
4. The system shall include help indicators accessible during use, which detail the operation of the interface at a level suitable for first-time users.

Hardware:

1. The system electronics shall ultimately be provided power from a 120V US power outlet.
2. The lab shall be instrumented with video cameras to provide the visual feedback.
3. Motors shall be controlled using micro controllers.
4. The system shall be lit by an independent light source.
5. The system shall be able to reset itself within 5 minutes after a user has finished an experiment, in order to prepare for the subsequent user.

Performance:

1. The system shall cost less than \$3,500.
2. The time delay from remote input to visual feedback shall be less than 2000 milliseconds on average.
3. The video streaming shall be broadcast with a quality of at least 240p.
4. The system shall take less than 2 hours to set up from an 'out of the box' state.
5. The system shall not require that the user need to purchase any hardware or software to operate the system.

## **2.2 Requirements Metrics and Interface Specifications**

Each requirement has a defined metric, which is a measurable quality or quantity that can be used to determine whether or not the final system meets the requirement. All of the requirements along with their metrics and the measured results of the final system can be seen in Appendix 12.5, the Requirements Matrix. The requirements are broken down into functional areas, and they incorporate the relevant interfaces for each subsystem. All requirements were both validated and verified via testing, analysis, and inspection.

## **3.0 Brief Summary of PDR Results**

### **3.1 Preferred Concept and Supporting Analysis**

During our PDR, we explored three different design options. In each of our design options, we did a trade-off analysis, including a Pugh Matrix with sensitivity analysis. The preferred concept from PDR was changed and developed for CDR. The initial chosen concept from PDR was to purchase motors and a microcontroller to motorize the existing equipment in the laboratory. The benefit of this design concept was that it eliminated the costs of purchasing the optical hardware. As this design progressed, the scheduling of lab hardware became an issue, so all of the optical hardware was purchased for dedicated project use. Another outcome of the PDR was the selection of specific hardware, which was later purchased. The hardware purchased included a Newport M-900 Spatial Filter System, Thorlabs 24" dovetail optical rail, NEMA 8 stepper motors, Logitech C920 webcam, Arduino Uno, Raspberry Pi (added later in the design process), and a web client software. All of the above components were chosen after much detailed analysis, which can be seen in Sections 5 and 6.

## **4.0 Top-level Design of Final Design Concept**

The final design concept is an integration of hardware with software. Figure 1 shows the physical block diagram of the system. The modem/router communicates with the Raspberry Pi via Ethernet connection. The Raspberry Pi acts as the webserver, host computer, and the video stream server. The Raspberry Pi interacts with the two webcams and the Arduino Uno via USB serial connection. Finally, the Arduino Uno sends the commands to the three motors, which are mounted to the physical motor mounts (mechanical system). Not explicitly shown in this figure are the shield and motor drivers that are attached to the Arduino Uno. Figure 4.1 illustrates how the commands are communicated from the webserver down to each individual motor. The mechanical system, which couples the motors to the optical equipment and the electronics, can be seen in Section 5.2.

Figure 2 shows the software block diagram for the system. The website interface is where the final user interface exists. There is a Monaserver and Avconv on the Raspberry Pi for both video streams. All of the files needed to process and stream the video feeds are saved on the Raspberry Pi. The Raspberry Pi also acts as the webserver, and therefore receives PHP commands over TCP/IP. Then, the Raspberry Pi writes commands to a text file, which are sent via a Python script to the Arduino Uno. Since this is an embedded system, the hardware and software work together to achieve the overall purpose.

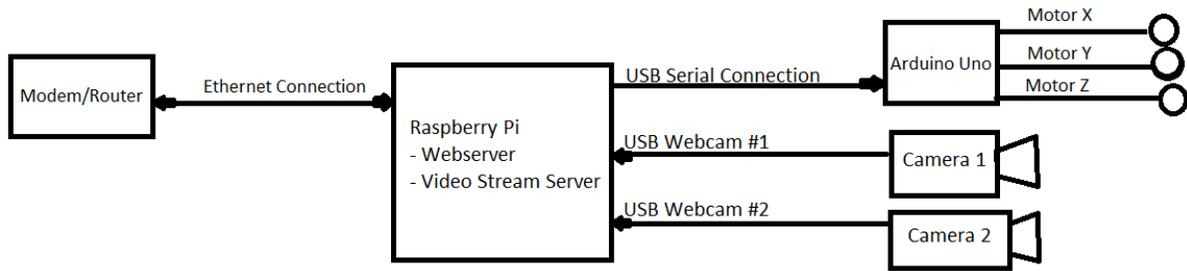


Figure 1: Physical Block Diagram

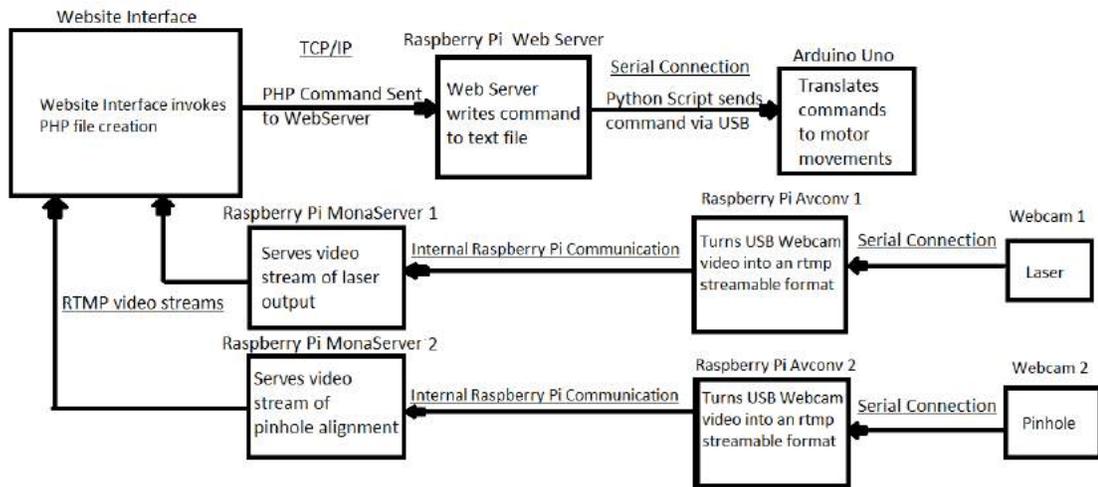


Figure 2: Software Block Diagram

## 5.0 Subsystem / Sub-assembly and Interface Design (Hardware)

### 5.1 Optical Subsystem Analysis

#### 5.1.1 Design Summary

For the optical subsystem, the experiment chosen to be performed was the alignment of a spatial filter. This experiment was chosen because its limited degrees of freedom and reliance on visual output translates well to online learning. The three degrees of freedom in this experiment are linear movement of the pinhole in the X, Y, and Z directions via micrometers on the spatial filter platform.

#### 5.1.2 Subsystem Requirements Decomposition

Only several of the requirements are applicable to the optical subsystem itself. Of these few requirements, several of them pertain more to the mechanical design but will be used in order to mount the optical subsystem. Two of these requirements are “The laser shall stay fixed within  $\pm 0.5$  degree” and “The system shall be mounted using a standard optical rail.” For the optical subsystem, these two requirements go hand in hand. By designing the elements of the system to be mounted to the optical rail using their corresponding mounts, little to no drift should be present in long-term applications or during operation. The next requirement is “The system shall be able to mount to an optics table with 1/4-20 holes spaced on a 1-inch grid.” The use of a standard optical rail will fulfill this requirement if the system is mounted directly to the table. However, the current design has the optical rail mounted to custom hardware, so the fulfillment of this requirement will be discussed further in the mechanical subsystem design. The next requirement is given by “The system shall take less than 2 hours to set up from an ‘out of the box’ state.” In the design of the optical subsystem, this was also covered by the use of standard optical hardware for mounting and operation. This hardware will provide options for quick disassembly due to the nature of optical mounts. The lens mounts will be able to be unscrewed from the posts, the laser able to be removed from its mount with little effort, and the spatial filter platform itself a single component to be uncoupled and removed from the system. The final requirement that needed to be considered in the optical subsystem design was “The system shall cost less than \$3,500.” After creating a bill of materials for all the optical hardware that will need to be purchased, with the exception of the 633 nm HeNe laser that will be provided by the University of Arizona College of Optical Sciences, the total with the cost of the other subsystems was found to be well below this limit. This is shown in Section 12.3.

### 5.1.3 Drawings, Schematics, and Relevant Visuals

Below is a solidworks assembly of the experimental set-up. There is a Helium-Neon Laser at one end, which then goes through a series of lenses that serve as the beam expander. These lenses are arranged as a reverse Galilean telescope that expands the beam to better fill the aperture of the next element - the microscope objective. The objective then focuses the beam through a pinhole, which spatially filters the beam. After the beam travels through the pinhole, it can be optionally collimated using a collimating lens. Finally, the image is captured on a screen. The cameras will be positioned to capture both the spatially filtered beam as well as the pinhole, which is a primary point of concern, since the pinhole is the mechanism by which the experiment is completed. See Section 5.1.4 for a detailed description of all components in the optical subsystem.

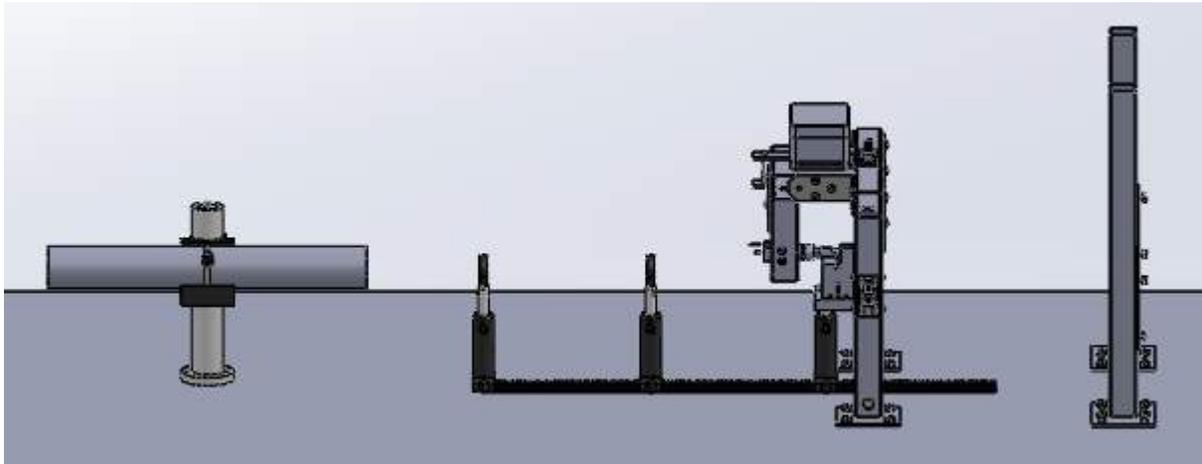


Figure 3: Layout of Optical Subsystem

### 5.1.4 Analyses, Specification Sheets, and Modeling Results

Starting from the source, the REO 633nm HeNe laser available via the University of Arizona College of Optical Sciences was chosen. This will provide a beam of sufficient power such that the camera can pick up the resulting image on the screen. As an alternative, consideration was given to using a laser diode in place of this full HeNe source. However, given the availability of the laser and its corresponding mounting hardware, it was decided to stay with the HeNe source. A set of v-mounts are used to hold the laser in place. These are held to a pair of 8" mounting posts and tightened via adjustment screws. Each of these posts is then attached to a corresponding mounting post base with  $\frac{1}{4}$  -20 screws. These mounting bases are

compatible with optical tables such as the Melles Griot table that will be used for this experiment. This table has a standard 1-inch grid of ¼-20 holes, so the mounting bases will allow the laser to be roughly aligned along the grid. For finer alignment, the v-mount allows for movement in tip and tilt via their adjustment screws which provide a total range of ±10° of pitch control.

The beam expander element of the setup was designed as a reverse Galilean telescope in order to expand the input beam to better fill the microscope objective. The objective lens will be discussed further with the spatial platform, but one of its key specifications was for a maximum input beam diameter no greater than 5.5mm. Accordingly, a magnification of 5x was chosen for the beam expander in order to expand the incoming beam of approximately 1mm diameter to a size of 5mm. This magnification will allow for some padding if the input beam is larger than expected while still filling the majority of the objective lens's aperture. The lenses to build the beam expander were chosen to be a -50mm plano-concave lens and a 250mm plano-convex lens. Both of these lenses have a 1-inch diameter, are made of BK7, and are uncoated in order to be cost-effective. Checking that these satisfy the magnification:

$$M = -\frac{f_o}{f_e} = -\frac{(250mm)}{(-50mm)} = 5$$

These focal lengths are very common in stock lenses, which will allow for greater availability from the available lab hardware. The shape factors were chosen because of their application in the telescope. The first element acts in infinite conjugate imaging, so the ideal shape factor would be plano-concave with the concave surface towards infinity. The same applies for the plano-convex element but in reverse, where it is imaging from one focal length away to infinity. The ideal shape factor for this use is a plano-convex with the convex side towards infinity. In order to function as a reverse Galilean telescope, the spacing of these elements will be equal to 200mm. These optics are mounted in fixed mounts because it was the most cost-effective option in order to attach optical posts with the 8-32 set screw at the top of the post. These optical posts will be held by optical post holders that allow for a total height adjustment range of nearly 3 inches. When mounted, the optics will be held at a certain height by a spring-loaded, hex-locking thumbscrew. Perfect orientation with the center of the lens surface perpendicular to the optical axis will not be required for this experiment. This selection of mounting

hardware will provide enough freedom to adjust each element to the proper height of the beam and rotate the element to be roughly aligned in rotation.

The spatial filter platform chosen was the Newport M-900. This was the model used in the hands-on lab experiment early in the semester and was found to provide all functionality necessary for our coupling purposes. Particular advantages of this model are the absence of crosstalk between micrometers due to the magnetic pinhole design. As one micrometer is moved, the pinhole slides along magnets such that the other micrometers do not move at all. This spatial filter platform comes with an objective lens as well as the pinhole. Both of these elements are interchangeable with other compatible objectives and pinholes if necessary. This platform comes packaged with a 25- $\mu\text{m}$  pinhole and a 10X microscope objective. Both of these elements can be swapped with other compatible pinholes and microscope objectives from Newport, but these additional parts will not be necessary for this experiment. The spatial filter platform is mounted directly to the same optical posts as the lenses through the use of M6 adaptor rings. These help to couple the M6 counterbored holes on the M-900 to the 8-32 set screws present on the optical posts.

As discussed previously, the collimation lens is now an optional component in the system, however it still may be used. The collimation lens was chosen as a standard 100mm uncoated BK7 plano-convex lens for similar reasons as the lenses for the beam expander- availability, cost, and the correct shape factor for an infinite conjugate application. This element was chosen to have a diameter of 2 inches because of its application as a collimating lens. The output from this lens will remain the same size at any distance, so the chosen diameter will allow the screen to be better filled. In order to collimate the light leaving the spatial filter platform, the lens will be placed at one focal length (100mm) behind the pinhole. The mounting hardware for this lens is similar to that of the lenses in the beam expander with the exception of a larger diameter fixed mount. The screen material that the resulting beam will display on came in a sampling pack with several different options. The material that worked in order to best reproduce the image through the camera with minimum saturation was chosen, being the Pro-White material.

The post holders for all of the above components will be mounted to rail carriers with  $\frac{1}{4}$ -20 screws. These attach from the bottom of the rail carriers through the bottom of the post holders. Each of these will then be attached to a dovetail optical rail, which is subsequently attached to the 1-inch grid of the

optical table with ¼-20 screws. The use of this rail with rail carriers will provide ease of motion that is constrained along the optical axis. The element positioning can be done using the scale engraved on the rail. This rail will also allow the whole system to be mounted along the same set of holes as the laser for rough alignment.

Manufacturer	Item Description	Part Number	Price per unit	Quantity	Cost
Newport	M-900 Spatial Filter System	M-900	899.95	1	899.95
Thorlabs	M6 Adaptor Ring	SD1	7.56	1	7.56
Thorlabs	Dovetail Optical Rail - 24"	RLA2400	144.00	1	144.00
Thorlabs	1" Lens Mount	LMR1	15.23	2	30.46
Thorlabs	2" Lens Mount	LMR2	24.68	1	24.68
Thorlabs	3" Optical Post	TR3	5.42	4	21.68
Thorlabs	3" Post Holder	PH3	8.27	4	33.08
Thorlabs	Rail Carrier	RC1	24.90	4	99.60
Thorlabs	2" 100mm EFL Lens	LA1050	28.50	(Optional) 1	28.50
Thorlabs	1" -50mm EFL Lens	LC1715	19.30	1	19.30
Thorlabs	1" 250mm EFL Lens	LA1461	19.40	1	19.40
REO	633mm HeNe Laser	Lab*	0.00	1	0.00
Thorlabs	Laser Post V-Mount	C1502	96.60	2	193.20
Thorlabs	8" Laser Post	P8	54.00	2	108.00
Thorlabs	Mounting Post Base	PB1	23.70	2	47.40
Carl's Place	Screen Sample Material	N/A	1.00	1	1.00
Thorlabs	25 x 1/2" 1/4-20 Cap Screw	SH25S050	7.75	1	7.75
N/A	Misc	N/A	40.00	1	40.00
<b>Subtotal</b>					<b>1,725.56</b>

Figure 4: Optical Bill of Materials. All relevant mechanical drawings for optical hardware are listed in Appendix 12.1.

## 5.2 Mechanical Subsystem Analysis

### 5.2.1 Design Summary

The mechanical subsystem was designed to provide a means to motorize each of the three axes of a pre-existing spatial filter assembly provided. Each axis controls a direction in which the pinhole can move. The goal is to align the pinhole with the focal point of a laser focused through an objective lens. The pinhole is determined to be aligned based on evaluating the visual output of the pinhole as shown on some type of screen. Thus, for remote control, in addition to each axis being motorized, there must be a screen to project the output on, a camera to record and transmit this visual data, and a controlled light source so that everything is visible. Thus, the mechanical subsystem must provide means for each of these. The laser will be mounted using optical hardware. The spatial filter and various lenses will be mounted to the optical rail, which will be mounted to the table. The spatial filter will be adjusted to an appropriate height for the motor mount assembly. The motor mount assembly was designed not to interfere with the rail or any components on it, for example, blocking the laser. It was also designed to be removable so as not to make the spatial filter a permanent component. Furthermore, it was designed to be very adjustable to

ensure that it would work. This adjustability also allows it to fold up fairly nicely for storage. The camera mount, independent lighting, and screen mount are designed to be adjustable to accommodate any position the rail would reasonably be in. All components will mount to a standard optics table in one form or another.

### 5.2.2 Subsystem Requirements Decomposition

The mechanical subsystem was designed to meet all of the mechanical requirements. Requirement “Mechanical setup shall fit within an area of 3 feet by 6 feet” will be met because the full assembly was modeled in Solidworks and met this size requirement. Requirement “System shall provide at least 2 degrees of freedom for the operator” is met because the spatial filter has 3 axes that provides 3 degrees of freedom for the operator. Requirement “System shall have end stops that prevent any components from moving too far as well as prevent any permanent deformation to the other parts” was initially meant for any linear translational components that we thought we might have. We did not end up having the linear translational components, thus making this requirement less applicable. However, there are physical stops built into the micrometers, which will be the only real moving components, so this requirement is also met. However, the physical stops are just a backup as the software written will stop the motors before the stops are reached. Requirement “Laser shall stay fixed within  $\pm 0.5$  degree” was also met after testing. Optical hardware will be used to mount the laser dependably and sturdily. Thus, the experiment will be able to be used many times without needing to adjust the laser. Requirement “The system shall be mounted using a standard optical rail” is met as all the optical equipment is compatible with optical rail and the custom parts mount around it. Requirement “The system shall be able to mount to an optics table with  $\frac{1}{4}$ -20 holes spaced on a 1 inch grid” is also met as each assembly has holes for mounting of this type. This can be seen in Figure 10 The screen mount assembly, as seen in Figure 9 respectively, utilizes 1-inch long slots so that any mounting position can be achieved in the direction of the slot. This allows the screen mounts to be set to the right width. The motor mount assembly, however, does not utilize slots as there is no need for adjustment as the mounting holes are on the same grid as the holes in the spatial filter to which the motor mount assembly must line up with to connect. However, the adjustability of the entire system allows for fine-tuning to obtain the correct positions upon this initial placement. The camera looking at the screen is mounted using a tripod and thus can be placed in the best position for viewing. The camera looking at the pinhole for rough alignment during operation is mounted to the structure. The adjustment within

the camera and the mounting structure allows for the camera to be placed in the optimal position as can be seen in Figure 12. The gooseneck light used for lighting will be able to clip wherever is most convenient. Additionally, a box housing the Arduino will be mounted to the structure for convenience of wiring. The box will be mounted simply using zip ties for convenience.

While these are the specified requirements, there were other design choices made to optimize functionality. Upon performing the spatial filter experiment, it was discovered that the z-axis micrometer was very close to the laser beam input; our fingers would often block the beam while adjusting the z-axis. In addition, the micrometers were very sensitive to non-axial forces; if the micrometer was pushed from the side, the pinhole would shift dramatically. In order to minimize non-axial forces as well as simplify coupling, the motors used to turn the micrometers were directly coupled so that they are axially aligned. Thus, in order to not block the laser beam input, the motors had to be fairly small. The NEMA 8 stepper motor was chosen to drive the z-axis micrometer; it is the smallest readily available stepper motor. In the current modeled motor mount assembly, the z-axis mount comes to the edge of the objective lens, but does not block any of it as can be seen in Figure 13. The NEMA 8 stepper motor was used for the other two axes simply for homogeneity; however, they could be changed to different motors with appropriate changes if it is deemed necessary. The micrometers require very little torque to turn, so the torque output of the NEMA 8 stepper motor is sufficient to turn each of the micrometers. Since the motors are directly coupled to the micrometers, the motors would have to have a high enough resolution to take tiny enough steps so that the spatial filter could be properly aligned. According to the calculations presented in 5.2.4, the stepper motors do have a sufficiently small step size. The motor mount assembly was designed to be very rigid so as to minimize any vibrations that the stepper motors would produce which would result in non-axial forces on the micrometers. However, based on observing the operation of the NEMA 8 motors, these vibrations should be very minimal and sufficiently damped by the mounting structure. In addition, mounting the optical rail directly to the table increases stability as well.

The spatial filter was designed to be adjusted by hand, therefore there were several difficulties that presented themselves when attempting to motorize it. Once such challenge was the axial translation of the x and y micrometers as they turned. While a hand can easily adjust to their new position, it is more difficult for a mounted motor to do so. There were a few design options, but

what seemed the best would be to have the motors translate with the micrometers. As such, the motors would have to translate, but not rotate, especially since the needed step sizes were so small. To accommodate the translational motion, nylon rollers were used. The nylon itself acts like a slick surface upon which the stepper motor can slide. Additionally, they are also rollers to ensure that the motor will be able to translate freely. However, now the rotation must be restricted. Thus, four rollers were used, one for each corner of the stepper motor, and were mounted in slots so as to be adjustable so that there would be no room for the stepper motor to rotate. While the stepper motor is not free to rotate, it is free to pivot about the rollers, which ended up being very beneficial; it further accommodates any misalignment of the motor axis with the micrometer. This setup can be seen in Figure 7. The z micrometer, on the other hand, did not translate. The same setup as for the other two motors would have worked for this motor except that one of the external screw heads would have blocked the laser's path. Thus, the motor was mounted and held in place with four screws that do not block the laser's path. These screws are a simple means of mounting and allow for adjustability. This setup can be seen in Figure 8.

Couplers are used to connect the stepper motors to the micrometers on the spatial filter assembly. The chosen couplers are appropriately sized to couple the motors to the micrometers. However, on the micrometer side, the couplers require drilling of the hole to have it easily slide on and off the micrometer. For the x and y micrometers, a size T drill was used and for the z micrometer, a size 31/64" drill was used to drill out the hole. The ease of attachment and disassembly is desired as the system is designed to be removable. The couplers attach to the motors and micrometers through the use of setscrews provided with the coupler; this allows for non-permanent attachment. Since the couplers are made like springs, the motor can also still turn the micrometer even if they are not perfectly axially aligned. However, the structure was designed to be highly adjustable so that the motors will be able to adjust so that this should not be much of a problem. One of the features of adjustability is linking plates between different members of the structure as can be seen in Figure 11. Each plate has a large hole drilled in it through which the screw passes. Then the screw is loosened, this allows the plate a wide range of movement for fine positioning. Once the desired location is obtained, the screw can then be tightened to maintain that position. In addition to fine positioning, this allows the structure to "fold up" and/or be disassembled for storage.

Slots are provided in the screen mount assembly for added adjustability. The screen mount assembly utilizes a square piece of metal with a threaded hole for mounting screws into the slots. This piece will be referenced as a screw mount. When the screw is loosened, the screw mount is free to slide up and down the housing tube. However, as the screw is being tightened, the square will start to wedge in the tube, preventing its rotation, and allowing the screw to thread into it. It will hold its position once the screw is tightened enough so that the tube housing is clamped between the screw mount and the screw. This can be seen in Figure 9. Shoulder screws were used for the screen assembly so that they could be tightened by hand. The slots allow for appropriate positioning of the edges of the screen in the vertical direction. The slots at the bottom of each screen mount allow for appropriate positioning of the edges of the screen in the horizontal direction. Combined, this should produce a smooth screen free of wrinkles and the like for optimal viewing. The screen is made of projector screen material for optimal viewing as well. Eyelets will be used on the two or four corners of the projector screen (depending on the material) as to prevent the holes necessary for mounting from tearing. The shoulder screws will then fit through the eyelets so that the screen may be mounted to the screen mounts. Our team machined all parts that need to be machined, thus we do not have labor costs for these custom machined parts. See the appendix for the drawings of these custom machined parts.

### 5.2.3 Drawings, Schematics, and Relevant Visuals

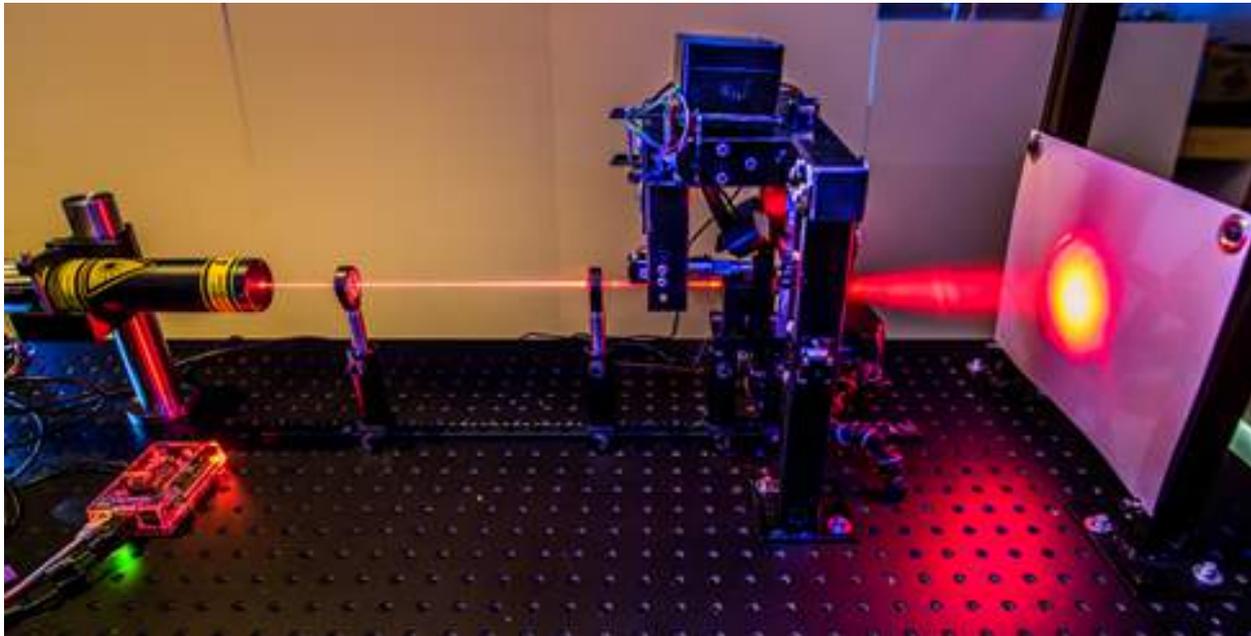
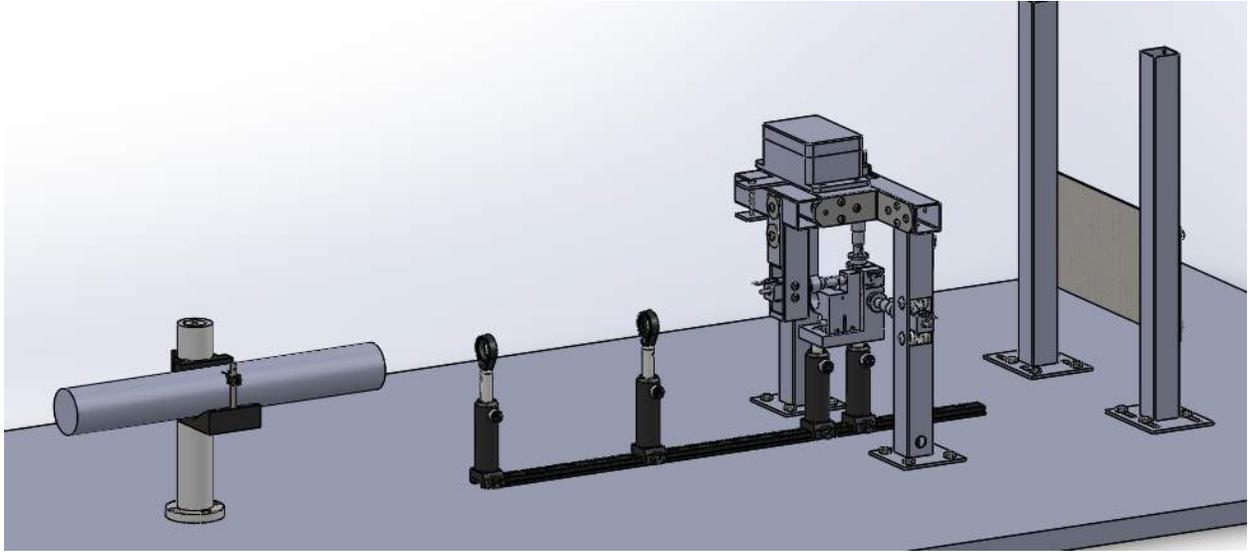


Figure 5: Overview of Assembly

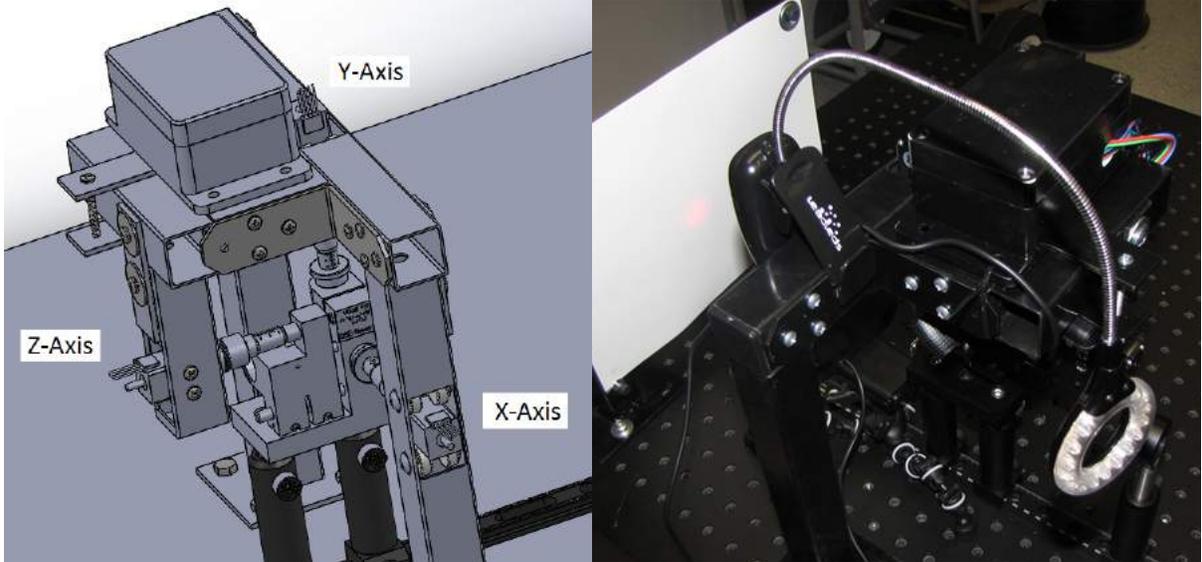


Figure 6: Motor Mount Assembly

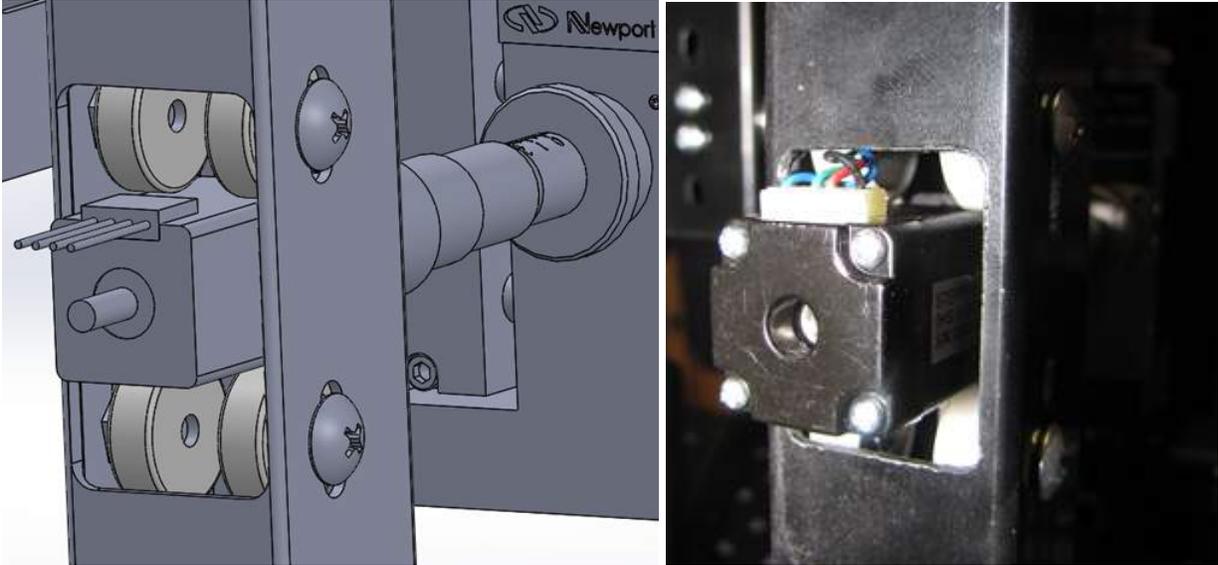


Figure 7: X Motor and Coupler (Translating)

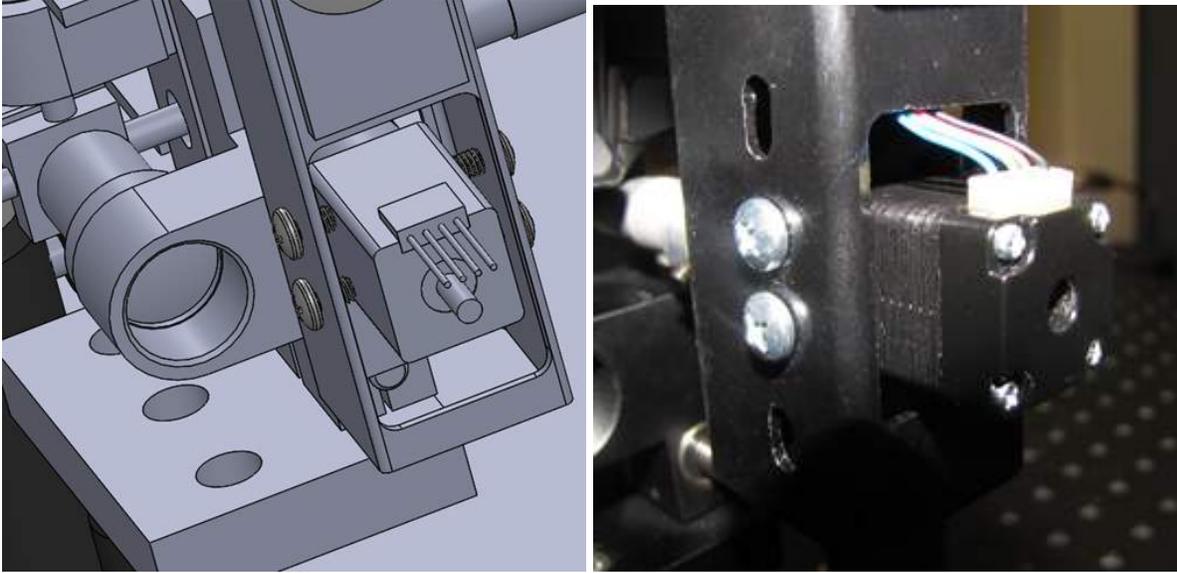


Figure 8: Z Motor and Coupler (Stationary)

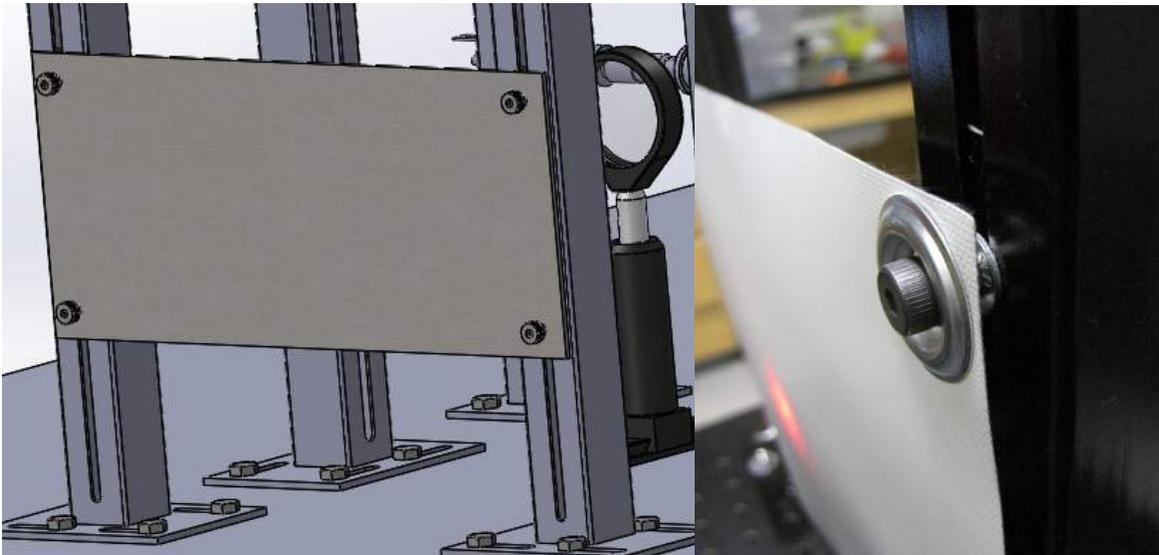


Figure 9: Screen Mount Assembly



Figure 10: Mounting Types

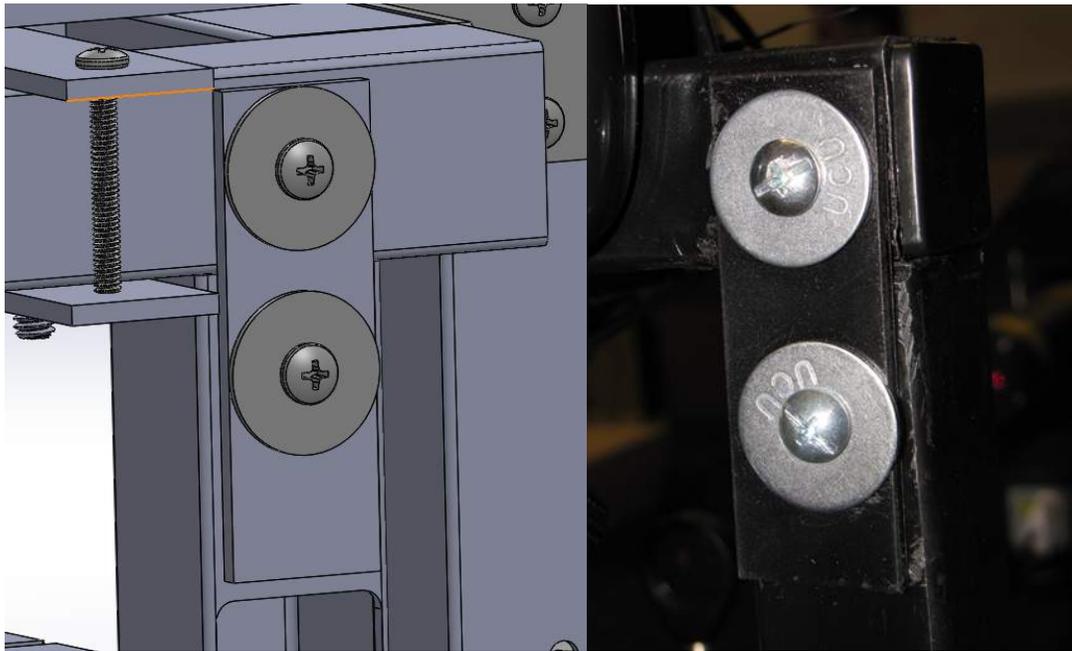


Figure 11: Adjustable Linking Plates

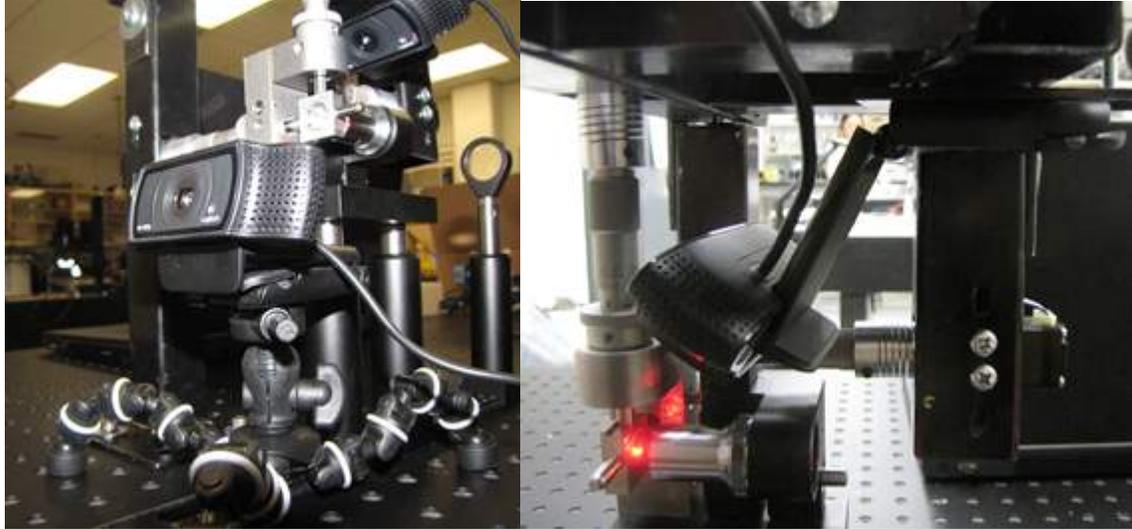


Figure 12: Mounted Camera

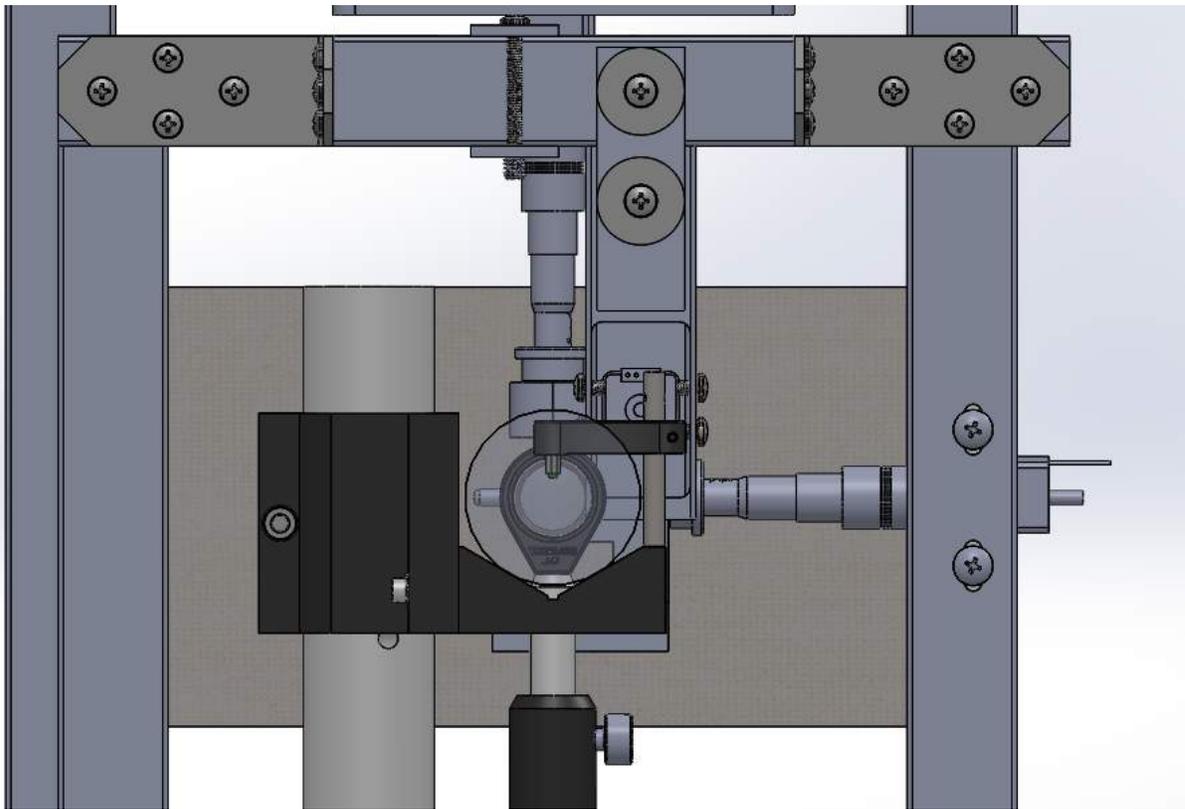


Figure 13: Unobstructed Laser Path

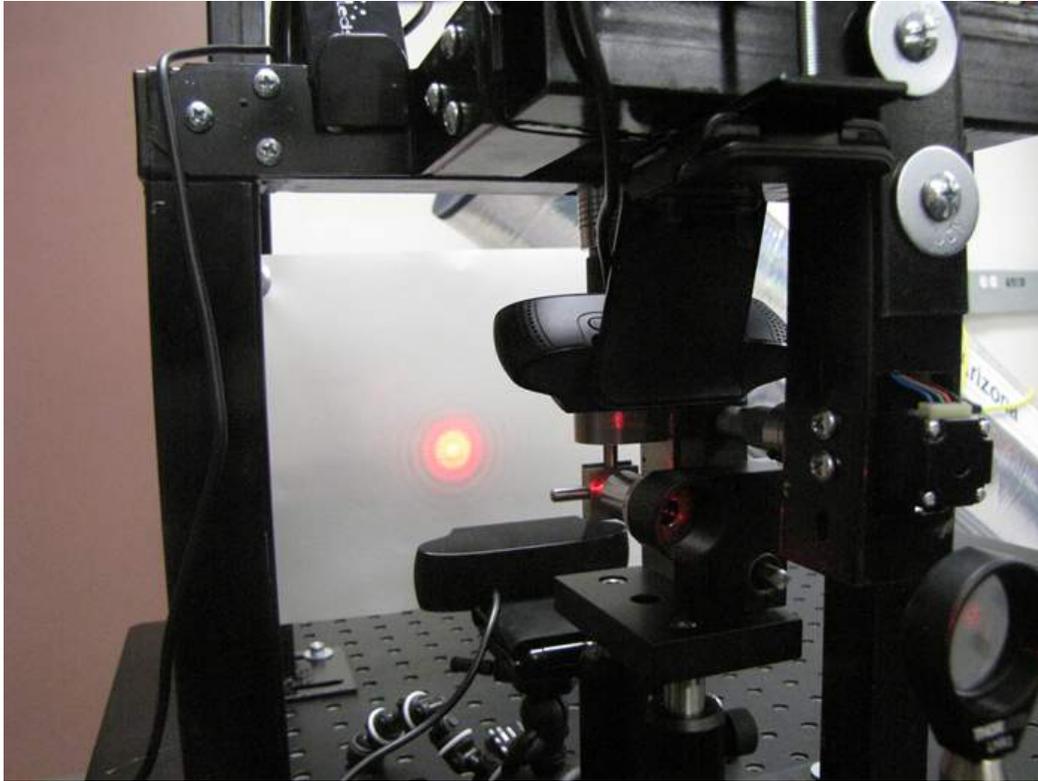


Figure 14: Alternate View of Assembly

#### 5.2.4 Analyses, Specification Sheets, and Modeling Results

The majority of the design decisions were made so that all the pieces fit together; thus most of the concerns were merely geometrical. Thus, by modeling each assembly in Solidworks, this helped ensure that everything will fit together properly. Additionally, adjustability was built in to further ensure everything worked properly. The only real analysis that was worth performing was to see if the stepper motors would produce a small enough step size for the spatial filter if the motors were directly coupled to the micrometers. A typical stepper motor has a 1.8-degree resolution, which means it takes 200 steps for 1 revolution. The given micrometers move the pinhole at a rate of 0.5mm per revolution. Thus, that means, through direct coupling, one step of the stepper motor results in 2.5  $\mu\text{m}$  of movement of the pinhole. The diameter of the pinhole we should be using is 25  $\mu\text{m}$ , so ten times the step size. Thus, direct coupling should be sufficient. However, if it is deemed necessary to have an even smaller step size, the stepper motor can be micro-stepped to 1/32 of a step size, which would translate to 0.078  $\mu\text{m}$  per step, which would be more than enough. The torque the stepper motor is able to produce in a step reduces with microstepping, so that must be taken into consideration if microstepping is pursued. Regardless, direct coupling is a sufficient means for driving the micrometers.

### Bill of Materials: Mechanical

Manufacturer	Item Description	Part Number	Price per unit	Quantity	Cost
Santa Rita Steel	16 GA. 1.25"x1.25" Steel Tubing - 11.5"		1 quote		1 N/A
Santa Rita Steel	16 GA. 1.25"x1.25" Steel Tubing - 9.875"		2 quote		2 N/A
Santa Rita Steel	16 GA. 1.25"x1.25" Steel Tubing - 2.69"		3 quote		2 N/A
Santa Rita Steel	16 GA. 1.25"x1.25" Steel Tubing - 5.25"		4 quote		1 N/A
Santa Rita Steel	16 GA. 1.25"x1.25" Steel Tubing - 4.25"		5 quote		1 N/A
Santa Rita Steel	16 GA. 1.25"x1.25" Steel Tubing - 15.875"		6 quote		2 N/A
Santa Rita Steel	1/8" thick 3" wide Steel Strip - 3"		7 quote		2 N/A
Santa Rita Steel	1/8" thick 3" wide Steel Strip - 4"		8 quote		2 N/A
Santa Rita Steel	1/8" thick 3" wide Steel Strip - 1"		9 quote		5 N/A
Santa Rita Steel	Tubing Subtotal Estimate			100	1 100
Ebay	NEMA 8 Stepper Motor	8HS15-0604S	18	3	54
uxCell	4mmx8mm Coupler	a14010900ux0520	6.96	3	20.88
McMaster Carr	8 Hole Angle bracket	17715A670	4.14	2	8.28
McMaster Carr	1 1/2" 8-32 Phillips Head Screw	90272A203	0.0645	2	0.129
McMaster Carr	10-24 Shoulder Screw	91259A533	1.06	4	4.24
McMaster Carr	#10 washer	92141A011	0.0233	4	0.0932
McMaster Carr	1/4" 8-32 Phillips Head Screw	91735A190	0.1034	18	1.8612
McMaster Carr	1.25"x1.25" Finishing Plug	9565K53	0.185	6	1.11
McMaster Carr	1/4" 10-24 Phillips Head Screw	90272A238	0.0362	6	0.2172
McMaster Carr	Oversized #10 Washer	91090A120	0.048	6	0.288
McMaster Carr	Square Finishing Plug 1.25"	9565K53	0.1852	2	0.3704
Krylon	Black Spray Paint for Metals and Plastics		5.99	1	5.99
Dritz	Extra-Large Eyelet Kit	660-65	5.99	1	5.99
Leadleds	Gooseneck lamp	LLD18B-BC-1	13.99	1	13.99

Figure 15: Mechanical Bill of Materials

## 5.3 Electrical Subsystem Analysis

### 5.3.1 Design Summary

The electrical subsystem was designed as an intermediate system to facilitate the necessary connection between the software and hardware. The main role is to receive electronic command signals from the software, process them, and in turn send control signals to each stepper motor used to control alignment of the spatial filter in a specific axis.

During operation the microcontroller will receive predefined input control signals via serial communication from the Raspberry Pi and process them accordingly. These control signals will contain the axis of the stepper motor, direction of movement, and movement amount. After processing the input controls the microcontroller will set values for the digital output pins that perform the instruction specified by the control signal. The firmware was developed in favor of reliability over speed, as a small change in code execution time would be insignificant in comparison to the video stream delay. The firmware source code can be found in Appendix 12.

### 5.3.2 Subsystem Requirements Decomposition

The design for the electrical subsystem was strongly focused on meeting or exceeding the system requirements. In particular the core of this system was based on the requirement "The motors shall be controlled using

microcontrollers.” This requirement lead to the choice of using an Arduino Uno to control the system hardware. Due to the small amount of current that the Arduino Uno is capable of delivering, the use of stepper motor drivers powered independently of the Arduino Uno would be required to provide the stepper motors with the current necessary for operation. The Pololu DRV8825 Stepper Motor Driver Carrier was chosen to perform this task as it is capable of handling current up to 2.2 A per phase, which is more than sufficient for the NEMA 8 stepper motors which can handle 0.6 A per phase.

”The system electronics shall ultimately be provided power from a 120V US power outlet” is conformed to by all of the electrical components. The Arduino Uno meets this as it can be powered by 5 V over a USB cable, or using an external AC-DC power supply with a recommended voltage range 7-12 V as stated in the technical specifications. The Pololu DRV8825 Stepper Motor Driver Carrier also meets this as they operate from 8.2 – 45 V.

Keeping to the overall modular design of the entire system and to help “The system shall take less than 2 hours to set up from an ‘out of the box’ state.” The electrical subsystem incorporates solely off-the-shelf components, which also greatly increases the reproducibility, and flexibility of the system. The Protoneer Arduino CNC Shield was chosen to allow the stepper motor driver carriers to be connected directly to the shield using the headers, and then connecting the populated shield using the female headers on the Arduino Uno as shown in Figure 5.3.3.1. This setup allows for a secure and compact connection without the need for any external wiring between components. The only external connections needed are the USB cable for serial communication between the Arduino Uno and Raspberry Pi, the power supply for powering the stepper motors, and the connection to the stepper motors themselves.

The Logitech C920 webcam that was chosen to provide video feedback to the user meets and exceeds the requirement “The video streaming shall be broadcast with a quality of at least 240p.” as it is capable of recording video at a resolution of 1920 x 1020, and can be scaled down as required.

For the system to be able to meet the requirement of being able to reset itself within 5 minutes of a user finishing the experiment, it is possible to incorporate a variety of methods to help enforce this, but the most reliable would be to have the system reset itself at certain time period that would correspond to the end of a user’s scheduled lab time. The resetting of the system will execute on a firmware level once the timer reaches the predetermined threshold.

Completing the reset sequence will only take seconds for the microcontroller to move the spatial filter to a predefined reset position.

### 5.3.3 Drawings, Schematics, and Relevant Visuals

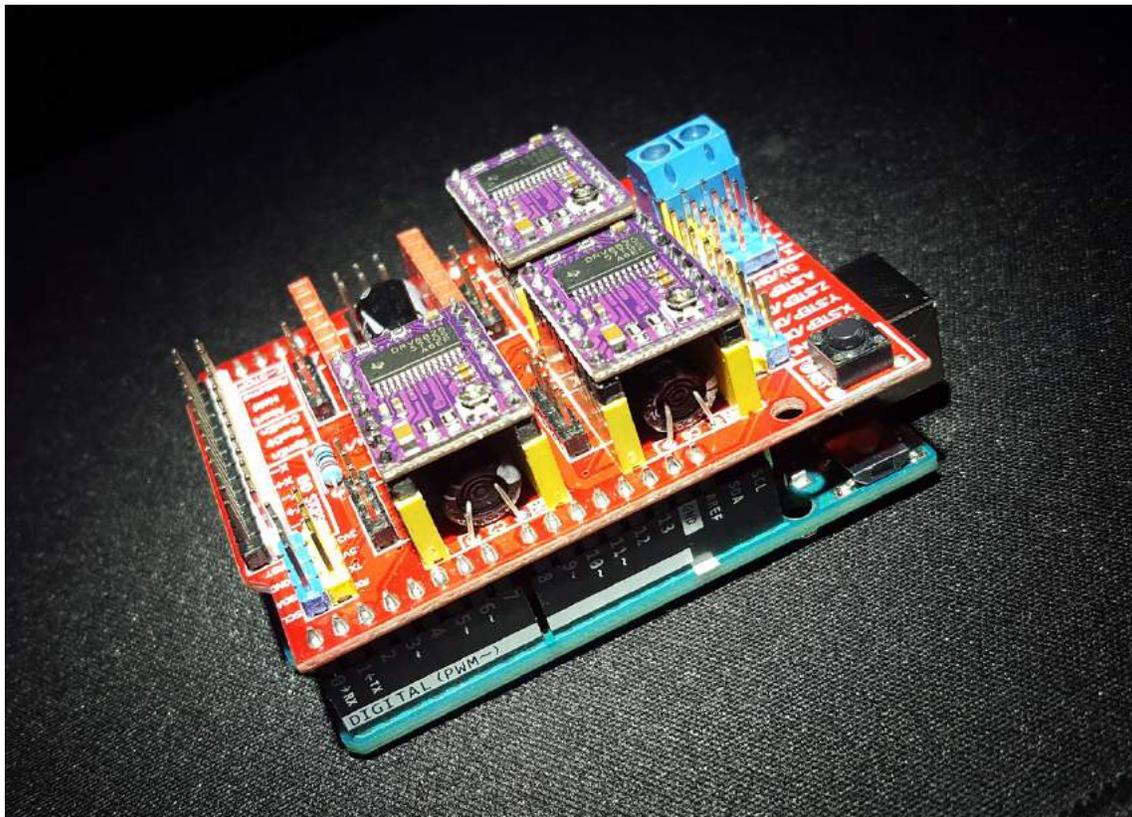


Figure 16: Arduino, Shield, and Stepper Driver Setup

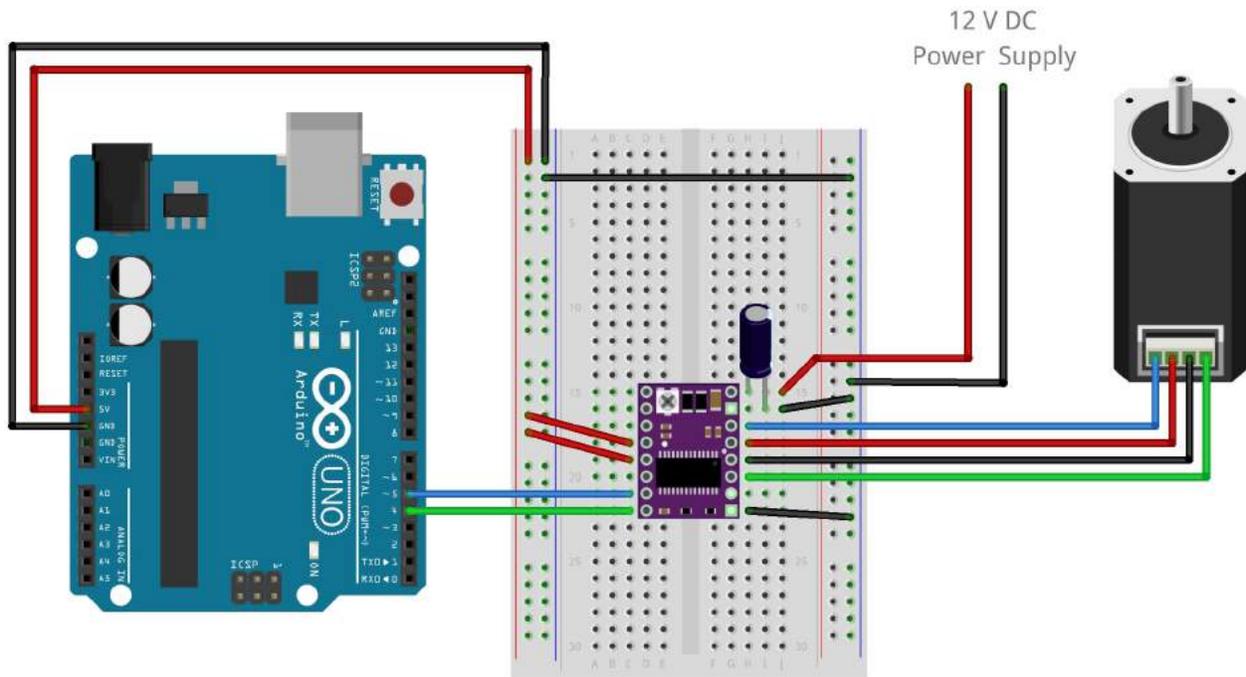


Figure 17: Stepper Motor and Driver Carrier Test Setup

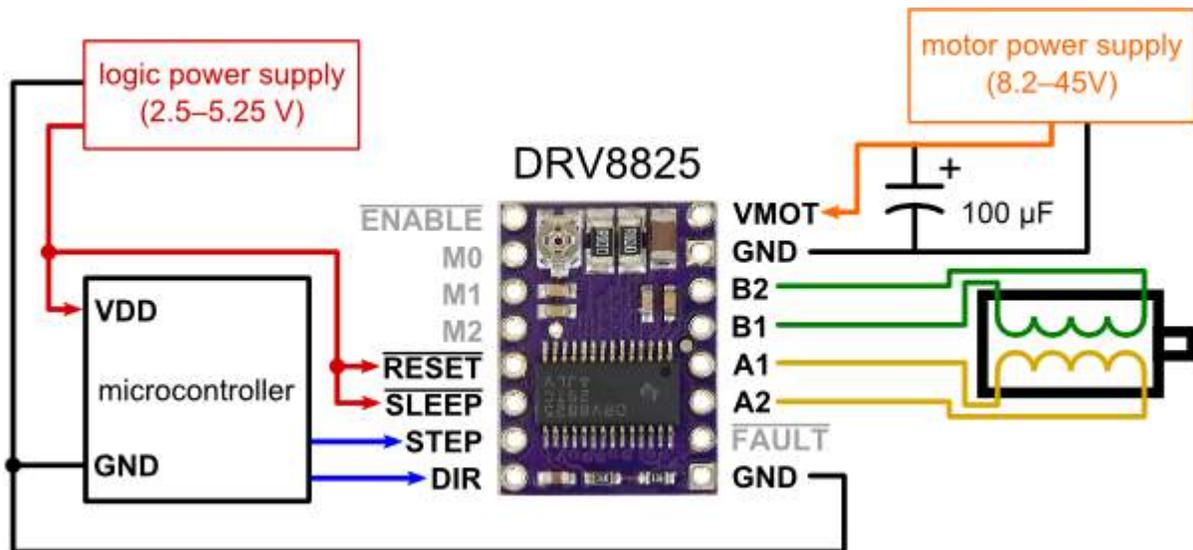


Figure 18: Minimal Wiring Diagram for Motor Driver

#### 5.3.4 Analyses, Specification Sheets, and Modeling Results

To show that it was possible for the NEMA 8 stepper motor to be controller using the Pololu DRV8825 Stepper Motor Driver Carrier and Arduino Uno, a test circuit as shown in Figure 17 was created following the wiring diagram in Figure 18. Some basic Arduino firmware, found in Appendix 12, was written to loop through a variety of different controls for the motor driver to display its capabilities. For this test pin 4, and pin 5 of the Arduino were set as outputs,

and connected to the direction pin, and step pin of the driver carrier. The reset and sleep on the driver carrier were wired directly to 5 V as these pins were not utilized and instead were held high. The enable pin was left open because the driver it enabled by default, setting the pin high would disable the driver. Mode pins M0, M1, and M2 were left open to operate the driver in full-step mode, but the driver can be operated in different microstep sizes down to 1/32 step depending on the combination of these 3 pins that are set high as shown in below in Figure 19. The blue, red, black, and green leads of the stepper motor were connected to B2, B1, A1, and A2 of the motor driver respectively. A 100 uF electrolytic capacitor was connected between VMOT and ground to prevent potential voltage spikes from damaging the driver when power is supplied. VMOT pin was then connected to an external 12 V power supply capable of delivering 1.5 A.

MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Figure 19: Motor Driver Step Size Selection Chart

The next phase of testing was to begin the implementation of controlling the stepper motor using input sent to the Arduino Uno via serial communication. In addition to the circuit used for the initial test, the universal serial bus (USB) built into the Arduino was used to establish serial communication with a computer. The firmware was designed to receive a control signal via the serial communication and perform a specific operation if the control signal matched one of the predefined commands. This Serial Monitor that is built into the Arduino software to send the command to the microcontroller and test the operation of the firmware. This test was completed successfully and leads to further development to incorporate control signals for all 3 stepper motors.

## 6.0 Algorithm Description and Interface Document (Software)

### 6.1 Design Summary

To develop the software subsystem, some research into methods of implementing GUI based user control of remote systems was conducted. The main software subsystem owner is Sarah Luciano, a graduate student in Computer Science that we are working with to integrate the overall system. We first explored several design options, including a remote desktop, a web-based application, as well as standalone application that runs through the Internet. Another alternative that was considered was an open source game-based server application called 'Blender'. Sarah ended up developing the website using a PHP development framework called Laravel. For our purposes we only needed to know that it was fully functional and communicates properly to our web server host.

The final choice we made to communicate between the website and the web server was to create a text file buffer that is written to with a PHP script (activated upon GUI button presses) that sends text information to the web server. That command is then translated from text into a binary command with a python script on the Raspberry Pi (the web server) and is then sent via USB serial connection to the Arduino Uno. The Arduino Uno is programmed to read in 8-bit binary commands from the USB connection and control the X, Y, and Z-axis motors accordingly.

Two video streams, giving the user a choice in how to view the system, were implemented to mitigate the "lost beyond reason" factor which could happen if a user was too far from the center of the pinhole. The first camera is pointed at the output screen which is what displays the spatially filtered laser and the second camera is pointed at the pinhole directly so that the user can navigate to a general range where they think the pinhole is, and locate the proper alignment from there. This is the most similar to an in lab experience since looking both at the output and the pinhole are vital to determining proper alignment. The video streams are both set at a fixed resolution of 480p (640x480) with a frame rate of 10fps each. Considering that 1 frame loss will result in a video delay of 100ms ( $1s/10frames = .1sec$  or 100ms) and with the two streams running (thus introducing delay in processing time), a total measured delay of around 250ms was measured and deemed acceptable since the requirements outlined that latency would be under 2000ms (2 seconds). This was considered acceptable for a 'real-time' interactive system. A program running on the Raspberry Pi called avconv captures the raw video. This program then converts the video into a streamable RTMP (Real Time Messaging Protocol) format and sends it internally to the video-streaming server. The video streaming server we chose was a lightweight open source piece of

software called MonaServer. One configuration file would need to be set up to manually assign the IP address and port. Once it was setup and running, any RTMP stream sent to the server would then be routed to the Modem/Router and accessible by outside users (as long as the same ports were opened/routed by the local system administrator).

## 6.2 Subsystem Requirements Decomposition

The online interface shall have a dedicated tab displaying the camera view using an in browser language, specifically HTML/PHP, in accordance to this source (Appendix H) which states that the average delay is close to 200ms at a resolution of 720p at 10 frames per second. The configuration, which is compatible with a Raspberry Pi, is only slightly degradation of quality that still meets the minimum requirements (250ms latency and 480p at 10 frames per second). The website created shall be operational on Google Chrome (tested web browser of choice), Safari (native OSX web browser), and Microsoft Edge (the successor to Internet Explorer). The system shall include help indicators accessible during use, which detail the operation of the interface at a level suitable for first-time users.

## 6.3 Drawings, Schematics, and Relevant Visuals

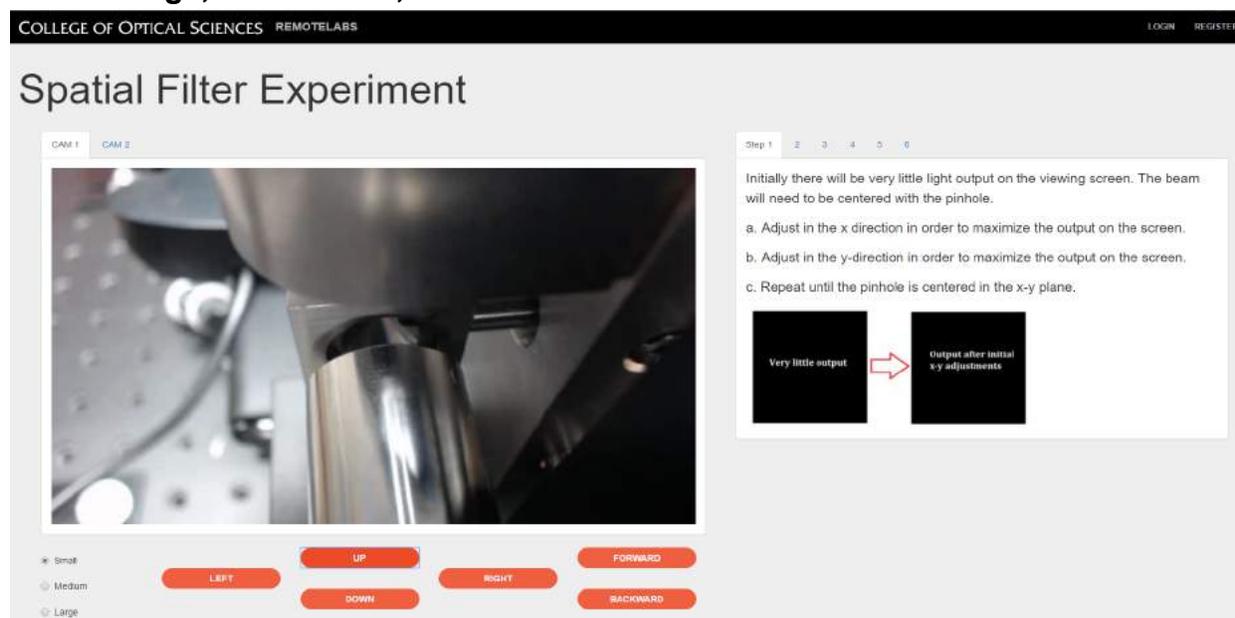


Figure 20: Final Version of the GUI

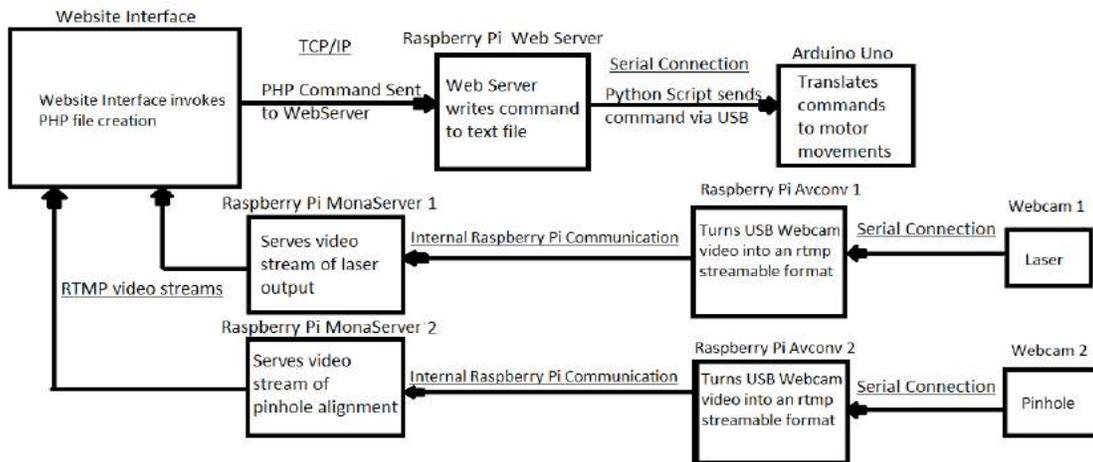


Figure 21: Final Software System Design

### 6.4 Analyses, Specification Sheets, and Modeling Results

The instructions that are sent from the host software on the lab computer to the Arduino microcontroller are as follows according to the table. These are then used to actuate motion in the system.

Command Encoding Table							
Motor #		Motor Direction			Magnitude of Movement		
AA	+	BB	+	CC		= AABBC	
Motor X	01	01	= Positive	01	= Small		
Motor Y	10	1X	= Negative	10	= Medium		
Motor Z	11	00	= No op	11	= Large		
no operation	00			00	no op		
				Actual Inst		= 00AABCC	
Example Instructions							
"Move Motor X in the positive direction one large step"							
011101							
Actual instructions will be 8 bits padded with two zeros since that is how Arduino accepts instructions							
00011101							

Figure 22: Command Encoding Table

## 7.0 Analysis

There were two main aspects of analysis to support system performance. First was the analysis of the delay of the video feedback. To calculate the delay, a phone stopwatch was placed in the field of view of camera that was streamed. Then, a photo was taken that had the phone screen as well as the computer screen with the streamed video feedback. The lag was calculated as the difference between the two stopwatch values.

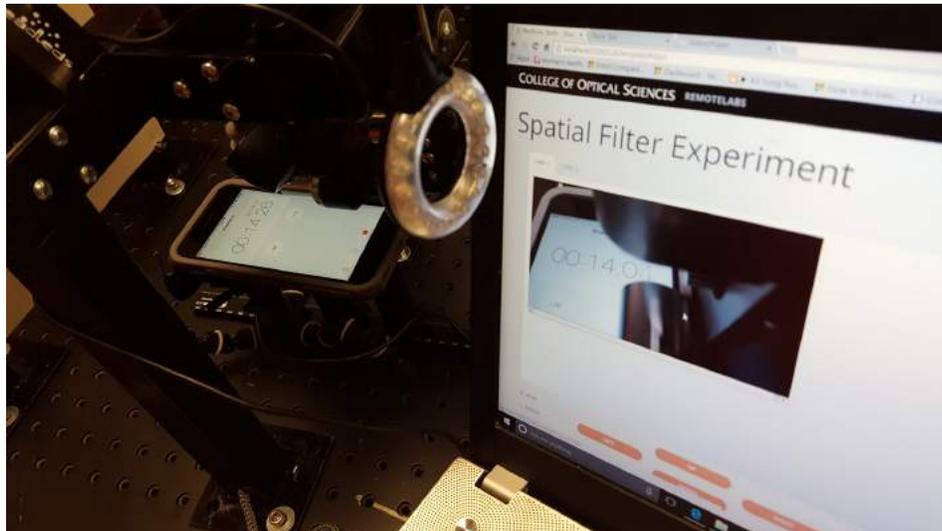


Figure 23: Photo of latency in software

Second, from the mechanical analysis, the majority of the design decisions were made so that all the pieces fit together; thus most of the concerns were merely geometrical. Thus, by modeling each assembly in Solidworks, this helped ensure that everything will fit together properly. Additionally, adjustability was built in to further ensure everything worked properly. The only real mechanical analysis that was worth performing was to see if the stepper motors would produce a small enough step size for the spatial filter if the motors were directly coupled to the micrometers. A typical stepper motor has a 1.8-degree resolution, which means it takes 200 steps for 1 revolution. The given micrometers move the pinhole at a rate of 0.5mm per revolution. Thus, that means, through direct coupling, one step of the stepper motor results in 2.5  $\mu\text{m}$  of movement of the pinhole. The diameter of the pinhole we should be using is 25  $\mu\text{m}$ , so ten times the step size. Thus, direct coupling should be sufficient. However, if it is deemed necessary to have an even smaller step size, the stepper motor can be micro-stepped to 1/32 of a step size, which would translate to 0.078  $\mu\text{m}$  per step, which would be more than enough. The torque the stepper motor is able to produce in a step reduces with microstepping, so that must be taken into consideration if microstepping is

pursued. Regardless, direct coupling is a sufficient means for driving the micrometers.

## **8.0 Development Plan and Implementation**

### **8.1 Development plan timeline**

The development plan presented in the CDR went through several modifications and changes as the semester progressed. Initially, the development plan set a release date for an alpha release of the whole system on March 29, 2016. This was going to be accomplished through simultaneous development of the software, electrical, and mechanical subsystems, and then 1 to 2 weeks of integration work. The development of each of the subsystems took longer than anticipated, which caused a delay in the development plan timeline. Around March 29, the system could be operated locally, but the remote connection had not yet been made.

After spring break, the development plan timeline was revised according to the current status of the project. The new development plan included finalizing the mechanical design and the software so that a remote connection could be made. A majority of the final tests did not take place until mid April. For a full description of the schedule along with screenshots of the Gantt chart, see the Project Management appendix. Despite delays in the development, all of the deliverables were met on time. The mid-semester status review, the final status review, the poster, and final report were all delivered on time.

### **8.2 Final product implementation**

The final product implementation was a mix of software and hardware, which enabled the system to provide for a complete end-to-end user experience. The user interacts with the software user interface to control the hardware, which is located in the sophomore laboratory in the College of Optical Sciences.

The software begins with a home page, which can be seen in Figure 25. After clicking the “Find out more!” button, the user is taken to a page to learn more about the spatial filter experiment, including the background and optics behind the spatial filter. After scrolling to the bottom, the user clicks the “Start” button, which starts the experiment (Figures 26 and 27). The next page is the main page that the user utilizes to perform the experiment. This page has several important features, including two video feedback streams, as well as directional control for each of the three motors that drive the x, y, and z-axes. Additionally, the steps to perform the laboratory are listed on the right side of the page, as seen in Figure 28.

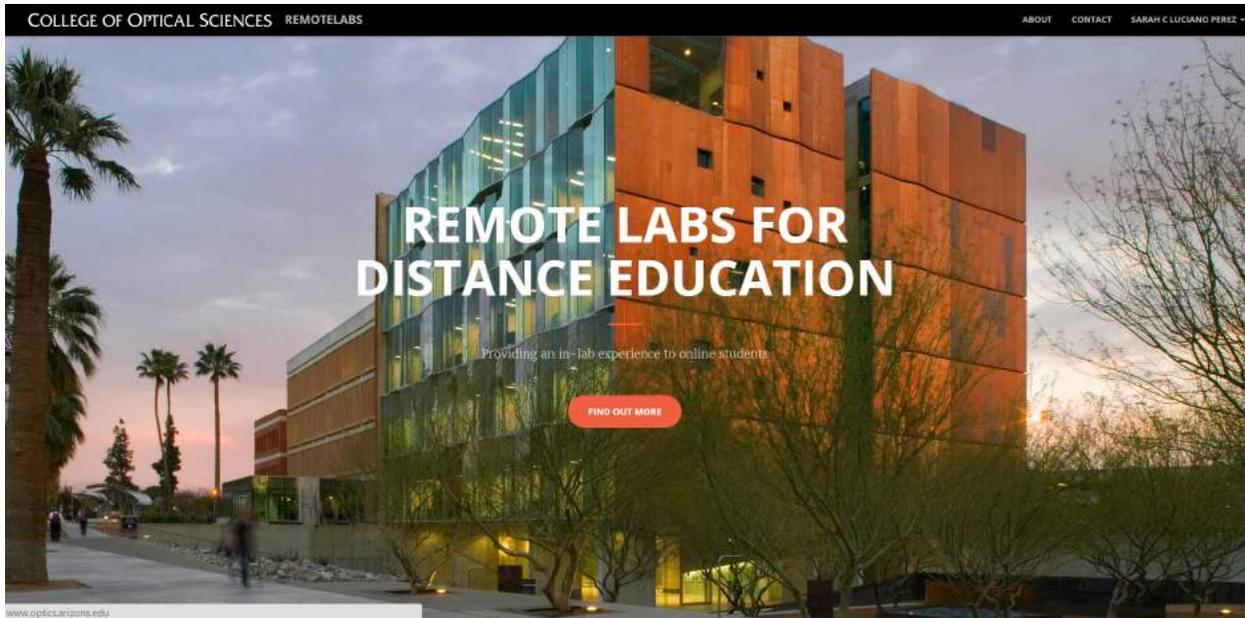


Figure 24: Home Screen

COLLEGE OF OPTICAL SCIENCES REMOTELABS

SARAH C. LUCIANO PEREZ

## Spatial Filter Experiment

Theory

### Spatial Filter Experiment

As a laser beam propagates through an optical system, it experiences small variations in intensity called spatial noise. These variations can be the result of scattering by optical defects, by optical defects, or other imperfections. An ideal beam free of spatial noise is referred to as a “clean” beam while one with spatial noise is called a “dirty” beam.

A spatial filter consisting of an objective lens and a pinhole can be used to clean up this dirty profile. The objective focuses the initial beam down to a spot of size:

$$\text{Beam Spot Diameter } [\mu\text{m}] \approx 1.27\lambda \frac{f}{D}$$

$\lambda$  = Wavelength of laser [ $\mu\text{m}$ ]  
 $f$  = Focal length of objective lens [mm]  
 $D$  = Input beam diameter [mm]

By focusing the beam, the lens performs an optical Fourier transform on the input beam. This allows a pinhole to be placed at the focal plane and used as a low-pass filter which clips the higher spatial frequencies in the beam. The maximum spatial frequency that can be passed is referred to as the cutoff frequency, determined by:

Figure 25: Spatial Filter Experiment Background Information

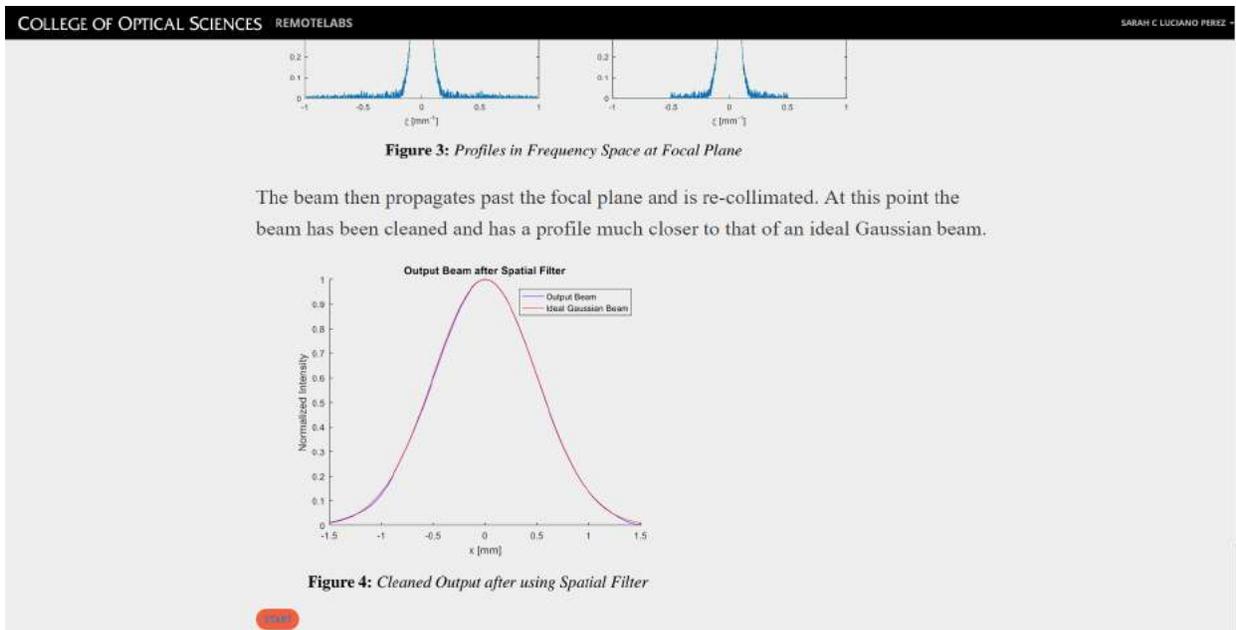


Figure 26: Spatial Filter Background Information with Start Button

COLLEGE OF OPTICAL SCIENCES REMOTELABS

LOGIN REGISTER

## Spatial Filter Experiment

CAM 1 CAM 2

Step 1 2 3 4 5 6

Initially there will be very little light output on the viewing screen. The beam will need to be centered with the pinhole.

- Adjust in the x direction in order to maximize the output on the screen.
- Adjust in the y-direction in order to maximize the output on the screen.
- Repeat until the pinhole is centered in the x-y plane.

UP DOWN LEFT RIGHT FORWARD BACKWARD

Small Medium Large

Figure 27: User Interface Page with Video Feedback and Controls

The hardware was designed to be an extension of the optical assembly to allow for a means to control the each axis of translation of the pinhole in the spatial filter assembly as well as a means to view the output. The mounting hardware for the cameras allows for a fairly wide range of mounting locations allowing for optimal positioning of the cameras for viewing. Additionally, the screen is very adjustable to allow for a high level of control over the size of the laser output on the screen by adjusting the depth of the screen. The motor mounts are adjustable and allow for optimal positioning of the motors at the cost of a little more setup time. Positioning

starts with the x-axis motor, then the y-axis motor, then the z-axis motor and is fairly easy to accomplish. The motor mounts allow for the axial translation of the x and y micrometers. The z micrometer does not translate, and thus that motor is stationary. However, due to the different construction of the z micrometer, it appears that it may take more torque to move the micrometer in one direction more than the other. This may be what is causing a different step size in one direction as opposed to the other. This can be corrected by adjusting the step size input to the motor so that the output step size (small, medium, large) in each direction is the same. This may require further fine-tuning in the future though. Other than that, the mechanical system functions exactly as it should. The adjustability in the mechanical assembly allows for easier wire routing and storage as well.

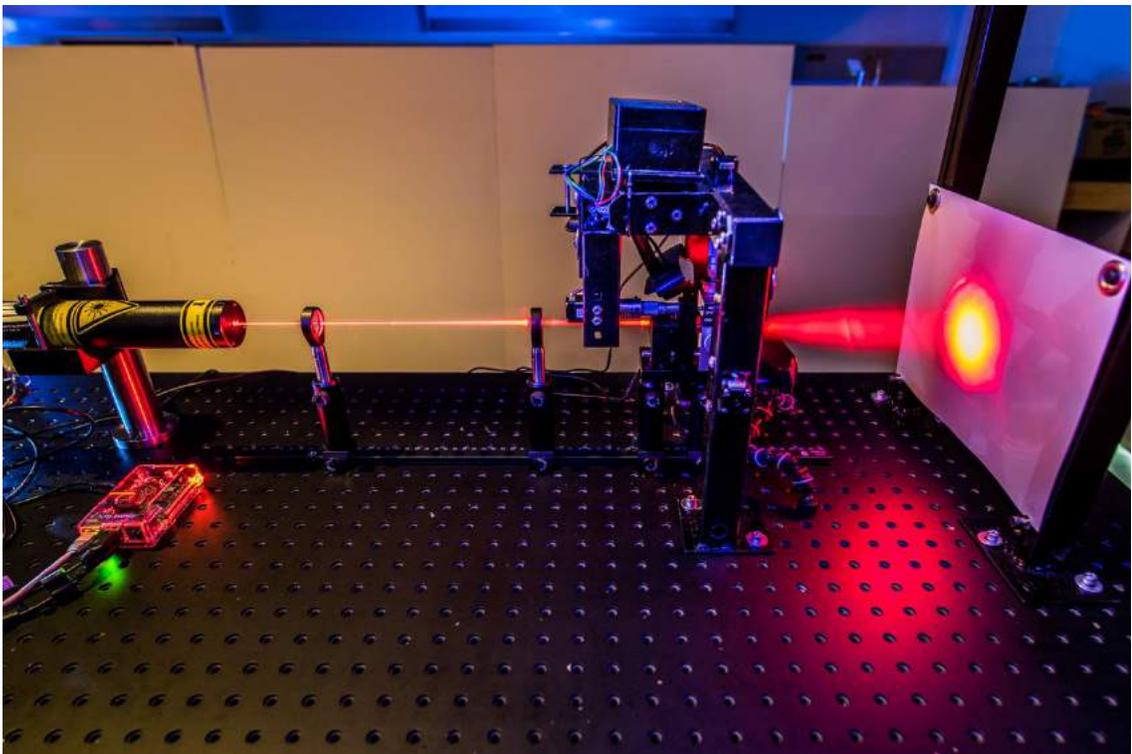


Figure 28: Full Spatial Filter Assembly with Visualization of the Beam

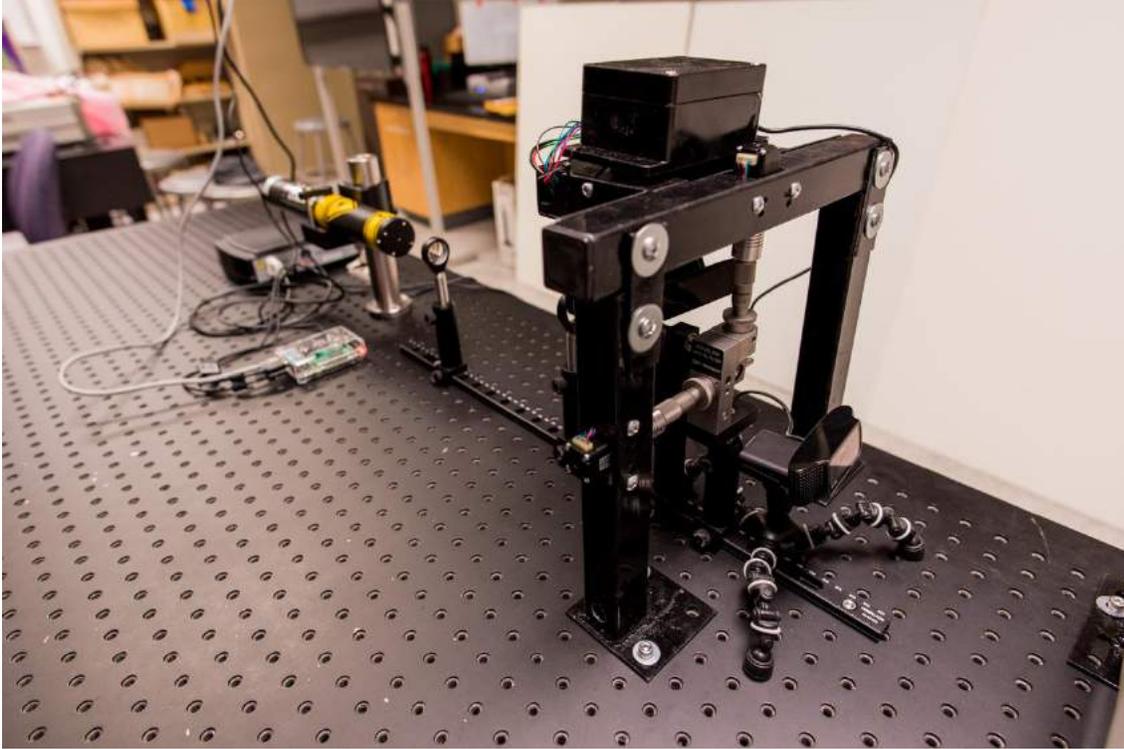


Figure 29: Alternate View of Mounting Brackets

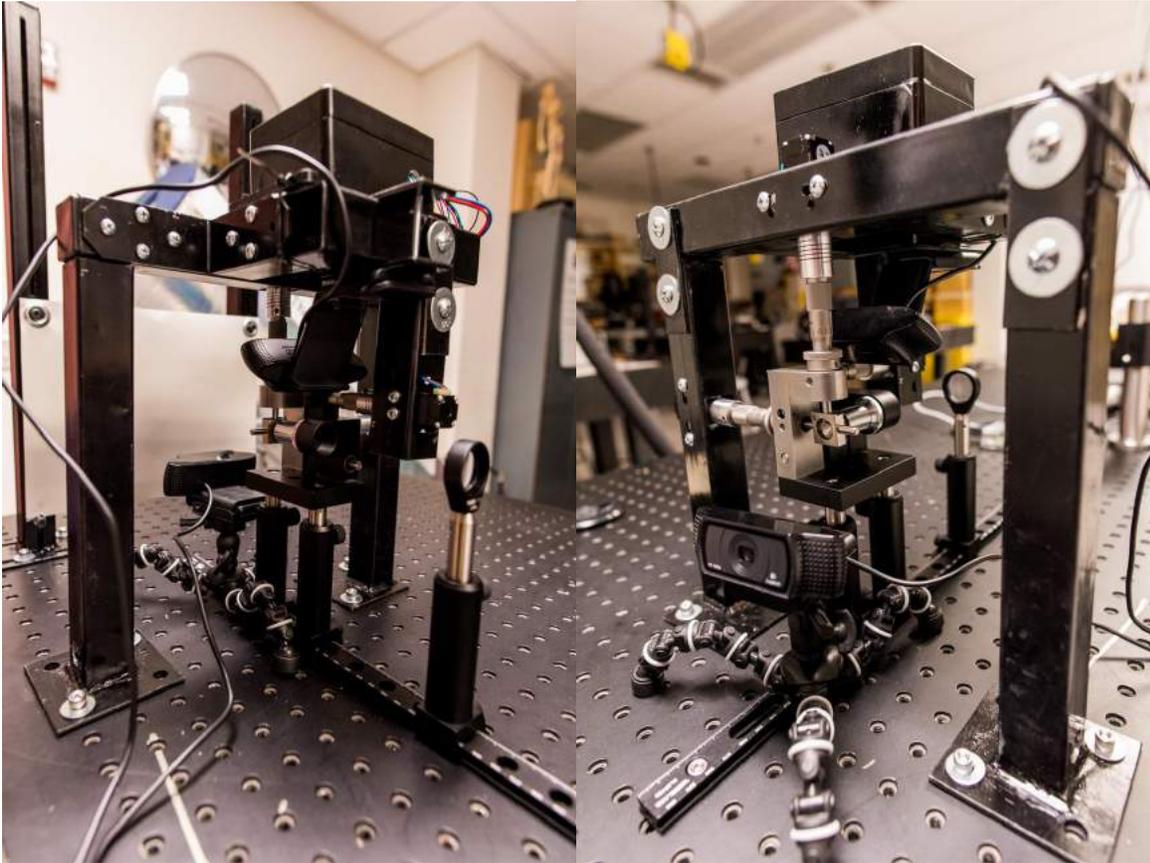


Figure 30: Close View of Spatial Filter with Camera Mounts

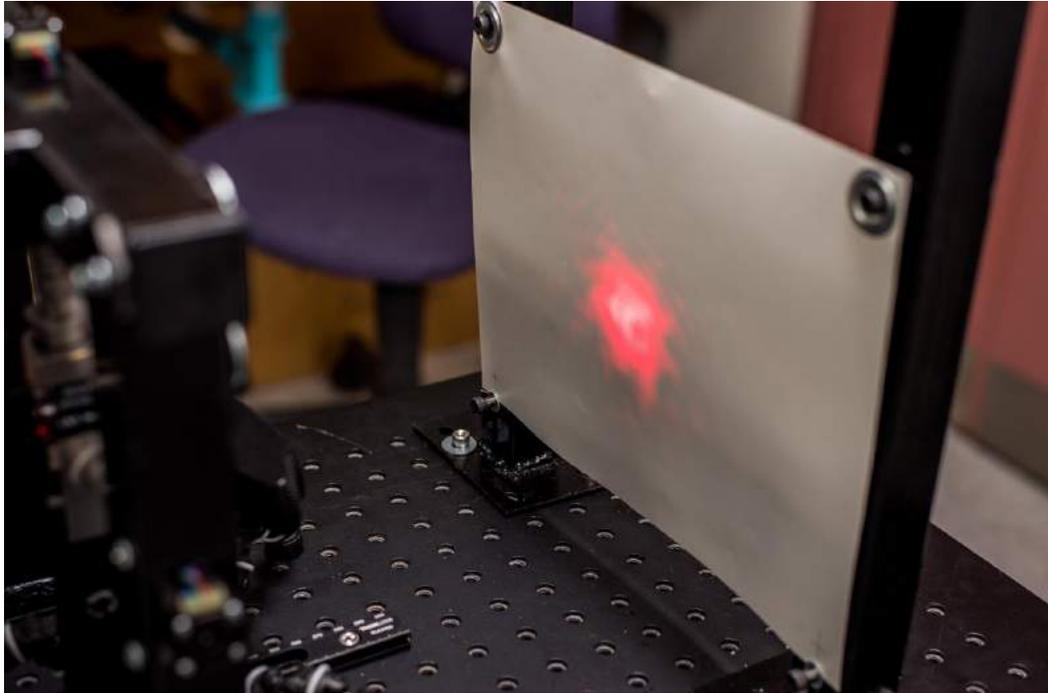


Figure 31: Dirty Beam Output on Screen

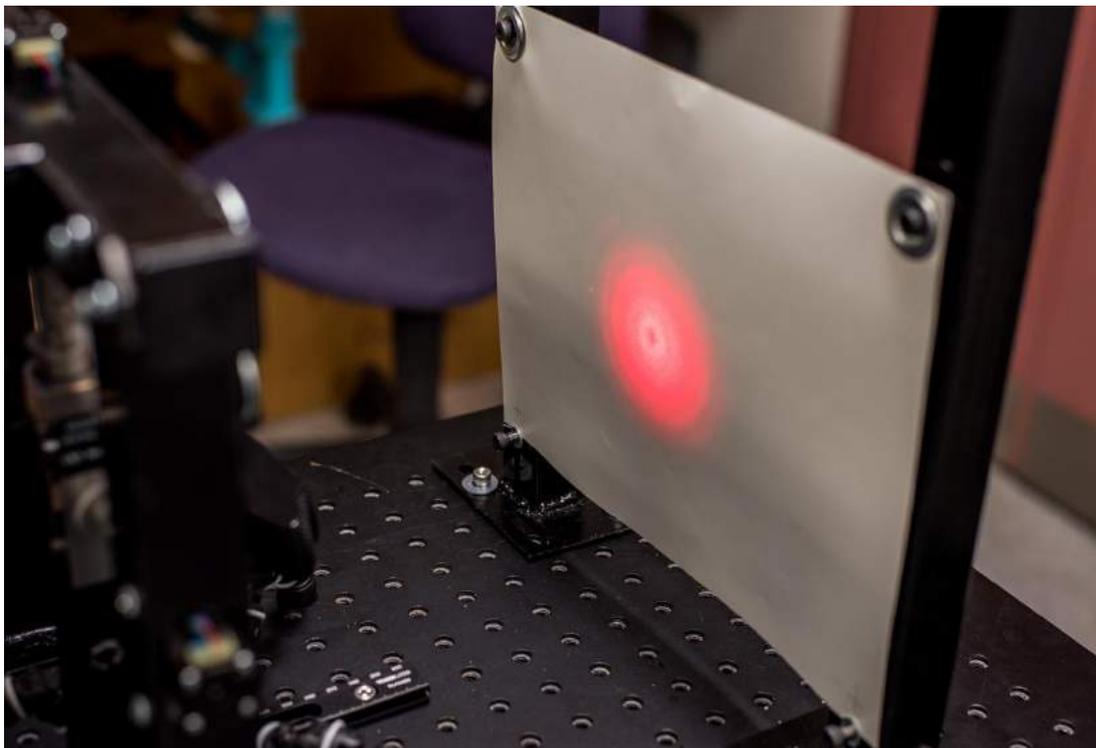


Figure 32: Beam Close to Alignment with Fringes

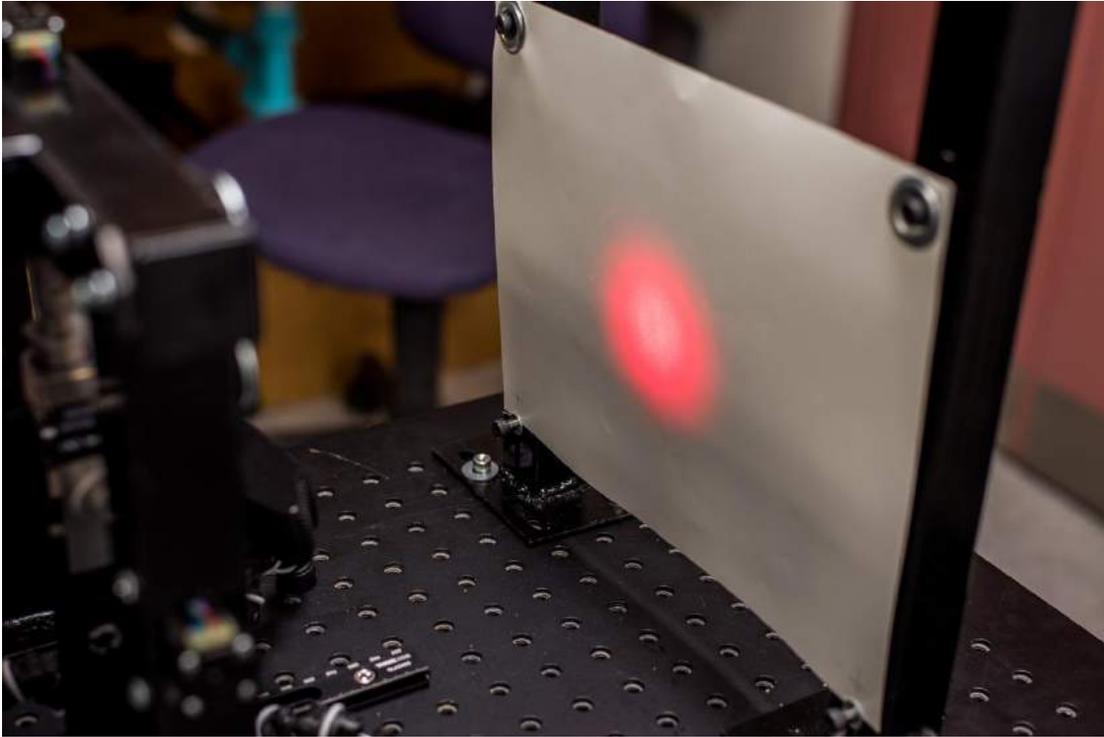


Figure 33: Clean Beam Output After Spatial Filter

## **9.0 Requirements Review / Acceptance Test Results / Performance**

### **9.1 Review of product against requirements**

In order to determine if the final design is acceptable to the specifications agreed upon by the sponsor, the final system must be compared against the requirements set during the preliminary system design. Appendix 12.5, Requirements Matrix, contains all of the requirements defined in the System Requirements Review, as well as the degree to which all requirements were fulfilled. The requirements matrix shows that the final system meets all of the requirements defined for the project.

### **9.2 Results of Acceptance Test**

Several of the requirements needed to be tested to ensure that the requirement was met. The tests performed followed the procedures outlined in the Acceptance Test Plan. All tests were passed, which therefore led to a successful final system end-to-end checkout. To see a detailed description of the test results, including the outcome and any additional follow up notes and comments, see Appendix 12.2, Acceptance Test Plan Results.

## **10.0 Closure**

### **10.1 Summary**

Distance education is a growing trend in today's society. One component, however, that has not been able to be translated well to online learning is the physical experience of a laboratory experiment. Our particular project was to motorize a preexisting optical engineering laboratory experiment so that it could be performed remotely over the internet anywhere in the world. We engineered this with the goal of providing as much of an in-lab experience as possible hopefully demonstrating the viability of this concept as an educational tool. Ultimately, the project does demonstrate that it is possible to translate physical laboratories to online experiences where the critical concepts of the lab are still effectively learned through the manipulation of a limited selection of the hardware components. However, as a supplement to distance students' restricted curriculum, this system can be an invaluable learning tool. In the future, this system could be scaled to incorporate more laboratory experiments with the ultimate goal of creating a fully online curriculum.

### **10.2 Challenges**

One of the challenges of the project as a whole was attempting to translate a physical experience to an online one. There will inevitably be something lost in the translation, however, how closely it imitates the physical experience is a different

story. We believe that for at least lower level optical experiments, where the experiment is mostly already set up for students, we believe that this very closely emulates the physical laboratory experiment. However, for higher-level optical experiments where the students are responsible for setting up the laboratory experiment, it becomes much more difficult to emulate this experience. Thus, there will be difficulty in convincing users that an online lab can provide a satisfactory interaction with physical concepts. This may inhibit getting support for future expansion. However, hopefully this project conveys this concept as a viable educational tool in certain situations.

### **10.3 Future Improvements**

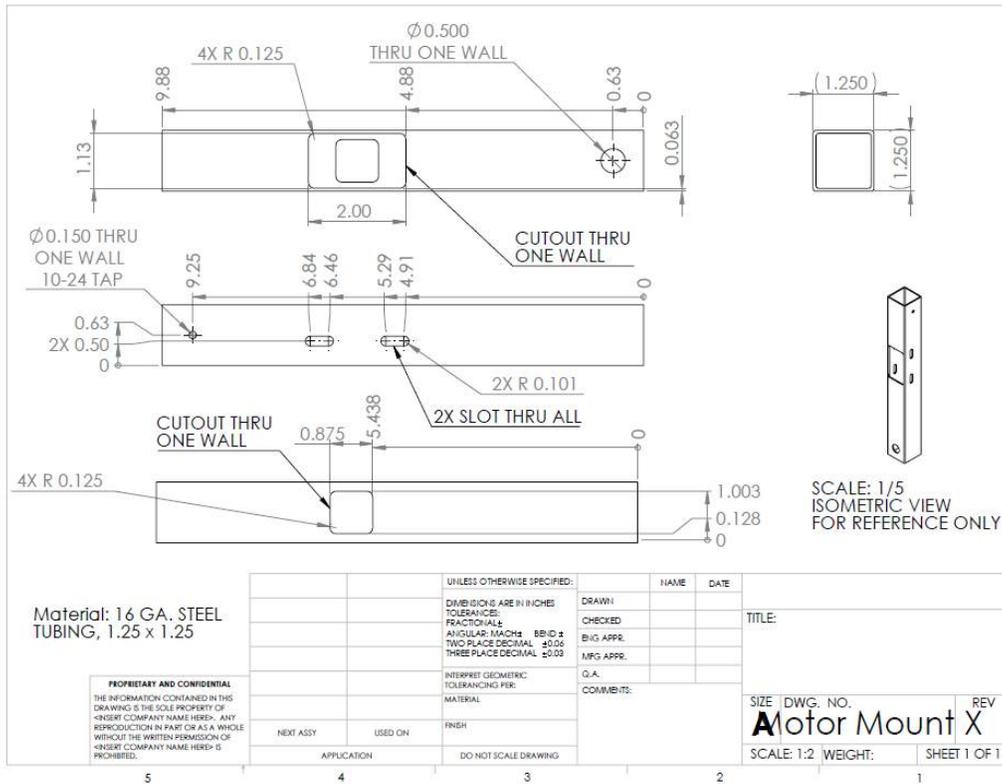
There are several future improvements that could be made to the RLDE system. First, the software could be expanded upon and to incorporate more features. These features could include a scheduling system that assigns time slots for the lab, a more interactive experience (by providing real time feedback based on movements), the movement of other hardware, an administrative login with grading tools, etc. Second, more hardware could be fabricated to allow for other experiments to be performed. Ideally, some of the future hardware could be more universal, such as a motor mount that moves an optical post or a stage, or a motor that allows for tip and tilt of a lens or mirror. Third, the entire system could be geared towards those with physical disabilities and may not have the fine motor skills needed to perform an optical engineering laboratory. Fourth, the cameras could be upgraded to high quality cameras that provide a better picture of the output. Currently, without the proper lighting the video feedback can be blurry.

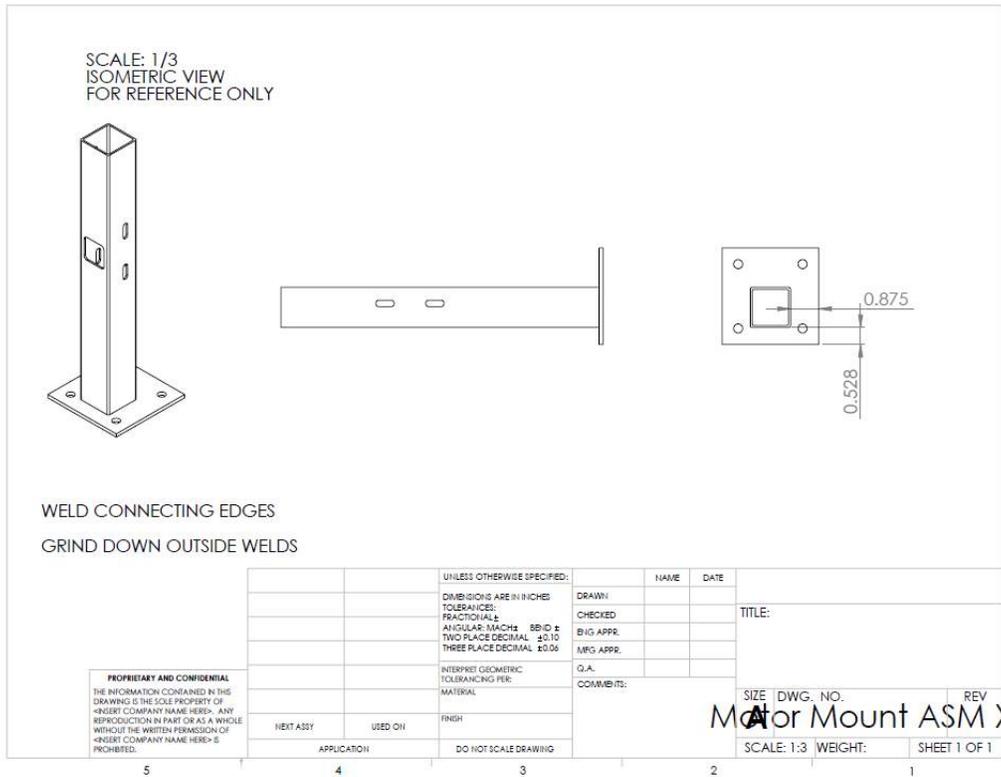
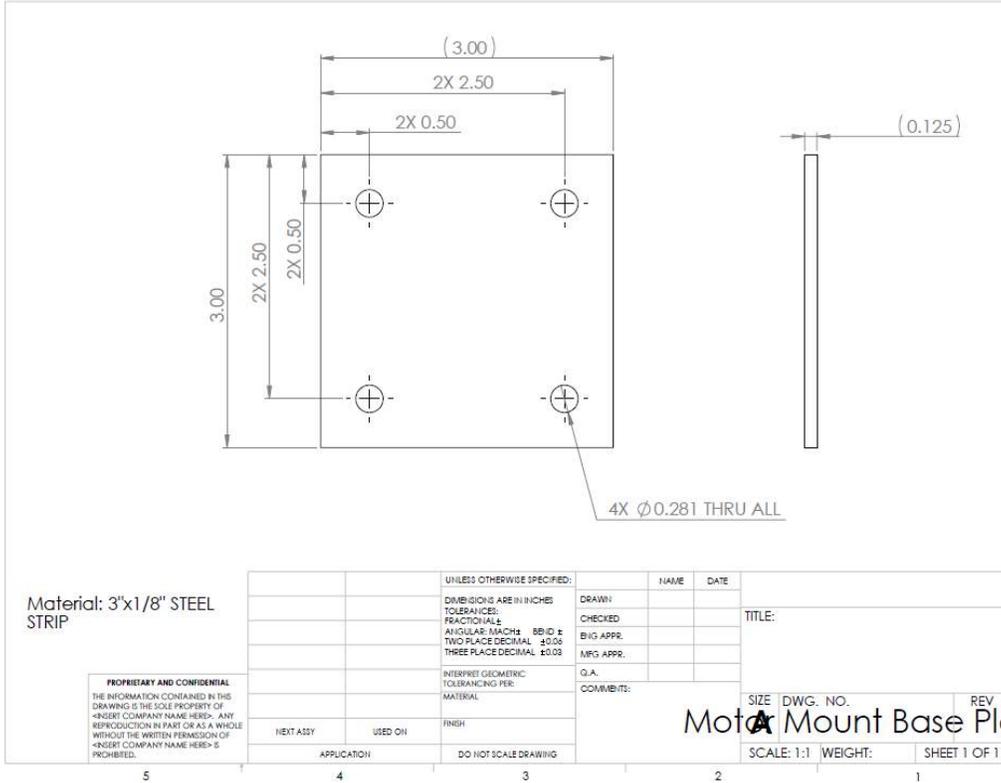
## 11.0 References

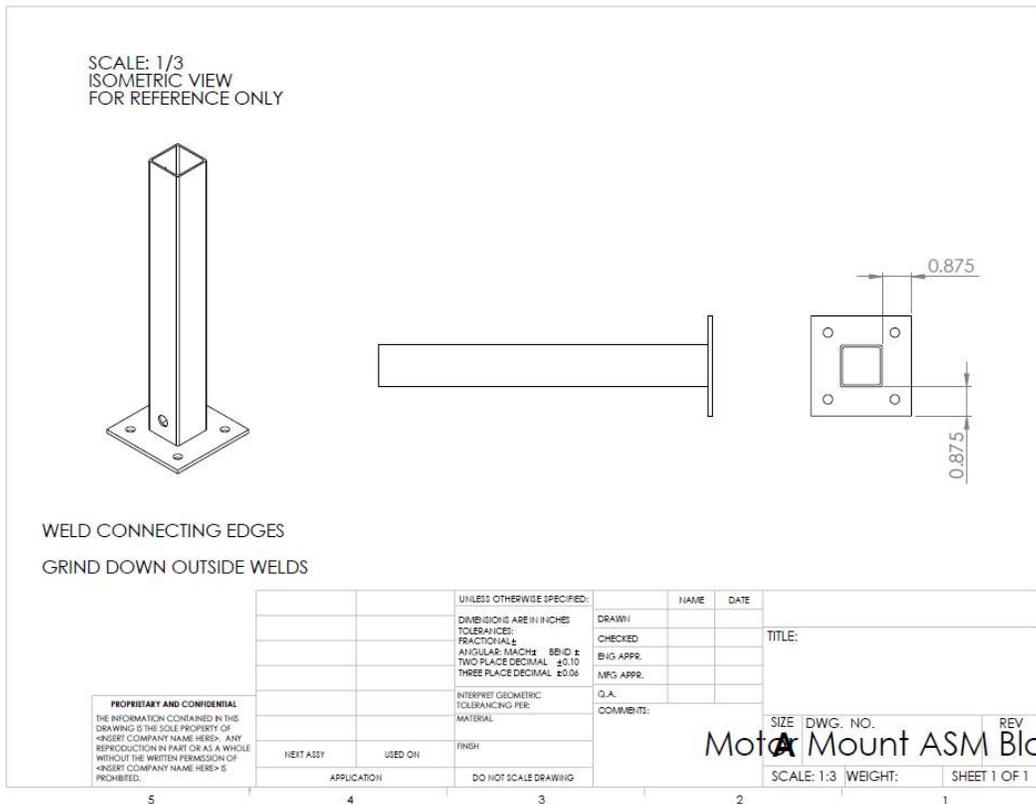
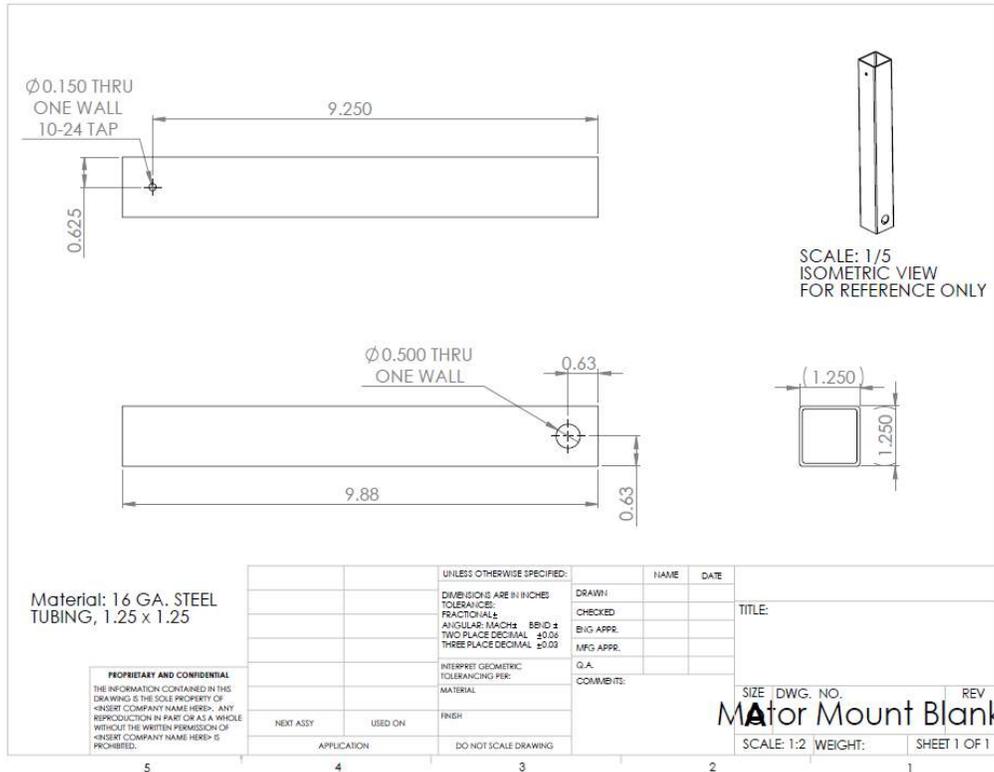
- Hua, Hong. "LAB 3: GAUSSIAN BEAM IMAGING SYSTEMS." *OPTI 471B: ADVANCED OPTICS LABORATORY*. Web. 1 Oct. 2015.  
<[http://wp.optics.arizona.edu/opti471b/wp-content/uploads/sites/27/2015/12/Lab3\\_BImaging\\_OSC471B\\_Hua.pdf](http://wp.optics.arizona.edu/opti471b/wp-content/uploads/sites/27/2015/12/Lab3_BImaging_OSC471B_Hua.pdf)>.
- Nofziger, Mike. "LAB 10: SPATIALLY FILTERING A LASER BEAM FOURIER TRANSFORM OF A GAUSSIAN." *OPTI380A Lab Handouts* (n.d.): n. pag. *OPTI380A Class Website*. 2005. Web. 1 Oct. 2015.  
<<http://fp.optics.arizona.edu/Nofziger/OPTI%20380A/OPTI380A%20Lab10.pdf>>.
- "Spatial Filters." *Newport*. Newport Corporation, n.d. Web. 1 Oct. 2015.  
<<http://www.newport.com/Spatial-Filters/144910/1033/content.aspx>>.
- "Open Source RTMP video server" MonaServer. <<http://www.monaserver.ovh/>>
- "Audio Video Conversion Software" avconv. <<https://libav.org/avconv.html>>
- "Raspberry Pi - Teach, Learn, and Make with Raspberry Pi." Raspberry Pi Home. N.p., n.d. Web. 02 May 2016.

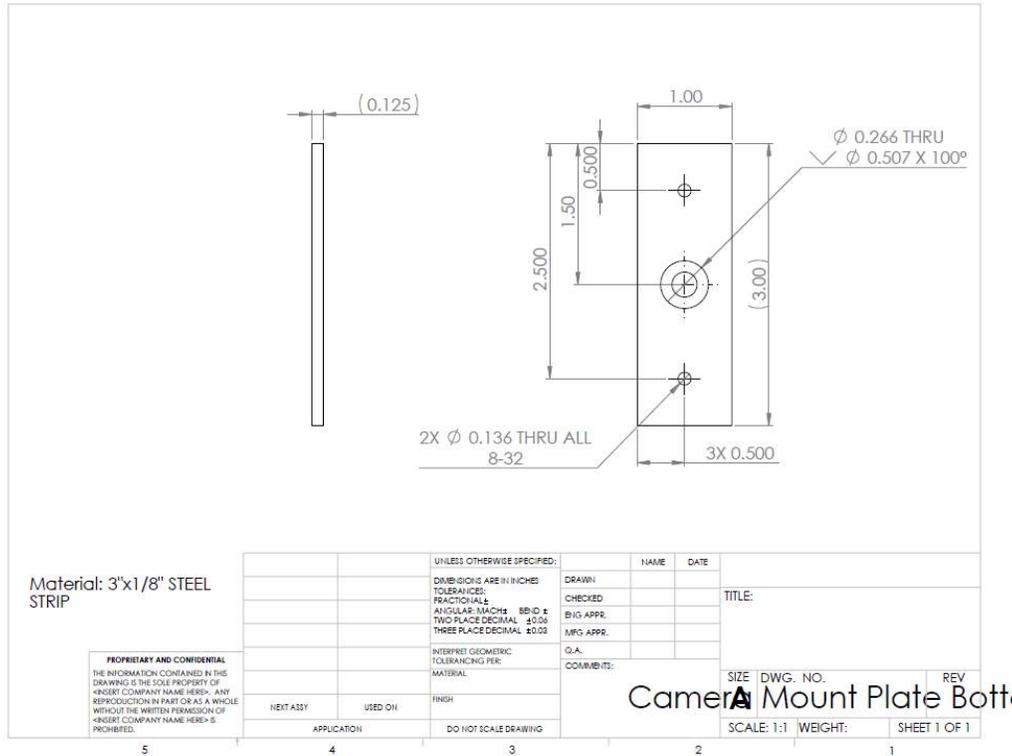
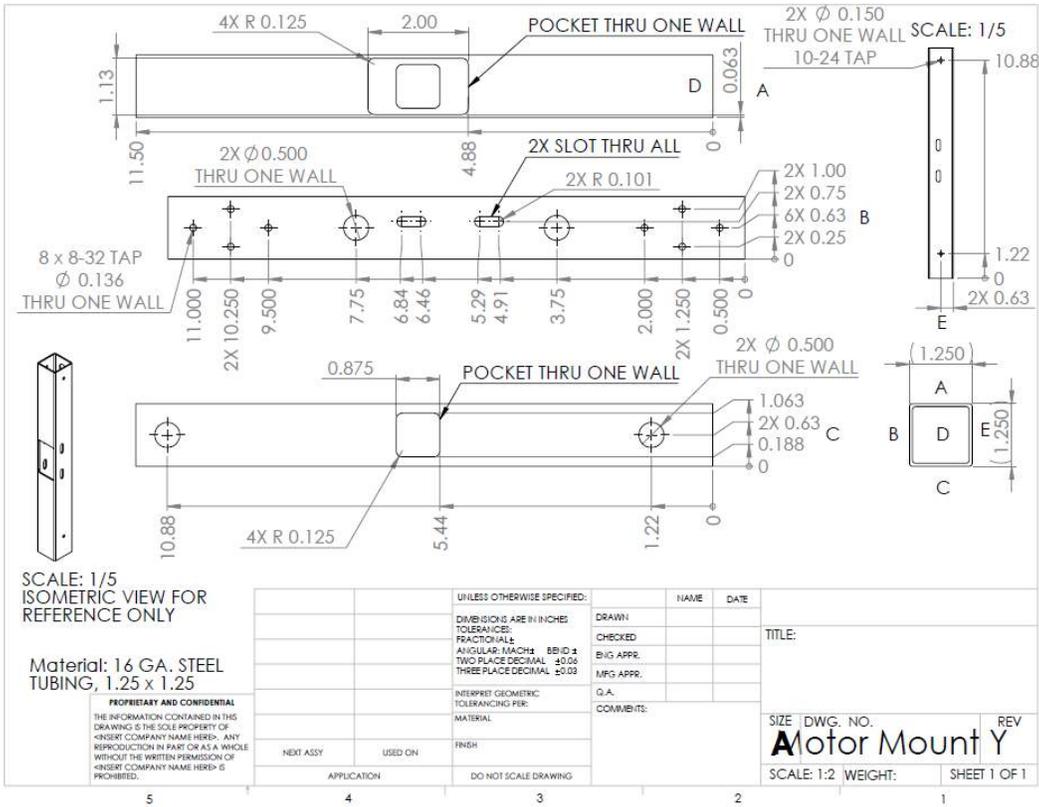
# 12.0 Appendices

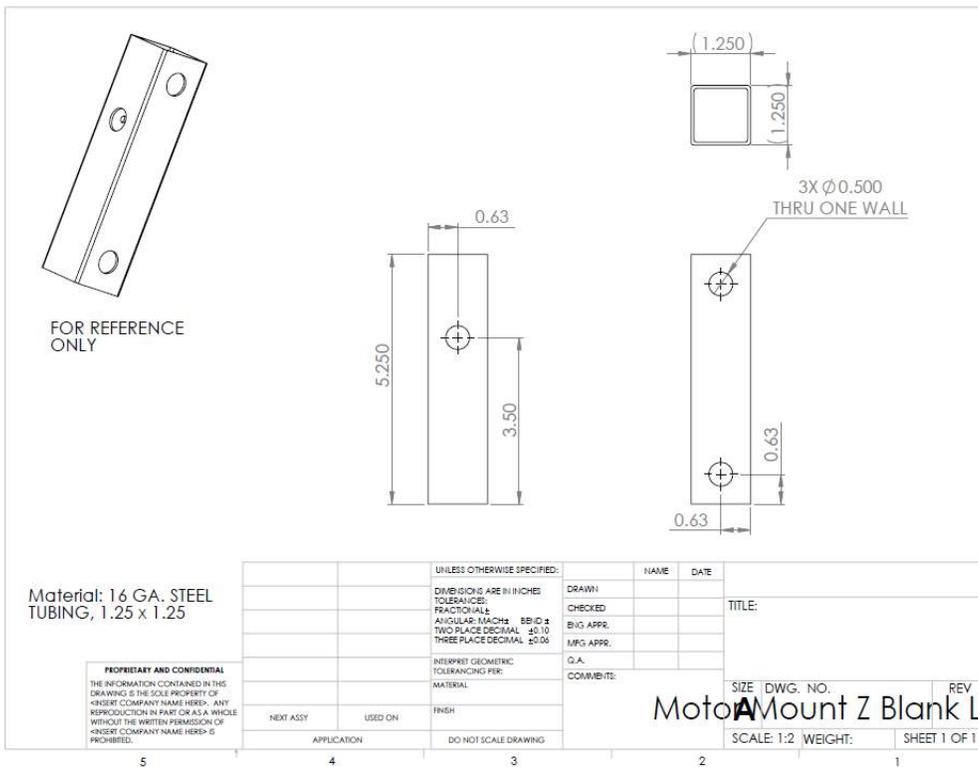
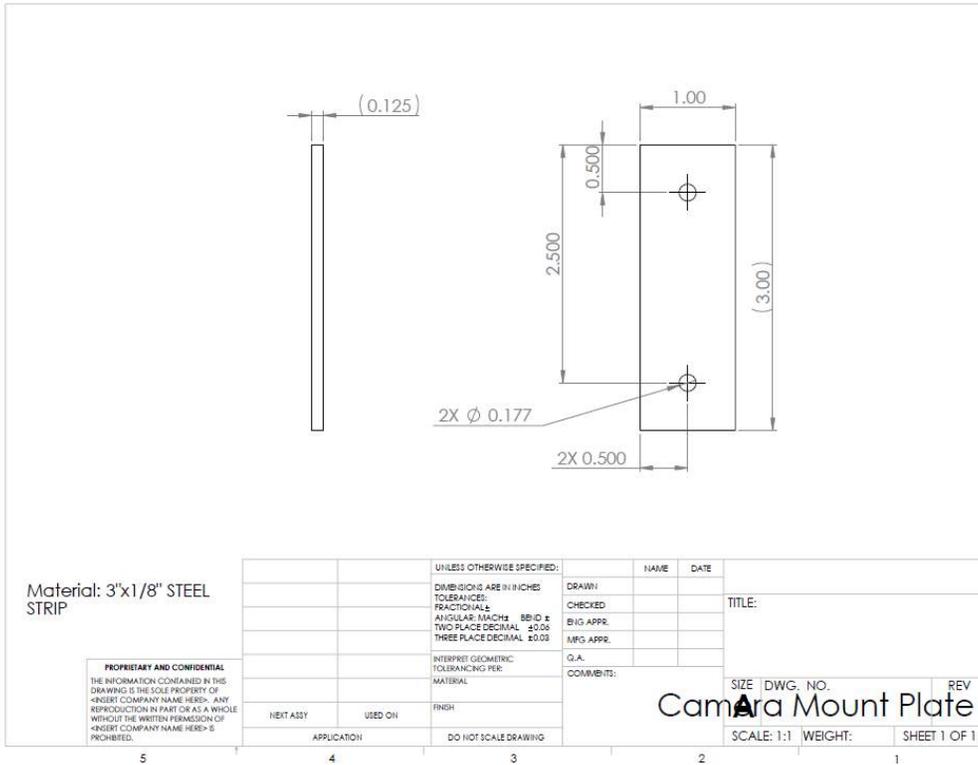
## 12.1 Engineering Drawings

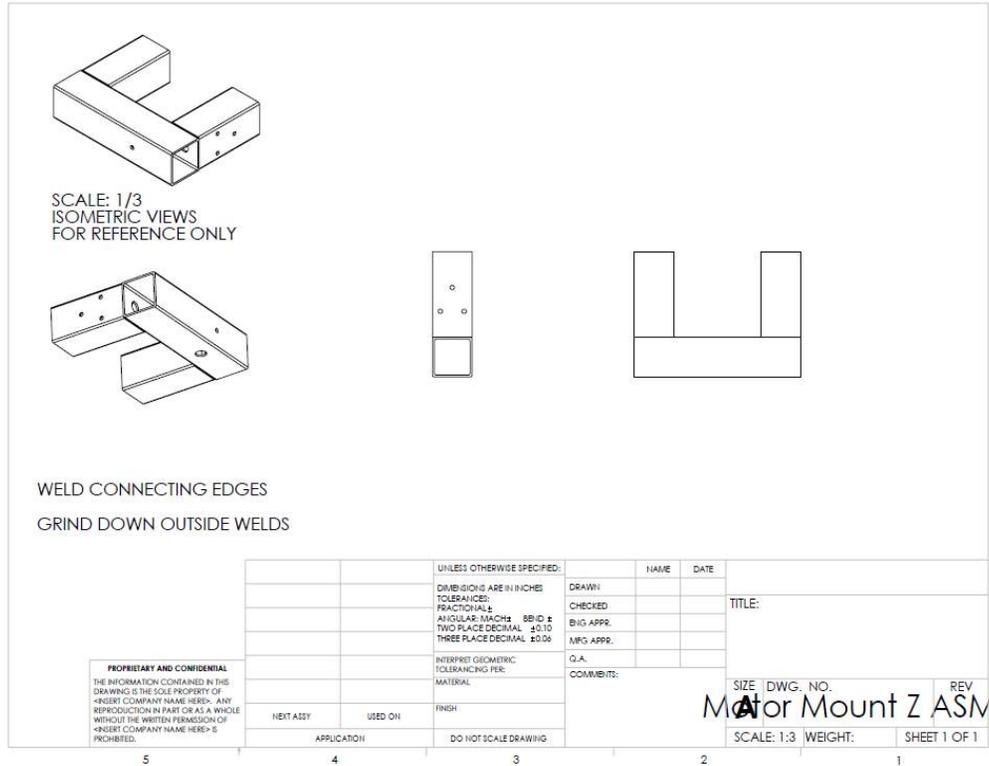
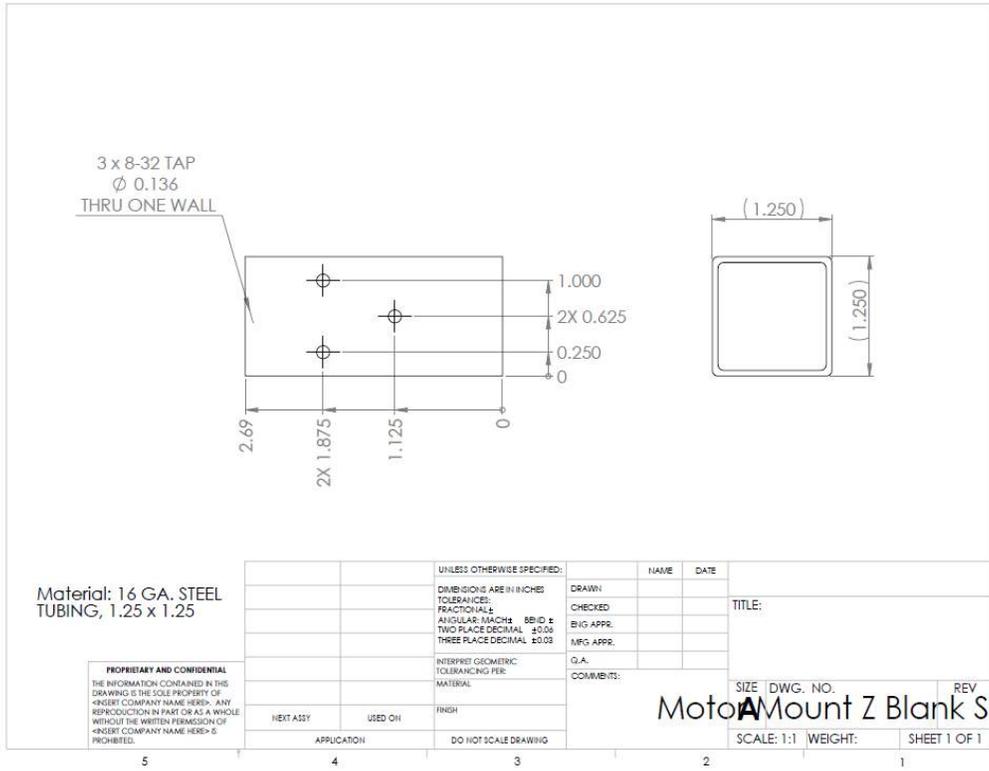




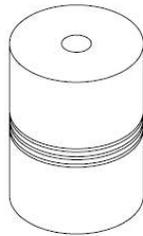




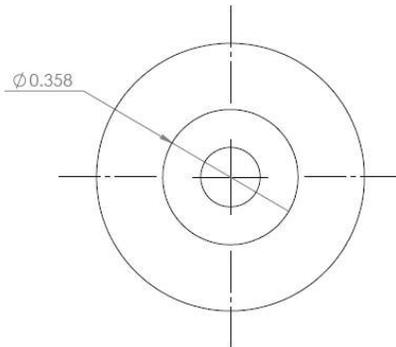








SCALE: 2/1  
ISOMETRIC VIEW  
FOR REFERENCE ONLY

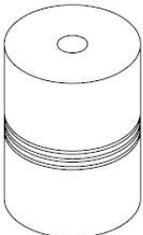


Material: 4mm x 8mm  
Aluminum Coupler

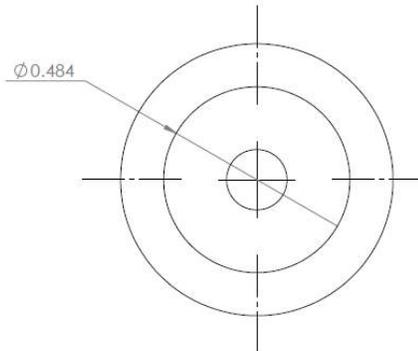
**PROPRIETARY AND CONFIDENTIAL**  
THE INFORMATION CONTAINED IN THIS  
DRAWING IS THE SOLE PROPERTY OF  
<INSERT COMPANY NAME HERE>. ANY  
REPRODUCTION IN PART OR AS A WHOLE  
WITHOUT THE WRITTEN PERMISSION OF  
<INSERT COMPANY NAME HERE> IS  
PROHIBITED.

		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES		DRAWN	
		TOLERANCES:		CHECKED	
		FRACTIONAL		B/G APPR.	
		ANGULAR: MACH: BB ID ±		MFG APPR.	
		TWO PLACE DECIMAL ±0.02			
		THREE PLACE DECIMAL ±0.03			
		INTERPRET GEOMETRIC TOLERANCING PER:		Q.A.	
		MATERIAL:		COMMENTS:	
		FINISH			
NEXT ASSY	USED ON				
APPLICATION		DO NOT SCALE DRAWING			

TITLE:	
SIZE DWG. NO.	REV
<b>A and Y Coupler</b>	
SCALE: 4:1	WEIGHT: SHEET 1 OF 1



SCALE: 2/1  
ISOMETRIC VIEW  
FOR REFERENCE ONLY

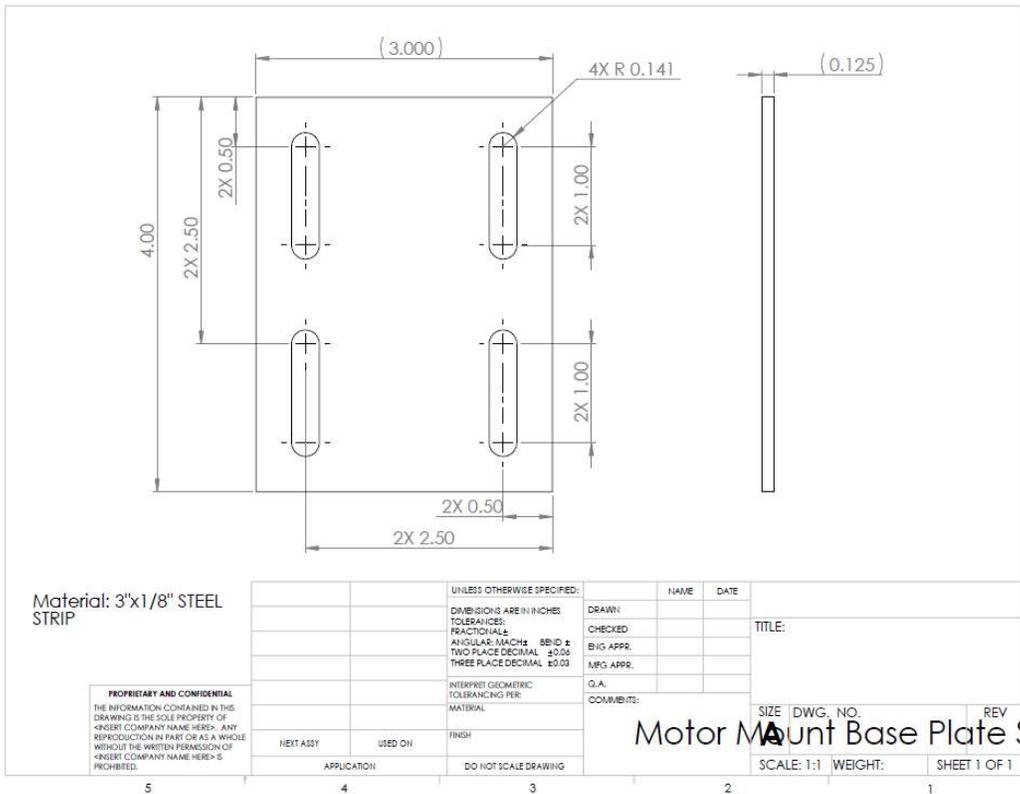
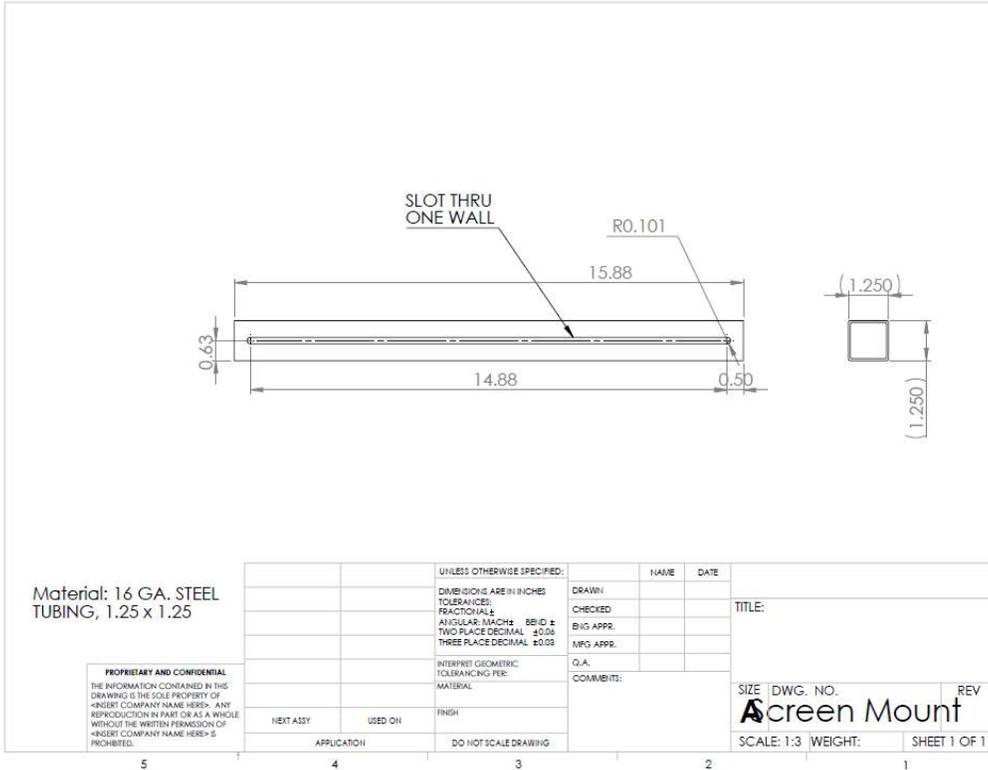


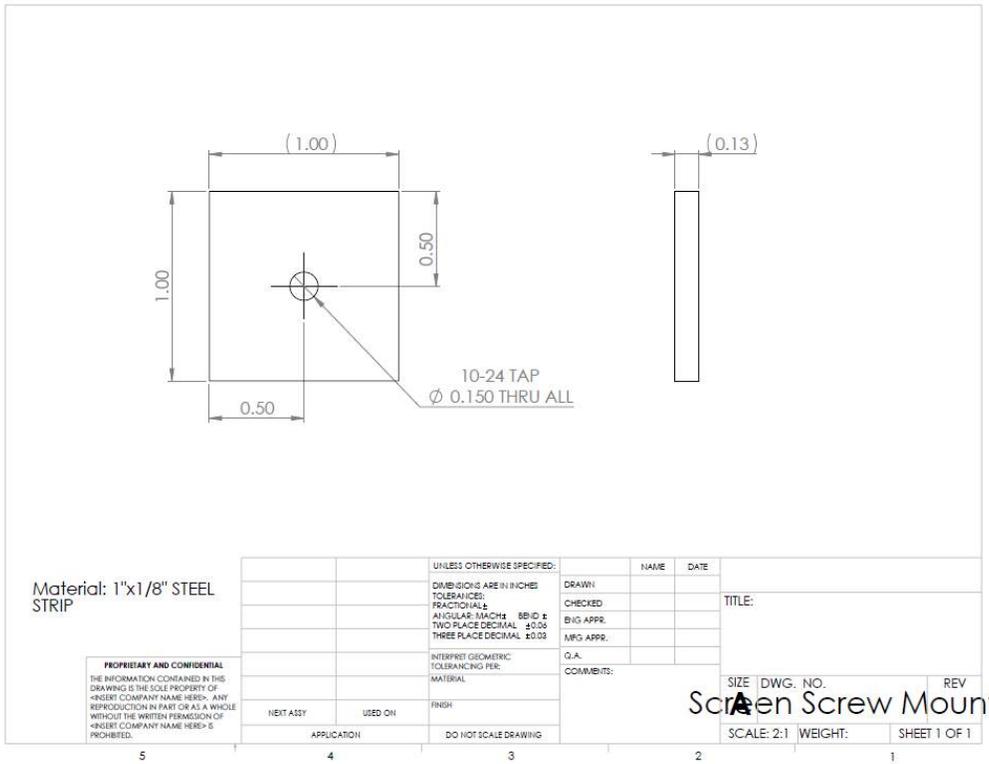
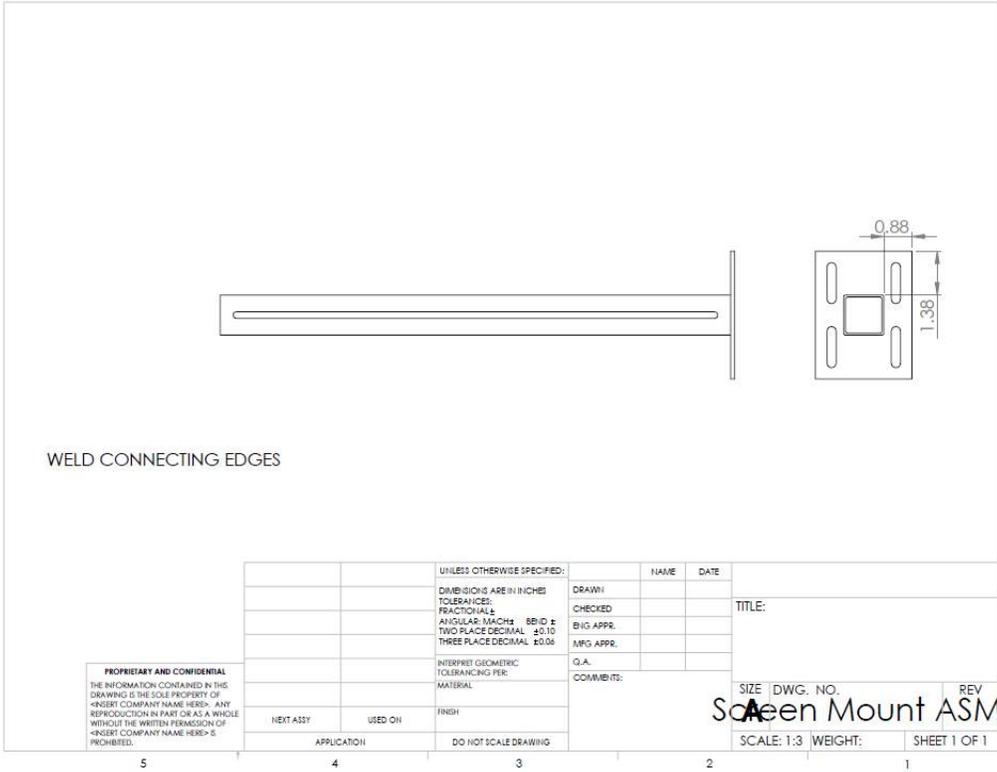
Material: 4mm x 8mm  
Aluminum Coupler

**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.

		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES		DRAWN	
		TOLERANCES:		CHECKED	
		FRACTIONAL		B/G APPR.	
		ANGULAR: MACH: BB ID ±		MFG APPR.	
		TWO PLACE DECIMAL ±0.02			
		THREE PLACE DECIMAL ±0.03			
		INTERPRET GEOMETRIC TOLERANCING PER:		Q.A.	
		MATERIAL:		COMMENTS:	
		FINISH			
NEXT ASSY	USED ON				
APPLICATION		DO NOT SCALE DRAWING			

TITLE:	
SIZE DWG. NO.	REV
<b>A Z Coupler</b>	
SCALE: 4:1	WEIGHT: SHEET 1 OF 1





## 12.2 Acceptance Test Plan Results

Test Number:	1.1
Test Description:	Check connection to system
Functional Group	1
Requirement:	The system shall allow the user to remotely connect to the device online.
Precondition:	Motor mounts fully integrated to experiment. Local motor movement verification completed.
Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to Arduino Uno via USB 2.0 A-male to B-male cord. Set up software environment utilized for connectivity on host computer. One member log on to Internet-based software remotely from an off-campus network. Move all three motors using controls on remote computer. One member in lab verifies that the motors move as expected and record results. Repeat Steps 1-7 for 3 replications.
Criteria to meet:	All three motors move using remote connection.
Outcome:	All three motors moved in both negative and positive directions.
Pass/Fail:	Pass
Follow up and Comments:	The remote connection was made through the U of A Wi-Fi. The firewall is open only to the U of A network; however, the port can be opened to outside networks with permission from Ashley Bidegain. As a note, z-axis step size appeared smaller than x and y-axis steps.

Test Number:	1.2
Test Description:	Verify end stops
Functional Group	1
Requirement:	System shall have end stops that will prevent any components from moving too far as well as prevent any permanent deformation to the other parts.
Precondition:	Test 1.1 passed.
Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to Arduino Uno via USB 2.0 A-male to B-male cord. Set up software environment utilized for connectivity on host computer. One member log on to Internet-based software remotely

	<p>from an off-campus network.  Move all three motors to limit in each direction using controls on remote computer.  Press forward and backward repeatedly to determine motor behavior.  While pressing the forward button for any three of the motors, the user will abruptly disconnect.  One member in lab verifies that the motors move as expected and does not move past any desired stop and record results.</p>
Criteria to meet:	All three motors move within range desired and specified by software. Upon abrupt disconnect, system stops immediately.
Outcome:	All three motors move within the desired range specified by the software. There is no movement upon abrupt disconnect.
Pass/Fail:	Pass
Follow up and Comments:	The rollers allow the motors to translate so that no physical harm is done to the system.

Test Number:	1.3
Test Description:	Laser sag test
Functional Group	1
Requirement:	Laser shall stay fixed within $\pm 0.5$ degree.
Precondition:	None
Procedures:	<p>Mount laser to rail.  Measure the height of the laser at a minimum of three points along the rail.  After measurements are taken, leave laser mounted and turned on.  Repeat Steps 1-3 for 3 replications, removing and remounting laser each time.  Mark laser location at a distance of at least 1 meter away from laser source.  Leave laser on for 24 hours.  Measure change in laser location from initially marked location.  Perform calculations to determine laser sag</p>
Criteria to meet:	Laser does not move by more than $\pm 0.5$ degree over the course of a 24-hour period.
Outcome:	No measurable sag over 48 hour time period.
Pass/Fail:	Pass
Follow up and Comments:	Tape marked on wall, laser checked 48 hours later.

Test Number:	1.4
Test Description:	Motor mount test
Functional Group	1
Requirement:	The system shall be able to mount to an optics table with 1/4-20 holes spaced on a 1-inch grid.
Precondition:	Motor mounts fabricated.
Procedures:	Set up spatial filter experiment on optical table. Attach motor mounts and brackets to optical table using hardware specified in the BOM. Couple spatial filter micrometers to motors. Verify all mounting points and interfaces with experiment.
Criteria to meet:	All mounting hardware mounts directly to optical table without altering any of the hardware. Motors couple to micrometers.
Outcome:	All hardware mounts to optical table and motors successfully coupled to micrometers.
Pass/Fail:	Pass
Follow up and Comments:	Some slight modifications had to be made to the original mechanical design to allow for adjustability in the mounting and set up.

### Test Procedures for Software Requirements – Functional Group 2

Test Number:	2.1
Test Description:	Camera feedback test
Functional Group	2
Requirement:	The online interface shall have at minimum one camera view broadcast.
Precondition:	Software is fully functional and has been proven to connect a remote user to the host computer.
Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to Logitech camera via USB 2.0 cord. Set up software environment utilized for connectivity on host computer. One member log on to Internet-based software remotely from an off-campus network. Verify that there is live visual feedback from the camera by calling another team member in the lab and communicating with them as they enter the video feed. Record results and verify for all camera angles used.
Criteria to meet:	Online interface window has at least one camera view broadcast.
Outcome:	Two video streams provide live feedback on the final

	software.
Pass/Fail:	Pass
Follow up and Comments:	One camera looking back at pinhole and the other camera looking at projection screen.

Test Number:	2.2
Test Description:	Browser verification
Functional Group	2
Requirement:	The software shall be accessible from Internet Explorer, Safari, or Google Chrome.
Precondition:	Software is fully functional and deployable.
Procedures:	<p>Turn on host computer.</p> <p>Set up spatial filter experiment on optics table.</p> <p>Connect host computer to experiment with all necessary USB connections.</p> <p>Set up software environment utilized for connectivity on host computer.</p> <p>One member log on to Internet-based software remotely from an off-campus network using the Internet Explorer browser</p> <p>Verify that the software is fully functional: video feedback, help indicators, movement of motors in all three axes, connect/disconnect capabilities.</p> <p>Repeat steps 1-6 utilizing Safari and Google Chrome as the browser of choice.</p>
Criteria to meet:	At least one of the three browsers allows for full functionality of the software.
Outcome:	Software is fully functional on all three browsers: Internet Explorer, Safari, and Google Chrome.
Pass/Fail:	Pass
Follow up and Comments:	Software has a flash component and will not work on mobile devices as is.

Test Number:	2.3
Test Description:	Internet connectivity and plugin test
Functional Group	2
Requirement:	The software shall run as an internet application.
Precondition:	Software is fully functional and deployable.
Procedures:	<p>Turn on host computer.</p> <p>Set up spatial filter experiment on optics table.</p> <p>Connect host computer to experiment with all necessary USB connections.</p> <p>Set up software environment utilized for connectivity on host computer.</p> <p>One member log on to Internet-based software remotely</p>

	from an off-campus network. Verify that the software performs all functions utilizing only browser plugins.
Criteria to meet:	Software is fully functional using only browser plugins and no other downloads.
Outcome:	Software provides video feedback and allows for the movement of the x, y, and z-axis micrometers using only web browser plugins.
Pass/Fail:	Pass
Follow up and Comments:	Uses flash.

Test Number:	2.4
Test Description:	Help indicators test
Functional Group	2
Requirement:	The system shall include help indicators accessible during use, which detail the operation of the interface at a level suitable for first-time users.
Precondition:	Software is fully functional and deployable.
Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to experiment with all necessary USB connections. Set up software environment utilized for connectivity on host computer. Set up a remote computer for connection to the host computer and experiment. Have a user who has never used the software before log on and explore the help indicators and perform the experiment. Ensure user can complete experiment and navigate the software controls using only the help indicators provided internally to the page. Repeat Steps 1-7 for 5 people.
Criteria to meet:	All 5 test subjects are able to successfully align the spatial filter and utilize the software without additional explanation.
Outcome:	All 5 test subjects were able to utilize the software to move the spatial filter.
Pass/Fail:	Pass
Follow up and Comments:	The full alignment of the spatial filter was sometimes questionable; however, all participants were able to move the hardware using the software controls.

### Test Procedures for Hardware Requirements – Functional Group 3

Test Number:	3.1
Test Description:	Power type test
Functional Group	3
Requirement:	The system electronics shall ultimately be provided power from a 120V US power outlet.
Precondition:	Motor mounts fabricated and all local electronics to the host computer finished and assembled.
Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to experiment with all necessary USB connections. Set up all motor mounts with motors installed. Plug all electronics in. Ensure power is not supplied from any source other than a 120V US power outlet.
Criteria to meet:	All electronics powered by 120V US power outlet and no battery or other power source is necessary.
Outcome:	All electronics ultimately plugged into a 120V US power outlet.
Pass/Fail:	Pass
Follow up and Comments:	A power converter was used for some of the electronics.

Test Number:	3.2
Test Description:	Camera verification test
Functional Group	3
Requirement:	The lab shall be instrumented with video cameras to provide the visual feedback.
Precondition:	Software is functional to provide video streaming.
Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to experiment with all necessary USB connections. Turn on camera. Log in to software from remote computer. Verify there is visual feedback using same steps as outlined in Test 2.1.
Criteria to meet:	Camera provides video feedback at user's end when connected remotely.
Outcome:	The two cameras both provided video feedback at the user's end.
Pass/Fail:	Pass
Follow up and Comments:	Streaming begins when startup scripts run. Video feed is 480p, can be increased but at the cost of more delay. The

	camera looking back the pinhole can take some time to focus and occasionally tries to focus on the background.
--	--

Test Number:	3.3
Test Description:	Microcontroller verification test
Functional Group	3
Requirement:	Motors shall be controlled using micro controllers.
Precondition:	Motors, motor drivers, Arduino Uno, shield, and wires are in stock
Procedures:	<p>Plug Arduino Uno into host computer.</p> <p>Connect Arduino Uno to motor drivers via the shield.</p> <p>Connect the motor drivers to each respective motor.</p> <p>Use computer to move each motor in both directions with smallest step size of 1.8 degrees.</p> <p>Ensure motors can be started and stopped and that they do not surpass any stops encoded.</p>
Criteria to meet:	All three motors move the correct amount when directed by the user input from the computer.
Outcome:	All three motors are able to move in both the positive and negative directions. The smallest step size of 1.8 degrees is achieved.
Pass/Fail:	Pass
Follow up and Comments:	The z-axis motor steps were smaller than the x and y-axis motors.

Test Number:	3.4
Test Description:	Reset time test
Functional Group	3
Requirement:	The system shall be able to reset itself within 5 minutes after a user has finished an experiment, in order to prepare for the subsequent user.
Precondition:	Experiment and motors with motor mounts fully functional from local host computer.
Procedures:	<p>Turn on host computer.</p> <p>Set up spatial filter experiment on optics table.</p> <p>Connect host computer to experiment with all necessary USB connections.</p> <p>Set up all motor mounts with motors installed.</p> <p>Align spatial filter using controls on host computer (or remote computer if the software is complete).</p> <p>Press the reset button and time the amount of time it takes for the system to return to a neutral configuration where the spatial filter is not aligned.</p> <p>Repeat Steps 1-6 for 5 replications.</p>
Criteria to meet:	The system returns to the neutral configuration in less than

	5 minutes for all 5 replications.
Outcome:	The system is able to reset in less than 5 minutes for all replications.
Pass/Fail:	Pass
Follow up and Comments:	N/A

### Test Procedures for Performance Requirements – Functional Group 4

Test Number:	4.1
Test Description:	Time delay test
Functional Group	4
Requirement:	The time delay from remote input to visual feedback shall be less than 2000 milliseconds on average.
Precondition:	Software is fully functional and deployable. Test 2.1 complete.
Procedures:	<p>Turn on host computer.</p> <p>Set up spatial filter experiment on optics table.</p> <p>Connect host computer to Logitech camera via USB 2.0 cord.</p> <p>Set up software environment utilized for connectivity on host computer.</p> <p>One member log on to Internet-based software remotely from an off-campus network.</p> <p>Verify that there is live visual feedback from the camera by calling another team member in the lab and communicating with them as they enter the video feed.</p> <p>Have another team member time the amount of time it takes from when the person places hand in camera until remote user declares he/she can see the hand.</p> <p>Repeat Steps 1-7 for 5 replications.</p>
Criteria to meet:	The delay is less than 2000 milliseconds for all 5 replications.
Outcome:	The delay for two video streams of 480p at 10 fps is on average 250ms. With higher resolution the delay increased.
Pass/Fail:	Pass
Follow up and Comments:	680x480 at 60 fps is close to 3000ms of delay.

Test Number:	4.2
Test Description:	Video quality test
Functional Group	4
Requirement:	The video streaming shall be broadcast with a quality of at least 240p.
Precondition:	Software is fully functional and deployable. Test 2.1 complete.

Procedures:	Turn on host computer. Set up spatial filter experiment on optics table. Connect host computer to Logitech camera via USB 2.0 cord. Set up software environment utilized for connectivity on host computer. One member log on to Internet-based software remotely from an off-campus network. Verify that there is live visual feedback from the camera. Check data usage for web page to verify a minimum of 240p.
Criteria to meet:	The video quality is greater than 240p.
Outcome:	Video quality is 480p.
Pass/Fail:	Pass
Follow up and Comments:	Video quality can be increased up to 1080p, at the expense of delay.

Test Number:	4.3
Test Description:	Set up time test
Functional Group	4
Requirement:	The system shall take less than 2 hours to set up from an 'out of the box' state.
Precondition:	All hardware is integrated and available in lab, motor mounts fabricated
Procedures:	Gather all hardware necessary for spatial filter experiment. Begin timer. Set up spatial filter experiment on optical table. Set up motor mounts and couple motors to micrometers. Plug in all electronics. Make all necessary connections to host computer. Stop timer and record time.
Criteria to meet:	Experiment fully set up and usable in less than 2 hours.
Outcome:	The experiment takes less than 30 minutes to set up from an out of the box state by an inexperienced user.
Pass/Fail:	Pass
Follow up and Comments:	Set up takes considerable less time after the first set up.

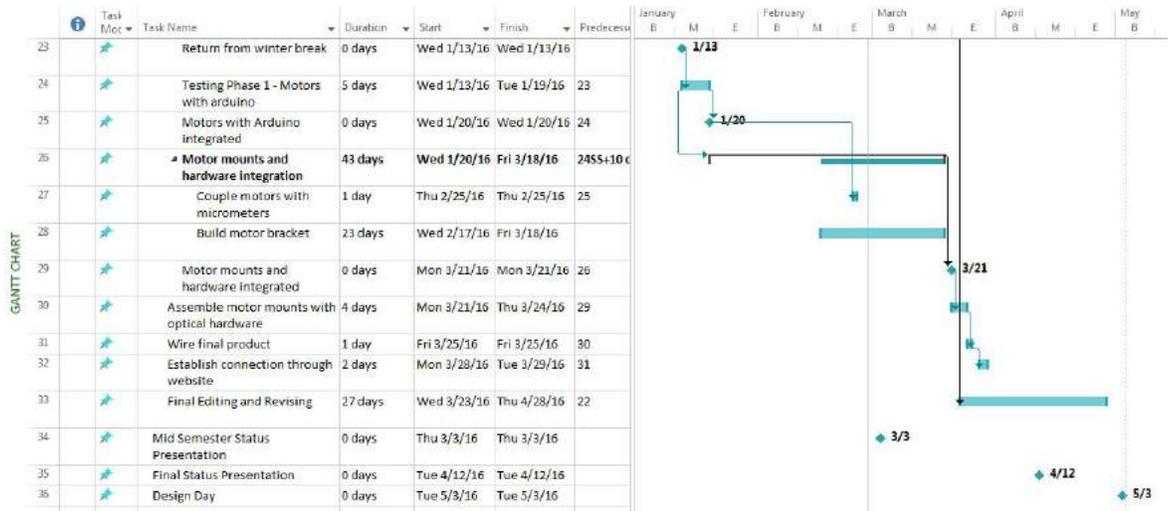
## 12.3 Budget and Suppliers

Beginning balance as of July 1, 2015:	<b>\$3,500.00</b>
Distributions Received - July 1, 2015 to present:	0.00
Transactions posted (revenue/expense) 7/1/15 to present:	2,613.39
Outstanding commitments (encumbrances) 7/1/15 to present:	0.00
<b>Current available balance:</b>	<b>886.61</b>

Date	Payee	Doc Ref	Distr. Amt.	Description			
<b>Distributions</b>							
	Invoiced 9/18/15;11/18/15						
<b>Total Distributions</b>				0.00			
Date	Object Code	Payee	Doc Ref	Rev/Exp	Encumb.	FRS	Description
11/13/2015	5490	Amazon	45137191969321	104.45		16-5	Arduino Uno Ultimate Starter Kit; 2 or 0.9deg Thin Nema 14 Stepper Motor Bipolar 0.5A 7Nm/10oz
11/13/2015	5490	Amazon	45562681971361	20.65		16-5	4mm x 8mm D18L25 CNC Motor Helical Shaft Coupler Beam Coupling Connect; 3D CAM 2 pos Flexible Couplings 5mm to 8mm
11/13/2015	5490	Ebay	451017191967943	24.31		16-5	US Ship Nema 8 Stepper Motor 1.8 deg Bipolar 0.6A
11/13/2015	5490	Ebay	45101701967943	6.09		16-5	DR CNC Stepper Motor Flexible Coupling Coupler 5x12mm
1/7/2016	5490	Thorlabs	2023810800001	208.64		16-7	mounting post v-clamp 3D1 1/4" (M6) Counterbore Adapter Ring for 8-32 (M4) Screws, 10 per Pack; RL2400 Dovetail Optical Rail, 24", Imperial, LMRR Lead Mount with Retaining Ring for 8" Optics, 8-32 Top; LMS2 Lead Mount with Retaining Ring for 8" Optics, 8-32 Top; TRS 812" Optical Post, SS, 8-32 Setcrew, 1/4"-20 Top, L = 3"; PHS 812" Post Holder, Spring-Loaded Hex-Locking Thumbscrew, L = 3; RCI Dovetail Rail Carrier, 1.00" x 1.00" (25.4 mm x 25.4 mm) 4; LA1050 N-BKT Plano-Convex Lens, Ø2", f = 100.0 mm, Uncoated 1; LC115 N-BKT Plano-Convex Lens, Ø1", f = 50.0 mm, Uncoated LAM61 N-BKT Plano-Convex Lens, Ø1", f = 250.0 mm., One PMMA Clamping Arm Included 2; P8 Ø1.5" Mounting Post, 1/4"-20 Top, L = 8" P81 Mounting Post Base, Ø2.48" x 0.40" Thick; SH25050 1/4"-20 Stainless Steel Cap Screw, 1/2" Long, Pack of 25 900 three-axis spatial filter
1/7/2016	5490	Thorlabs	47933271965707001	659.01		16-7	Generic 30 pc allen wrench, Villos raspberry pi 2 complete starter kit, 4x8 mm coupler, ditz extra large eyelet kit, leadleads multipurpose gooseneck 18-LED 4x8 mm coupler
1/11/2016	5490	Newport	2023810739001	932.36		16-7	
1/20/2016	5490	Amazon	2045756663001	107.74			
1/20/2016	5490	Amazon/luccell	11204575665001	6.95			
1/22/2016	5490	Ebay/PPaypal STEPPERONLINE	2041322530001	45.42		16-8	US Ship Nema 8 stepper motor 1.8deg bipolar 0.6A 5.7oz.
1/22/2016	5490	McMaster-Carr	2041322568001	18.67		16-8	8 hole angle bracket, 10-24 shoulder screw
1/27/2016	5490	Santa Rita Steel	2045756677001	24.54			Steel tubing; 3" wide Steel Strip + Labor
1/27/2016	5490	Ace Hardware	2045756676001	2.02			Socket Head Screws; Hex Head Screw; Hex Nut; Phillips Head Screw
1/28/2016	5490	Amazon	2043488437001	46.68		16-8	Electronic Express Hook up Wire Kit; Steel Pan Machine Screw; Led/holesalers 12 Volt 4 Amp 48 Watt ULAC Power Adapter Jack
1/28/2016	5490	Amazon	2043488442001	12.13		16-8	Qunqi A4988 Driver CNC Shield Expansion Board for Arduino V3 Engraver
3/9/2016	5490	DigiKey	712107216826001	39.25			CONN PIN 24-30AVG CRIMP 15GOLD
4/1/2016	5490	Amazon		169.07			Logitech HD Pro Webcam C320, 1080p Widescreen Video Calling and Recording, Arduino UNO R3 Board Module With DIP ATmega328P(Blue)
5/1/2016		Independent Contractor		50.00			Photography Shoot
5/1/2016		FedEx		74.81			Poster for Design Day
<b>Total Expenses / Encumbrances</b>				2,613.39	0.00		
				Expenses	Encumbrances		
<b>Payroll distributions - salaries detail</b>							
Date	ObjCode	Incumbent	PPE date	Rev/Exp	Encumb.	FRS	memo

## 12.4 Project Management

The project management was done using Microsoft Project, which allowed for the development of a schedule and Gantt chart. The schedule was used to coordinate the timing of milestones, deliverables, testing stages, and integration work. As with any project, the timeline did slip slightly and get pushed back; however, there was slack built into the schedule to accommodate this, which is why all deliverables were turned in on time. Below is a Gantt chart showing the spring semester schedule at the beginning of January. The takeaway from the below Gantt chart is that there was a hopeful deadline of March 29, 2016 to establish a website connection and have a fully functioning alpha release of the system. Unfortunately this optimistic self-imposed deadline was not met. This took away potential revision time to ameliorate the system.



Around mid-March the schedule was revised to reflect the current status of the project. A majority of the final system tests and integration and verification was pushed into April. Below is the revised Gantt chart.



In conclusion, the project management helped to keep the team on track to finish the final design by design day. Although several optimistic goals were not met, the requirements were met, and the final design was delivered on Design Day, which meets the ultimate goal of the year-long project.

## 12.5 Requirements Matrix

Requirement Number	Requirement	Measured Performance Metric (Results)
Functional Requirements		
F1	The system shall allow the user to remotely connect to the device online, through a user interface.	Final system includes login page, video feedback, and motor controls via webpage with

		fully developed user interface.
F2	The system shall allow the user to control the lab hardware remotely.	The webpage provides for controls that allow the user to move the spatial filter in the x, y and z-axes in both the forward and reverse directions with various step sizes.
F3	The system shall provide real-time visual feedback of the experiment, as it is being performed.	The system is implemented with two video cameras, which provide video feedback that the user views on the main webpage.
F4	The system shall be disassemblable.	The mechanical design is built using tube steel, nuts, and bolts, so that the entire system can be disassembled.
F5	The experiment instructions shall be posted online.	The webpage contains an instruction page as well as a help page to explain how to use the software.
F6	The system shall operate in real time.	The total delay from video feedback and control input is on average ~250ms, which provides for a near real-time system.
F7	Sarah Luciano shall assist in providing the computer programming for the online interface.	Sarah developed the software and code necessary to create the webpage and online user interface.
F8	The system shall have safety precautionary measures in place to prevent physical harm to humans in the lab or the equipment.	There are rollers on the motor mounts so that the hardware cannot be destroyed. Additionally, firmware end stops are implemented so that the motors cannot be turned too far in any given direction.

F9	The system shall have a controller that operates all of the lab hardware and equipment.	After the system is set up, all hardware (the spatial filter) is controlled through the software.
F10	The system shall give feedback messages to provide the user with a more complete understanding of the current system configuration.	The software provides help indicators, as well as a paragraph describing how to use the software.
F11	The system shall be operable 24 hours per day.	After the system is initialized, the software allows a user to control the hardware at any given time. Scheduling will be incorporated in future revisions of the project.
Mechanical Requirements		
M1	Mechanical setup shall fit within an area of 3 feet by 6 feet.	The mechanical setup measures 1.2 feet by 3.5 feet.
M2	System shall provide at least 2 degrees of freedom for the operator.	The final system provides control of three degrees of freedom in the spatial filter.
M3	System shall have end stops that will prevent any components from moving too far as well as prevent any permanent deformation to the other parts.	There are rollers on the motor mounts so that the hardware cannot be destroyed. Additionally, firmware end stops are implemented so that the motors cannot be turned too far in any given direction.
M4	Laser shall stay fixed within $\pm 0.5$ degree.	There was no measurable sag in the laser over a 48 hour time period.
M5	The system shall be mounted using a standard optical rail.	All optical components of the system mount to a 76mm optical rail.
M6	The system shall be able to mount to an optics table with 1/4-20 holes spaced on	Both the optical and fabricated mechanical subsystems mount to an

	a 1-inch grid.	optical table with 1/4-20 holes spaced on a 1-inch grid.
Software Requirements		
S1	The online interface shall have several windows showing the different camera views.	The software has two tabs that alter which camera view is presented in the webpage.
S2	The software shall be accessible from Internet Explorer, Safari, or Google Chrome.	The software is accessible from all 3 web browsers: Internet Explorer, Safari, or Google Chrome.
S3	The software shall run as an Internet application.	The software is a webpage hosted on a Raspberry Pi.
S4	The system shall include help indicators accessible during use, which detail the operation of the interface at a level suitable for first-time users.	The software provides help indicators, as well as a paragraph describing how to use the software and align a spatial filter.
Hardware Requirements		
H1	The system electronics shall ultimately be provided power from a 120V US power outlet.	The final system electronics all plug into a 120V US power outlet.
H2	The lab shall be instrumented with video cameras to provide the visual feedback.	The system contains two video cameras to provide a view of the laser output on the screen as well as the spatial filter pinhole.
H3	Motors shall be controlled using micro controllers.	The motors are controlled using an Arduino Uno microcontroller, a shield, and motor drivers.
H4	The system shall be lit by an independent light source.	The system has a clamp light to provide light for the camera view on the pinhole.

H5	The system shall be able to reset itself within 5 minutes after a user has finished an experiment, in order to prepare for the subsequent user.	The software allows for the system to reset to a default state in less than 5 minutes.
Performance Requirements		
P1	The system shall cost less than \$3,500.	The final system cost came to a total of \$2489.00.
P2	The time delay from remote input to visual feedback shall be less than 2000 milliseconds on average.	The time delay of the final system is on average 250ms.
P3	The video streaming shall be broadcast with a quality of at least 240p.	The final system implementation operates at 480p for two video streams at 10 frames per second each.
P4	The system shall take less than 2 hours to set up from an 'out of the box' state.	The system takes an inexperienced user less than 30 minutes to set up from an out of the box state.
P5	The system shall not require that the user need to purchase any hardware or software to operate the system.	The web page is fully contained on the Raspberry Pi, and the software is accessible from a web browser.

## 12.6 Source Code

Source Code 12.6.1 Python Script - serial\_to\_arduino.py

```
import serial
import time
import os
# serial library imported for serial connection to arduino
# time library imported so we can wait for serial connection to initialize with arduino

# Opens the serial connection with the arduino with baud rate of 9600
arduino = serial.Serial('/dev/ttyACM0', 9600)

# sizes are l, m, s
```

```

# directions are l, r, u, d, f, b
# 8 bits are used for commands
# we use a dictionary to retrieve the binary command
binaryCommands = { 'l,l': 0b00011011, 'l,r': 0b00010111, 'l,u': 0b00101011,
                  'l,d': 0b00100111, 'l,f': 0b00111011, 'l,b': 0b00110111,
                  'm,l': 0b00011010, 'm,r': 0b00010110, 'm,u': 0b00101010,
                  'm,d': 0b00100110, 'm,f': 0b00111010, 'm,b': 0b00110110,
                  's,l': 0b00011001, 's,r': 0b00010101, 's,u': 0b00101001,
                  's,d': 0b00100101, 's,f': 0b00111001, 's,b': 0b00110101,
                  'reset': 0b11111111, 'zero': 0b11001100 }

# Give the allotted amount of time for the arduino to settle the serial connection
time.sleep(2)

print('Running...')

# loop forever
while True:
    # try to open file if one exists with the name.
    time.sleep(.3)
    try:
        # open the file we will be reading from
        f = open('move.txt', 'r')
        data = f.readline() # this removes the newline and carriage return chars
        f.close()
        os.remove('move.txt')
        # now data should be size + direction
        x = str(binaryCommands[data])
        arduino.write(x)
    log = open('MoveLog.txt', 'a')
    log.write(data + '\n')
    log.close()
    except:
        pass

```

#### Source Code 12.6.2 Steams.sh

```

#!/bin/bash
# Begin Streams

echo "Begin Streams"

{

```

```
cd /home/pi/MonaServer/MonaServer
```

```
sudo ./MonaServer&
```

```
cd /home/pi/MonaServer2/MonaServer
```

```
sudo ./MonaServer &
```

```
avconv -f video4linux2 -s 640x360 -r 15 -fflags nobuffer -i /dev/video0 -f flv  
rtmp://128.196.207.136:1945/ -an &
```

```
avconv -f video4linux2 -s 640x360 -r 15 -fflags nobuffer -i /dev/video1 -f flv  
rtmp://128.196.207.136:1946/ -an &
```

```
cd /var/www/test/public
```

```
sudo python serial_to_arduino.py &
```

```
} &> /dev/null
```

```
echo "Streams are now running!"
```

In order to get the avconv program onto the Raspberry Pi the following command must be run

```
"sudo apt-get install libav-tools"
```

To get the USB webcams working on the Raspberry Pi the following command must be run

```
"sudo apt-get install fswebcam"
```

#### Source Code 12.6.4 Arduino Firmware

```
const int xStep = 2; // Define step pin for x-axis  
const int xDir = 5; // Define direction pin for x-axis  
const int yStep = 3; // Define step pin for y-axis  
const int yDir = 6; // Define direction pin for y-axis  
const int zStep = 4; // Define step pin for z-axis  
const int zDir = 7; // Define direction pin for z-axis  
const int enable = 8; // Define enable pin for stepper drivers  
const int maxSteps = 500; // Max number or steps in any direction from alignment  
  
int inputControl = B00000000; // Input control initialised to 0 (no input)  
int motorSel = B00000000; // Motor select initialised to 0 (no motor selected)  
int stepDir = B00000000; // Step direction initialised to 0 (no direction)
```

```

int moveSize    = B00000000; // Movement size initialised to 0 (no step size)
int motorMask   = B11110000; // Motor mask set to upper 4 bits (4, 5, 6, 7)
int dirMask     = B00001100; // Direction mask set to bit 3 and 2.
int sizeMask    = B00000011; // Size mask set to bit 0 and 1.
int xStepCount  = 0;         // Track steps in x-axis
int yStepCount  = 0;         // Track steps in y-axis
int zStepCount  = 0;         // Track steps in z-axis

boolean inputDone = false; // Input reading done set to false

// Setup for serial communication
void setup()
{
  Serial.begin(9600); // Set baud rate
  pinMode(xStep, OUTPUT); // Set xStep pin as output
  pinMode(xDir, OUTPUT); // Set xDir pin as output
  pinMode(yStep, OUTPUT); // Set yStep pin as output
  pinMode(yDir, OUTPUT); // Set yDir pin as output
  pinMode(zStep, OUTPUT); // Set zStep pin as output
  pinMode(zDir, OUTPUT); // Set zDir pin as output
  pinMode(enable, OUTPUT); // Set enable pin as output
}

void serialEvent()
{
  while (Serial.available() > 0) // While there is serial communication
  {
    // Read a new byte (8 bits) from the serial communication
    inputControl = Serial.parseInt();

    inputDone = true; // A byte has been read
  }
}

void loop()
{
  digitalWrite(enable, LOW);
  if(inputDone) // If a byte has been read
  {
    Serial.println(inputControl, BIN);

    if (inputControl == B11111111)
    {
      // Reset could take place here
      // Will most likely be done by a script sending controls to return to the origin
    }
    else if (inputControl == B11001100)
    {

```

```

// This is where the zero/origin/alignment point can be set
xStepCount = 0; // Sets origin in x-axis
yStepCount = 0; // Sets origin in y-axis
zStepCount = 0; // Sets origin in z-axis
}
else
{
// Mask all bits except upper 4. Bit 7&6 must be 0, bit 5&4 are motor select
motorSel = inputControl & motorMask;

// X-Axis Motor selected
if (motorSel == B00010000)
{
// Masks all bits except stepDir bits (2 and 3)
stepDir = (inputControl & dirMask);
// If step direction is positive (Clockwise)
if (stepDir == B00000100)
{
// Set x direction high for CW
digitalWrite(xDir, HIGH);
// Mask all bits except stepSize bits (0 & 1)
moveSize = inputControl & sizeMask;
// If movement size is small
if(moveSize == B00000001)
{
// Completes 1 step for small step
digitalWrite(xStep, HIGH); // Set xStep high
delay(10); // Hold high for 10 ms
digitalWrite(xStep, LOW); // Set xStep low
delay(10); // Hold low for 10 ms
xStepCount++; // Increase x-axis step count
}
// If movement size is medium
else if(moveSize == B00000010)
{
// Completes 4 steps for medium movement
for(int i = 0; i < 4; i++)
{
digitalWrite(xStep, HIGH); // Set xStep high
delay(10); // Hold high for 10 ms
digitalWrite(xStep, LOW); // Set xStep low
delay(10); // Hold low for 10 ms
xStepCount++; // Increase x-axis step count
}
}
// If movement size is large
else if(moveSize == B00000011)
{
// Completes 10 steps for large movement

```

```

for(int i = 0; i < 10; i++)
{
    digitalWrite(xStep, HIGH); // Set xStep high
    delay(10);                // Hold high for 10 ms
    digitalWrite(xStep, LOW); // Set xStep low
    delay(10);                // Hold low for 10 ms
    xStepCount++;            // Increase x-axis step count
}
}
}
// If step direction is negative (Counter-Clockwise)
else if (stepDir == B00001000)
{
    // Set x direction low for CCW
    digitalWrite(xDir, LOW);
    // Mask all bits except stepSize bits (0 & 1)
    moveSize = inputControl & sizeMask;
    // If movement size is small
    if(moveSize == B00000001)
    {
        // Completes 1 step for small step
        digitalWrite(xStep, HIGH); // Set xStep high
        delay(10);                // Hold high for 10 ms
        digitalWrite(xStep, LOW); // Set xStep low
        delay(10);                // Hold low for 10 ms
        xStepCount--;            // Decrease x-axis step count
    }
    // If movement size is medium
    else if(moveSize == B00000010)
    {
        // Completes 4 steps for medium movement
        for(int i = 0; i < 4; i++)
        {
            digitalWrite(xStep, HIGH); // Set xStep high
            delay(10);                // Hold high for 10 ms
            digitalWrite(xStep, LOW); // Set xStep low
            delay(10);                // Hold low for 10 ms
            xStepCount--;            // Decrease x-axis step count
        }
    }
    // If movement size is large
    else if(moveSize == B00000011)
    {
        // Completes 10 steps for large movement
        for(int i = 0; i < 10; i++)
        {
            digitalWrite(xStep, HIGH); // Set xStep high
            delay(10);                // Hold high for 10 ms
            digitalWrite(xStep, LOW); // Set xStep low

```

```

        delay(10);          // Hold low for 10 ms
        xStepCount--;      // Decrease x-axis step count
    }
}
}
}
// Y-Axis Motor selected
else if (motorSel == B00100000)
{
    // Masks all bits except stepDir bits (2 and 3)
    stepDir = (inputControl & dirMask);
    // If step direction is positive (Clockwise)
    if (stepDir == B00000100)
    {
        // Set y direction high for CW
        digitalWrite(yDir, HIGH);
        // Mask all bits except stepSize bits (0 & 1)
        moveSize = inputControl & sizeMask;
        // If movement size is small
        if(moveSize == B00000001)
        {
            // Completes 1 step for small step
            digitalWrite(yStep, HIGH); // Set yStep high
            delay(10);                // Hold high for 10 ms
            digitalWrite(yStep, LOW); // Set yStep low
            delay(10);                // Hold low for 10 ms
            yStepCount++;             // Increase y-axis step count
        }
        // If movement size is medium
        else if(moveSize == B00000010)
        {
            // Completes 4 steps for medium movement
            for(int i = 0; i < 4; i++)
            {
                digitalWrite(yStep, HIGH); // Set yStep high
                delay(10);                // Hold high for 10 ms
                digitalWrite(yStep, LOW); // Set yStep low
                delay(10);                // Hold low for 10 ms
                yStepCount++;             // Increase y-axis step count
            }
        }
        // If movement size is large
        else if(moveSize == B00000011)
        {
            // Completes 10 steps for large movement
            for(int i = 0; i < 10; i++)
            {
                digitalWrite(yStep, HIGH); // Set yStep high
                delay(10);                // Hold high for 10 ms
            }
        }
    }
}

```

```

        digitalWrite(yStep, LOW); // Set yStep low
        delay(10);                // Hold low for 10 ms
        yStepCount++;            // Increase y-axis step count
    }
}
}
// If step direction is negative (Counter-Clockwise)
else if (stepDir == B00001000)
{
    // Set y direction low for CCW
    digitalWrite(yDir, LOW);
    // Mask all bits except stepSize bits (0 & 1)
    moveSize = inputControl & sizeMask;
    // If movement size is small
    if(moveSize == B00000001)
    {
        // Completes 1 step for small step
        digitalWrite(yStep, HIGH); // Set yStep high
        delay(10);                // Hold high for 10 ms
        digitalWrite(yStep, LOW); // Set yStep low
        delay(10);                // Hold low for 10 ms
        yStepCount--;            // Decrease y-axis step count
    }
    // If movement size is medium
    else if(moveSize == B00000010)
    {
        // Completes 4 steps for medium movement
        for(int i = 0; i < 4; i++)
        {
            digitalWrite(yStep, HIGH); // Set yStep high
            delay(10);                // Hold high for 10 ms
            digitalWrite(yStep, LOW); // Set yStep low
            delay(10);                // Hold low for 10 ms
            yStepCount--;            // Decrease y-axis step count
        }
    }
    // If movement size is large
    else if(moveSize == B00000011)
    {
        // Completes 10 steps for large movement
        for(int i = 0; i < 10; i++)
        {
            digitalWrite(yStep, HIGH); // Set yStep high
            delay(10);                // Hold high for 10 ms
            digitalWrite(yStep, LOW); // Set yStep low
            delay(10);                // Hold low for 10 ms
            yStepCount--;            // Decrease y-axis step count
        }
    }
}
}

```

```

}
}
// Z-Axis Motor selected
else if (motorSel == B00110000)
{
// Masks all bits except stepDir bits (2 and 3)
stepDir = (inputControl & dirMask);
// If step direction is positive (Clockwise)
if (stepDir == B00000100)
{
// Set z direction high for CW
digitalWrite(zDir, HIGH);
// Mask all bits except stepSize bits (0 & 1)
moveSize = inputControl & sizeMask;
// If movement size is small
if(moveSize == B00000001)
{
// Completes 1 step for small step
digitalWrite(zStep, HIGH); // Set zStep high
delay(10); // Hold high for 10 ms
digitalWrite(zStep, LOW); // Set zStep low
delay(10); // Hold low for 10 ms
zStepCount++; // Increase z-axis step count
}
// If movement size is medium
else if(moveSize == B00000010)
{
// Completes 4 steps for medium movement
for(int i = 0; i < 4; i++)
{
digitalWrite(zStep, HIGH); // Set zStep high
delay(10); // Hold high for 10 ms
digitalWrite(zStep, LOW); // Set zStep low
delay(10); // Hold low for 10 ms
zStepCount++; // Increase z-axis step count
}
}
// If movement size is large
else if(moveSize == B00000011)
{
// Completes 10 steps for large movement
for(int i = 0; i < 10; i++)
{
digitalWrite(zStep, HIGH); // Set zStep high
delay(10); // Hold high for 10 ms
digitalWrite(zStep, LOW); // Set zStep low
delay(10); // Hold low for 10 ms
zStepCount++; // Increase z-axis step count
}
}
}
}

```

```

    }
  }
  // If step direction is negative (Counter-Clockwise)
  else if (stepDir == B00001000)
  {
    // Set z direction low for CCW
    digitalWrite(zDir, LOW);
    // Mask all bits except stepSize bits (0 & 1)
    moveSize = inputControl & sizeMask;
    // If movement size is small
    if(moveSize == B00000001)
    {
      // Completes 1 step for small step
      digitalWrite(zStep, HIGH); // Set zStep high
      delay(10);                // Hold high for 10 ms
      digitalWrite(zStep, LOW); // Set zStep low
      delay(10);                // Hold low for 10 ms
      zStepCount--;            // Decrease z-axis step count
    }
    // If movement size is medium
    else if(moveSize == B00000010)
    {
      // Completes 4 steps for medium movement
      for(int i = 0; i < 4; i++)
      {
        digitalWrite(zStep, HIGH); // Set zStep high
        delay(10);                // Hold high for 10 ms
        digitalWrite(zStep, LOW); // Set zStep low
        delay(10);                // Hold low for 10 ms
        zStepCount--;            // Decrease z-axis step count
      }
    }
    // If movement size is large
    else if(moveSize == B00000011)
    {
      // Completes 10 steps for large movement
      for(int i = 0; i < 10; i++)
      {
        digitalWrite(zStep, HIGH); // Set zStep high
        delay(10);                // Hold high for 10 ms
        digitalWrite(zStep, LOW); // Set zStep low
        delay(10);                // Hold low for 10 ms
        zStepCount--;            // Decrease z-axis step count
      }
    }
  }
}
PORTD = PORTD | B00000011;

```

```
inputControl = B00000000; // Set input control signal back to nothing
inputDone = false; // Set false so we're waiting for another input
}
}
```