

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 05-10-2013		2. REPORT TYPE		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Pointing and Jitter Control for the USNA Multi-Beam Combining System				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Patrick, Zachary Max				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Naval Academy Annapolis, MD 21402				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) Trident Scholar Report no. 419 (2013)	
12. DISTRIBUTION / AVAILABILITY STATEMENT This document has been approved for public release; its distribution is UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The ability to accurately combine energy from multiple low power beams on a specific target is critical to making directed energy weapons effective and practical. At the United States Naval Academy Directed Energy Research Center, a project is underway to develop a three beam combining system that employs fast steering mirrors (FSMs) for pointing and jitter control of individual beams. In the previous work, an adaptive H-infinity optimal controller has been developed to control a single beam using a beam position detector for feedback. This project will apply the H-infinity adaptive controller and other controllers to the multiple-beam combining system in a multiple-input, multiple-output feedback control environment. Instead of using a position detector, a high-speed video camera will be employed to provide centroid estimation and feedback for pointing control algorithms.					
15. SUBJECT TERMS Directed Energy, Laser Control, Beam Combining					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 90	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)

A TRIDENT SCHOLAR PROJECT REPORT

No. 419

Pointing and Jitter Control for the USNA Multi-Beam Combining System

by

Midshipman 1/C Zachary M. Patrick, USN



UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

This document has been approved for public
release and sale; its distribution is limited.

USNA 1531-2

U.S.N.A. --- Trident Scholar project report; no. 419 (2013)

Pointing and Jitter Control for the USNA Multi-Beam Combining System

by

Midshipman 1/C Zachary M. Patrick
United States Naval Academy
Annapolis, Maryland

(signature)

Certification of Adviser(s) Approval

Captain Joseph Watkins, USN
Permanent Military Professor, Mechanical Engineering Department

(signature)

(date)

Professor Tae Lim
Aerospace Engineering

(signature)

(date)

Professor Richard O'Brien
Weapons and Systems Engineering Department

(signature)

(date)

Acceptance for the Trident Scholar Committee

Professor Maria J. Schroeder
Associate Director of Midshipman Research

(signature)

(date)

USNA 1531-2

Abstract

The ability to accurately combine energy from multiple low power beams on a specific target is critical to making directed energy weapons effective and practical. At the United States Naval Academy Directed Energy Research Center, a project is underway to develop a three beam combining system that employs fast steering mirrors (FSMs) for pointing and jitter control of individual beams. In the previous work, an adaptive H-infinity optimal controller has been developed to control a single beam using a beam position detector for feedback. This project will apply the H-infinity adaptive controller and other controllers to the multiple-beam combining system in a multiple-input, multiple-output feedback control environment. Instead of using a position detector, a high-speed video camera will be employed to provide centroid estimation and feedback for pointing control algorithms.

Keywords

-Directed Energy

-Laser Control

-Beam combining

Background

1. Terminology

- Irradiance – Power per area, W/cm^2
- Fluence – Energy per area, J/cm^2
- Platform Jitter – The unintended motion of the centroid of a Directed Energy beam about an aim point caused by motion of the platform supporting the beam system
- H_∞ – Optimal controller design method
- MISO – Multiple Input Single Output
- MIMO – Multiple Input Multiple Output
- SISO – Single Input Single Output
- Beam Profiler – Device which captures and displays data regarding a beam's irradiance

2. Motivation

Directed energy weapons (DEW) can deliver energy at the speed of light, engage targets with unrivaled accuracy, and give the user control over a variable level of lethality. These weapons grant their users new options in a modern environment where long range missiles, small boats, and unmanned aerial vehicles (UAV) have all but replaced conventional ship to ship engagements. Though these are seemingly weapons of the future, they have become more realistic with the development of powerful solid state and free electron lasers. These directed energy systems are being developed by the United States Navy for implementation on surface

vessels and aircraft. Most recently, the Navy demonstrated the capabilities of the Laser Weapons System (LaWS) onboard the *USS Dewey*.¹

These weapons are not without issues that include several natural phenomena and technical limitations that severely affect the destructive ability of the laser beam. Atmospheric conditions such as turbulence and airborne particles, platform jitter, lack of feedback from the target, and current laser technology represent just a few of these limiting factors. This project is motivated by the need to address these three issues: correcting jitter, controlling the paths of multiple beams, and using a realistic source of feedback for practical applications. Feedback of an error signal is absolutely necessary for a DEW as the laser must not only hit the target, but must impact the target at precisely the same location for up to several seconds, depending on the target and the environment, in order to disable or destroy the target.

Jitter is defined as the “motion of the centroid of irradiance of a laser beam spot relative to a reference (aim point on the target).”² Jitter with respect to a platform can be pictured in the following manner. As a platform experiences vibration, the mounted laser experiences these same vibrations. This vibration at the source causes the spot of the beam to move around the desired point on the target. A simple example of this is holding a laser pointer and watching the effect a shaking hand has on the movement of the laser spot. If the spot cannot be held still, the power it delivers is spread across a larger area over a period of time and therefore its irradiance and effectiveness are reduced.

Current laser technology offers a number of options for directed energy weapon systems. A wide variety of lasers exist that have different sources of energy such as chemical exothermic

¹ Steele, Jeanette. Navy Unveils Laser Weapon for Ships. 08 April 2013. 2013.

² Perram, Glen P., et al. An Introduction to Laser Weapon Systems

reactions, solid-state materials, and the manipulation of electrons in a magnetic field known as a free electron laser.³ Solid-state lasers are the most plausible source for use by the US Navy in the near term because they offer “improved efficiency [and] reduction in weight and volume”⁴ as compared to chemical or free electron lasers. Solid state lasers, however, cannot currently provide high enough power levels to destroy a target using a single beam. On solid-state directed energy weapons, multiple solid state lasers are combined (either coherently or incoherently) so that the total power is significant enough to inflict damage on a target. For example, the Navy’s Laser Weapons System (LaWS) uses six 5.5 kW lasers combined incoherently to provide approximately 32 kW of power output.⁵ Multiple beam systems can deliver required power levels, but they also complicate the system. Each beam adds new error as it traverses a different optical path. Additionally, the problem of control increases significantly because it is difficult to distinguish one beam from another at the target in order to determine the contribution of each beam to the overall jitter and correct accordingly.

The final issue to be investigated is that of obtaining an error signal. The target does not instantly blow up when hit by a laser unlike when impacted by a bullet or missile. The energy required to incapacitate or destroy the target is built up over time (on the order of seconds) and thus the ability to maintain a specific aimpoint on the target is necessary for a directed energy system to work efficiently. An error signal is necessary to update the beam control system on how well the aimpoint is being maintained. This information is vital because it is used by the controller, a computer with programmed procedures to correct a known error, to help increase the effectiveness of the weapon. The controller alters the beam path via mirrors as well as

³ *Ibid.*

⁴ ONR. Solid State Laser. 10 Feb 2012. <http://www.onr.navy.mil/en/Media-Center/Fact-Sheets/Solid-State-Fiber-Laser.aspx>

⁵ Pawlak, Robert J., and Stephen R. Horman. Laser Weapons System (LaWS) Demonstrator

pivoting the entire weapon to track a moving target. In previous laboratory experiments, beam position data has been taken at the target and used as an error signal to close the loop of basic jitter control algorithms. This type of measurement is not possible in a real application because the target will be an enemy threat. Visual feedback from a camera or other form of analysis must be taken at the platform that hosts the directed energy weapon to be practical in the real system. This project uses an imaging system and algorithm that can be placed on the host platform. For the purposes of this investigation, the camera is placed in a stable location near the target as this research deals with the algorithms to control the beam vice the imaging system itself. Image analysis techniques will be employed to calculate the location of beam centroids within the image frame as well as to calculate the area of beam spots. These values will be used to generate an error signal for the controllers.

3. Related and Previous USNA Work

Lasers and their military applications have been researched heavily by the United States Government for many years. By the year 1980 \$2.5 billion dollars of government funding was invested in lasers technology.⁶ The 1983 Strategic Defense Initiative placed special emphasis on using directed energy weapons. Space based x-ray lasers and chemical lasers were researched for ballistic missile defense purposes.⁷ None of these designs were ever made into a weapon system, but the knowledge obtained during their research proved invaluable to later ventures. Examples of these later projects include: the Air Force's Airborne Laser Test Bed (ALTB) and Airborne Laser (ABL), the Navy's Laser Weapons System (LaWS), Maritime Laser

⁶ Perram, Glen P., et al. An Introduction to Laser Weapon Systems

⁷*Ibid.*

Demonstrator (MLD) and Free Electron Laser (FEL) and the Army's Tactical High Energy Laser program (THEL) to name a few.

The Navy's current prototype laser system based on solid state lasers (SSLs) is LaWS. LaWS utilizes six beams to produce 32 kW of power as it leaves the weapon's aperture.⁸ The purpose of the LaWS prototype is to address issues of multiple beam combining and control in a package that can be integrated with the current Phalanx gun mount and targeting system. The MLD program used a single 15 kW beam and was the first laser tested at sea onboard a ship. MLD addressed beam control during poor sea state, propagation in a maritime environment, and integration with ship's navigation and radar systems.⁹

At the United States Naval Academy (USNA) significant work has been done in the field of directed energy weapons. Several midshipmen have completed independent, Bowman or Trident Scholar research projects on the control and effects of directed energy weapons. MIDN Malinowski (class of 2011) developed a control algorithm for a known platform disturbance using position feedback from the target to adjust a single mirror and to reduce the jitter of a beam. ENS Moran, a past Trident Scholar, successfully created an adaptive controller that mitigated platform jitter in real time, adaptively tuning the controller based on sensor input of the platform vibration. Three other previous Trident Scholars, now Ensigns Roberts, Roush and Dunn developed systems capable of determining the platform jitter and predicting the path of the beam in real time while tracking an aim point, allowing jitter correction without reference to the target.

⁸Pawlak, Robert J., and Stephen R. Horman. Laser Weapons System (LaWS) Demonstrator

⁹ Northrop Grumman. Maritime Laser Demonstration. 10 Feb 2012.

http://www.as.northropgrumman.com/products/maritime_laser/assets/MLD_Datasheet.pdf

4. Theory

4.1. Platform Jitter

Platform jitter is important because it reduces the effectiveness of the DEW in disabling or destroying the target. The overall irradiance is decreased by platform jitter since it causes the beam spot to deviate from the intended aim point. Jitter is quantified as the angle of the deviated beam from its intended path. Figure 1 shows a basic laser setup and how jitter affects the position of the spot of the beam at the target. The top left diagram is the ideal case where there is no jitter. After jitter is experienced, the platform is displaced by θ radians as shown in the lower left diagram. The diagram on the right shows the effect of jitter at the target along the line of sight. The deviation from the desired aim point due to platform jitter is represented by equation 1. The tangent of jitter angle θ can be approximated as θ by using the small angle theorem.

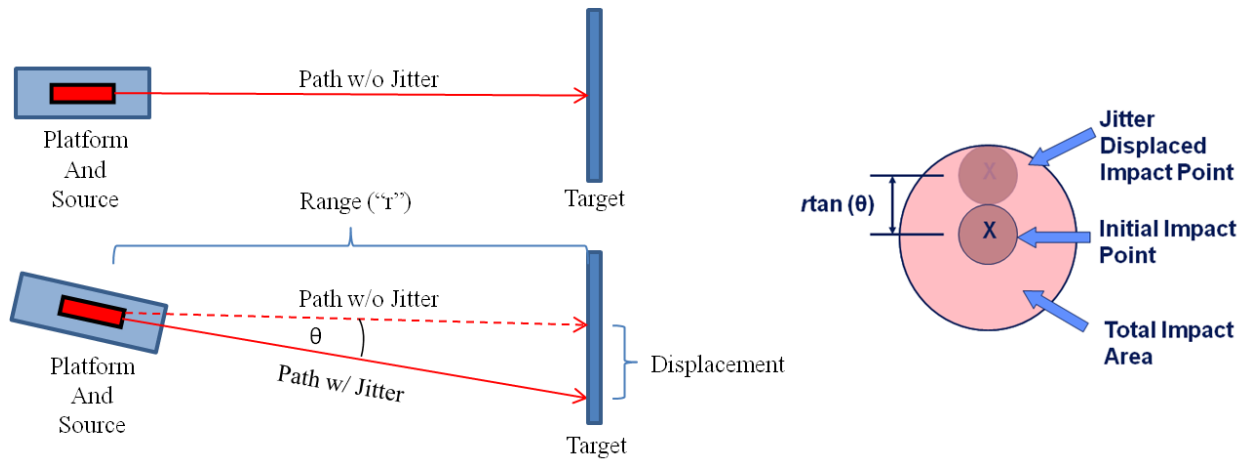


Figure 1: The Effect of Platform Jitter on Beam Spot Position¹⁰

¹⁰ Dunn, Nicholas. Development of an Isolated Platform Directed Energy Beam Control System. USNA.

$$\text{Displacement} = \text{Range} * \tan(\theta) \quad (1)$$

The plot in Figure 2 shows an example of the effects of jitter on a multi-beam system's irradiance on a 2000 μm radius circular target. Between one and two seconds the beams experience uncontrolled jitter that causes their aim points to move off center. The resulting losses in irradiance are plotted both individually for each beam and as a combined irradiance value in black. The jitter is controlled at 2.04 seconds to attenuate these effects and increase irradiance on target. The loss of irradiance due to jitter is difficult and costly to overcome by increases in total power at this time due to the limitations of available laser technology. As a result, jitter must be attenuated to make directed energy weapons more effective onboard the Navy's vessels.

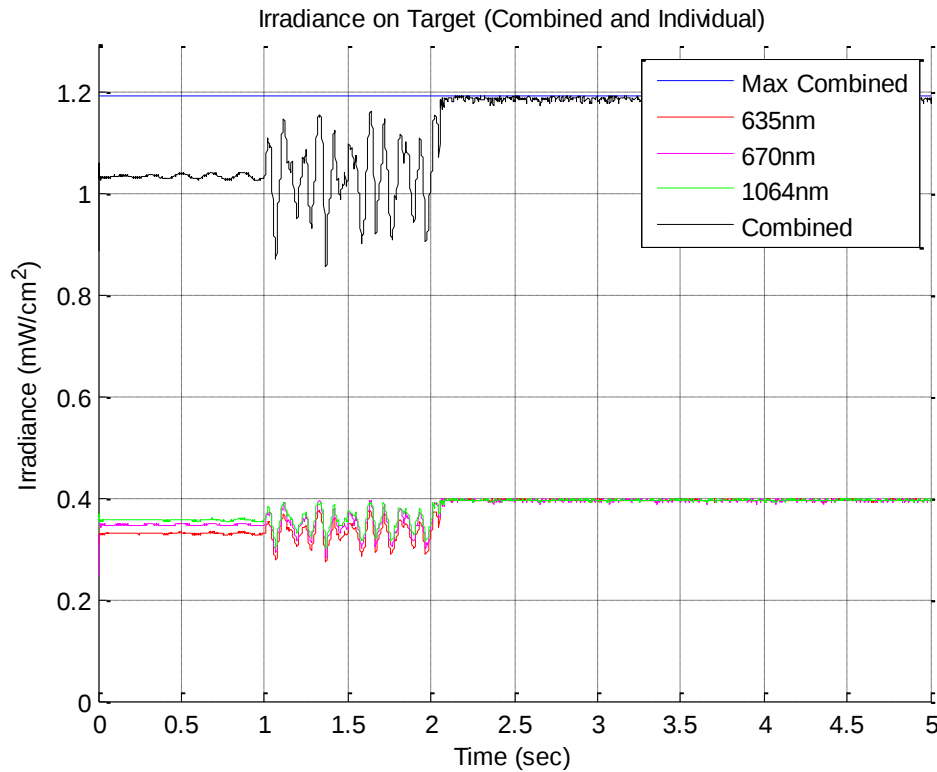


Figure 2: Plot of Multi-Beam System Irradiance during Jitter Control

4.2. Control Theory

A controller is an algorithm (implemented in computer code) that uses measurements of a system's response (feedback) to make that system respond in a desired manner. In this particular case, the system is the experimental apparatus shown in the equipment section. The controller uses the area measurement from the Hamamatsu camera to adjust one fast steering mirror for each beam in an attempt to reduce the area to a minimum value. This minimum value corresponds to the area achieved when the beams are perfectly overlapping without jitter. This control problem is known as disturbance rejection and entails reducing the sensitivity of the system output (beam spot area) to the disturbance (platform motion). A block diagram of the disturbance rejection model, Figure 3, breaks down each component of the system and shows where the disturbances come into effect. There are several quantities represented as functions of "s" (i.e. $R(s)$). These can be thought of as a series of inputs and outputs as one flows along the diagram. For example $E(s)$ is an input to the controller which produces the output $U(s)$. However, this output $U(s)$ becomes the input for the plant block. The (s) term in these quantities means that the function is shown in the frequency domain (Laplace transform) as opposed to having a function that is based on time. The frequency domain allows for complex differential equations to be modeled as simpler algebraic functions that can then be analyzed in the block format shown below. The input of the block diagram, $R(s)$, is what the user would like to see as the final output of the system (the minimum possible area in this case). A summing junction computes the difference between the desired output and the current output $Y(s)$ (measured beam spot area) including the effect of the disturbance (platform jitter). This difference is the error signal $E(s)$ which is sent to the controller (computer algorithm) to create a control input $U(s)$

(voltage). This signal is sent to the actuator (fast steering mirror) that causes the system output to change. This process repeats on the order of 1000 times each second.

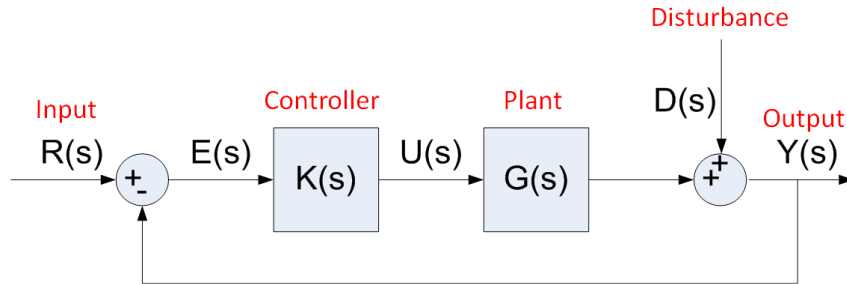


Figure 3: Disturbance Rejection Block Diagram¹¹

The Bode sensitivity function in equation 2 relates the output of the system to the disturbance. The system can be made less sensitive to the disturbance by making the magnitude of the function as small as possible in the frequency range of interest. For example, in this project platform disturbances are identified as specific frequencies of vibration. By reducing the system sensitivity at these frequencies, jitter is reduced and irradiance is increased. The sensitivity is reduced by increasing the action of the controller, $K(s)$, at these frequencies.

$$Sensitivity = \frac{1}{1 + K(s)G(s)} \quad (2)$$

¹¹ O'Brien, Richard T., Jr., R. Joseph Watkins, *Adaptive H-infinity Vibration Control*

4.3. H_∞ Theory

H_∞ controllers are used because they guarantee robustness. Robustness means that the controller maintains stability (the most basic requirement for successful operation) where there is uncertainty in the system model (the actual system responds differently than the system model). Additionally, H_∞ controllers designed at the Naval Academy are capable of identifying and counteracting disturbance frequencies even if they are unknown to the controller initially. H_∞ methods utilize a weighting transfer function (equation 3) to implement desired performance requirements on the system. The designer of the controller determines a pair of damping ratios (ζ_1 and ζ_2) to attenuate a specific disturbance frequency ω_n . This weight in turn penalizes the frequency by increasing its importance to the controller. The controller distinguishes these as the most important frequencies to attenuate. In short, the weighting transfer function lets the designer identify a disturbance frequency and make it appear important so the controller minimizes its effect.

$$w(s) = \frac{s^2 + 2\zeta_1\omega_n s + \omega_n^2}{s^2 + 2\zeta_2\omega_n s + \omega_n^2} \quad (3)$$

The result of H_∞ control is the reduction of the energy transmitted by frequencies at the output. Figure 4 shows the power spectral density (power/unit of frequency) vs. disturbance frequency from previous work at USNA. H_∞ control reduces the power spectral density at the disturbance frequencies as seen in this figure. For example, a 10Hz signal is reduced from 40 dB/Hz to approximately 15 dB/Hz. This results in a significant reduction of the effect of platform jitter due to that disturbance frequency. H_∞ control will be implemented in this project

to determine if the resulting control is worthwhile given the increased computational power required to run the complex H_∞ models.

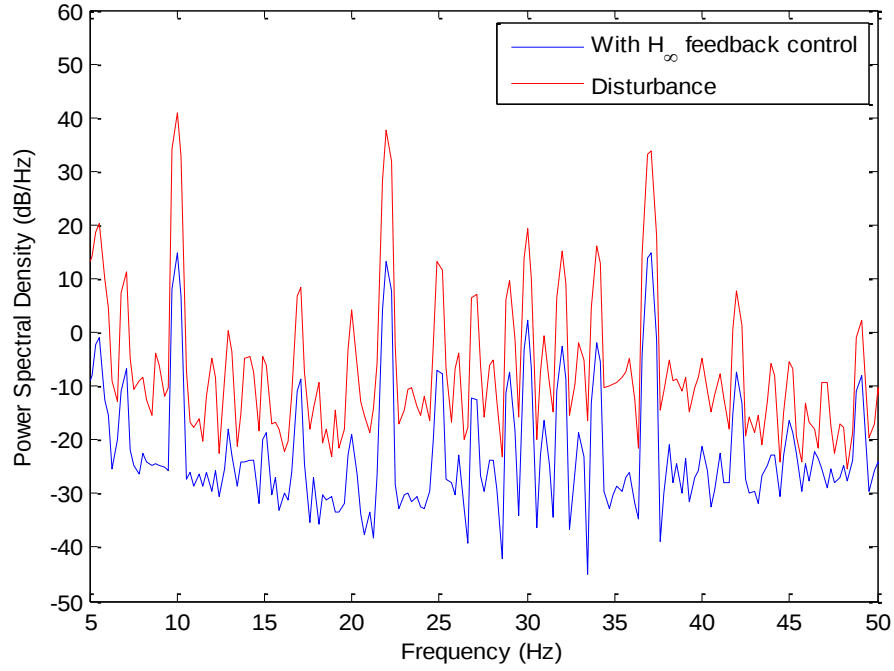


Figure 4: Disturbance Attenuation with H_∞ ¹²

¹² O'Brien, Richard T, and R.J. Watkins. *H ∞ Jitter Control for a Platform-Mounted Laser*

Table of Contents

Methods.....	14
SISO PI Control	14
SISO H-Infinity Control	20
PI Control One Beam/Camera Feedback	21
Centering of Three Beam Using Pointing Control Algorithm	23
Sequential Area Control after Centering of Beams via Pointing Algorithm	29
Combined Command Jitter Control	34
Results and Analysis	35
SISO PI Control	34
SISO H-Infinity	45
Target Area vs. Image Area Experiment	46
Pointing Control Algorithm	47
Sequential Area Control	49
Conclusions and Recommendations	53
Equipment	56
Appendices	67
Works Cited.....	90

Methods

1. Single-Input Single-Output (SISO) PI Control

1.1. *Purpose:*

The two main objectives of this portion of the project were: automate a calibration process for the sensors, and establish optimal performance benchmarks for the system using classical PI controllers.

1.2. *Physical Lab Setup:*

1.2.1. Source/Target Table:

The source and target tables contained the same arrangement described in the Lab Setup Overview of the Equipment Section. The reflection PSMs provided the feedback for the FSMs. The band-pass filters allowed only one wavelength to pass to each PSM sensor and therefore the individual position of each beam on the target could be measured with calibration.

1.3. *SIMULINK Model*

1.3.1. First Level:

The first layer of the SIMULINK model *MMM3* (Multiple Mirror Model 3), Figure 5, contained 4 blocks: inputs, jitter control, switching, and output. The input and output blocks were typical of all models used in the USNA Directed Energy Lab. The only changes

made to these blocks were the routing of signals for the third FSM and the addition of calibration blocks for the PSMs and FSMs.

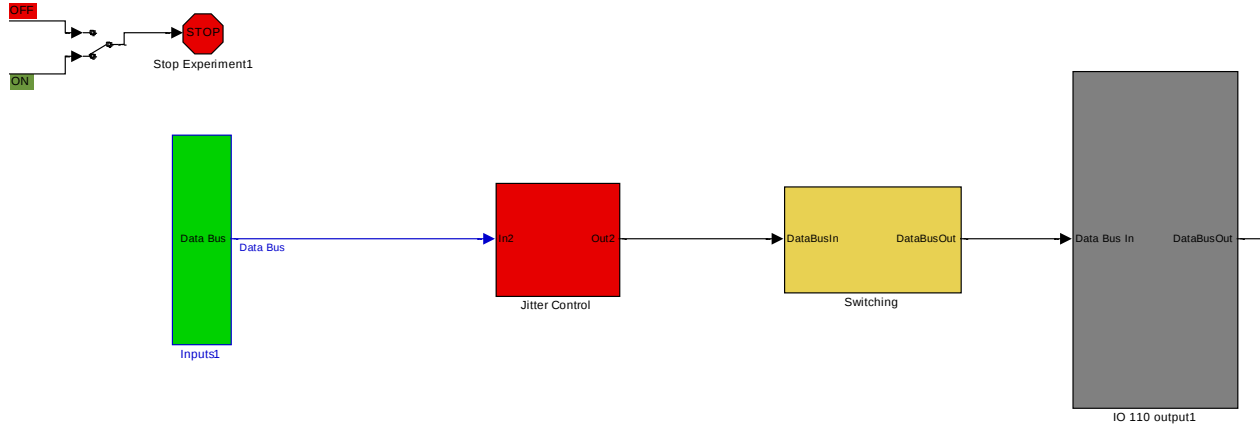


Figure 5: MMM3 First Layer

1.3.2. Jitter Control:

The jitter control block contained all of the controllers for this system and was masked. The mask allowed the user to choose which PSM provided feedback to each mirror. Within the jitter control block were three PI controllers used for jitter control of the beams, Figure 6. Proportional and integral tuning values were defined within a mask on each block. These values were defined within the run script and their variable names put in the mask. Each controller block contained controllers for the x and y axes, Figure 7.

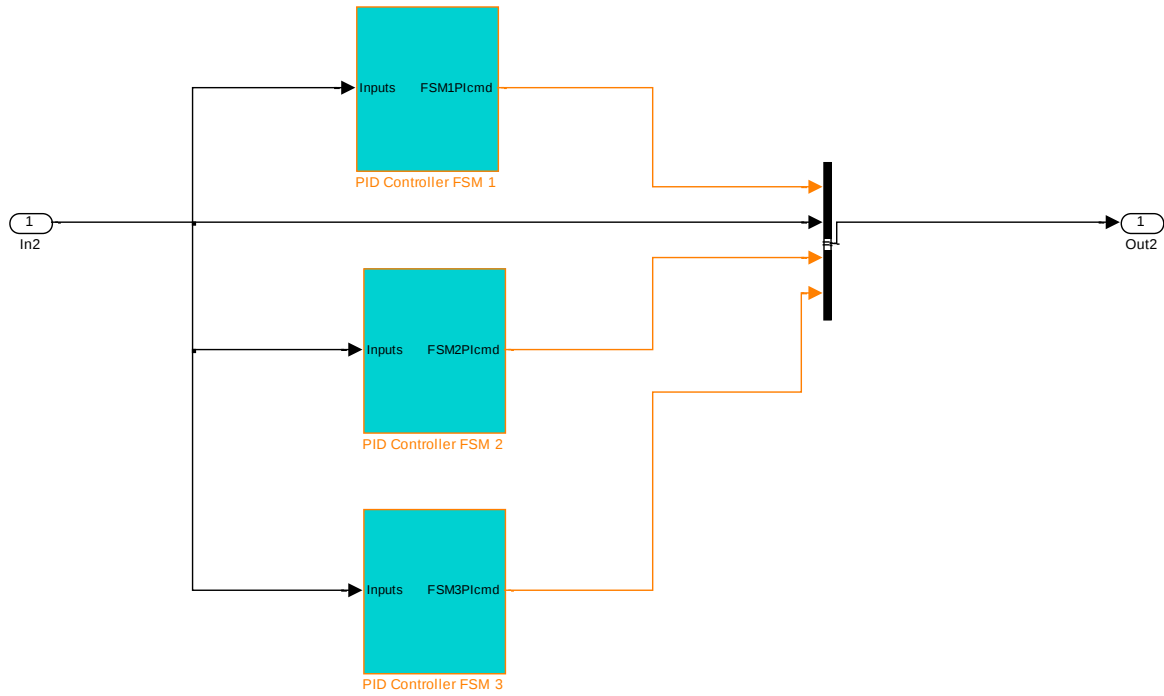


Figure 6: Jitter Control Block

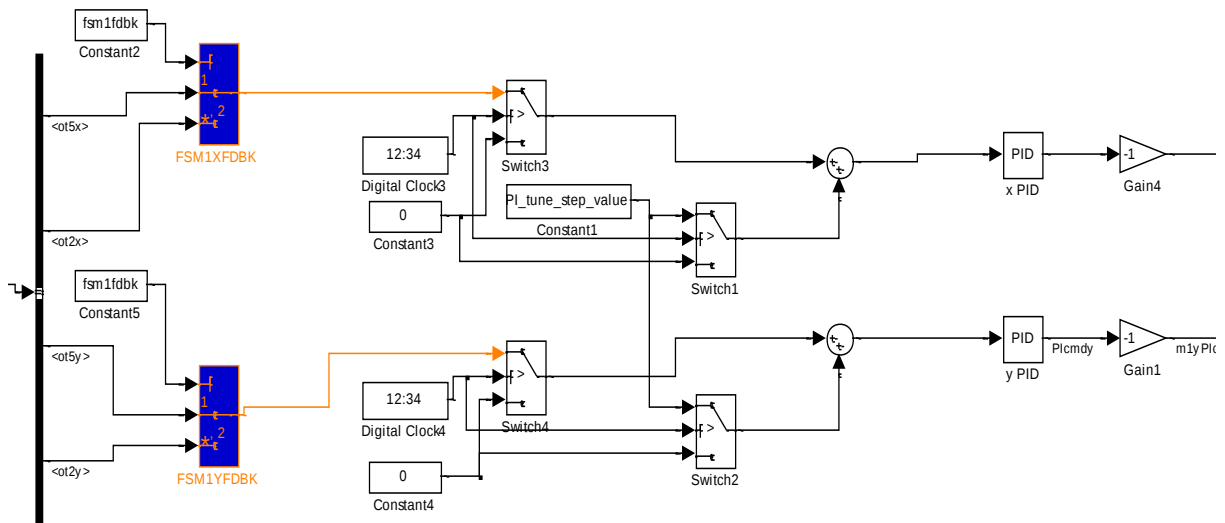


Figure 7: PID Controller Block

1.3.3. Switching:

The switching block was masked to provide a variety of control options, Figure 8. Control for each axis of each FSM could either be turned off, set to PI control, or set to a test sinusoidal signal. If the control was turned off, a constant value could be entered into the field below as an input (default was zero).

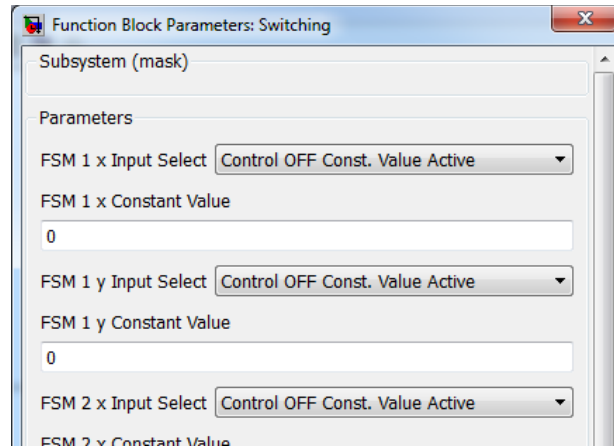


Figure 8: Switching Block Mask

1.4. **Calibration:**

The objective of this test was to use the feedback (reflection) PSMs to center each beam on the target PSM. However, the feedback PSM positions did not match the target position exactly. This meant that a beam might travel slightly further to the feedback PSM and might be slightly off center. In order to account for this, the zero of the feedback PSM had to be shifted to match that of the target PSM. Additionally, a longer optical path meant the displacement from platform motion (rotation) was greater at the feedback PSM than at the target. Thus, the data had to be scaled to have the reflection readings match the target. By having the feedback PSM data represent the target, the user could determine the individual positions of beams on the target

despite ambiguity on the target PSM. The three step process for achieving this was within the *MMM3* calibration script file: gross centering of beams, precise centering of beams and shift of feedback PSM zeroes, and movement of beams to scale feedback PSM data. The gross centering of the beams allowed the user to turn on each beam and ensure that it fell on the surface of its corresponding feedback PSM as well as the target PSM. Next, each beam was turned on individually and the corresponding FSM centered it on the target using PI control with feedback from the target PSM. At the end of this test, the average mirror command and feedback PSM readings were stored as calibration values. The corresponding average mirror command was stored as the new calibration bias value in the output block for the mirror in order to ensure the beams were centered on the target at the start of a test. The average feedback PSM value was subtracted in the input block to correct for the measurement bias errors so that the zero of the feedback PSM would match the zero of the target PSM. Figure 9 shows where these values were stored within the input and output model blocks. The scale values were used to modify the data outside of the model. However, they could be implemented in the model by simply adding gain blocks after the shifting blocks.

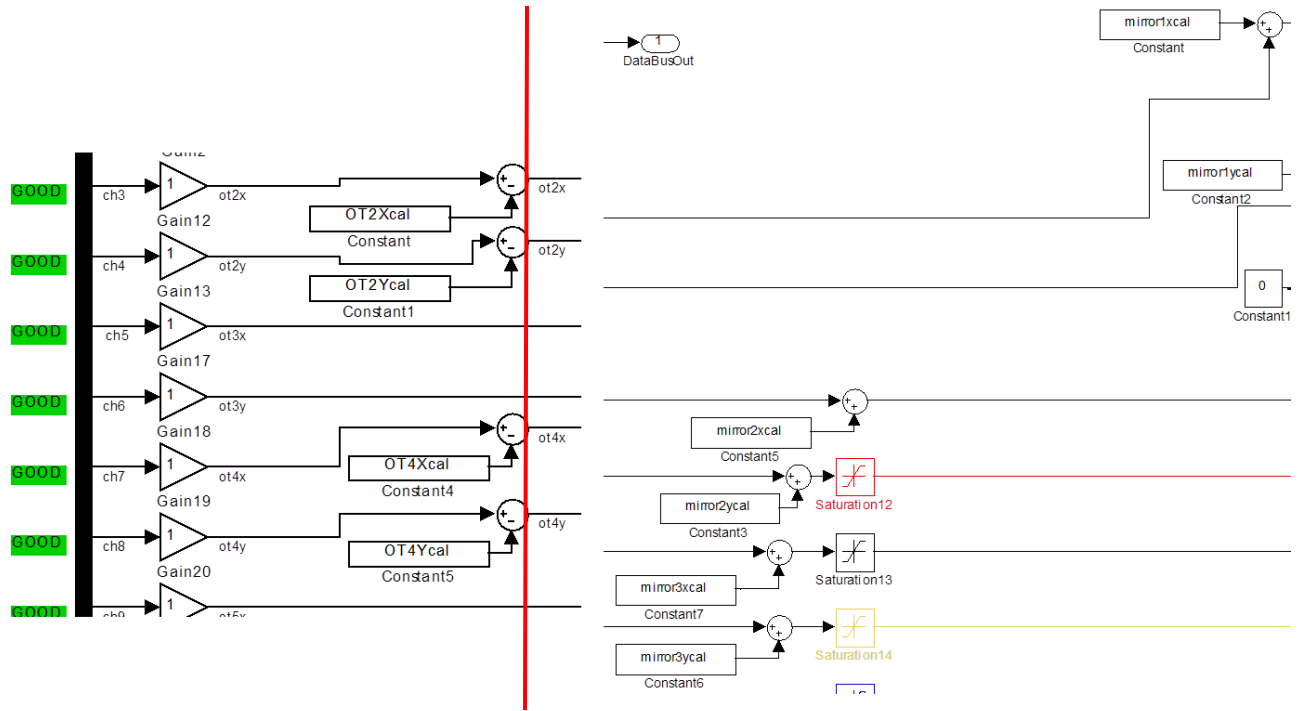


Figure 9: Input (left) and Output (right) Calibration Blocks

1.5. *Experimental Procedure:*

The calibration routine was run before each set of experiments and every 15 minutes thereafter to maintain accuracy due to possible changes in the experimental setup (e.g. the floating platform isolators leaked some nitrogen). A run consisted of controlling one to three beams at a time and then analyzing the jitter angles and power spectral density plots based on each beam's position data from the feedback PSMs. Runs were completed at a sample rate of 1 kHz. The *Run_MMM3* m-file was used for runs after initial calibration was completed. Note that all m-files generated for this investigation are reproduced in the appendix.

2. SISO H_∞ Control

2.1. *Purpose:*

The purpose of this test was to determine the ability of the system to run two and three beam systems controlled with H_∞ controllers originally designed in a previous Trident's work and modified for the multi-beam project. Feedback control was conducted using three independent SISO H_∞ control loops.

2.2. *SIMULINK Model*

2.2.1. *Jitter Control Block:*

The jitter control system for this experiment contained three of the SISO H_∞ controllers, one for each beam. Each of the blocks received feedback from its respective PSM. Each beam traversed an independent path from the laser/collimator, the corresponding fixed flat mirror and FSM, to the reflection PSM. Therefore, the control output from each block was routed to the corresponding FSM controlling the beam measured by the feedback PSM.

2.3. *Experimental Procedure:*

Calibration used the *MMM3* model and the SISO PI Control. After these calibration values were obtained, initialization programs were run to determine the necessary parameters for the H_∞ controllers.¹³ *ACC12CCEXPVerDual.m* was the supervisory program which initialized and ran all the required calculations. The sample time for the experiment was

¹³ Moran, Shane. An Adaptive H-Infinity Algorithm for Jitter Control and Target Tracking in a Directed Energy Weapon. USNA, 2012.

defined by this program. The maximum sampling frequency for simultaneously controlling three beams was 500Hz while two beams could be controlled at 1kHz due to CPU computational speed limitations.

3. **PI Control of One Beam via Camera Feedback**

3.1. ***Purpose:***

The purpose of this was to test the camera setup and determine the sensitivity of the system when using a video camera.

3.2. ***Physical Lab Setup:***

3.2.1. *Source/Target Table:*

The source and target tables contained the same arrangement described in the Lab Setup Overview of the Equipment Section. However, the beam splitters were removed and the target PSM was exchanged for a plain piece of paper. The Hamamatsu camera was installed on the target table so that the target paper was in its field of view. A small zoom lens was used to achieve the best possible resolution of the target area. See the appendix for more information regarding camera setup and use.

3.3. ***SIMULINK Model***

3.3.1. *First Level:*

The first layer of the SIMULINK model *PICamOneCent*, Figure 10, contained the typical blocks used in this project. The camera feed was imported and the centroid position obtained from the MATLAB **Blob Analysis Block** used as feedback for the PI controller contained within the jitter control block. The image was analyzed as described in the camera setup section of the appendix. Additionally, the feedback was further modified to allow for positive and negative position data. The **Blob Analysis Block** used the original image's origin at the top left corner. The x and y axes started at the origin and were defined only in the positive direction (right and down). The controller algorithm was programmed to point the beam at the center of the image. Gain blocks were used to deal with a discrepancy in the y-axis orientation as the camera's y-axis was inverted compared to that of the FSM. The switching block in this model allowed the user to assign which FSM, and therefore which beam, would be controlled using the PI controller. The camera's response to different wavelengths was determined using this method.

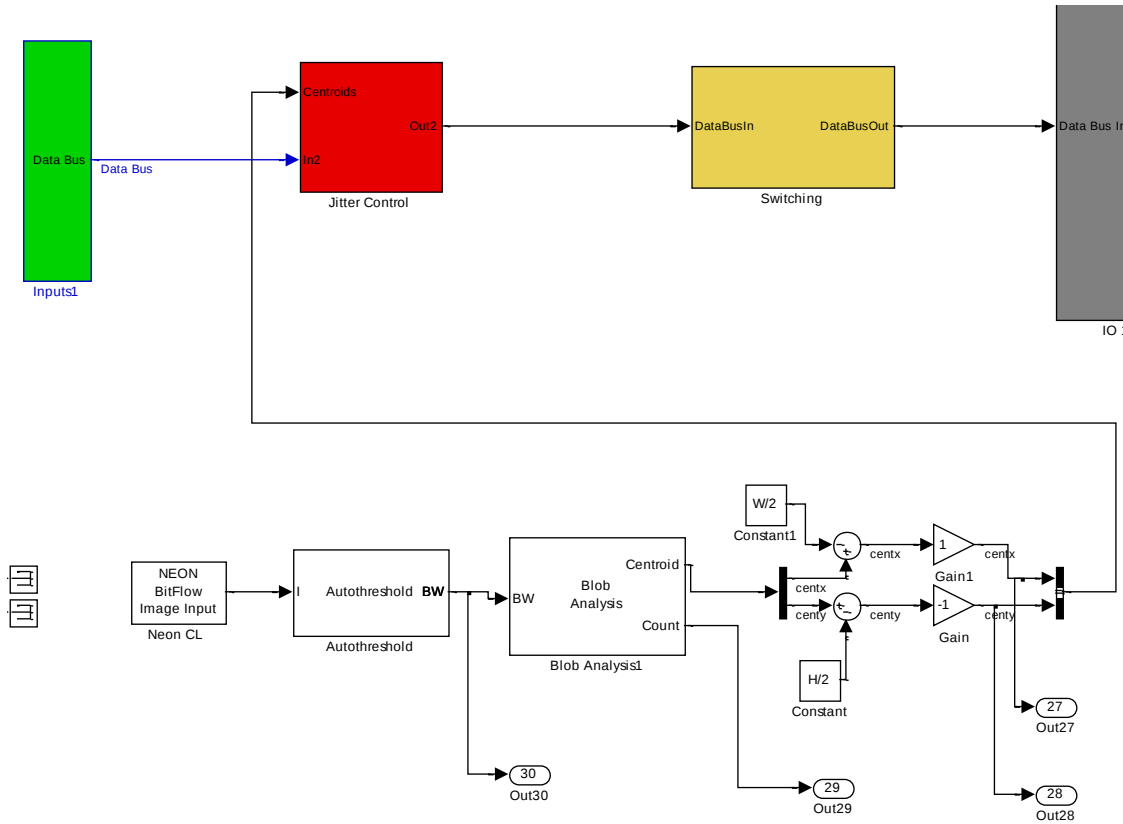


Figure 10: PI Camera Feedback One Centroid Model

3.4. *Experimental Procedure:*

The program for running the single beam algorithm was *run_PICamOneCent*. The model ran at a sample rate of 1kHz.

4. Centering of Three Separated Beams Using Pointing Control Algorithm

4.1. *Purpose:*

This subroutine's purpose was to take three distinct beam spots and center them on an aim point. It was intended to address a situation where the beams are initially pointed downrange and have moderate error in aim (pointing), (see Figure 11). It is important to note

that while the beams are different wavelengths (for the purpose of determining the actual location for this investigation), the camera as currently configured cannot distinguish one beam from another. The algorithm developed was intended to deal with this ambiguous situation, as the actual DEW system will most likely be composed of beams of the same or nearly the same wavelength.

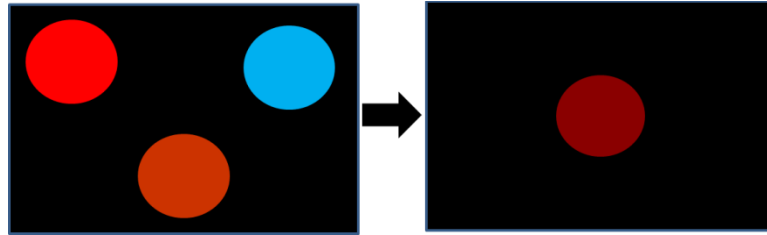


Figure 11: (Simulation) Initial Orientation of Beams (Left) and Centered Overlapping Beams (Right)

4.2. *SIMULINK Model:*

4.2.1. *First Level:*

The first level of the *ThreeCentroidPointingandAreaControl* SIMULINK model contained the blocks for control using camera feedback as well as a MATLAB function block which contained the pointing control algorithm. Several more of the available outputs of the **Blob Analysis Block** were utilized in this model. The area, major axis, and orientation outputs were part of the area control algorithms explained in later sections. Only the additional MATLAB function block and output block were pertinent to the pointing control algorithm which centered the beam.

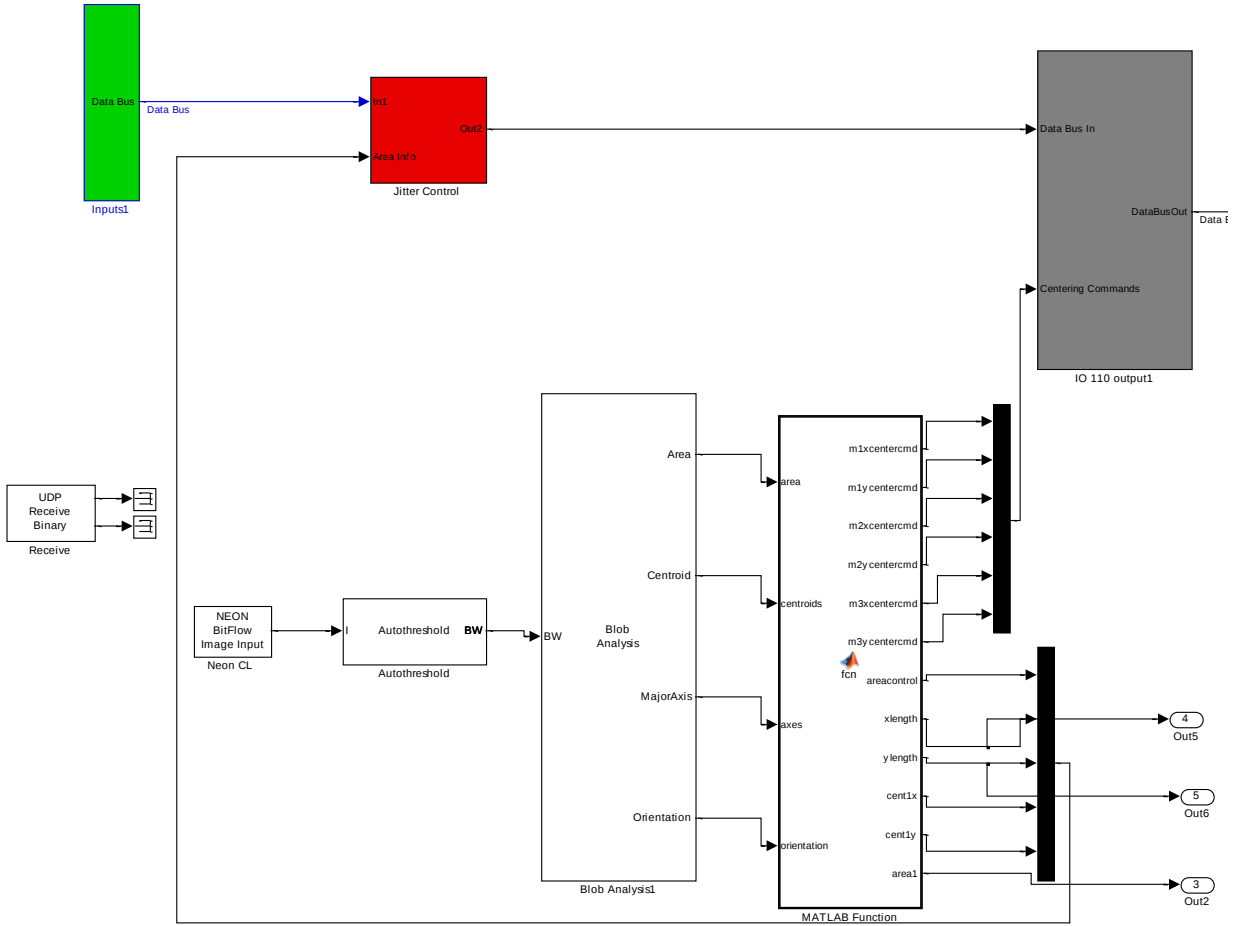


Figure 12: Pointing Control Model

4.2.2. MATLAB Function Block/Pointing Algorithm:

The MATLAB function block contained the entire algorithm for centering the beams (see the appendix for the fully commented code). The algorithm begins by determining the initial position of all three beam spots. These positions are taken with the origin at the center of the image and are averaged over a period of 10 samples. The averaging of initial positions is important in order to counteract the effects of small changes in spot position due to jitter. More samples meant a better average position, but also slowed the centering process. Next, the algorithm's identification phases deals with the ambiguous situation of not knowing which beam

spot corresponds to which FSM. A command was given to the first FSM in the positive direction of the x-axis in order to cause course movement in a corresponding beam spot. By checking the new average position of each spot the algorithm could determine which beam had been moved by FSM 1. Next the algorithm would use FSM 1 to center that beam spot using the linear relationship to be described later. FSM 2 was then given a large command in the direction of the y-axis and its corresponding spot was similarly identified and centered. The final spot left was assigned to FSM 3 and centered.

4.2.3. Output Block:

The output block in the SIMULINK model was changed slightly to accommodate the new centering commands that were an output of the pointing control algorithm. The centering commands were run as an input into the block and then distributed to each corresponding mirror axis, Figure 13.

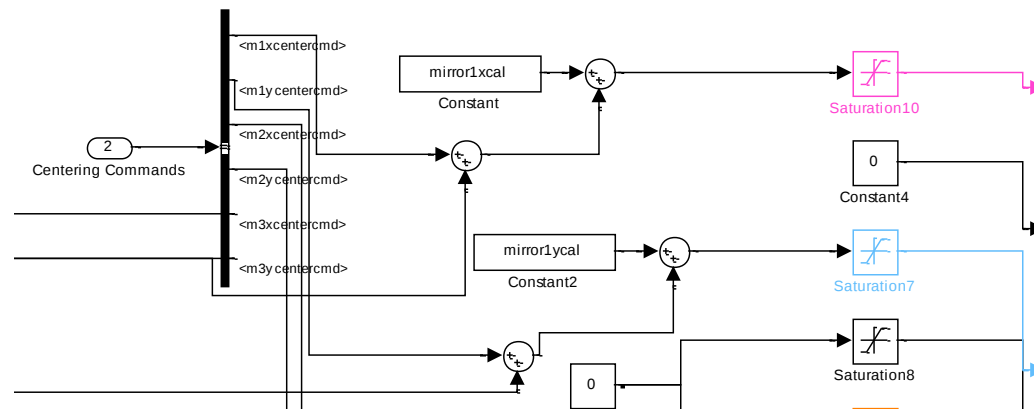


Figure 13: Output Block Centering Command Signal Routing

4.3. *Linear Relationship of FSM Movement to Beam Spot Movement:*

A linear relationship was used in order to center the beams as described in the pointing control algorithm section on page 25. This relationship determined how many pixels the beam centroid would move in the camera's image given a specific mirror command. First, it was determined that there was no significant cross coupling between the x and y axes. This was done by giving a command to only the x or y axis and monitoring the change in position for evidence of x axis commands causing y axis position change and vice versa.

After collecting data sets on all mirrors it was determined that the x and y axes on all but FSM 3, Figure 14, could be related to change in beam position by 850 pixels per 1 volt of command, Figure 15. FSM 3's x-axis subscribed to this relationship, but its y- axis was significantly more sensitive. It was determined that 1 volt of commanded change in FSM 3's y axis would result in a change of beam spot position of 2000 pixels.

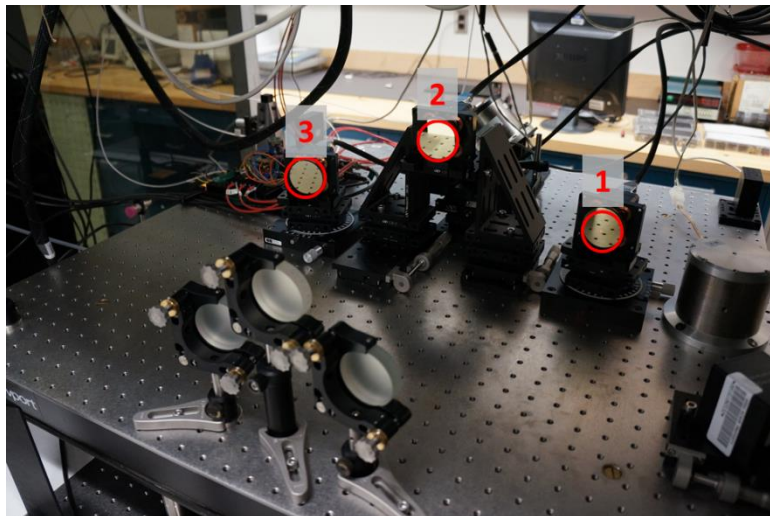


Figure 14: FSM Positions and Numbers

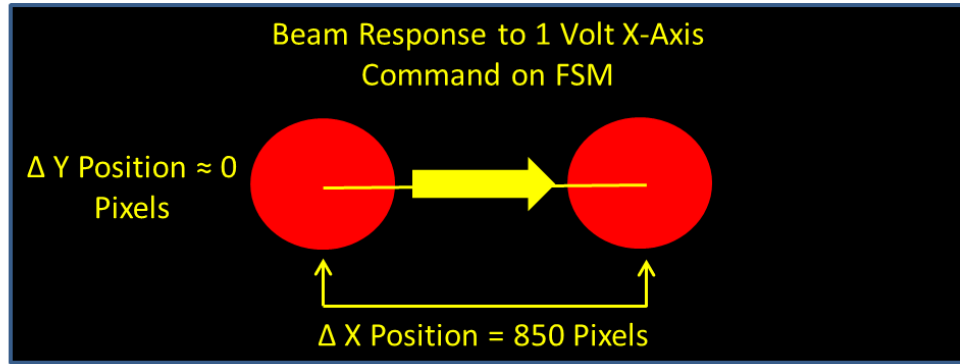


Figure 15: FSM Command and Beam Position Change Relationship

4.4. *Experimental Procedure:*

The procedure for this model did not require additional calibration unless the user desired to have diagnostic data on each beam from the beam splitter/reflection PSM setup. If this was desired the *MMM3* model and corresponding calibration file had to be used first. The target paper that originally replaced the target PSM had to be attached to the side of the PSM so that both could be utilized. First, the *MMM3* calibration was done. Note that, unlike before, the beams were now initially separated by use of the flat mirrors. This meant that the calibration process was very sensitive since the beams were on the extreme edges of the target PSM's sensor. Once the calibration values had been determined the PSM stage would be moved along its x-axis so that the beam spots fell on the piece of paper and not the PSM. Ideally, this piece of paper would be mounted so that it was at approximately the same position as the PSM sensor once moved into position. The target paper had to be used because the PSM sensor absorbed much of the laser light, making it impossible for the camera to see any beam spots. Once calibration was complete, the run file for *ThreeCentroidPointingandAreaControl* could be used to conduct runs. Occasionally, the **Blob Analysis Block** would experience failures and result in inadequate centering of the beams. If this occurred the run had to be restarted. These issues

were largely fixed by adjusting the individual beam power to ensure that the beam brightness was uniform across all three spots as seen by the camera. An effort was made to develop an image analysis block specific to this project, (see appendix Intensity Centroid Detection Script). However, the resulting program called too many extrinsic functions and could not be successfully implemented via a MATLAB *s-function*. This block was successful in dealing with the brightness/thresholding issue mentioned above during simulations, but could not be used on the real time target computer via s- function.

5. Sequential Area Control after Centering of Beams via Pointing Algorithm

5.1. *Purpose:*

This portion of the *ThreeCentroidPointingandAreaControl* model was designed to reduce the area of the spot formed by the combined beams. Similar to the pointing algorithm, this control had to deal with ambiguity because there were no distinguishing factors between beams.

5.2. *SIMULINK Model:*

5.2.1. MATLAB Function Block Activation of Area Control and Rendering of Area Data:

The MATLAB function block on the first level of the model, Figure 12, had an additional role after it completed the centering algorithm. After obtaining a data sample the block calculated the width and height of an elliptical region whose borders were determined by the combined beam spot. This data was done in two calculations using the ellipse major axis length and orientation data provided by the blob analysis block. The major axis was defined

along the longest diameter of the ellipse and its orientation was defined as the angle it formed with the x-axis of the image, Figure 16.

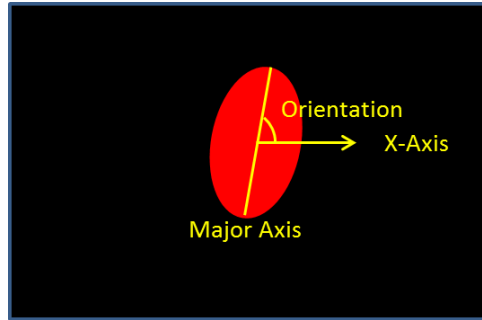


Figure 16: Ellipse Major Axis and Orientation

The x and y components of the major axis length were then determined using Equation 4 and Equation 5 below. This data was fed to the jitter control block for use as feedback. The x-length data and y-length data could be directly assigned to FSM x and y axes given that there was no cross coupling present between FSM axes. The MATLAB function would continuously run these calculations even after it had centered the beams. Additionally, after centering the beams, it would activate the area controllers.

$$Xlength = abs(axes(1,1) * \cos(orientation(1,1))) \quad (4)$$

$$Ylength = abs(axes(1,1) * \sin(orientation(1,1))) \quad (5)$$

5.2.2. Sequential Area Controllers Jitter Control Block:

The Jitter Control block in this model contains a PI controller for each axis and MATLAB functions for assigning the feedback sign and creating sequential control, Figure 17. The overall structure of this block was designed to send control to one mirror at a time for a

specific period of time and then switch control to another mirror. The integration filters of the PI controllers had to be reset after control of a beam was completed. Additionally, the feedback was defined as negative or positive.

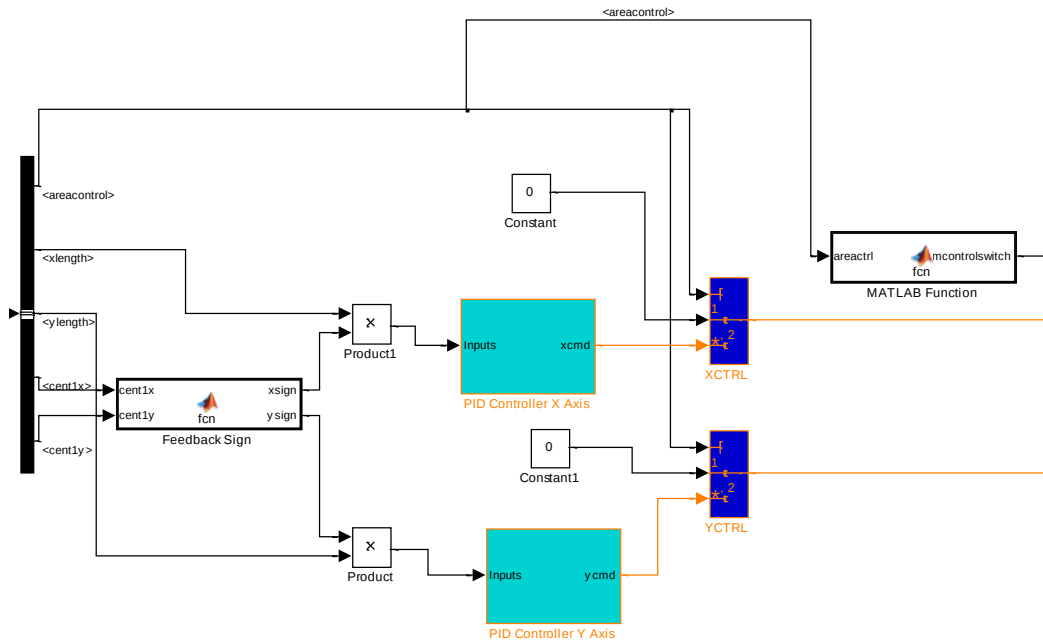


Figure 17: Jitter Control Block

First, the block determined the feedback sign. This was necessary because the width and height may increase or decrease even though the direction the beam is moving has not changed (without ever reaching zero). For example, if the beam under control is left of the point of minimization and is moving to the right, the area will decrease and then, as the beam passes the point of minimization the area will began to increase. By detecting when the beam passed through the point of minimization, the sign of the feedback could be determined. This switch in sign made it possible for the PI controller calculations to determine when the beam had passed through its optimal position. It is important to note that this block could be eliminated via use of an *a priori* positive minimum area provided to the controller. With the installed equipment, the area varied

depending on thresholding and was difficult to determine and thus this method was not used here. As a result it was determined that the algorithm results may be improved by using the movement of the combined beams' centroid to determine positive vs. negative change. If the centroid was seen shifting to the left of or below the desired aim point the feedback was defined as negative. If the centroid moved right of or above the aim point it was positive. The **mcontrolswitch** MATLAB function in this block generated a switching signal which would change which mirror was controlled every tenth of a second. This block was adjusted for one experiment so that only one beam would be controlled. The other two beams were only given commands to maintain the position that the centering algorithm had calculated during this experiment. After the sign was determined the block passed the signal to the PI controller to determine the necessary correction. The control signal was passed to the switching block at that point. The integration filters of the PI controllers were reset every tenth of a second in synch with the change of mirror under control. This was accomplished using a pulse generator within the PI controller block, Figure 18. Integration filters were not reset during the aforementioned “one-beam control” runs.

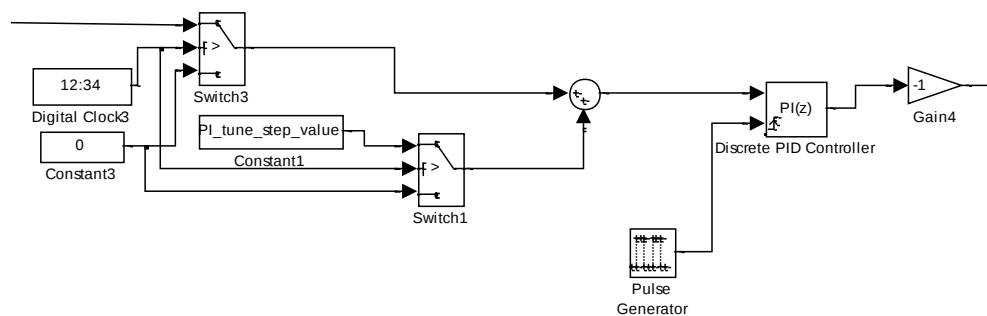


Figure 18: Pulse Generator to Reset Integration Filter

5.2.3. Sequential Area Controllers Jitter Control Block:

The switching block used the mirror control switch signal generated in the jitter control block to route signals to the appropriate mirror at the appropriate time. Six switches for each axis of each mirror were set in turn to activate a mirror when the correct mirror control signal was received, Figure 19. Additionally, when a mirror was not actively being controlled, it needed to remain in its last controlled position in order to maintain the minimization of the area it had achieved. The MATLAB function blocks labeled **Mirror Steady** accomplished this by storing the final command and holding the mirror statically in that position.

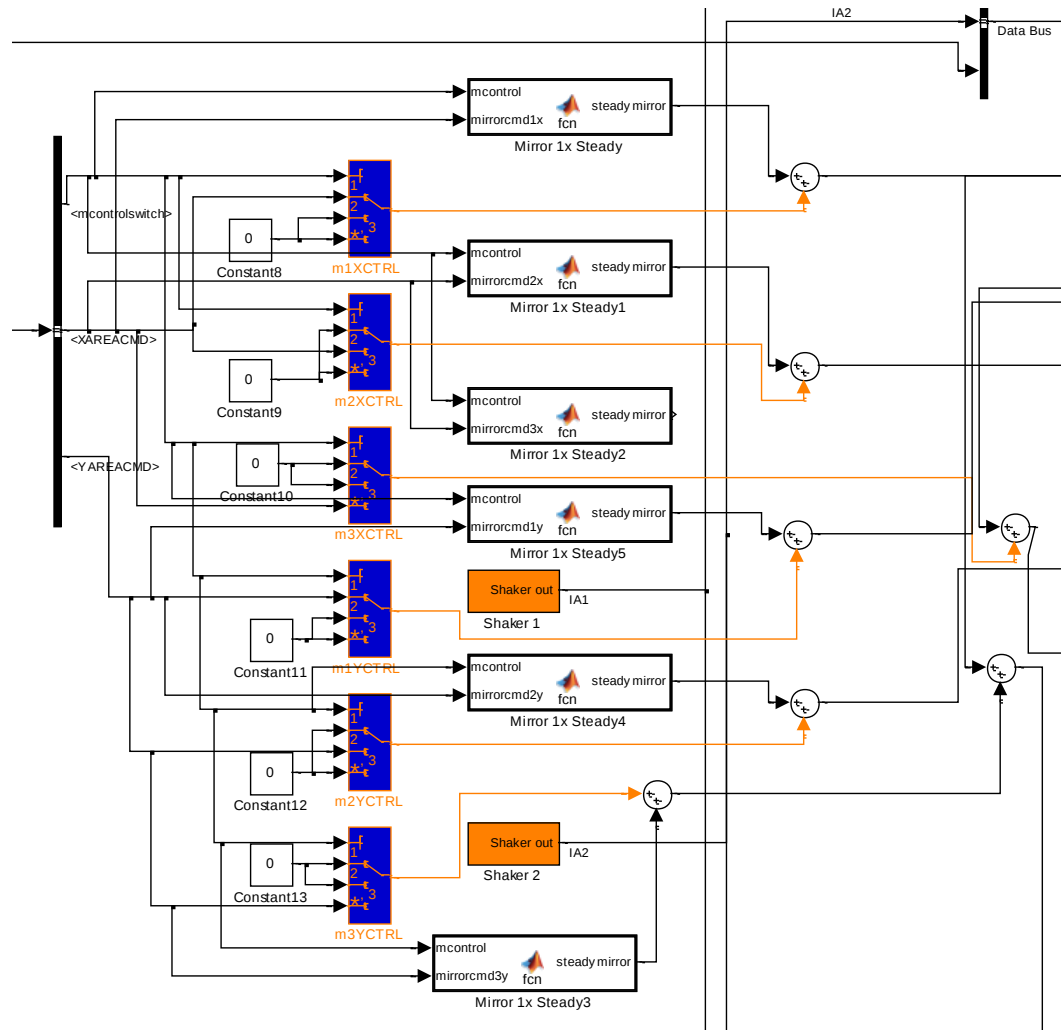


Figure 19: Switching Block for Sequential Area Control

6. Combined Command Jitter Control

6.1. *Purpose:*

The combined command jitter control model's purpose was to minimize jitter experienced by the spot of the combined beams. This was to be accomplished by applying identical jitter control commands across all three mirrors. The beam movement due to jitter was similar across all three beams due to the nature of the supporting platform and made this type of control possible. However, at higher frequencies the beam corresponding to FSM 2 often experienced higher magnitudes of disturbance. This meant that the combined correction was not correct for all three beams. In a real system, with atmospheric jitter impacting each optical path differently, this type of control may not be realistic in which case individual jitter control will be necessary.

6.2. *SIMULINK Model:*

The *CombinedAreaCtrlModel* was identical to the *ThreeCentroidPointingandAreaControl* with the exception of the sequential jitter control and switching blocks. Instead, the jitter control block contained two PI controllers (one for each axis) which were provided x and y position data for the centroid of the combined beam spot. The output of these controllers was then provided to all three mirrors without modification.

6.3. *Experimental Procedure:*

Similar to other models, the calibration blocks had to be zeroed or variable values determined using *MMM3* and its calibration file. The model could be run using the corresponding m file once calibration was complete or could be run without the calibration file if calibration was determined to be unnecessary.

Results/Analysis

1. PI SISO Control Results

1.1. *Methods of Analysis*

The analysis for this portion of the project was done individually for each beam using Power Spectral Density (PSD) plots and jitter angle plots. The Power Spectral Density of the frequencies found within the beam's movement was calculated both uncontrolled and while under control. The controller was cut on after 2.1 seconds to allow recording of the uncontrolled disturbance prior to a controlled run to verify proper response of the system. The uncontrolled PSD was calculated from a 30 second run while the platform was undergoing the disturbance vibration with no control. Comparing these plots would reveal the reduction in the PSD due to PI control using feedback from the reflection PSM. Jitter angle was calculated in these plots and those following using the distance to the target from the FSM and the displacement on the target. The jitter angle plots were a useful performance metric, as opposed to displacement distances, because they were not specific to a target range.

1.2. *One Disturbance Frequency Results*

The platform was disturbed by a 2 volt magnitude/17Hz frequency during this section, see equipment for actuator force information (approx. 1.5lbf in this case).

1.2.1. 635nm Beam

The 635nm beam performance is contained in Figure 20 and Figure 21. FSM 1 reduced the jitter angle of the beam by 97.24% by controlling the jitter to 2.4 μ rad. Previous FSM performance in the lab would indicate that this is a reasonable number. The PSD plot showed that the control reduced the 17Hz disturbance frequency by approximately 20dB. The additional peaks at around 5Hz in these PSD plots (and all those following) represents the first

and second fundamental translation modes of the platform which are excited by any applied transverse force from the inertial actuators.

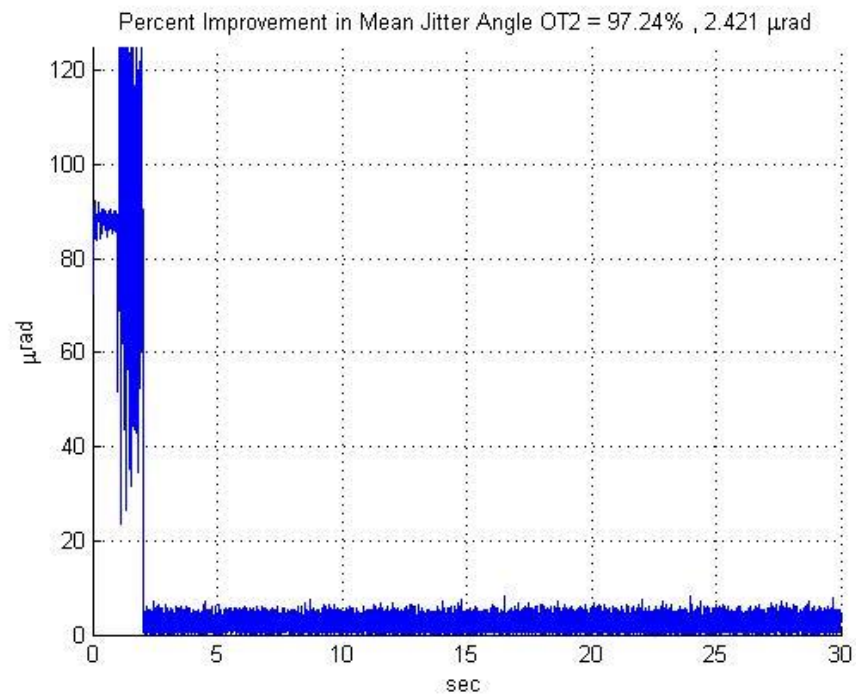


Figure 20: 635nm/FSM 1 Jitter Angle Performance under PI Control (after ~2 seconds)

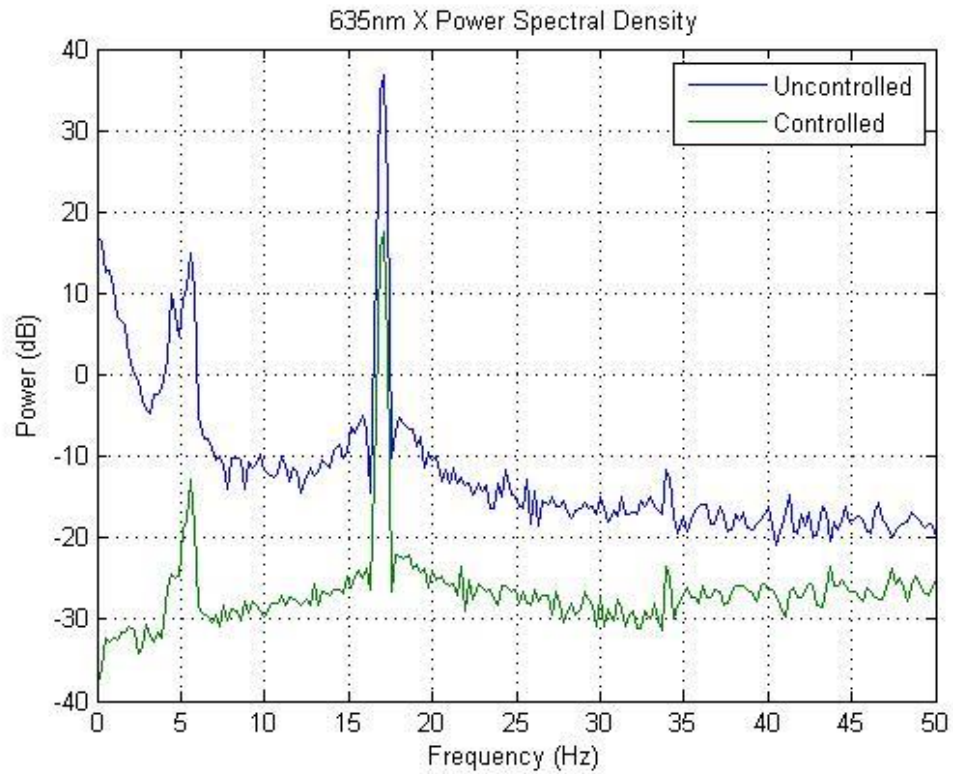


Figure 21: 635nm/FSM 1 PSD of Frequencies

1.2.2. 670nm Beam

The 670nm beam results are found in Figure 22 and Figure 23. FSM 2 reduced the jitter angle of the beam 96.01% by controlling the jitter to 2.5 μ rad. Similar to the 635nm beam, the 17Hz disturbance frequency's PSD was reduced by 20dB.

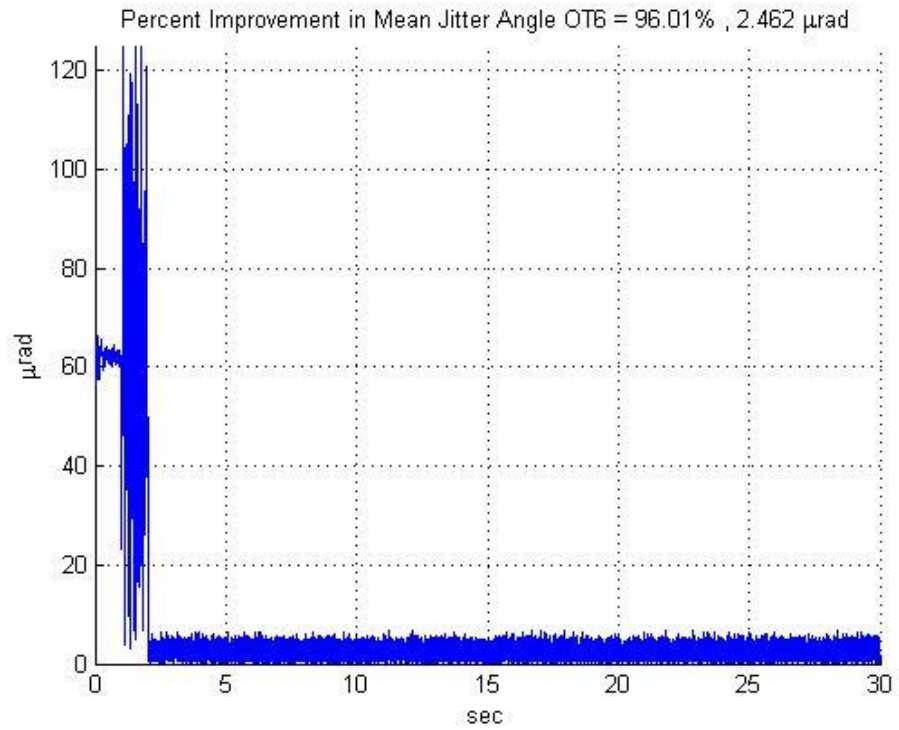


Figure 22: 670nm/FSM2 Jitter Angle Performance under PI Control (after ~2 seconds)

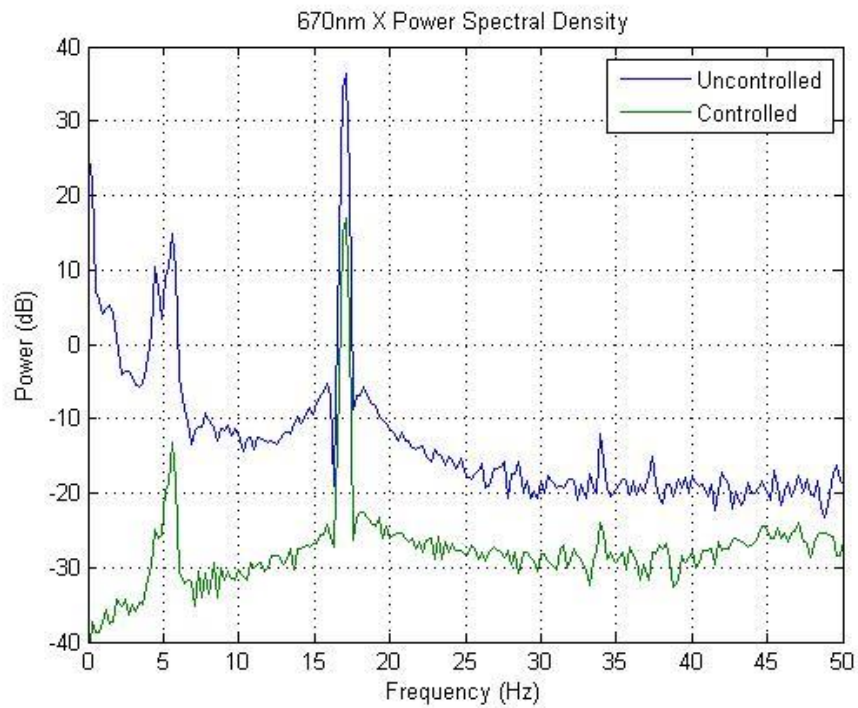


Figure 23: 670nm/FSM2 PSD of Frequencies

1.2.3. 1064nm Beam

For these runs a 1064nm beam substituted the 405nm beam due to a quality issue with the 405nm fiber. The fiber was not transmitting a significant amount of light from the source to the collimator. The 1064nm beam served as a substitute while a new fiber was acquired. The 1064nm beam results are found in Figure 24 and Figure 25. FSM 3 achieved a 94.22% improvement in jitter angle with a mean jitter angle of 2.9 μrad during control. A similar reduction of 20dB can be seen in the PSD of the 1064nm beam. The additional peak at 34Hz represents the second harmonic of the 17Hz disturbance frequency. This peak was not present in the PSDs of the other beams and is attributed to FSM 3's sensitivity. FSM 3 was sensitive throughout testing and had to be tuned daily to maintain stability. Future work will address this characteristic of FSM 3 by evaluating the transfer functions of all three FSMs.

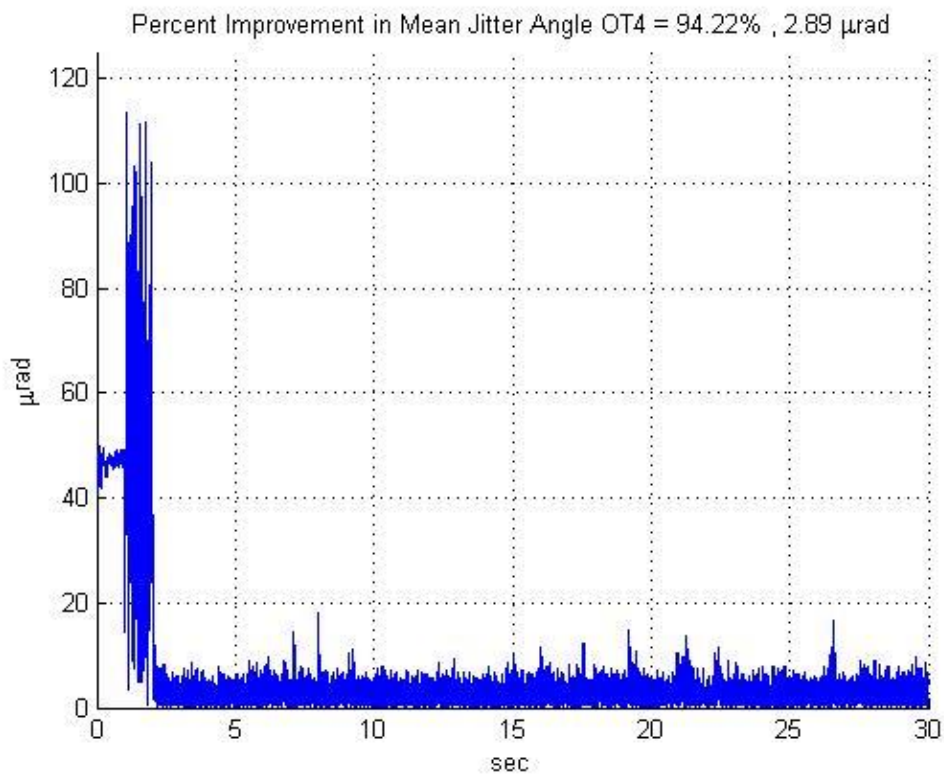


Figure 24: 1064nm/FSM3 Jitter Angle Performance under PI Control (after ~2 seconds)

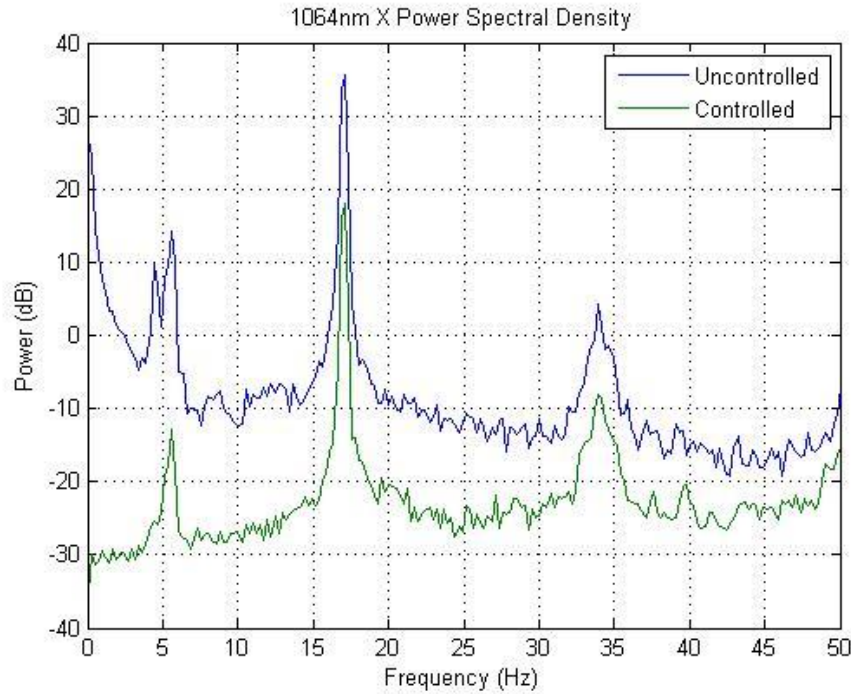


Figure 25: 1064nm/FSM3 PSD of Frequencies

1.3. *Three Disturbance Frequencies Results*

The platform was disturbed by 17Hz, 23Hz and 47Hz frequencies at magnitudes of 2 volts, 2 volts and 1 volt respectively.

1.3.1. 635nm Beam

The 635nm beam performance is contained in Figure 26 and Figure 27. FSM 1 reduced the jitter angle of the beam by 89.8% by controlling the jitter to 3.1 μ rad. The PSD plot showed that the control reduced the PSD of the 17Hz disturbance frequency by 20dB, 23Hz by 15dB, and 47Hz by 10dB.

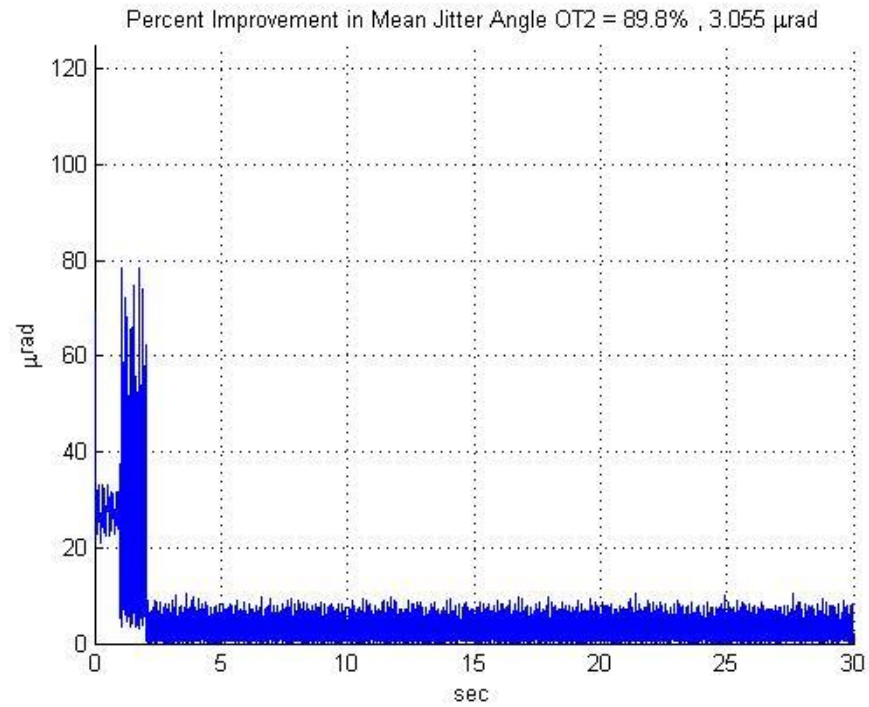


Figure 26: 635nm/FSM 1 Jitter Angle Performance under PI Control (after ~2 seconds)

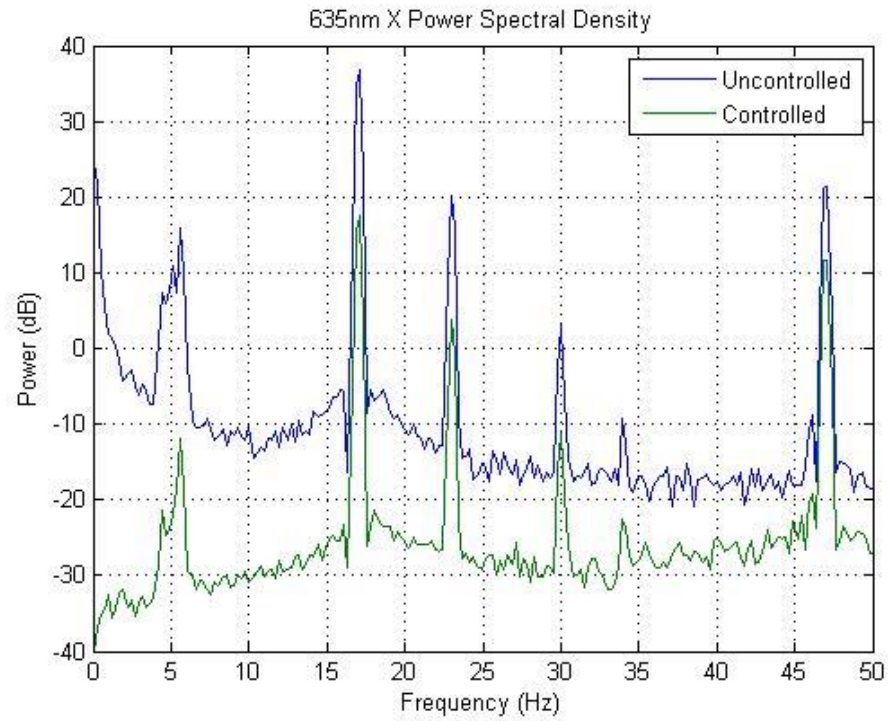


Figure 27: 635nm/FSM 1 PSD of Frequencies

1.3.2. 670nm Beam

The 670nm beam results are found in Figure 28 and Figure 29. FSM 2 reduced the jitter angle of the beam 87.09% by controlling the jitter to 3.6 μrad . The PSD of the disturbances were reduced as follows: 17Hz by 20dB, 23Hz by 15dB and 47Hz by 8dB.

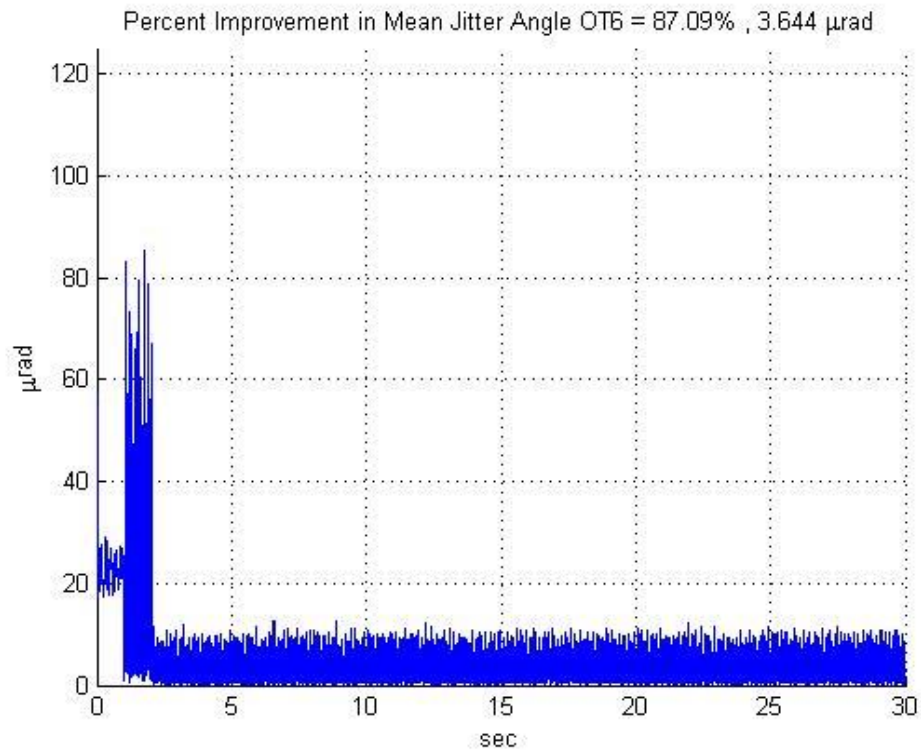


Figure 28: 670nm/FSM2 Jitter Angle Performance under PI Control (after ~2 seconds)

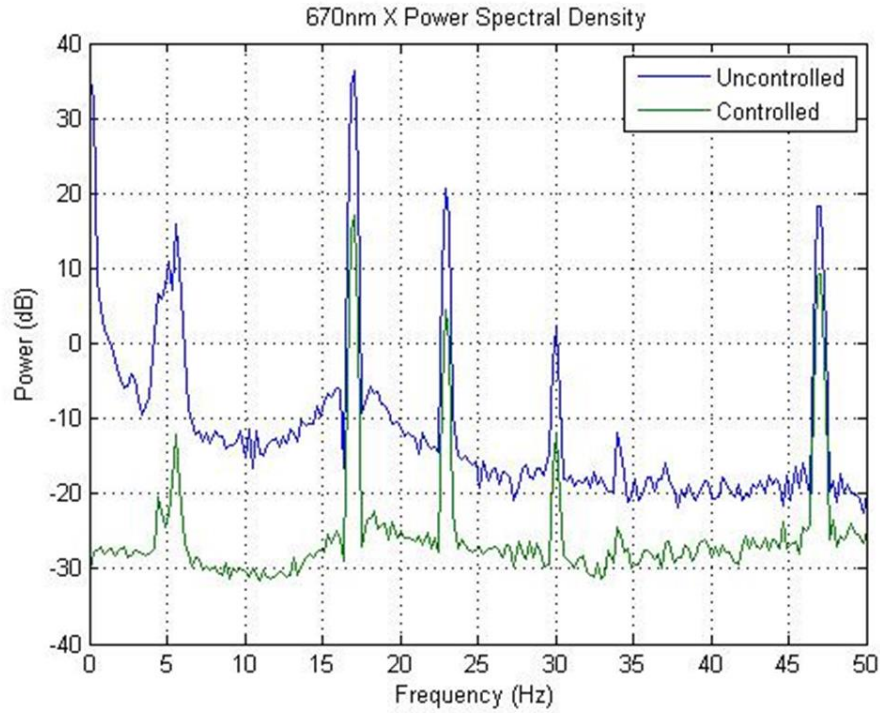


Figure 29: 670nm/FSM2 PSD of Frequencies

1.3.3. 1064nm Beam

For these runs a 1064nm beam substituted the 405nm beam due to a quality issue with the 405nm fiber as described in 1.2.3 of the results section. The 1064nm beam results are found in Figure 30 and Figure 31. FSM 3 achieved an 88.38% improvement in jitter angle with a mean jitter angle of 3.4 μ rad during control. The powers of the disturbance frequencies were reduced as follows: 17Hz by 20dB, 23Hz by 14dB and 47Hz by 5dB.

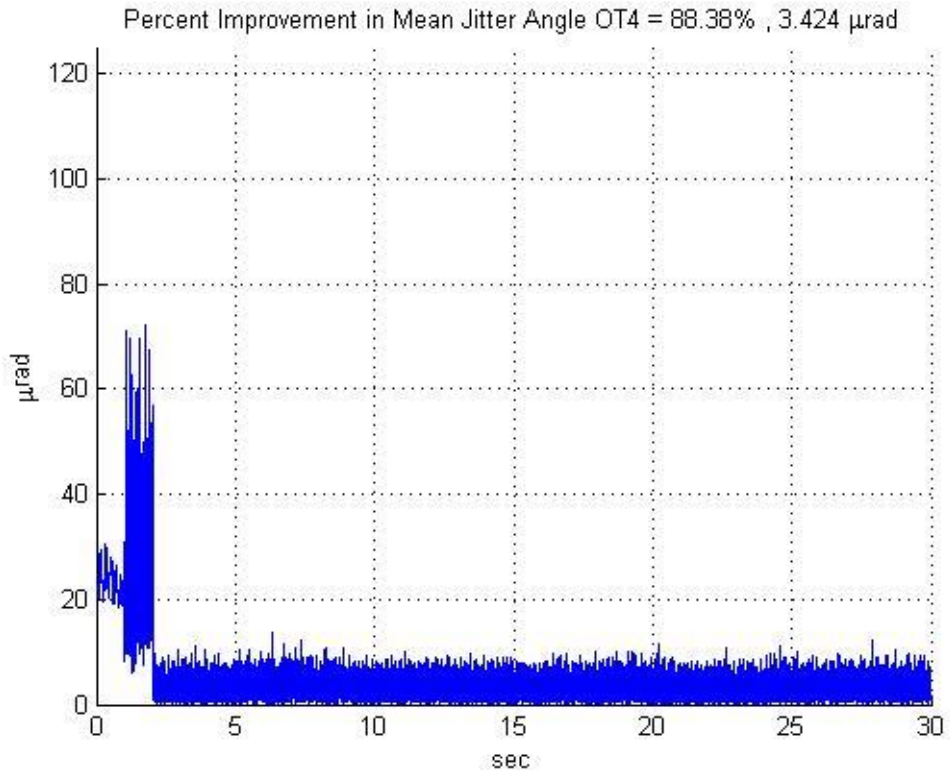


Figure 30: 1064nm/FSM3 Jitter Angle Performance under PI Control (after ~2 seconds)

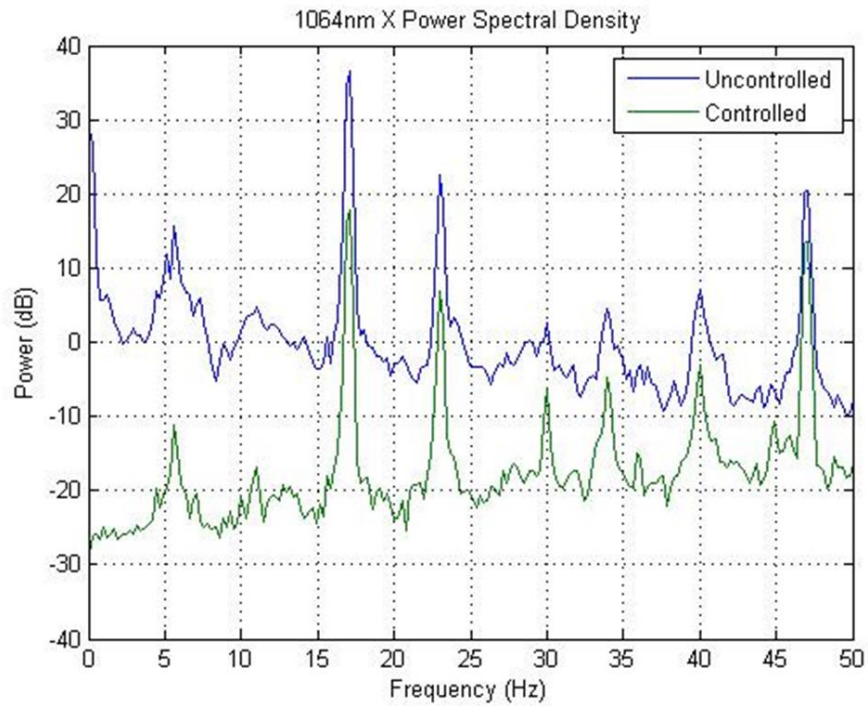


Figure 31: 1064nm/FSM3 PSD of Frequencies

1.4. *PI SISO Analysis*

The PI SISO results established reasonable performance benchmarks for the new system. The three disturbance frequencies showed that FSM 3 was more susceptible to the disturbances. FSM 3 experienced higher magnitudes of disturbances in the second and third harmonics of the 17Hz and 23Hz frequencies. FSM 3 was also more sensitive to tuning and exhibited instability on a regular basis that required the mirror to be tuned regularly.

2. **H_∞ SISO Control Results**

The analysis for this portion of the project was done individually for each beam using Power Spectral Density plots and jitter angles. It became apparent during experiments that the controller had two issues. The first issue with the control was that of frequency identification. Figure 32 demonstrates how during a 17Hz disturbance the controller took approximately 3 seconds to properly identify the disturbance and calculate the controller. Once under control the resultant mean jitter angle was approximately 10 μ rad. The second issue with the H_∞ control was that it demanded significant computing power. The target computer was able to run two beams at a 1 kHz sample rate and three beams at 500 Hz. Expanding this to a larger system would require significant computing power (e.g. a 10 beam system). Figure 32 is representative of one of the beams during the two beam 1kHz sample rate experiment.

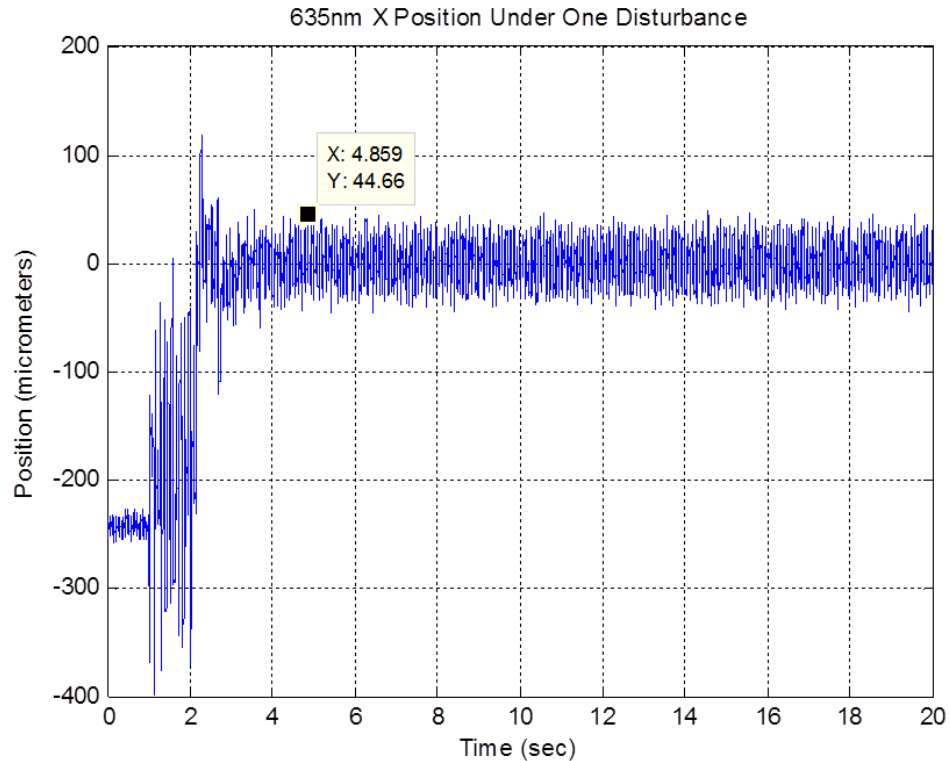


Figure 32: 635nm/FSM1 Position Data under H_{∞} Control

3. Results of Target Area (square mm) vs. Image Area (square pixels) Experiment

This experiment determined the relationship between area measurements of the camera in square pixels and actual area on the target in square millimeters. Additionally, it determined the minimum discernible jitter angle for the 300 (W) x 225 (H) pixel resolution of the camera's image. A ruler marked to millimeter accuracy was placed in the camera's field of view to determine actual vertical and horizontal lengths covered by the camera, Figure 33 and Figure 34. Figure 33 showed that the 300 pixel image width corresponded to 8 millimeters on the target. Figure 34 showed the 225 pixel height corresponded to 6.5 millimeters on the target. This meant that the 67,500 square pixel area corresponded to 52 square millimeters. A conversion factor of 7.70×10^{-4} square mm per square pixel was determined for future calculations during area control.

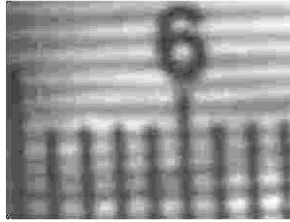


Figure 33: Image Width Measurement

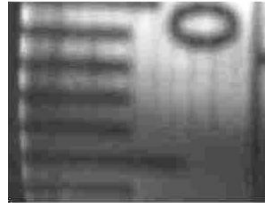


Figure 34: Image Height Measurement

4. Centering of Three Separated Beams Using the Pointing Control Algorithm

Three beams were brought together using the linear approximation described in the methods section. Runs were done with and without disturbances. Disturbance runs used 10Hz, 17Hz, 29Hz, and 47Hz frequencies at 3 volt, 2 volt, 2 volt, and 1 volt magnitudes respectively. The feedback PSMs were used to determine the accuracy of the centering algorithm under these conditions. Figure 35 indicates the positions of the beams after centering with no disturbances. The 670nm beam was the farthest from the center at a radial distance of 2318 micro-meters. The 405nm beam was closest to the center at 523 micro-meters. Using the conversion factor determined in the Target Area vs. Image Area experiment it can be determined that the algorithm missed the desired aim point for the 670nm beam by 38 pixels horizontally and 8.5 pixels vertically.

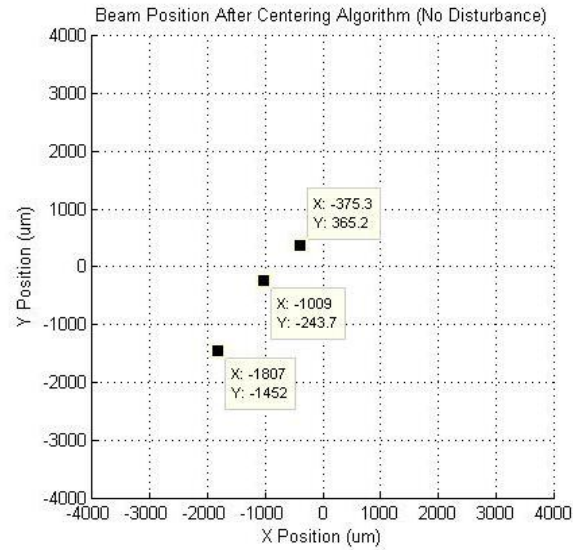


Figure 35: Position of Beams on target after Centering Algorithm (No Platform Disturbance)

Figure 36 shows the algorithm performance under disturbance. The farthest beam was, again, the 670nm beam at 1975 micro-meters. The 635nm beam was closest to the aim-point at 941 micro-meters.

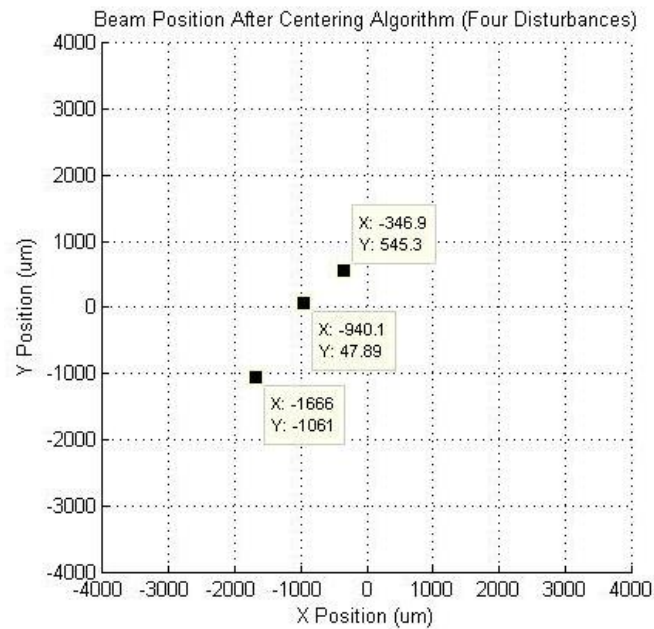


Figure 36: Beam Positions on target after Centering Algorithm (Platform Disturbances)

The performance of the algorithm during disturbance and no-disturbance runs led to the conclusion that the algorithm was stable and effective when disturbances were present. This is attributed to the fact that the algorithm averaged the position of each beam over many samples in order to eliminate any symmetrical platform jitter (i.e. a beam oscillating over a given pixel range would have its position averaged near the center of the oscillations).

The major issues with the algorithm were due to the limitations of the **Blob Analysis Block**. Approximately one out of five runs resulted in the block failing to identify a centroid or not detecting a change in beam position quickly enough. Additionally, the beams' power levels had to be balanced with respect to the response of the camera, (see equipment section). This was required to prevent the threshold block from eliminating a beam due to it being too faint. If this occurred, the pointing algorithm would not work as it was designed to identify three beams.

5. Sequential Area Control after Centering of Beams via the Pointing Algorithm

Sequential area control performance was evaluated by examining the reduction in area, both in square-pixels and mm^2 , during control. Two separate approaches to minimizing the area of the combined beams were utilized. The first left two beams fixed in place and controlled the largest beam, 405nm, to reduce area. The second approach controlled all three beams to reduce the area.

5.1. One Beam Control/Two Beam Fixed - No Disturbance

The control of one beam resulted in a reduction of the total combined spot area of 21.5% as shown in Figure 37. Additionally, the smallest achievable area is that of the largest

individual beam spot area at 1975 square pixels. This means that 1459 square pixels out of a possible reduction of 4797 square pixels (uncontrolled area minus largest individual area) were achieved. This represents 30% of the maximum possible reduction in area. Finally, the minimum area achieved was held stable by way of the feedback sign algorithm discussed in the methods section.

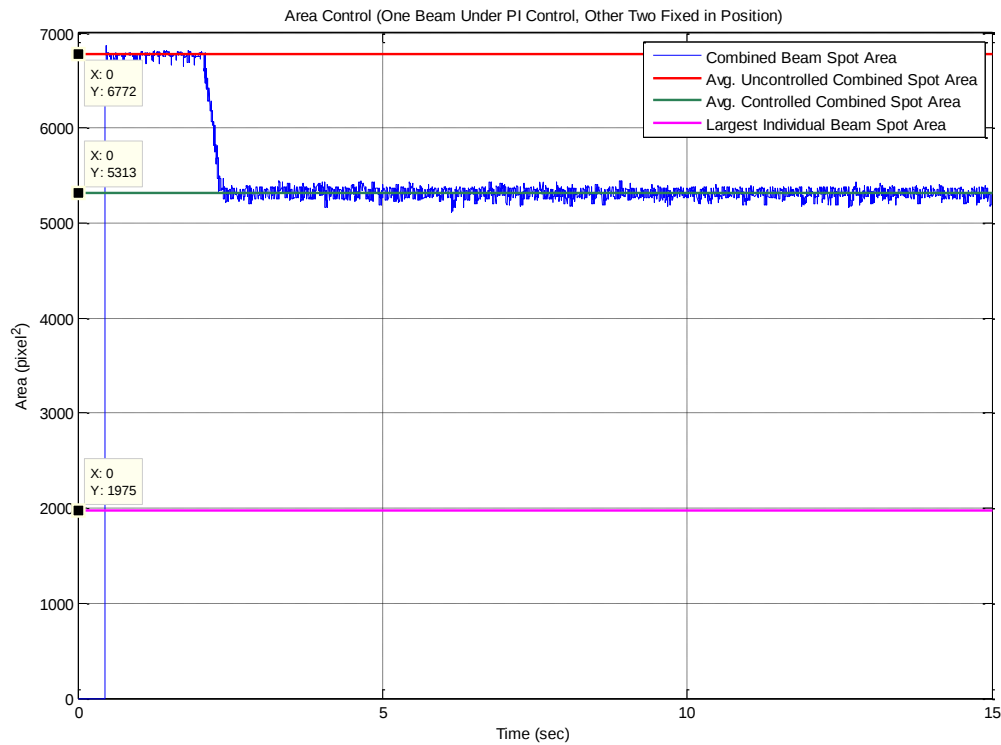


Figure 37: Area Control Algorithm (One Beam Controlled to reduce three beam spot area/No Platform Disturbances)

5.2. Two Beam Control/One Beam Fixed Disturbance Results

The area control under disturbance saw a similar minimization of the area and stable holding of that minimum. The platform was disturbed by 10Hz, 17Hz, 23Hz, and 47Hz disturbances at the magnitudes discussed in the methods section. The starting area was reduced

by 26% and 38% of the maximum possible area reduction was achieved. Movement in the combined centroid data due to jitter and not control affected the feedback sign algorithm and should be considered in future work to improve performance.

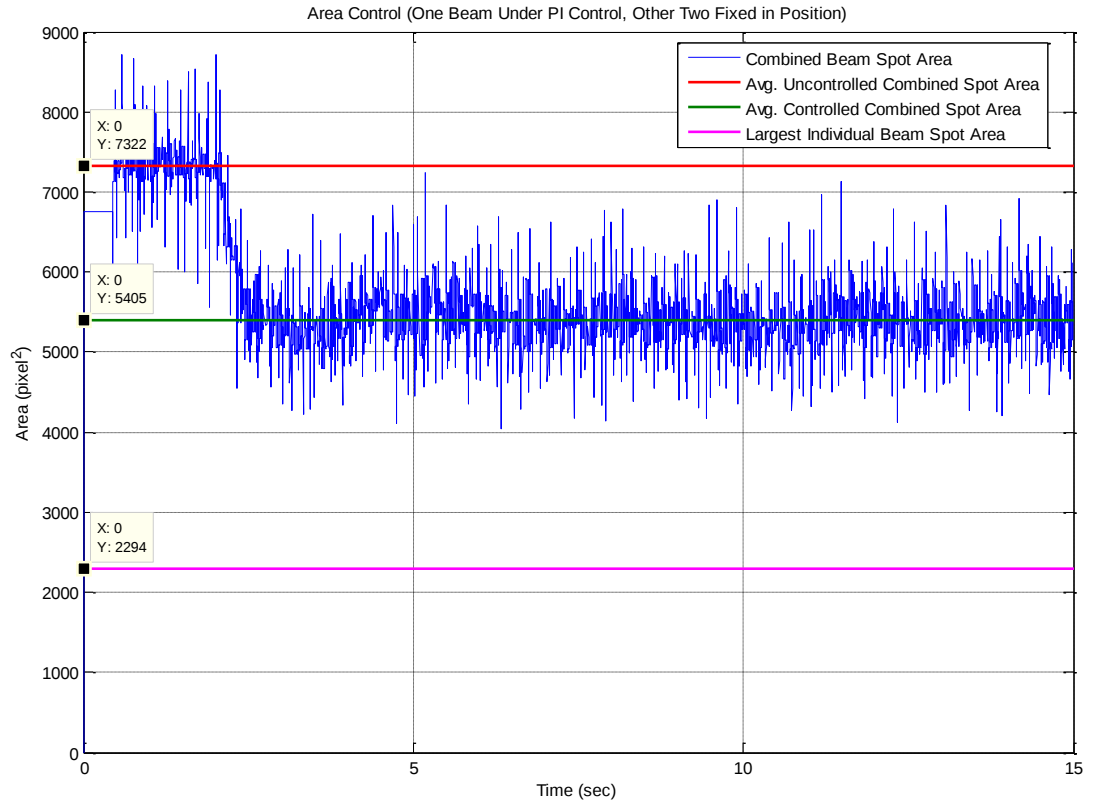


Figure 38: Area Control Algorithm (One Beam Controlled to reduce three beam spot area/Four Disturbances)

5.3. *Three Beam Area Control Results*

Three beam area control was less effective than the single beam variant. Figure 39 shows an example run where the area was reduced by 262 square-pixels. This represents the peak performance of this system. On average, the three beam control only achieved a reduction of about 150 square-pixels. The three beam control's tendency to oscillate around the large

initial starting area is attributed to the timing of the reset of the integration filters described in the methods section. The filters appear to reset at a different rate than the mirror control switch which results in “jumps” by each beam. These, in turn, caused the area to rise far above the initial starting area. In order for this method to work, the control switch that determines which mirror is controlled must coincide perfectly with the integration filter reset.

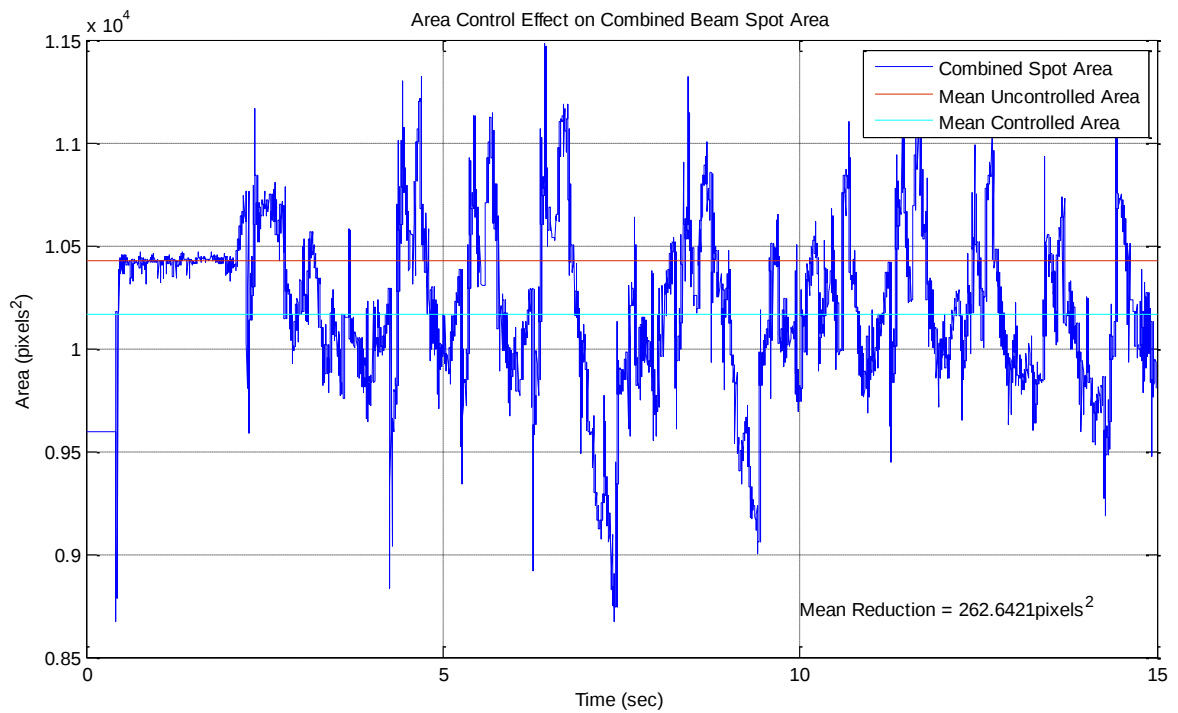


Figure 39: 3 Area Control (three beams controlled to reduce combined spot area)

Conclusions/Recommendations

The purpose of this project was to create a multi-beam control system and research methods for pointing and jitter control. The setup used was effective in combining multiple beams and allowed for individualized control of each beam. Beams were controlled using feedback from PSMs and classical PI control to set performance metrics for the system. PI controllers were capable of reducing jitter angles by 88% while the platform was disturbed by four frequencies and 94% when one disturbance frequency was present. More computationally complex H_∞ controllers were implemented to test the computation limits of the system. It was determined that three beams could be controlled at a 500 Hz sampling frequency and two beams at 1kHz. Next, pointing control algorithms and area control methods were developed. The feedback for both of these forms of control was provided by an imaging system using a 1 kHz, 12 bit resolution camera. This feedback is more realistic for the deployed system than using PSMs in the lab setting. The centering algorithm developed for pointing control demonstrated that using a linear relationship between mirror commands and beam displacement is a possible approach to pointing a multi-beam system using a sparse array of FSMs. This algorithm's performance was not negatively affected by jitter and brought beams within 2.5 mm of the desired aim point. Area controllers were developed and demonstrated that the total area of the beams could be reduced by sequentially controlling each beam or by controlling one beam and leaving two in place. Moving three beams resulted in oscillations from the PI controller filter resets. These oscillations saw peak reductions of 15%, but averaged only 2.5 % of the starting area. Control of one beam with two beams fixed after centering achieved 30% of the possible reduction in area without disturbances and 38% of the possible reduction in area when four disturbance frequencies were present. This level of reduction in area could be significant in real

world applications when trying to achieve short dwell times on target by increasing irradiance. For example, consider a system with 10 beams at 10kW each and an initial individual beam diameter of 3 inches. For this example it is assumed that jitter moves the beams in the same direction and the beams begin perfectly overlapped. If the system experiences no jitter, the beams will overlap in a 45.34 square centimeter area at 8.6 km and have an irradiance of 1908 watts per square centimeter. When five micro-radians of jitter are present, the long term beam radius is increased. The result is an increase in area to 206 square centimeters and a reduction in irradiance to 420 watts per square centimeter assuming a simple Gaussian distribution to the jitter. If this area can be reduced by 38% of the total area reduction possible, as shown in this investigation, the area can be reduced to 145 square centimeters and the irradiance increased to 597 watts per square centimeter. This represents a 42% increase in irradiance. This increase means reduced dwell times on a target. For example, an anti-ship missile requiring 5 seconds of dwell time without area control in order to disable it will require only 2.9 seconds of dwell to accomplish the same effect.

Recommendations for future work:

- Research methods to make the plant more controllable (e.g. polarization of beams, camera with sensor capable of imaging multiple wavelengths with each sample allowing the beams to be individually distinguished when overlapped). A more effective solution to the beam identification problem would significantly reduce the complexity of pointing and jitter control in a multiple beam system thus maximizing the effectiveness of the sparse array FSM arrangement.
- If beam identification cannot be done: develop Kalman filters or other methods to track the beams while moving. This could prove useful in dealing with beams that are constantly crossing due to increased jitter or atmospheric effects.

- Centroid identification is a critical component using an imaging system for feedback. The MATLAB **Blob Analysis Block** is effective for simple image analysis however, for the multi-beam feedback scenario, the block is not fast enough nor does it meet the required accuracy. Further research is needed to develop an analysis tool for detecting and determining the position of multiple beams. The appendix of this report contains the block this research was able to create. This block was effective, but contained too many extrinsic functions to be effectively applied in a MATLAB s-function with the current CPU limitations.

Equipment

1. Lab Setup Overview

At the Naval Academy's Directed Energy Research Center there are two Newport optical tables connected by a tunnel, Figure. One optical table serves as the source table and contains an optical plate which can be disturbed using inertial actuators. The other table contains target sensors and materials for testing. The tunnel and tables are surrounded by acrylic sheets for laser safety.

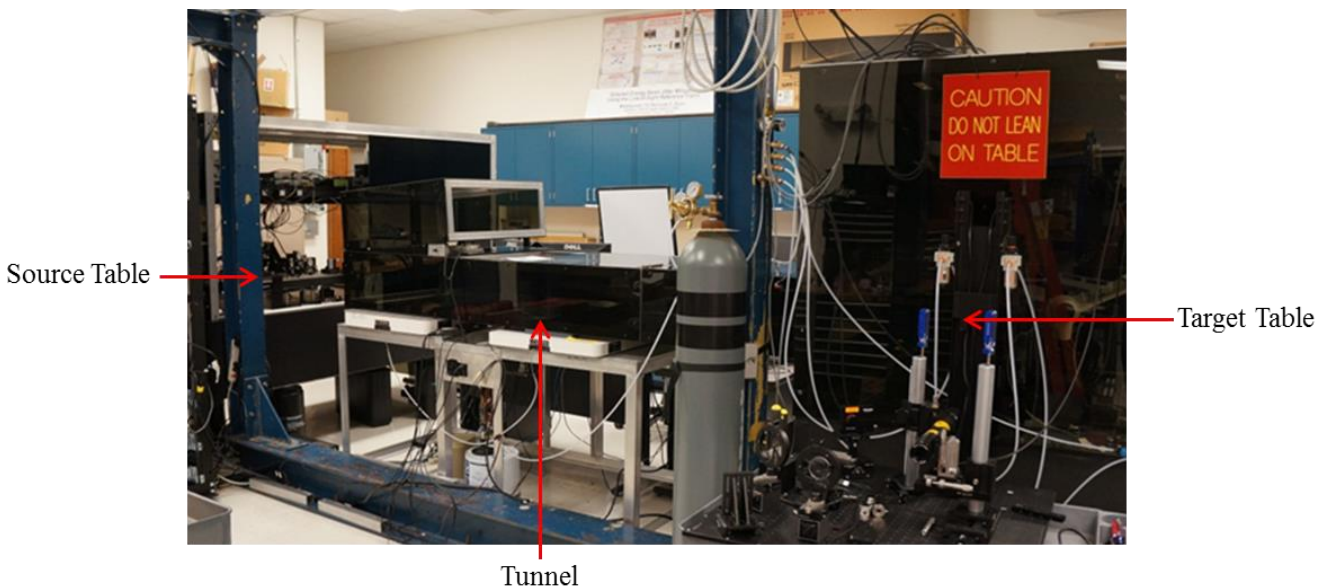


Figure 40: USNA Lab Setup

The source table in this project contains three fiber coupled lasers of different wavelengths with collimators, three flat mirrors, three fast steering mirrors, and two inertial actuators, Figure 41. Each beam is individually reflected off of a single flat mirror onto a corresponding fast steering mirror. The fast steering mirrors are the final component of the optical train. The platform on which all of this is mounted is floated via a spring/isolator setup and pressure from a tank of nitrogen. The inertial actuators subject the platform to vibrational

disturbances to simulate motion of the kind expected from a surface vessel or aircraft. The vibrations consist of several frequencies that are determined by the user before an experiment takes place.

The target table, Figure 42, contains position sensing modules (PSM) for sensing the position of each beam and beam splitters to reflect the beams while also allowing them to pass through to the target PSD. The detectors that receive reflections from the beam splitters have band pass filters so that only one beam wavelength can be seen by each of the PSDs. The readings of these PSDs were calibrated to match those of the target PSD (see method of investigation) so that the individual position of each beam on the target PSD was known. This served as a mode of control feedback in the first SISO controllers and later as a diagnostic when using video based feedback. The optical path of the beams across the equipment is nominally shown in Figure 43.

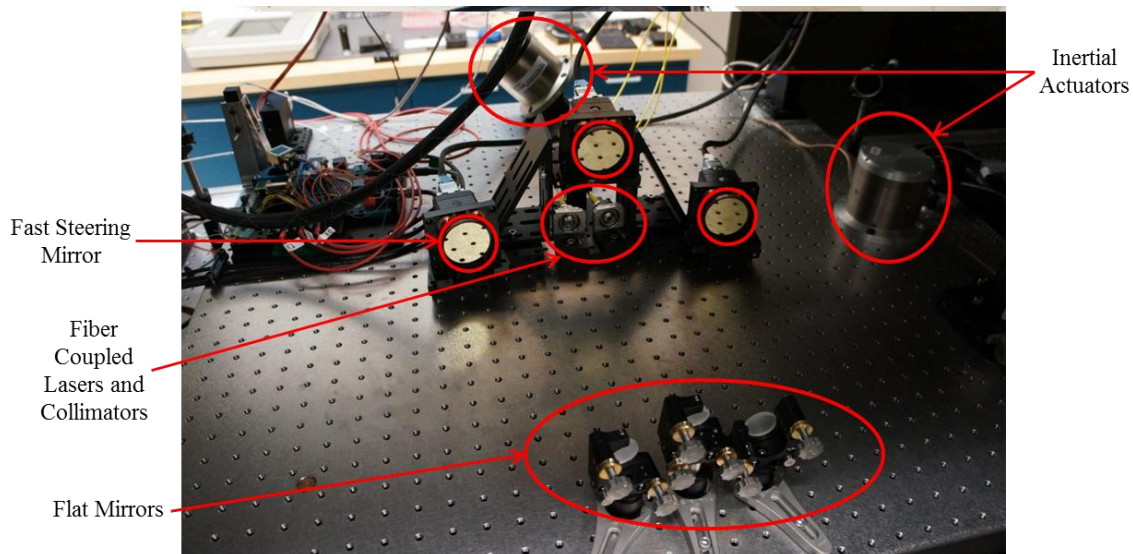


Figure 41: Source Table

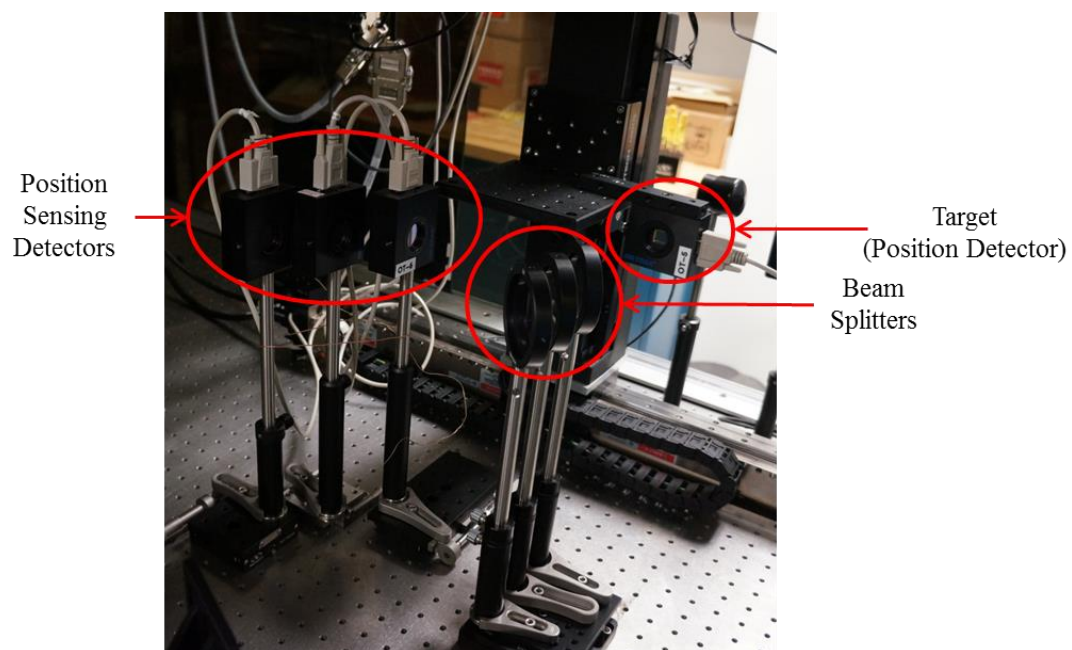


Figure 42: Target Table

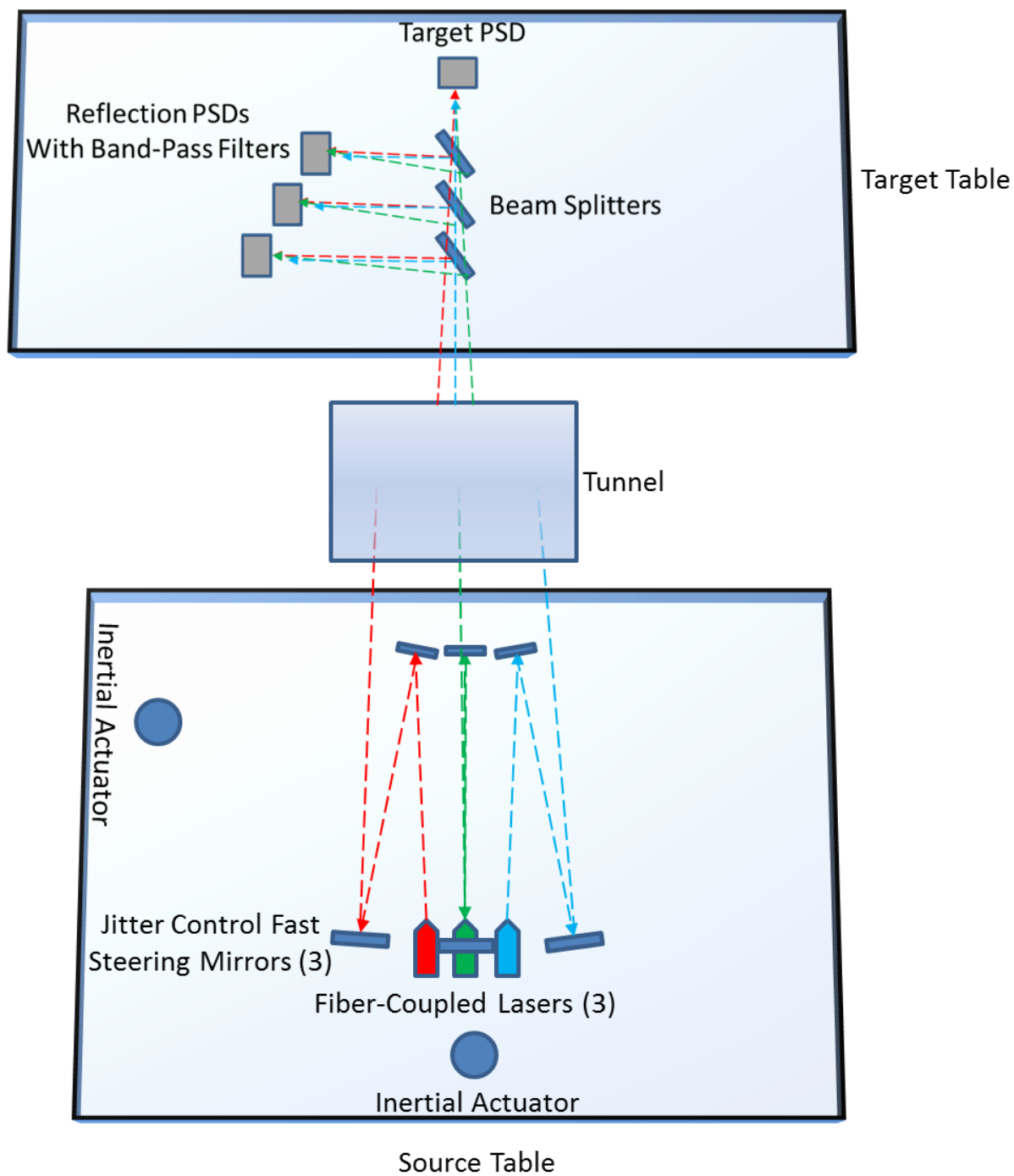


Figure 43: Arrangement of Equipment with Beam Paths

2. Fiber Coupled Lasers

The fiber coupled laser Multiple Channel Laser System (MCLS) used in this project is supplied by ThorLabs, Figure 44. It contains four different wavelengths whose power can be adjusted via the control knob. Additionally, the temperatures of the power sources can be adjusted for stability control. The specifications for each laser channel are given in Table 1.



Figure 44: MCLS Control Box¹⁴

Table 1: Channel Specifications

LOCATION	CH.1	CH.2	CH.3	CH.4
WAVELENGTH (NM)	405	635	675	1064
MEASURED POWER OUTPUT (mW)	5.23	4.07	3.12	25.32
LASER CLASS	3B	3B	3B	3B
FREESPACE/FC	FC/PC	FC/PC	FC/PC	FC/PC
DIODE MANUFACTURER	LASERCOMP	OPNEXT	HITACHI	AXCELPHO

¹⁴4-Channel Fiber-Coupled Laser Source. (n.d.). Retrieved January 2013, from THORLABS:
http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=3800

	OEN			TON
DIODE	CS4050205M	HL6322G	HL6714G	M9-A64-0200

3. Fast Steering Mirror

The fast steering mirrors (FSMs) serve to mitigate jitter and poor aim by allowing for correction of the beam's position at high frequencies. Each fast steering mirror has two axes of rotation, azimuth and elevation, allowing correction of a beam's horizontal and vertical position. The mirrors come from Optics In Motion, Figure 45. The specifications for the mirrors as given by the manufacturer are contained in Table 2.

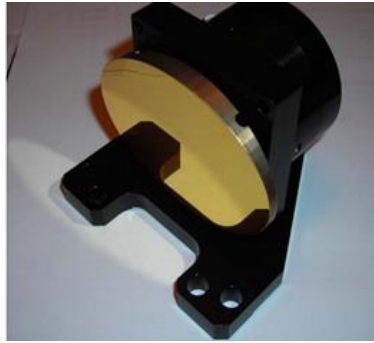


Figure 45: Fast Steering Mirror¹⁵

¹⁵(n.d.). Retrieved January 2013, from Optics In Motion: <http://www.opticsinmotion.net/OIM102-3%20INFO.pdf>

Table 2: Fast Steering Mirror Specifications¹⁶

Specification	Typical	Units
Dynamic Performance		
Mirror Angular Range (mechanical)	+/- 1.5	degrees
Angular resolution	<2	urads
3dB Bandwidth (small angle 10mV)	> 500	Hz
Linearity	1%	% Full Scale
Step Response (1 mrad step)	<10	ms
Mirror Substrate		
Material	Fused silica	
Mirror substrate size	2.1" x 3.0"	
Coating	Protected Gold	
Reflectivity	>95% from 700nm – 10um	
Wavefront quality	$\lambda/4$ @ 633nm	waves rms
Clear Aperture	1.8" x 2.7"	inches
Electrical		
Peak power	30	Watts
Mechanical		
Mirror head size	2.3 X 2.3 X 2.2	inches
Controller size	2.0 X 4.0 X 6.1	inches

4. Collimators

Collimators are used to collimate or align the paths of a beam's waves in a parallel fashion. This, in theory, means the beam will not converge or diverge and therefore will stay the same diameter along its path regardless of distance. In practice the beam is very close to collimated, but not perfectly. ThorLabs collimators are used in conjunction with ThorLabs fibers

¹⁶Ibid.

and the Thorlabs MCLS to assure compatibility between components. The Thorlabs collimators used are the FiberPort PAF-X-18-A (405 nm beam), FiberPort PAF-X-18-B (635 nm, 675nm), and FiberPort PAF-X-18-C (1064 nm). The output characteristics of these collimators are detailed below, Table 3.



Figure 46: FiberPort Collimator¹⁷

Table 3: Collimator Specifications¹⁸

Item #	EFL	Input MFD ^a	Output 1/e ²		Divergence	Lens Characteristics		AR Ranged	Length L ^e (in/mm)
			Waist Dia.	Dist. ^b		CA ^c	NA		
PAF-X-18-A	18.4 mm	3.5 μ m	3.01 mm	7936 mm	0.190 mrad	5.5 mm	0.15	400 - 600 nm	0.87/22.8
PAF-X-18-B	18.4 mm	5.0 μ m	3.98 mm	7347 mm	0.272 mrad	5.5 mm	0.15	600 - 1050 nm	0.87/22.8
PAF-X-18-C	18.4 mm	10.4 μ m	2.95 mm	2629 mm	0.565 mrad	5.5 mm	0.15	1050 - 1600 nm	0.87/22.8
a. Mode-Field Diameter, calculated using the following equipment:									
-A: 460HP at 450 nm, -B: 780HP at 850 nm, -C: SMF-28e+ at 1550 nm, -D: SM2000 at 2000 nm									
b. Maximum Waist Distance is defined as the maximum distance from the lens a Gaussian beam's waist can be placed.									
c. Clear Aperture									
d. Wavelength of the Antireflection Coating									
e. Length from tip of the connector bulkhead to face of the FiberPort flange.									

¹⁷FiberPort. (n.d.). Retrieved January 2013, from ThorLabs:
http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2940

¹⁸FiberPort. (n.d.). Retrieved January 2013, from ThorLabs:
http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2940

5. Position Sensing Modules

Position Sensing Modules (PSMs), Figure 47, are used to determine the location of the centroid of “power density” of a spot of light on a sensor.¹⁹ Coated layers respond to light and generate current corresponding to the distance from incident point to electrodes on the outer border of the sensor. These currents are inversely proportional to the distance between the incident point and the electrodes.²⁰ By examining these values, precise horizontal and vertical positions can be determined.

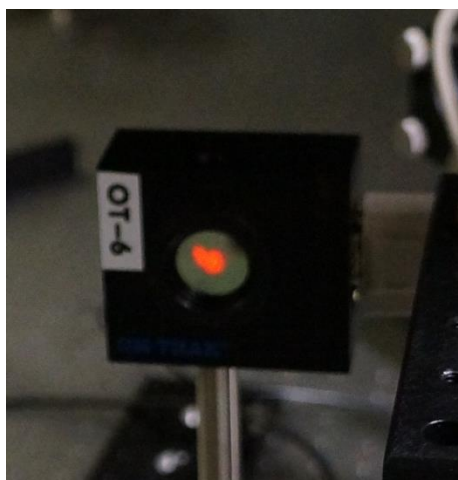


Figure 47: Position Sensing Module

6. Video Camera

This project employs the use of a visual cue from a camera for feedback. The Hamamatsu C11440, Figure 48, was chosen for its high sample frequency ($>1\text{kHz}$) and high resolution. The pertinent data and response are contained in Table 4 and Figure 49.

¹⁹PSD Theory. (n.d.). Retrieved January 2013, from On-Trak Photonics: <http://www.on-trak.com/theory.html>

²⁰Ibid.



Figure 48: Hamamatsu Camera

Table 4: Camera Specifications²¹

Imaging Device	CMOS image sensor FL-280
Wavelength Range	300nm-1000nm
Active Pixels	1920x1440
Pixel Size	3.63x3.63 μm

²¹C11440. (n.d.). Retrieved January 2013, from Hamamatsu: <http://sales.hamamatsu.com/index.php?id=13226509>

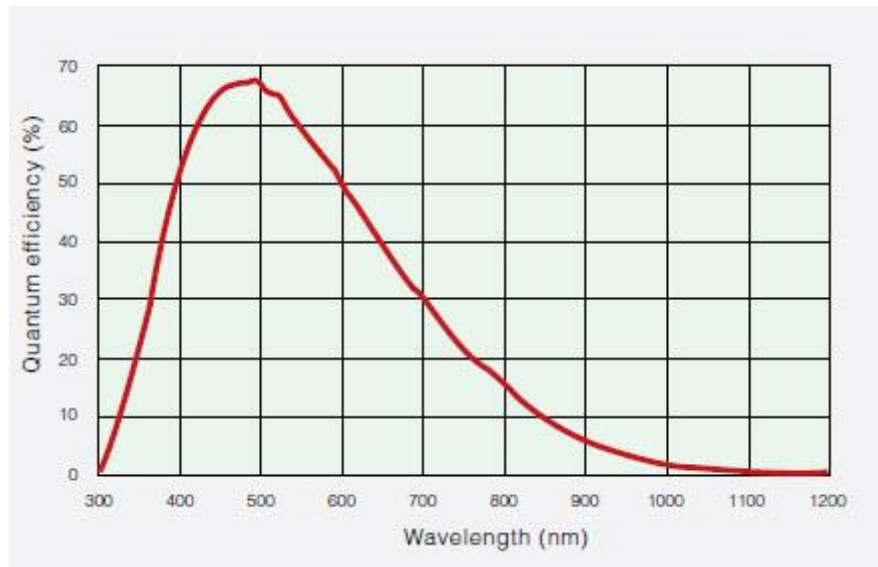


Figure 49: Camera Response²²

7. Video Camera

The actuators' forces are described by the following chart, Figure 50. The resistance of these actuators was 4 ohms. Commands were given in volts which must be converted using the resistance to amps for use in the chart.

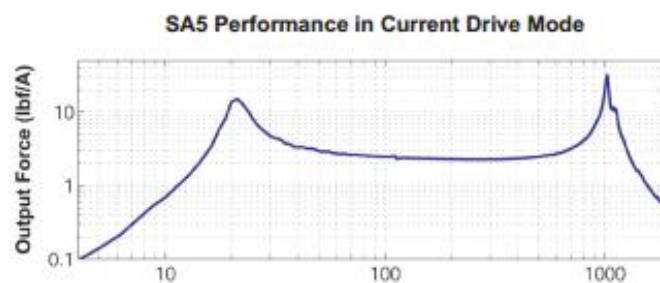


Figure 50: Inertial Actuator Performance Chart²³

²²C11440. (n.d.). Retrieved January 2013, from Hamamatsu: <http://sales.hamamatsu.com/index.php?id=13226509>

²³CSA Engineering. April 2013. www.csaengineering.com/literature

Appendices

7. Hamamatsu Camera Usage/Models

7.1. XPC Target Send/Receive Models

The XpcCamHost_VideoReceive/Send models allowed the user to send video data from the target PC and view it in real time on the server computer. The send model was loaded on the target PC and connected to the receive model which was running on the server.

7.1.1. Send Model:

Within the send model there were four major blocks, Figure 51. The first was the Neon CL BitFlow block. This block used a predefined configuration file and a number of settings to determine the image size and windowing of the camera data, Figure 52. The “columns” and “rows” options defined the image resolution in pixels. Typically, a resolution of 300 by 225 pixels (W by H) was used to maintain a 1 kHz sampling rate. Regardless of size, the ratio of H/W was maintained at 0.75. This was to ensure no distortion of the image. The starting column and row determined which pixel was selected to be the origin of the image. The origin was defined by the block as the top left corner of the image.

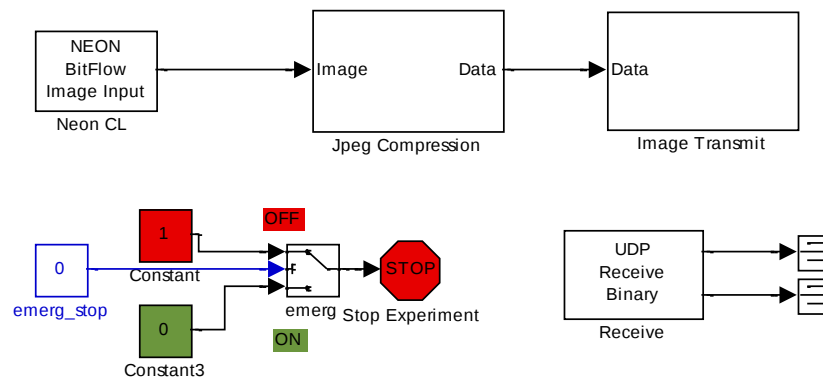


Figure 51: XPC Target Video Send Model

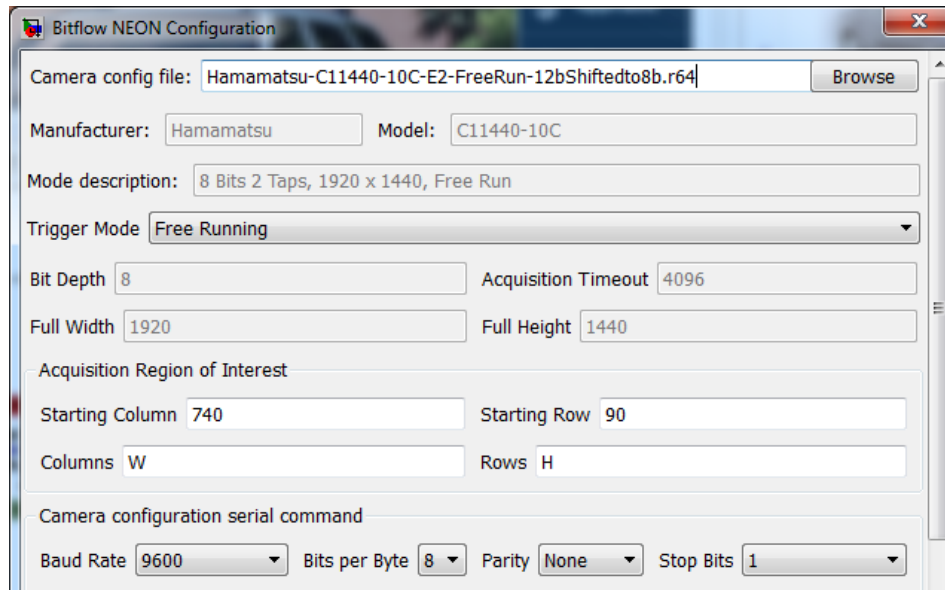


Figure 52: Bitflow Configuration Block

The JPEG Compression block's purpose was to reduce the amount of data contained in the image (compression) and turn the color image grayscale. Within the block's options the user could define maximum output size of the image and desired color space.

The image transmit block connected to the image receive block in the video receive model loaded on the server PC. The settings for the block were defined as seen in, Table 5. The UDP Receive block output ports were routed into terminators. However, they could be used to keep the previous output when no new data arrived.

Parameter	Value
IP Address to send to	192.168.0.2 (can change if LAN is altered)
Remote IP port to send to	25001
Use the following local IP port	-1 (automatic port assignment)
Sample time	Ts (defined outside of the model in a script)

Table 5: Image Transmit Parameters

7.1.2. Receive Model:

The receive model, Figure 53, was run on the server computer in order to import the sent data. The image receive block was setup to receive from all IP addresses in order to prevent a bad link between the target and server PCs. The video viewer block allowed the user to see the video in real time. If the user desired, a video file could be created using the multimedia file block. However, this block was often deleted from the model to decrease execution time.

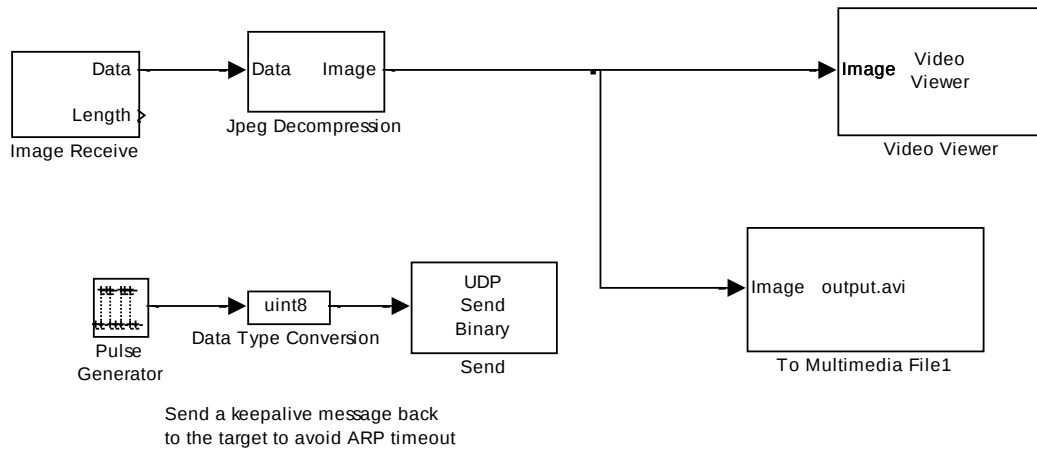


Figure 53: Video Receive Model

8. Image Analysis

8.1. *MATLAB Blob Analysis Block*

Centroid position and area calculations were accomplished via the MATLAB blob analysis block, Figure 54. This block received a black and white image which had been generated by thresholding the camera's grayscale image. The block was asked to search for 3 spots and determine their centroids' locations in the image. Additionally, it was used for giving the dimensions and total area of the spot generated when all three beams were centered. These were all given as outputs of the block by checking corresponding boxes in the main block

settings. The blob properties menu was critical to the performance of this block. The blob area had to be set precisely in order to obtain reasonable results. The expected minimum area was set to 40 pixels despite the beams' actual sizes being in the thousands of pixels. The block tended to eliminate too much of the beam spot and give poor results when larger minimums were set. No maximum blob areas were given for similar reasoning. The block was generally very temperamental and it is recommended that it be replaced with the centroid script generated during this project for analysis.

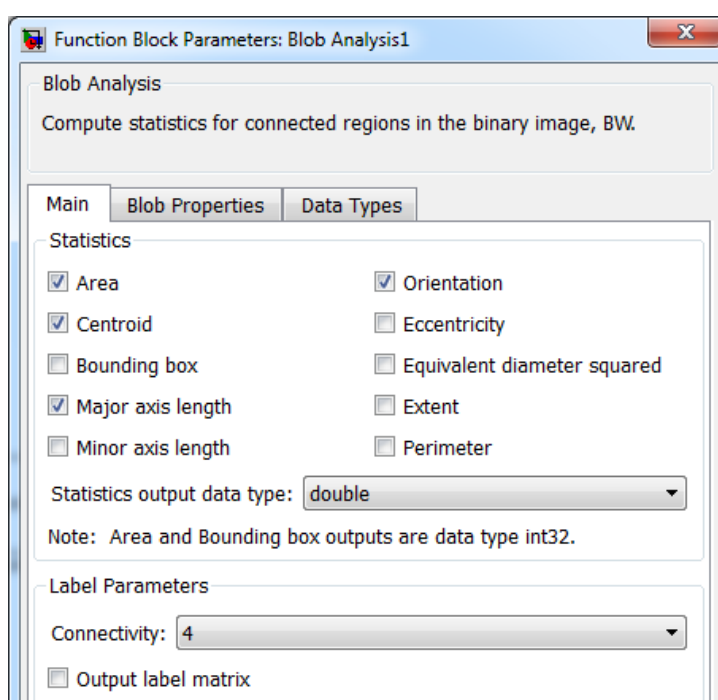


Figure 54: Blob Analysis Block

8.2. *Intensity Centroid Detection Script*

This project generated a MATLAB script file that would region the beam spots and determine centroid location, Figure 55. If desired, the region function in the script could generate area calculations. The script was designed to be given grayscale image data. This script should give better results than the MATLAB blob analysis block if its threshold and area

opening functions are properly tuned. In order to be implemented on the target PC, the MATLAB functions must be coded since they are currently extrinsic to the script.

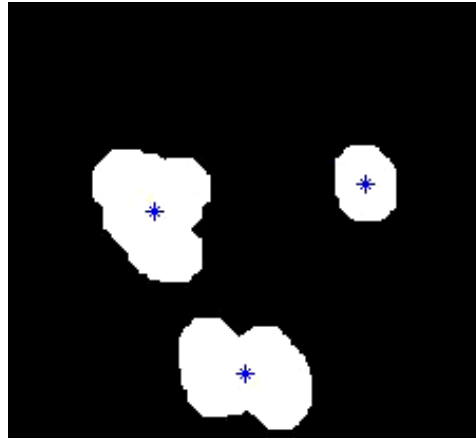


Figure 55: Centroid Script Result for Three Distinct Beams

8.3. *Kalman Filter for Tracking Beam Spots*

A simple Kalman filter was developed from MATLAB tutorial work done at <http://studentdavestutorials.weebly.com>. The purpose of this filter was to deal with situations where beam spots crossed. In these situations it was assumed that the FSM controllers had identified which beam spot corresponded to a given FSM. At that point, the controllers could begin to control the beams based on centroid position. However, if the beam spots crossed one another it would be impossible for the controllers to track individual beam position once spots became overlaid. The Kalman filter and supplementary Hungarian algorithm were intended to deal with this situation by tracking the spots and, via a simple model of the system, predicting the beam's position in crossing situations. After the beams had crossed, these anticipated position predictions could be used to assign the individual beam spots back to their corresponding FSM using the Hungarian algorithm. The algorithm tracked the actual and predicted positions of the spots before during and after they crossed. This allowed it to track the

spots and correctly assign numbers back to the right spots when they “reappeared” after becoming ambiguous during the crossing situation.

9. **Equipment Usage**

9.1. ***FiberPort Collimators***

The collimators were used to focus the fiber coupled laser light at infinity. By doing so, the beam had little to no divergence and would traverse its entire optical path at the same beam diameter. The smallest diameter black screws on the back, surrounding the fiber “port,” were used to adjust the position of the front lens of the collimator. If the beam was converging (coming to a focus) the black screws needed to be turned clockwise. Conversely, a diverging beam (increasing in size) called for the screws to be turned counter clockwise. All three screws needed to be adjusted in small and equal increments to ensure proper positioning of the lens. The silver screws on the back adjusted a set of springs which placed a force on the lens. This spring force is what held the lens in place after adjustments were made. Once the springs were completely compressed, however, it was not possible to make further adjustments to the lens position via the black screws. As a result, the user had to turn the silver screw counterclockwise to reduce spring compression and allow the lens to move. If the lens were moved so far forward that the spring was fully extended there would be no force holding the lens in place. If this occurred the silver screws had to be turned clockwise to increase pressure on the lens. Around the outer circumference of the collimator were three other screws which served to adjust the lateral and vertical position of the lens. The third screw approached the other two at a forty-five degree angle and, when tightened, locked them in place. It was important to use this locking screw because without it vibrations would tend to shift the focusing lens over time.

9.2. *Laser Graphical User Interface (GUI)*

A GUI, Figure 56, was created for controlling the fiber coupled lasers without needing access to the source box. This was useful for maintaining safety and for increasing efficiency during calibration when beams had to be manipulated regularly. This GUI communicated with the source by virtual serial connection over a USB cable. It is important to note that MATLAB was able to handle this serial communication in the GUI, but not in large run script files for SIMULINK models. This meant that script files could not automatically manipulate the laser source. As a result, all changes had to be made manually via the GUI. The control window allowed the user to enable with a left-click and to disable with a right-click. The current meters could be slid or the arrows could be used to make changes in small increments. The user could obtain real time laser status data by using the refresh values buttons. It is important to note that the system enable had to be enabled before individual beams would turn on. The source box, when used with the lab's inter-lock system, rejected some aspects of this GUI. The source box would no longer send laser status data or confirmation of enables/disables. The basic enable/disable and current adjustment functions were useable, but actual power levels had to be checked on the source box display.

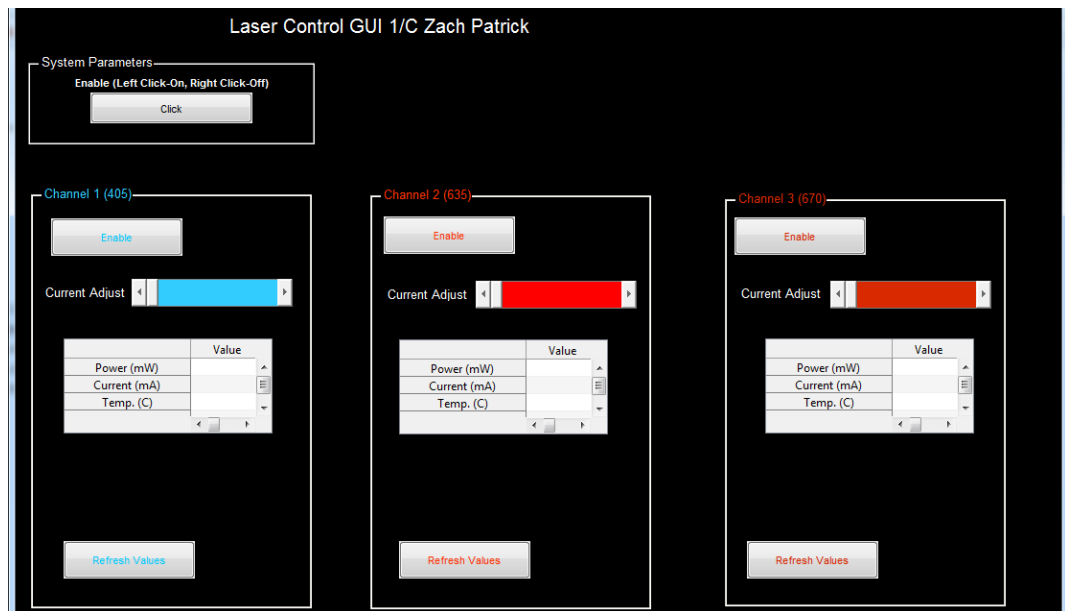


Figure 56: Laser GUI Window

10. MATLAB Code

10.1. Calibration Code

```

%% Patrick's Three Laser, Four PSD, Three FSM, Three Flat Mirror, Calibration
File
clear
%% Model Values
Ts=0.005;
H = 225;
W = 300;
%sample time
fintime = 7;           %Length of data run.
mirror1xcal = 0.0;
mirror1ycal = 0.0;
mirror2xcal = 0.0;
mirror2ycal = 0.0;
mirror3xcal = 0.0;
mirror3ycal = 0.0;
OT6Xcal     = 0.0;
OT6Ycal     = 0.0;
OT2Xcal     = 0.0;
OT2Ycal     = 0.0;
OT4Xcal     = 0.0;
OT4Ycal     = 0.0;
calibrate=0;
%-----
% Shaker input (sinusiod, max 4 signals)
shakeramp   = [0      0      0      0];

```

```

shakerfreq      = [0      0      0      0];
shaker_switch   = 7;
shakerampb      = [0      0      0      0];
shakerfreqb     = [0      0      0      0];
shakerphase     = [0      0      0      0];
shakeramp2      = [0      0      0      0];
shakerfreq2     = [0      0      0      0];
shakerphase2    = [0      0      0      0];
shaker_start    = 0;
shaker_end      = fintime;
shaker_start2   = 1.00;
shaker_end2     = fintime;
chirp_on        = 0;
IA_chirp_gain   = 1;
IA_init_freq    = 1;
IA_final_freq   = 1000;
IA_targ_time    = 120;
noise_power     = 0.00;
noise_power2    = 0.00;

cal_ot1x = 0;    cal_ot1y = 0;
cal_ot2x = 0;    cal_ot2y = 0;
cal_ot3x = 0;    cal_ot3y = 0;
cal_ot4x = 0;    cal_ot4y = 0;
cal_ot5x = 0;    cal_ot5y = 0;
% cal_ot6z = 0;    cal_ot6x = 0;
% cal_ot7z = 0;    cal_ot7x = 0;
cal_tgttx = 0;    cal_tgtty = 0;
%Rate Sensors
cal_pitch_rate = 0;
cal_roll_rate = 0;
cal_yaw_rate = 0;
%Accelerometers
r_OA=0.1225; % m
r_OB=0.118;% m
r_OC=0.1235; % m
cal_Ox = 0;
cal_Oy = 0;
cal_Oz = 0;
cal_Ay = 0;
cal_Az = 0;
cal_Bx = 0;
cal_Bz = 0;
cal_Cx = 0;
cal_Cy = 0;
%-----
% PID gains for PI Controller (Control A)
fsm1px = 0.00010; % 1.25*
% fsm1px = (1.0)*0.06*0.45*1.0; % factor of (?)* added 16 July 12 by ROB

fsm1lix = fsm1px*1.45/0.001; %*1.2/0.001
fsm1ldx = 0.000000; % go smaller
fsm1lpy = 0.00010; %1.25*
% fsm1py = 0.2*0.45*0.5;

fsm1liy = fsm1py*1.45/0.001;

```

```

    fsmldy = 0;

    fsm2ix = fsm1px*1.95/0.001;
    fsm2iy = fsm1py*1.95/0.001;
    avg_m1xc = 0;
    avg_m1yc = 0;
    % PI test mode for critical gains
    PI_tune_step_value= 0; %step value
    x_PI_tune = 3; %time for x axis step
    y_PI_tune = 3; %time for y axis step
%-----
%   Test Parameters for sinusoid (max 4 signals)
    x_test_amp= [0.025*1      0      0      0];
    x_test_freq=[1          0      0      0];
    y_test_amp= [0.015*1      0      0      0];
    y_test_freq=[2          0      0      0];
%-----
%   Test Parameters
%   time in sec, value in mrad (max = 13.1 mrad)
    y_step_time = 0.5; y_step_value = -0.3;
    y_step_value = y_step_value*10/26.2; %convert to volts
    x_step_time = 0.5;
    x_step_value = 0.3; x_step_value = x_step_value*10/26.2;
    imp_delay = 1;
    imp_delay=round(imp_delay/Ts); %delay time to impulse in sec
    imp_mag = -0.3;
    imp_mag = imp_mag*10/26.2+0.03*0; % impulse mag in mrad
    init_freq = 1;
    final_freq = 1000; targ_time = 120; %Chirp Parameters
    chirp_gain = 0.262; chirp_gain=chirp_gain*10/26.2;
    calruntime = 10;
%-----

%-----
%Plot Parameters
    plot_time=2.0; %length of plot in seconds
    delay_time=shaker_start+1; %delay before start of example plot
    adapt=2+delay_time+plot_time; %modify adaption to be after delay
    x_plot_bias=200; y_plot_bias=200; %amt to bias example signal
    pbiasy = 300; pbiasx = 300;
    pidstart = (adapt-0.1)+1*0; % PID control start, sec, before adaption
    req_theta_start=pidstart;
    % parallel controllers cmd - 1=single, 2 = parallel A and B
    par_cntlrsA = 1; par_cntlrsB = 1;
%-----
%   STOP EVALUATION HERE!
%-----

%% Load Model
tg=xpctarget.xpc;
C1 = (get(tg, 'Application')); C2='MMM2onecentroid'; C3 = get(tg, 'Connected');
C4 = 'Yes'; C5 = 'loader';
TF1=strcmp(C1, C2); TF2=strcmp(C3, C4); TF3=strcmp(C1, C5);
if ~TF1;
    if ~TF3
        unload(tg);
    end
end

```

```

        load(tg, 'MMM2onecentroid');
    else
        load(tg, 'MMM2onecentroid');
    end
end
if ~TF2
    error('Connection with target cannot be established - aborting');
end

%% First Calibration
reply = input('connect model (if not connected)\nset Jitter Control OFF, Target Tracking OFF\npress enter ');
set_param('MMM2onecentroid', 'SimulationCommand', 'update')
tg.StopTime=999;

+tg
reply1 = input('Set OT5 to position x = 1.074mm and y = -0.582mm...press enter');
reply2 = input('Turn on each beam and check that it is visible on OT5 and its reflection PSD (2,4,6) \n if not adjust flat mirrors on target table and then OT 2,4, or 6...press enter');
-tg
clear tt oo
reply3 = input('Turn on Only the 635nm Beam and Set Mirror 1 to \n Calibrate Feedback (In Jitter Control) and x/y PID Control (Switching)...press enter');
pic = 0;%input('Enter 1 if using PI ctrl: ');
calibrate= 1; %Set to '1' to use Calibration Constants
Ts = 0.001; %sample time
Fs = 1/Ts; %sample Frequency

fintime = 12; %Length of data run
MeanOff = 0; % subtract running mean for HINF
% 1 = on

%-----
% Actuator input (sinusoid, max 4 signals)
% amp in volts, freq in Hz
%-----
sfreq = 10; %Actuator First Frequency
aa = 0; %Make '1' to Run all Frequencies for Actuator 1
bb = 0; %Make '1' to Run all Frequencies for Actuator 2
%-----
%Actuator 1 (Pitch/Yaw)
%First Set of Frequencies
shakeramp = [0 0 0 0];

%Second Set of Frequencies
shakerampb = [0 0 0 0];
%shakerampb = [3 2*0 2*0 0];

% shakerfreqb= [16 25 27 47]; %[sfreq 13
27 47];
% shakerfreqb= [12 25 27 47]; %[sfreq 13
27 47];
shakerfreqb = [0 0 0 0]; %[sfreq 13 27
47];

```

```

% shakerfreqb= [9      25      38      47]; %[sfreq      13
27      47];
%shakerfreqb= [16      22      38      47]; %[sfreq      13
27      47];

shaker_switch=299; % Shaker switch time for frequencies

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% H-infinity axis attention values
attenX = 33;%21.5;
attenY = 34;%21.5;
dattX = 3;
dattY = 3;%.75;
nmrX = 0.1;
nmrY = 0.1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% shakerfreq= [sfreq      16      28      47]; %Moran 10Nov
shakerphase = [0 0 0 0];
shaker_start = 1; %start time of vibrations in secs
shaker_end = fintime;
noise_power = 0.02*aa; %noise power for Band Limited White Noise
% (usually use about 0.01)

%-----
%** YOU DO NOT USE THIS SHAKER 1 CONTROLS BOTH SHAKERS
%Actuator 2 (Roll)
shakeramp2 = [3*0 2*0 1*bb 1*bb];
shakerfreq2 = [10 23 41 51]; %[17 23 41
51];
shakerphase2 = [pi/4*1 pi/3*0 0 0];
shaker_start2 = shaker_start; %start time of vibrations in secs
shaker_end2 = fintime;
noise_power2 = 0.02*bb; %noise power for Band Limited White Noise
% (usually use about 0.01)

%-----
%Chirp Parameters (set chirp_on to 1 for chirp signal, 0 to input freq)
chirp_on = 0; IA_chirp_gain=0.8;
IA_init_freq = 1; IA_final_freq = 150; IA_targ_time = 101; %Chirp
Parameters
%-----
%Distance from Last FSM face to Target
dist_targ = 4.4967; %m
%dist_targ = 8; %m
%Distance from Laser Source to FSM

%dist_FSM = (0.365+0.427)*1.00; %m
dist_FSM = 0;
% dist_FSM = 0.2275*0; %m
%-----
% Distance from Plate's "Center of Rotation" to FSM
w = 0.0635; %m (originally 0.0635)
h = 0.1175; %m (originally 0.1175)
d = 0.3048; %m (originally 0.3048)
%-----
FSM_position = [0,0,0];

```

```

%-----
Run_Mean      = 0;          %Set to '1' to Subtract Running Mean from Rate Sensor
Data
a_Run_Mean    = 0;          %Set to '1' to Subtract Running Mean from Accelerometer
Data
Filter_Mean = 1;          %Set to '1' to Subtract the jitter free "drift" signal
from ARS data

% Target tracking curve fit parameters

%X
%Target tracking slope x
Mx = -0.9212; %(curve fit=-0.9212)
% %Target tracking offset x
Bx = -5.11; %(curve fit = -3.714)

%Y
%Target tracking slope y
My = 0; %(curve fit=)
%Target tracking offset y
By = 0; %(curve fit = )

% First PI Target Tracking Controller (X-Axis)
targtrackxprop = 0.044;
targtrackxint  = .81;
targtrackxderiv = 0;
% First PI Target Tracking Controller (Y-Axis)
targtrackyprop = 0.044;
targtrackyint  = .81;
targtrackyderiv = 0;
% Second PI Target Tracking Controller (X-Axis)
targtrackxprop2 = .26;
targtrackxint2  = 4.6;
targtrackxderiv2 = 0;
% Second PI Target Tracking Controller (Y-Axis)
targtrackyprop2 = .26;
targtrackyint2  = 4.6;
targtrackyderiv2 = 0;

%-----
trigger=2; %Trigger for Beam Profile, 1=Trigger ON, 2=Trigger OFF
% Control Selection:
%-----
% Select Rotations from either PSDs or Rate Sensors for use with Control
% 1 = PSD Calc
% 2 = Rate Sensors (Integration Only)
% 3 = Rate Sensors with Prediction Algorithm
% 4 = Rate Sensor with Accels and Prediction Algorithm
PSD_or_Rate_Sensors = 1;

%-----

```

```

%   Select Target Position Control or Required FSM Theta Control
%       1 = Tgt Position with PI
%       2 = Req Theta
Tgt_Pos_or_Req_Theta = 1;

%-----
%   Select FeedBack or FeedForward Control for use with Target Position
%       Control Above (1 must be selected above)
%       1 = FeedBack; 2 = FeedForward
Back_or_Forward = 1;
if (Back_or_Forward == 2)
    PredFilter = 0;
else
    PredFilter = 0;
end
A_x_ffd_sel = Back_or_Forward; %(x axis at tgt)
A_y_ffd_sel = Back_or_Forward; %(y axis at tgt)
%-----
%   Select Target tracking source
%1=OFF(Beacon Laser on OT4) 2=ON (OT5 feedback)
OT5FBX = 2;
OT5FBY = 2;
trackstart = 1.5; %delay before tracking starts

% Control Parameters:
%-----
% % H-infinity axis attention values
% attenX = 20;%21.5;
% attenY = 20;%21.5;
% dattX = 1;
% dattY = 3;
% nmrX = 0.1;
% nmrY = 0.15;
% PID gains for PI Controller (Control A)
%(Kcr_x=0.0158, Pcr_x=0.002 and y crit gain = 0.031)

if Back_or_Forward == 1;
    fsm1px = 0.0050; % 1.25*
%     fsm1px = (1.0)*0.06*0.45*1.0; % factor of (?)* added 16 July 12 by ROB

    fsm1lx = fsm1px*2.45/0.001; %*1.2/0.001
    fsm1dx = 0.000000; % go smaller
    fsm1py = 0.0050; %1.25*
%     fsm1py = 0.2*0.45*0.5;

    fsm1iy = fsm1py*2.45/0.001;
    fsm1dy = 0;

    fsm2ix = fsm1px*1.95/0.001;
    fsm2iy = fsm1py*1.95/0.001;

elseif Back_or_Forward == 2;
    if PSD_or_Rate_Sensors == 1;
        %PSM ideal gains:
        fsm1px = 0.03*0.45;

```

```

    fsmlix = fsm1px*1.4/0.001;
    fsmldx = 0;
    fsm1py = 0.04*0.45;
    fsmliy = fsm1py*1.5/0.001;
    fsmldy = 0;
    else
        %ARS ideal          PSM ideal          FB ideal
        fsm1px = 0.007*0.45*2.5;          %0.03*0.45          %0.04*0.45
        fsmlix = fsm1px*1.9/0.001; %fsm1px*1.4/0.001 %fsm1px*1.2/0.001
        fsmldx = 0;
        fsm1py = 0.005*0.45*1;          %0.04*0.45          %0.1*0.45
        fsmliy = fsm1py*3.1/0.001; %fsm1py*1.5/0.001 %fsm1py*1.2/0.001
        fsmldy = 0;
    end
end

% Use these for tuning the PI Controller
PI_tune_step_value = 0; %step value
x_PI_tune          = 3; %time for x axis step
y_PI_tune          = 3; %time for y axis step
%-----
% LMS parameters for LMS Controller (Control B)
% mux=0.012; leakx=1; % x axis adaption rate and leakage factor
% muy=0.020; leaky=1; % y axis adaption rate and leakage factor
mux          = 0.012; leakx          = 0.999; % x axis adaption rate
and leakage factor
muy          = 0.018; leaky          = 0.998; % y axis adaption rate
and leakage factor
w0x          = 0; w0y          = 0; % initial tap gains
biasx        = 0.005*1; biasy        = 0.005*1; % estimate of bias
correction
ax_to_mx     = 1; ay_to_my = 14; % estimate of gain
correction for FSM to accel
ot2y_to_m2y  = -1/10;
mu_y_error   = 0.05; leak_y_error = 1.0;
adapt_y_error = 0.0;
mu_x_error   = 0.05; leak_x_error = 1.0;
%-----
% Reference Signal Selection
% 1=OT-1, 2=Accel-2 (Bx and Ay), 3 = rate sensor (pitch, roll)
% 4 = Az 5 = Bz
x_ref_sel=2; y_ref_sel=2;
zz=1; % number of delays for the predictor ref signal
%-----
% Error source selection
% 1=mirror postion, 2=OT3 position, 3=OT2 position
%
x_error_sel = 2; y_error_sel=2;
accel_lag   = 1.05;
OT2y_lag    = 1;
%-----
% parallel controllers cmd - 1=single, 2 = parallel A and B
par_cntlrsA = 1; par_cntlrsB = 1;
%-----
% Test Parameters for sinusoid (max 4 signals)
% amplitude in Volts, frequency in Hz
x_test_amp  = [0.02 0 0 0];

```

```

x_test_freq =[2      0      0      0];
y_test_amp  = [0.02   0      0      0];
y_test_freq =[1      0      0      0];
% time in sec, value in mrad (max = 13.1 mrad)
y_step_time = 1;
y_step_value = 0.1; y_step_value = y_step_value*10/26.2; %convert to
volts
x_step_time = 1;
x_step_value = 0.1; x_step_value = x_step_value*10/26.2;
imp_delay = 1; imp_delay=round(imp_delay/Ts); %delay time to impulse
in sec
imp_mag = -0.3; imp_mag = imp_mag*10/26.2+0.03*0; % impulse mag in
mrad
init_freq = 1; final_freq = 1000; targ_time = 120; %Chirp Parameters
chirp_gain = 0.262; chirp_gain=chirp_gain*10/26.2;
stepOTxstart = 1;
stepOTystart = 1;

% FSM_Acal_x = 2.62*1.3; FSM_Acal_y = 2.62*1.40;
FSM_Acal_x = 2.62*1; FSM_Acal_y = 2.62*1;
set_param('MMM2onecentroid', 'SimulationCommand', 'update')
tg.StopTime=fintime;
tg = xpctarget.xpc
+tg
pause(fintime+0.5);
-tg
% Calculate Mirror 1 Calibration Values Based On Average of Final 2 seconds
% Of Control Efforts From PID Control (Feedback from Target OT5)
mirror1xcal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,3));
mirror1ycal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,4));
% Calculate New Reference Point on OT2 for "Zero"
% This Makes OT2's X And Y Zeroes match OT5's
OT2Xcal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,23));
OT2Ycal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,24));
reply4 = input('Turn Mirror 1 Back to Control Feedback (Jitter Control)
and Turn Its Control Off...press enter');
reply5 = input('Turn on Only the 670nm Beam and Set Mirror 2 to \n
Calibrate Feedback (In Jitter Control) and x/y PID Control
(Switching)...press enter');
set_param('MMM2onecentroid', 'SimulationCommand', 'update')
tg.StopTime=fintime;
tg = xpctarget.xpc
+tg
pause(fintime+0.5);
-tg
mirror2xcal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,5));
mirror2ycal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,6));
OT6Xcal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,15));
OT6Ycal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,16));
reply6 = input('Turn Mirror 2 Back to Control Feedback (Jitter Control)
and Turn Its Control Off...press enter');
reply7 = input('Turn on Only the 1064nm Beam and Set Mirror 3 to \n
Calibrate Feedback (In Jitter Control) and x/y PID Control
(Switching)...press enter');
set_param('MMM2onecentroid', 'SimulationCommand', 'update')
tg.StopTime=fintime;
tg = xpctarget.xpc

```

```

+tg
pause(fintime+0.5);
-tg
mirror3xcal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,21));
mirror3ycal = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,22));
OT4Xcal      = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,25));
OT4Ycal      = mean(tg.OutputLog(((fintime-2)*(1/Ts)):end,26));

% Set Shakers to Shake For Shift And Scale Calibration of Reflected PSD's
% (OT2,4,6)
shakeramp = [3      2*0      0      0];
shakerfreq= [17      25      37      43];

reply8 = input('Turn on Only the 635nm Beam and Set Mirror 1 to \n
Calibrate Feedback (In Jitter Control) and x/y PID Control
(Switching)...press enter');
set_param('MMM2onecentroid', 'SimulationCommand', 'update')
    tg.StopTime=fintime;
tg = xpctarget.xpc
+tg
pause(fintime+0.5);
-tg

% Define Required Data From Speedgoat Log For Shift And Scale of ot2 x/y
ot2x      = tg.OutputLog(:,23);
ot2y      = tg.OutputLog(:,24);
ot5x2     = tg.OutputLog(:,1);
ot5y2     = tg.OutputLog(:,2);

% Calculate Required Shift and Scale Values for OT2x to Match OT5x
shift2x    = mean(ot5x2)-mean(ot2x);
ot2xshift  = ot2x+shift2x;
scale2x    = 1.0;
for i = 1001:12001;
    if ot5x2(i) == 0 || ot2xshift(i) == 0
        scale2x = 1.0;
    else
        scale2x = (abs(ot2xshift(i))/abs(ot5x2(i)));
    end
    scaling2x(i,1)=scale2x;
end
scale2x = mean(scaling2x);
ot2xshiftscale = ot2xshift*scale2x;

% Calculate Required Shift and Scale Values for OT2y to Match OT5y
shift2y    = mean(ot5y2)-mean(ot2y);
ot2yshift  = ot2y+shift2y;
scale2y    = 1.0;
for i = 1001:12001;
    if ot5y2(i) == 0 || ot2yshift(i) == 0
        scale2y = 1.0;
    else
        scale2y = (abs(ot2yshift(i))/abs(ot5y2(i)));
    end
    scaling2y(i,1)=scale2y;
end

```

```

scale2y = mean(scaling2y);
ot2yshiftscale = ot2yshift*scale2y;

reply9      = input('Turn Mirror 1 Back to Control Feedback (Jitter Control)
and Turn Its Control Off...press enter');
reply10     = input('Turn on Only the 670nm Beam and Set Mirror 2 to \n
Calibrate Feedback (In Jitter Control) and x/y PID Control
(Switching)...press enter');
set_param('MMM2onecentroid', 'SimulationCommand', 'update')
    tg.StopTime=fintime;
tg = xpctarget.xpc
+tg
pause(fintime+0.5);
-tg

% Define Required Data From Speedgoat Log For Shift And Scale of ot6 x/y
ot6x      = tg.OutputLog(:,15);
ot6y      = tg.OutputLog(:,16);
ot5x6     = tg.OutputLog(:,1);
ot5y6     = tg.OutputLog(:,2);

% Calculate Required Shift and Scale Values for OT6x to Match OT5x
shift6x    = mean(ot5x6)-mean(ot6x);
ot6xshift  = ot6x+shift6x;
scale6x    = 1.0;
for i = 1001:12001;
    if ot5x6(i) == 0 || ot6xshift(i) == 0
        scale6x = 1.0;
    else
        scale6x = (abs(ot6xshift(i))/abs(ot5x6(i)));
    end
    scaling6x(i,1)=scale6x;
end
scale6x = mean(scaling6x);
ot6xshiftscale = ot6xshift*scale6x;

% Calculate Required Shift and Scale Values for OT6y to Match OT5y
shift6y    = mean(ot5y6)-mean(ot6y);
ot6yshift  = ot6y+shift6y;
scale6y    = 1.0;
for i = 1001:12001;
    if ot5y6(i) == 0 || ot6yshift(i) == 0
        scale6y = 1.0;
    else
        scale6y = (abs(ot6yshift(i))/abs(ot5y6(i)));
    end
    scaling6y(i,1)=scale6y;
end
scale6y = mean(scaling6y);
ot6yshiftscale = ot6yshift*scale6y;

reply11    = input('Turn Mirror 2 Back to Control Feedback (Jitter Control)
and Turn Its Control Off...press enter');

```

```

reply12      = input('Turn on Only the 1064nm Beam and Set Mirror 3 to \n
Calibrate Feedback (In Jitter Control) and x/y PID Control
(Switching)...press enter');

set_param('MMM2onecentroid', 'SimulationCommand', 'update')
    tg.StopTime=fintime;
tg = xpctarget.xpc
+tg
pause(fintime+0.5);
-tg

% Define Required Data From Speedgoat Log For Shift And Scale of ot4 x/y
ot4x      = tg.OutputLog(:,25);
ot4y      = tg.OutputLog(:,26);
ot5x4     = tg.OutputLog(:,1);
ot5y4     = tg.OutputLog(:,2);
tt        = tg.TimeLog;
% Calculate Required Shift and Scale Values for OT4x to Match OT5x
shift4x   = mean(ot5x4)-mean(ot4x);
ot4xshift = ot4x+shift4x;
scale4x   = 1.0;
for i = 1001:12001;
    if ot5x4(i) == 0 || ot4xshift(i) == 0
        scale4x = 1.0;
    else
        scale4x = (abs(ot4xshift(i))/abs(ot5x4(i)));
    end
    scaling4x(i,1)=scale4x;
end
scale4x = mean(scaling4x);
ot4xshiftscale = ot4xshift*scale4x;

% Calculate Required Shift and Scale Values for OT4y to Match OT5y
shift4y   = mean(ot5y4)-mean(ot4y);
ot4yshift = ot4y+shift4y;
scale4y   = 1.0;
for i = 1001:12001;
    if ot5y4(i) == 0 || ot4yshift(i) == 0
        scale4y = scale4y;
    else
        scale4y = (abs(ot4yshift(i))/abs(ot5y4(i)));
    end
    scaling4y(i,1)=scale4y;
end
scale4y = mean(scaling4y);
ot4yshiftscale = ot4yshift*scale4y;

% Comparison Plots
%OT6x
figure(1)
title('OT6x Shift/Scale vs. OT5x');
plot(tt,500*ot6xshiftscale,'r',tt,500*ot5x6,'b');
hold on
plot(tt,500*(ot6xshiftscale-ot5x6).^2,'g');
legend('OT6x ShiftScale','OT5x','Error^2');
xlabel('Time (sec)');

```

```

ylabel('X Position (micro-meters)');

figure(2)
title('OT4y Shift/Scale vs. OT5y');
plot(tt,500*ot4yshiftscale,'r',tt,500*ot5y4,'b');
hold on
plot(tt,500*(ot4yshiftscale-ot5y4),'g');
legend('OT6x ShiftScale','OT5x','Error^2');
xlabel('Time (sec)');
ylabel('X Position (micro-meters)');

```

10.2. Pointing Algorithm Code

```

% function
[mlxcentercmd,mlycentercmd,m2xcentercmd,m2ycentercmd,m3xcentercmd,m3ycentercm
d,areacontrol,cent1xiddat,cent1yiddat,cent2xiddat,cent2yiddat,cent3xiddat,ce
nt3yiddat,m1corr,m2corr,m3corr,fram,cent1x,cent1y,cent2x,cent2y,cent3x,cent3y,
cent1xstartdat,cent1ystartdat,cent2xstartdat,cent2ystartdat,cent3xstartdat,ce
nt3ystartdat]= fcn(area,centroids,count)
function
[mlxcentercmd,mlycentercmd,m2xcentercmd,m2ycentercmd,m3xcentercmd,m3ycentercm
d,areacontrol,xlength,ylength,cent1x,cent1y,area1]=
fcn(area,centroids,axes,orientation)
persistent framenum cent1xstart cent1ystart cent2xstart cent2ystart
cent3xstart cent3ystart cent1xid cent1yid cent2xid cent2yid cent3xid cent3yid
cent1xid2 cent1yid2 cent2xid2 cent2yid2 cent3xid2 cent3yid2 mlxcenter
mlycenter m2xcenter m2ycenter m3xcenter m3ycenter areactrl m1cor m2cor m3cor

W = 300;
H = 225;

if isempty(framenum)
    framenum = 0;
    cent1xstart = 0;
    cent1ystart = 0;
    cent2xstart = 0;
    cent2ystart = 0;
    cent3xstart = 0;
    cent3ystart = 0;
    cent1xid = 0;
    cent1yid = 0;
    cent2xid = 0;
    cent2yid = 0;
    cent3xid = 0;
    cent3yid = 0;
    cent1xid2 = 0;
    cent1yid2 = 0;
    cent2xid2 = 0;
    cent2yid2 = 0;
    cent3xid2 = 0;
    cent3yid2 = 0;

```

```

    mlxcenter    = 0;
    mlycenter    = 0;
    m2xcenter    = 0;
    m2ycenter    = 0;
    m3xcenter    = 0;
    m3ycenter    = 0;
    areactrl     = 1;
    mlcor        = 0;
    m2cor        = 0;
    m3cor        = 0;
end
%%Three Distinct Beam Case
% Get Centroid Positions
if framenum > 4 && framenum <= 15
    cent1xstart = centroids(1,1)-(W/2);
    cent1ystart = -1*(centroids(1,2)-(H/2));
    cent2xstart = centroids(2,1)-(W/2);
    cent2ystart = -1*(centroids(2,2)-(H/2));
    cent3xstart = centroids(3,1)-(W/2);
    cent3ystart = -1*(centroids(3,2)-(H/2));
    framenum = framenum + 1;

% Move Mirror 1x To ID beam
elseif framenum > 15 && framenum <= 30
    framenum = framenum + 1;
    mlxcenter = 0.025;

elseif framenum > 30 && framenum <= 200
    framenum = framenum + 1;
    cent1xid = (cent1xid+(centroids(1,1)-(W/2)))/2;
    cent1yid = (cent1yid+(-1*(centroids(1,2)-(H/2))))/2;
    cent2xid = (cent2xid+(centroids(2,1)-(W/2)))/2;
    cent2yid = (cent2yid+(-1*(centroids(2,2)-(H/2))))/2;
    cent3xid = (cent3xid+(centroids(3,1)-(W/2)))/2;
    cent3yid = (cent3yid+(-1*(centroids(3,2)-(H/2))))/2;

elseif framenum > 200 && framenum <= 201
    framenum = framenum + 1;
    if abs(cent1xid-cent1xstart) > 4
        mlcor = 1;
        mlxcenter = mlxcenter+ (-1*cent1xid)/800;
        mlycenter = (-1*cent1yid)/800;
    end
    if abs(cent2xid-cent2xstart) > 4
        mlcor = 2;
        mlxcenter = mlxcenter + (-1*cent2xid)/800;
        mlycenter = (-1*cent2yid)/800;
    end
    if abs(cent3xid-cent3xstart) > 4
        mlcor = 3;
        mlxcenter = mlxcenter + (-1*cent3xid)/800;
        mlycenter = (-1*cent3yid)/800;
    end
end

elseif framenum > 201 && framenum <=225;
    m2ycenter = -1*0.025;

```

```

    framenum = framenum + 1;

elseif framenum > 225 && framenum <= 400;
    framenum = framenum + 1;
    cent1xid2 = (cent1xid2+(centroids(1,1)-(W/2)))/2;
    cent1yid2 = (cent1yid2+(-1*(centroids(1,2)-(H/2))))/2;
    cent2xid2 = (cent2xid2+(centroids(2,1)-(W/2)))/2;
    cent2yid2 = (cent2yid2+(-1*(centroids(2,2)-(H/2))))/2;
    cent3xid2 = (cent3xid2+(centroids(3,1)-(W/2)))/2;
    cent3yid2 = (cent3yid2+(-1*(centroids(3,2)-(H/2))))/2;

elseif framenum > 400 && framenum <= 401;
    framenum = framenum + 1;
    if abs(cent1yid2-cent1ystart) > 4 && m1cor ~= 1;
        m2cor = 1;
        m2xcenter = m2xcenter + (-1*cent1xid)/950;
        m2ycenter = m2ycenter + (-1*cent1yid)/950;
    end
    if abs(cent2yid2-cent2ystart) > 4 && m1cor ~= 2;
        m2cor = 2;
        m2xcenter = m2xcenter + (-1*cent2xid)/950;
        m2ycenter = m2ycenter + (-1*cent2yid)/950;
    end
    if abs(cent3yid2-cent3ystart) > 4 && m1cor ~=3;
        m2cor = 3;
        m2xcenter = m2xcenter + (-1*cent3xid)/950;
        m2ycenter = m2ycenter + (-1*cent3yid)/950;
    end
    if (m1cor == 1 && m2cor == 2) || (m1cor == 2 && m2cor == 1);
        m3cor = 3;
        m3xcenter = (-1*cent3xid)/950;
        m3ycenter = (-1*cent3yid)/2000;
    end
    if (m1cor == 1 && m2cor == 3) || (m1cor == 3 && m2cor == 1);
        m3cor = 2;
        m3xcenter = (-1*cent2xid)/950;
        m3ycenter = (-1*cent2yid)/2000;
    end
    if (m1cor == 2 && m2cor == 3) || (m1cor == 3 && m2cor == 2);
        m3cor = 1;
        m3xcenter = (-1*cent1xid)/950;
        m3ycenter = (-1*cent1yid)/2000;
    end

    areactrl = 2;

else
    framenum = framenum + 1;

end
m1xcentercmd = m1xcenter;
m1ycentercmd = m1ycenter;
m2xcentercmd = m2xcenter;
m2ycentercmd = m2ycenter;
m3xcentercmd = m3xcenter;
m3ycentercmd = m3ycenter;

```

```

areacontrol = areactrl;
blobarea    = area(1,1);
xlength     = abs(axes(1,1)*cos(orientation(1,1)));
ylength     = abs(axes(1,1)*sin(orientation(1,1)));
area1       = area(1,1);
%   cent1xiddat = cent1xid;
%   cent1yiddat = cent1yid;
%   cent2xiddat = cent2xid;
%   cent2yiddat = cent2yid;
%   cent3xiddat = cent3xid;
%   cent3yiddat = cent3yid;
%   cent1xstartdat = cent1xstart;
%   cent1ystartdat = cent1ystart;
%   cent2xstartdat = cent2xstart;
%   cent2ystartdat = cent2ystart;
%   cent3xstartdat = cent3xstart;
%   cent3ystartdat = cent3ystart;
cent1x = centroids(1,1)-(W/2);
cent1y = -1*(centroids(1,2)-(H/2));
%   cent2x = centroids(2,1)-(W/2);
%   cent2y = -1*(centroids(2,2)-(H/2));
%   cent3x = centroids(3,1)-(W/2);
%   cent3y = -1*(centroids(3,2)-(H/2));
%   m1corr = m1cor;
%   m2corr = m2cor;
%   m3corr = m3cor;
%   fram    = framenum;

```

Works Cited (In Order Cited)

Steele, Jeanette. Navy Unveils Laser Weapon for Ships. 08 April 2013. 2013.

Perram, Glen P., et al. An Introduction to Laser Weapon Systems

ONR. Solid State Laser. 10 Feb 2012. <http://www.onr.navy.mil/en/Media-Center/Fact-Sheets/Solid-State-Fiber-Laser.aspx>

Pawlak, Robert J., and Stephen R. Horman. Laser Weapons System (LaWS) Demonstrator

Perram, Glen P., et al. An Introduction to Laser Weapon Systems

Pawlak, Robert J., and Stephen R. Horman. Laser Weapons System (LaWS) Demonstrator

O'Brien, Richard T., Jr., R. Joseph Watkins, *Adaptive H-infinity Vibration Control*

O'Brien, Richard T, and R.J. Watkins. *H ∞ Jitter Control for a Platform-Mounted Laser*

Moran, Shane. An Adaptive H-Infinity Algorithm for Jitter Control and Target Tracking in a Directed Energy Weapon. USNA, 2012.

4-Channel Fiber-Coupled Laser Source. (n.d.). Retrieved January 2013, from THORLABS: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=3800

(n.d.). Retrieved January 2013, from Optics In Motion: <http://www.opticsinmotion.net/OIM102-3%20INFO.pdf>

FiberPort. (n.d.). Retrieved January 2013, from ThorLabs: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2940

FiberPort. (n.d.). Retrieved January 2013, from ThorLabs: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2940

PSD Theory. (n.d.). Retrieved January 2013, from On-Trak Photonics: <http://www.on-trak.com/theory.html>

C11440. (n.d.). Retrieved January 2013, from Hamamatsu: <http://sales.hamamatsu.com/index.php?id=13226509>