NORTHERN ARIZONA UNIVERSITY

ASTRAEA

NAVY PRECISION OPTICAL INTERFEROMETER

Requirements Specification

Version: 2.0

Team Members	*Clients*	*Mentor*
Adam Schilperoort	Jim Clark (NRL)	Isaac Shaffer
Brandon Horner	Teznie Pugh (Lowell)	
Michael Partridge		
Peter Kurtz		
Trey Tangeman		

Approved: ________________	Approved: ________________
Date: ____________________	Date: ____________________
Client	Astraea

December 11, 2019

Contents

1 Introduction

Improving astrometry, or the measurement of position, motion, and magnitude of stars, is a never-ending goal of astronomers and astrophysicists around the world. By building devices which can achieve higher precision astrometrics, scientists can improve upon previous models of stars and develop a more accurate representation of the cosmos. Breakthroughs in this area could lead to a better understanding of the universe, or potentially redefine the laws of physics altogether.

In the twentieth century, improvements in astrometry enabled physicists such as Albert Einstein and Stephen Hawking to make revolutionary contributions to the field of astrophysics. Following in these early innovator's footsteps, modern day astrophysicists are researching the properties of dark matter, modelling black holes, and searching for exoplanets. Improvements in observational technology could drastically accelerate research in these areas. Not only could this technology benefit scientists, but also the general public — for example, accurate star charts provide precise positioning for GPS satellites, which helps improve navigation around the planet.

From exploring foreign worlds to answering humanity's biggest questions, the impact of astrometry cannot be understated. Unfortunately, in the past few decades modern astrometry reached a point of diminishing returns, making it exponentially more expensive to improve on the existing technology. As a result, an alternative to the conventional single-aperture telescope design needed to be developed. Navy Precision Optical Interferometer (NPOI) tackled this design problem by developing an optical telescope system which utilizes multiple telescopes receiving images simultaneously, combining their images to achieve a much larger telescope. This design is capable of providing the highest angular resolution on Earth of any optical design at a significantly lower cost than equivalent conventional designs.

NPOI, an astronomical long-baseline optical interferometer has been in operation on Anderson Mesa, outside of Flagstaff, Arizona, since 1994. The 437 meter baseline array has a unique capacity for detecting and determining motions and orbits of binary systems, which is its current research focus. The site's unique precision is achieved through a high-tech mirror control system which simultaneously gathers light from three to six telescopes separated by some distance and combines their respective images for analysis. NPOI's partners are Lowell Observatory, United States Naval Observatory (USNO), and Navy Research Laboratory (NRL). NRL's goal was to develop technology able to measure the positioning of stars better than any other instrument on Earth to allow precise navigation from stars in the event GPS systems fail. USNO became involved to generate star charts for astronomy and astrophysics analysis. Lowell joined due to the unique ability to study binary star systems.

2 Problem Statement

Currently at NPOI, there are several spots on the array where a siderostat station can be placed. A siderostat station consists of a flat siderostat mirror and a smaller Narrow Angle Tracker (NAT) mirror. During observations, star light is reflected off the siderostat to the NAT then into the pipes on the array to be combined with starlight from other siderostat stations. The siderostat mirrors are moved by stepper motors, motors that move in discrete steps. The NAT mirrors are pushed with piezoelectric actuators. The mirrors at each of these stations are controlled by racks of computers. These computer racks act as the intermediary between an observer and a siderostat station. When the observer wants to move the mirrors to look at a different object in the night sky, the observer has to send a command to a specific siderostat station, as shown in Fig. 1, where a computer known as SidCon will pick up the signal over an optical fiber network. The command is then passed to another computer, PowerCon, where computations are performed to convert the command sent by the observer to a small voltage. The small voltage is then passed to either a Parker microstepper or an Ultravolt piezoelectric actuator. The Parker microstepper amplifies the signal which is then sent to a stepper motor on a siderostat. The Ultravolt piezoelectric actuator amplifies the signal from 0 to 5 volts to 1000 to 0 volts in a negative correspondence. The signal is then sent to a piezoelectric actuator attached to the NAT.

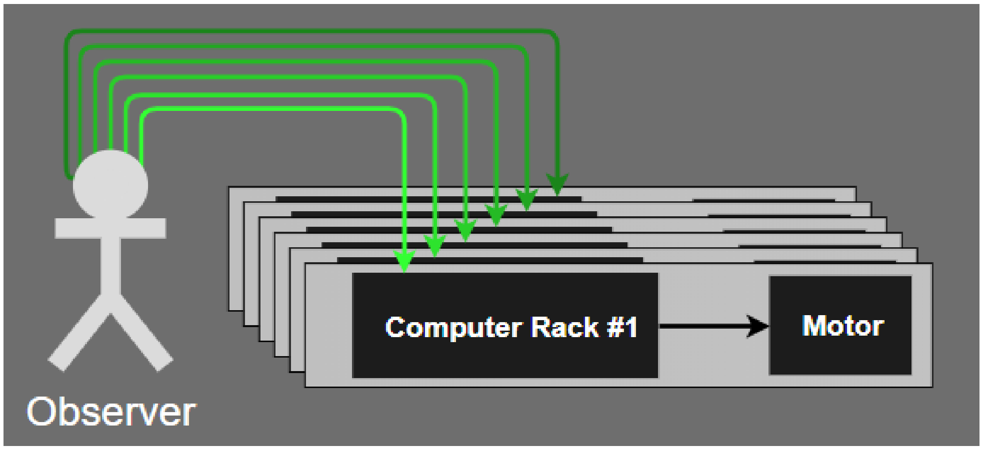


Figure 1: Current observer-station interaction

There are several major issues plaguing NPOI's siderostat stations currently. These problems fall into two different categories: hardware and software.

The current hardware at each siderostat station is between 20 to 30 years old. This antiquated hardware is also custom-made, and the individuals who made the hardware are no longer available at NPOI. One of the side effects of the custom hardware is that there is poor insulation. Some of the hardware components need to be isolated away from other hardware to avoid cross communication between electronic signals. This can cause loss of precision in the system as the mirrors move more or less and are not actually at the location the observer specified. Over the past few years there has been an increase in hardware failure rates. Whenever a piece of the custom hardware failed, it would be replaced with a prefabricated backup. As of the end of 2019, the last prefabricated backup has been used; the next hardware failure at a siderostat station cannot be fixed and observations will cease.

On the software front, each operational siderostat station has a computer rack. Each rack has its own operating system, which is some Linux distribution. There is no standardized version or distribution of Linux on every rack. Although there are different versions of Linux on the racks, each rack contains its own copy of the software that handles receiving the signal from the observer and the operations to convert the signal from the observer to the required output for the Parker microstepper or the Ultravolt piezoelectric actuator. This software is bloated with unnecessary lines of code, causing the software to be heavy and redundant. This becomes cumbersome on the system when any updates are made to the software, as any updates have to be done physically at each station which costs time and money.

There are several key items in the siderostat control system that are in dire need of updating or replacement:

- Failing hardware
- Hard to replace custom hardware
- Redundant software

With these key items, the following sections will discuss the solutions.

3 Solution Vision

Given the state of the hardware systems at NPOI, the team have been tasked with creating a solution that aims to update the existing systems. Many of the problems stem from the fact that the hardware is custom made, outdated, and is redundant across all stations. The many computers at each station that handle observer commands are going to be replaced with a single, centralized computer built with modern consumer hardware, or what is going by the name BrainCon.

The job of BrainCon is to host a server over the network at NPOI from an environmentally secure location, and then handle observer commands to operate the siderostat and NAT at each station – all from a single location.

BrainCon will achieve the operation of the siderostat and NAT by communicating with a microprocessor at the respective station, which will have the software necessary to control the micro steppers on the siderostat and the piezoelectric actuators on the NAT. The microprocessor will go by the name NeuronCon and is envisioned to be a Raspberry Pi.

BrainCon aims to solve the problems at NPOI by replacing the many-to-many software system with a one-to-many system. This involves replacing the custom motherboards and proprietary PCI cards that effectively control the siderostat stepper motors and piezoelectric actuators on the NAT. The new software system will alleviate software redundancy and allow for easier troubleshooting and maintenance of the system. On the hardware end, BrainCon and NeuronCon will remove the failing custom hardware and reduce the points of failure.

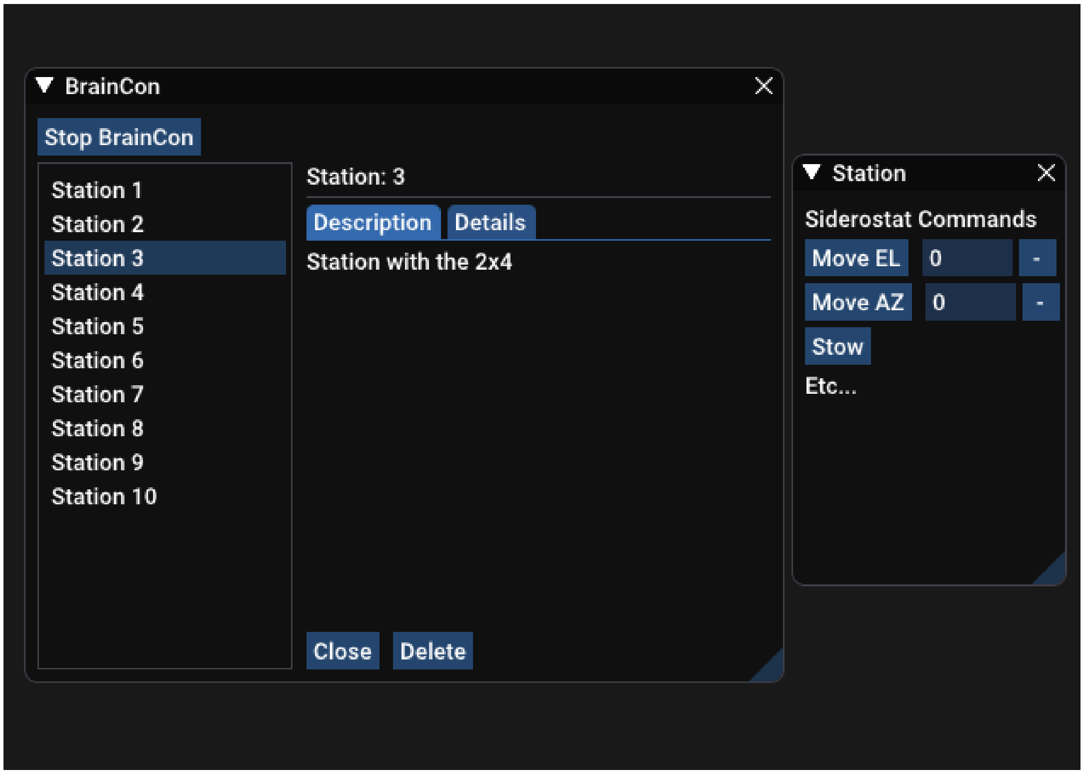
In terms of specifics, BrainCon will aim to have the following functionality:

- Communicate back and forth with the observer over a server, providing feedback and receiving commands.
- Communicate back and forth with the NeuronCon at each respective station over a server, forwarding commands and receiving feedback.
- Provide a graphical user interface, which will have the system status and interface for providing commands.

BrainCon is the central point of command in the system, where NeuronCon drives the motors. The functionality of a NeuronCon needs to be described as well:

- Listen for commands over the network.
- Drive the stepper motors on the siderostat by sending signals to the Parker micro stepping driver.
- Drive the piezoelectric actuators on the NAT by outputting appropriate voltages, which then get stepped up.
- Create a feedback loop between NeuronCon and the limit switches, sending the feedback to BrainCon. This allows for BrainCon to adjust the siderostat and provide its own soft limits.

With this envisioned solution in mind, a prototype of what the graphical user interface for BrainCon might look like has been developed:



Station: 3

Station with the 2x4

Move EL: 0

Move AZ: 0

Figure 2: Prototype graphical user interface

As seen in the figure, BrainCon will allow the user to interact with each individual station and manage the commands that are sent. Since this is just a prototype, this iteration of the user interface has less than the bare minimum functionality needed.

4 Project Requirements

In this section, the functional, performance, and environmental constraints will be discussed. These requirements will outline the details of how the envisioned solution will look and the kind of functionality it will provide.

Since the envisioned solution is effectively split into two discrete pieces of hardware and software, the requirements for BrainCon and NeuronCon will be discussed separately. BrainCon handles the communication and direction of commands, so the software implications will be discussed there. On the other hand, the technology on NeuronCon is very low-level, so the hardware specifics for driving the hardware at each station will be discussed.

4.1 BrainCon

Given that BrainCon is an off-the-shelf computer that handles observer commands and communicates with the NeuronCon at each observing station, these are the functional requirements that were determined to be necessary. Here is how they expand out into other requirements:

1. Network with observer
    a. TCP server hosted on BrainCon
        i. Network wrapper API
            1. Persistent connections
            2. Modular design
        ii. Multi-threaded network
            1. Job system
            2. Thread pool
            3. Thread safety
    b. Graphical user interface
        i. Graphical user interface API
        ii. User interface design
            1. Configuration
            2. Individual station control
2. Send commands to NeuronCon
    a. BrainCon job creation
    b. Packet handling
        i. Parse destination and command
        ii. Data Processing
            1. Handle conversion to stepper motor counts
        iii. Handle connection to NeuronCon
            1. Convert destination to IP address
3. Handle feedback loops
    a. Designated thread listening for feedback
    b. Handle the feedback
        i. Error detection
            1. Issue commands to stop failing system
        ii. Keep track of metrics reported by feedback loops
        iii. Produce system status reports on user interface

Now that the baseline functional requirements for BrainCon have been established, the details on how these functions might be implemented and how they will interface with the user will be discussed.

4.1.1 Network with observer and AlignCon

Networking with the observer and AlignCon are major functions that BrainCon provides. Networking with the observer from BrainCon will define how a user actually interacts with the system. As a result, this has caused this requirement to expand out into a few other requirements:

1. TCP server hosted on BrainCon

The type of server protocol that will be used to network not just with the observer, but with all the stations as well, is the TCP protocol. This entails that the BrainCon software will be fairly low-level networking code, as no framework is being used.

A TCP server will be created; in particular, a network wrapper API will be programmed, which allows for ease of use and modularity in the code. The modularity means that networking with a NeuronCon can be done at a station or the observer computer. In either scenario, the implementation would stay the same.

The network wrapper API will simply take an IP address and a port and create and bind a socket to listen and accept requests on. This is where the importance of TCP presents itself. Having the ability to keep a connection between, for example, a station and BrainCon means packets can freely be sent back and forth – effectively creating a feedback loop.

On top of having this network wrapper API, the TCP server on BrainCon will be multi-threaded. BrainCon is going to use a thread pool system for multi-threading. Essentially, there will be a job queue that the main thread – the thread which will receive packets on the TCP socket – will enqueue a job on based off the type of packet received. Then, on the job queue, there will be a number of threads, or a thread pool, waiting to dequeue a job and handle it. This system will allow for standardization of handling and processing data, while doing so on multiple threads and in a safe way, making networking between many stations and the observer trivial.

2. Graphical user interface

Given that BrainCon will have this networking API, which is multi-threaded and allows for networking between all the stations and the observer, the missing link is the method of interaction between BrainCon and the observer. In order to provide a user interface that is easy-to-use and does not require programming experience, a graphical user interface will be created.

The first requirement for creating a graphical user interface is deciding on a graphical user interface API. The BrainCon graphical user interface will use a lightweight, immediate-mode API. The API provides the necessary ability to draw graphical user interface but does so without needing a massive framework or IDE to compile.

Using the graphical user interface API, BrainCon will need to be able to provide the interface necessary for the observer to submit commands and see system status. The interface will have a fairly simple design, where there will be a configuration button that allows the user to add, delete, and edit stations in a configuration file hosted by BrainCon. There will be a new window created for each station.

For each window, which corresponds to a station, the user can then select a type of command to submit to it, which will be listed as a grid of buttons with the name of each command. The BrainCon software will then receive a command and process it and redirect it to the appropriate station. Some examples of command are: stow siderostat, move EL+/- (elevation positive or negative), move AZ+/- (move azimuth positive/negative).

In terms of status reporting, the graphical user interface will report a few metrics, which it calculated based on feedback from NeuronCon. To begin with, each station window will have a light, which toggles between red and green, that will indicate whether a connection can be established to the desired stations selected. Next, the user interface will have few text boxes that list the following metrics for each station:

- Stepper motor counts
- Siderostat position
- Station status

The stepper motor count will be a simple integer which will be received from NeuronCon. The siderostat position is a more complex metric, which will require BrainCon to compute the position of the siderostat based on feedback from NeuronCon. The status of the station will be a simple message, such as “OK,” or an error message reporting that there is an issue.

4.1.2 Send commands to NeuronCon

The next piece to the BrainCon software is the communication interface between BrainCon and NeuronCon. Since the network interface was outlined above, this section will only discuss the additional functionality required to send commands to NeuronCon.

1. BrainCon job creation

The main hurdle for achieving communication between BrainCon and NeuronCon depends on the processing of packets received on BrainCon. Every time BrainCon obtains a packet, it will need to parse the packet to determine the type of command and determine if any data processing needs to occur before sending out the command. The workflow of packet handling will be discussed in this section.

After a packet is received by BrainCon over the network, it will first push the packet as a job onto the job queue. The job will contain the packet data and the destination, or which station the commands want to be redirected to. A single worker thread will then pull the job off the job queue, where the worker thread will then proceed to handle the packet.

2. Packet handling

The first step in packet handling is parsing the packet. Parsing the packet involves separating the packet header from the packet data. For the most part, the packet header will be ignored, as BrainCon is not using a predefined protocol, such as HTTP, but instead using its own packet structure. Inside the packet data, the type of command will be parsed out, along with its parameters. For example, a command that states to move the siderostat five counts in elevation would translate to a packet type of elevation move and a packet parameter of five counts.

Once the packet is parsed, the worker thread will determine if any computations are needed based on the command type. For example, the worker thread would need to translate an elevation move of five counts to a number of counts for the stepper motor, as well as pack which motor the counts needs to be done on in the output packet. With this determined, the worker thread will take the destination passed to it in the job queue and create an outgoing packet. The outgoing packet will contain at least two fields: the number of motors to step the stepper motor and which axis the counts correspond to.

Finally, the worker thread will send the outgoing packet to the respective NeuronCon by establishing a connection to the IP address associated with the destination, which will be stored inside the BrainCon configuration file. NeuronCon will receive the packet and continue from there. See the NeuronCon section to see how it will handle commands from BrainCon.

4.1.3 Handle feedback loops

The final high-level functional requirement for BrainCon is receiving feedback from NeuronCon and other systems such as AlignCon, which provides the feedback for the angular position of the NAT. This will involve listening for feedback from the devices and handling it. This functional requirement expands straightforwardly:

1. Designated thread listening for feedback

Rather than having each thread that is handling a command listen for feedback after sending a packet, a designated thread will listen on a socket on BrainCon. Having a single thread assigned to this task allows for NeuronCon, for example, to send feedback as desired, since there is not a one-to-one correlation between sending a command and waiting for feedback.

This functionality will be implemented by using the network API that was outlined in the first functional requirement for BrainCon. As it is designed modularly, BrainCon simply needs to configure the IP addresses that correspond to each feedback loop in the system. BrainCon can then simply listen for packets, push the packet data as a job to be handled by the other worker threads, and immediately go back to listening for feedback. Using this workflow, BrainCon can maintain a state of the system and make more accurate assumptions.

2. Handle the feedback

After the feedback is pushed onto the job queue, a worker thread will pull it off and handle the packet. Inside a feedback packet, it will simply have the data for the feedback it is providing and a type field indicating the kind of feedback. For example, NeuronCon will have a feedback loop that reports the step count.

Using the feedback data, BrainCon can maintain metrics about the system. BrainCon will use these metrics to aid in issuing commands to the stations, as well as providing messages or errors to the observer through the user interface. Each station will have a series of toggling lights or text fields indicating types of feedback in the user interface, as outlined in the first functional requirement on the graphical user interface.

Based on the feedback metrics, BrainCon can determine if the system is failing or if a soft limit has been reached. A soft limit is simply a software limit that BrainCon maintains, which helps keep the observer aware that the siderostat or NAT are reaching their hard limits. If a failure is reported, which might result from a hard limit being reached, BrainCon will need to issue the commands necessary to stop the failing system. Error reporting will be handled in the graphical user interface as stated above.

Lastly, BrainCon will need to do calculations based on the feedback it receives. For example, BrainCon will need to do statistical analysis of the position of the siderostat, based on feedback from the system, which it will use to do calculations to recalculate the step size. This new step size would then be sent to the microcontroller over the network.

4.2 NeuronCon

Given that NeuronCon is a microcontroller that handles input commands from BrainCon over the network, communicates with piezoelectric actuator for the NAT and stepper controllers for the siderostat, and responds to feedback from limit switches, there are a handful of requirements that were determined necessary for this system:

1. Network with BrainCon
    a. Send feedback to BrainCon
        i. Send packets containing relevant data
    b. Receive inputs from BrainCon
        i. BrainCon command processing
            1. Packet parsing
            2. Data processing
2. Control mirror positioning
    a. Drive stepper motor
        i. Send pulses to Parker microstepper
        ii. Receive feedback from limit switches
    b. Drive piezoelectric actuator
        i. Send command to Ultravolt Piezoelectric Actuator Control
        ii. Receive feedback from limit switches
3. Track position of NAT and siderostat
    a. Motor step counts
    b. Piezoelectric actuator voltage
    c. Log limit switches
4. Be electronically insulated from EM interference

Given this outline of functional requirements for NeuronCon, each subsection may be elaborated on to fully detail the necessary functionality of each part of the system.

4.2.1 Network with BrainCon

Because BrainCon is handling all major feedback, data processing, and distribution of observer commands to individual stations, NeuronCon will need to interface with BrainCon through the network to operate the mirrors and provide feedback information to the system. Networking with BrainCon includes the following:

1. Send feedback to BrainCon

Because NeuronCons will be distributed individually to each station, feedback for mirror positioning necessary for BrainCon software will have to be handled at each station and reported back to BrainCon over the network. When BrainCon issues a command to NeuronCon to move a stepper motor or a piezoelectric actuator, software on NeuronCon will need to issue the action and update local variables corresponding to mirror positioning, reporting the updated variables and affirmation of step completion as a packet to BrainCon. Limit switches on the Sid and Nat, acting as absolute positioning feedback, will need to trigger an event on the NeuronCon which then sends a packet to BrainCon informing its software a limit switch has been reached. The software on BrainCon will then be able to utilize this feedback for its mirror control algorithms.

2. Receive inputs from BrainCon

Individual NeuronCons act as liaisons for BrainCon, allowing for individualized control of stations. To allow for remote control of each station's piezoelectric actuators and stepper motors, NeuronCon will need to receive packets from BrainCon, parse the data provided, and respond to commands given. Inputs received from BrainCon may tell the NeuronCon to push a stepper motor a certain number of counts, apply a certain voltage to the piezoelectric actuators, or halt.

4.2.2 Control mirror positioning

Previously it has been mentioned that NeuronCon is responsible for controlling mirror positioning for the Sid and NAT. Because NeuronCon does not directly interface with the stepper controls on the Sid or the Piezoelectric actuators on the NAT, to change the siderostat positioning, NeuronCon must interface with a 'middle man'. Software on the microcontroller will convert input commands from BrainCon into a specific number of steps and direction for each motor. Because these steps are accomplished by issuing voltages through GPIO pins on the microcontroller, variables will be set to memory map a pin to its corresponding pin on a Parker Microstepping Drive or pin on an Ultravolt A-series Piezoelectric Actuator. These are outlined with the following requirements:

1. Drive stepper motor

To move the siderostat, NeuronCon must move one or both of the siderostat stepper motors. To accomplish this, the NeuronCon must interface with a Parker Microstepping Drive for each axis of control. Each motor step corresponds to a 5V, 200ns pulse applied to the Parker device, so for NeuronCon to control one siderostat mirror, it must be able to apply these pulses to two different Parker Microstepping Drives.

Drive piezoelectric actuator

To move the NAT, NeuronCon must interface with two Ultravolt A-series Piezoelectric Actuator Control devices. Each piezoelectric actuator (one for each axis) accepts a voltage from

0-1000, so upon initialization of the NAT, each piezoelectric actuatorshould be initialized to 500V for maximum range. Because the Ultravolt device is responsible for stepping up the voltage to the acceptable range, NeuronCon only needs to apply a voltage in the 0-5V range, 2.5V corresponding to 500V on each piezo. Thus, to change the position of the NAT according to BrainCon commands, the NeuronCon simply needs to increase or decrease the applied DC voltage in a 0-5V range.

Additionally, NeuronCon will need to interface with and receive signals from optical limit switches. These limit switches are placed on the Sid and NAT and will trigger when either mirror reaches a limit. While the stepper motor control will be highly precise, the mirror has mechanical issues which prevent the stepper control from being able to reliably predict the positioning of the mirror, making sensing limits incredibly important. Detecting absolute positioning of either mirror and reporting this information back to BrainCon is essential for knowing the different mechanical requirements of each station. Reaching a limit will also inform BrainCon to decelerate the mirror to prevent from damaging hardware, so it is essential that NeuronCon can detect these switches and report the information to BrainCon.

4.2.3 Track positioning of NAT and siderostat

Utilizing the limit switches, motor counts for siderostat stepper motors, and voltages for the NAT's piezoelectric actuators, it is possible to approximate and track the positioning of the NAT and Sid. While the positioning data may be somewhat unreliable, it is important software on the microcontroller reports back to the observer GUI what the approximate position is. To accomplish this, each stepper motor pulse, applied voltage to a piezoelectric actuator, and activated limit switch must be logged and reported over the network to BrainCon, which then reports this information back to the observer GUI. Through successful tracking of Sid and NAT behavior, 'expected' vs 'observed' behavior of the mirrors can be a valuable asset for detecting mechanical problems in the system, allowing the site maintenance staff to monitor and fix issues.

4.2.4 Electronic insulation

One of the major issues of the previous system is that it behaved strangely due to electronic crosstalk. Thus, the client has expressed interest in insulating the Pi from the rest of the devices at each station so that crosstalk is virtually impossible. This requirement means not only placing the device in an individualized case, but physically separating it from other devices.

In summary, NeuronCon will need to network with BrainCon, control mirror positioning, track positioning of NAT and Sid, and be electronically insulated. Interfacing with BrainCon requires receiving and interpreting packets sent from BrainCon and reporting data back. Pushing mirrors requires applying a signal to a Parker Microstepping Drive and an Ultravolt Piezoelectric Actuator Control, while tracking positioning requires monitoring limit switches and logging any mirror commands. Electronic insulation can be accomplished through an insulated, physically separated box. Now, performance requirements for some of the functional requirements will be considered.

5 Performance Requirements

This section will explain the quantitative components associated with some functional requirements listed above. These requirements were determined by the client and the project team in order to maintain the current precision of the interferometer.

5.1 BrainCon

The central point of command in the system, or BrainCon, is a modern Dell server rack. Here are some performance requirements that have been deemed necessary to meet the software requirements of the system:

1. Multithreaded job system
    a. BrainCon can use hyperthreading with its 6 cores. As a result, the multithreaded job system must be able to operate with 12 threads.
2. Network with NeuronCon
    a. Must be able to process and send packets to NeuronCon in under 50ms.
    b. Ensures that the stepper motor does not go too far passed a limit switch if one is reached.

These software performance requirements for BrainCon ensure that NeuronCon receive commands in a timely manner. The importance of this kind of performance is that NeuronCon could potentially blast passed a limit switch if the system is not fast enough to detect it or not fast enough to provide the appropriate commands to NeuronCon.

5.2 NeuronCon

The microprocessor in the system will be the bottleneck of the system, as it has the slowest processing speed in the system.

1. Quick Processing
    a. At a minimum, NeuronCon must operate at 250Hz to handle computational requirements listed in the functional requirements.
    b. To be safe, 25,000Hz will be the minimum clock speed requirement for the project.

2. Accurate electrical pulses to control mirror positioning
   - a. Must send 5V, 200ns pulses to two Parker Microstepping Drives which will in turn drive two stepper motors.
   - b. Must send 0-5V pulses to two Ultravolt Piezoelectric Actuator Controls which will in turn drive two piezoelectric actuators on the NAT.
3. Precise mirror control
   - a. The NeuronCon solution must maintain a precision of 59 arcseconds when driving the Sid.
   - b. A precision of 10 arcseconds must be maintained when controlling the NAT.

A standard Raspberry Pi has a clock speed of 5MHz. If 99% of the system was being utilized, there is 50,000Hz leftover. This means that one percent of a standard Raspberry Pi's processing power will be two times higher than the minimum clock speed requirements of the system. This suggests that the NeuronCon solution will have no problem keeping up with the clock speed requirements. These performance requirements ensure that NeuronCon can handle the feedback from limit switches quick enough to deliver feedback to BrainCon in a timely manner. The precision at NPOI is very important, so maintaining precise mirror control is a must.

6 Environmental Requirements

BrainCon must interface with existing hardware. This includes the local network and the observer computer. BrainCon must receive commands from the observer computer, perform the necessary calculations, and pass them on to appropriate NeuronCon at each station. It receives and sends these commands over the local network, via Ethernet.

NeuronCon must interface with existing hardware. This includes the local network, limit switch wiring, and sending appropriate output to the motor steppers and piezoelectric actuators. NeuronCon must communicate with BrainCon over the local network, via Ethernet. It must receive feedback from the limit switches; this input is determined by the limit switches already in place, thus NeuronCon must have the limit switches directly wired to its pins. The stepper and piezoelectric actuator accept their own uniquely formatted input. The format of this input is determined by the stepper and piezoelectric actuator currently in use. NeuronCon must be able to send movement command input to these motors in the exact format they expect. NeuronCon must be able to withstand temperatures of -20 to 115 °F, as determined by its location at the site. It also needs to be electronically insulated to avoid electronic interference with the other components in the station, specifically the piezoelectric actuator controller and the microsteppers.

7 Potential Risks

In this section, potential risks to the project's success are considered based on how likely the risk is to occur and how much of an impact that risk would have on the overall success of the development of the project. Each risk has a likelihood of happening, which ranges from low to high. Low likelihood means the risk is unlikely to occur in our project, medium likelihood means that it is uncommon to occur, and high likelihood means the risk is very likely to occur. The impact of a risk ranges from a low level of impact to high level of impact. A low level of impact would set the team back a matter of hours to a day. A medium level of impact would delay the project progress from a day up to two weeks. A high level of impact would consist of destruction of hardware delaying the project for two or more weeks. These are risks that are deemed most relevant to the project by the client and the project team.

7.1 Risk Overview

Risk	**Likelihood**	**Severity**
Software updates	Medium	Medium
Raspberry Pi communicating through existing network	Low	Medium
Overlooked or incorrect interface hardware	Low	Medium
Weather or fire hazards	Low	High
Humidity	Low	High

Figure 3: Risk Analysis

7.2 Risk Mitigation

1. Software updates

The first risk is that software updates may break the current system. Updating the OS on BrainCon or NeuronCon could potentially set the project back. Using a version control manager like GitHub can mitigate long delays and ease software updates.

2. NeuronCon communicating through existing network

The next risk is that the NeuronCon solution may not be able to communicate to BrainCon over the existing network at NPOI. There is a very small chance of this being a problem, as the network has Ethernet ports available for plug and play. However, until there are on-site tests, this remains a risk. If an alternative method is required to communicate over the network, it would set us back a couple weeks to order new parts. This risk can be mitigated by communicating concerns to the clients and coming up with a solution for early testing. It is worth noting that Lowell has a dedicated IT staff which will be able to assist with troubleshooting any technology issues.

3. Overlooked or incorrect interface hardware

A handful of cords in the current system are custom made. While the team has looked into the correct interface hardware, there is a small chance that further interface hardware is needed. This risk again, would just require ordering new hardware. This risk can be mitigated by further detailing the connections required for each device to assure everything is in place. Technological demos will also aid in the understanding of what needs to be bought for the project or manufactured by the team or an electrical engineer.

4. Weather or fire hazards

To mitigate lightning damage, a well-insulated computer case would potentially help save the NeuronCons if a siderostat station was struck. If the project receives more funding, fire damage can be avoided through fireproof computer cases. These cases are usually water resistant as well, so sprinkler systems would not be an issue. Flooding could be mitigated by raising the elevation of the NeuronCons and BrainCon within their respective rooms. There is little that can be done in the case that a tornado sweeps over the interferometer. However, if NPOI is narrowly missed by a tornado, the wind could tear the roofs down above the NeuronCons or BrainCon. To mitigate this risk, some internal structure could be created above the computer racks to support cave ins from the roofs. BrainCon and the NeuronCons should also be placed away from the windows and doors of each building.

5. Humidity

The buildings that house the hardware for the siderostat and NAT are humidity controlled. If an observer was to accidentally leave the door open for too long, the humidity levels could raise substantially. Overexposure to high humidity could cause corrosion to circuit boards, requiring them to be repaired or replaced. The mitigation for this risk is also to supply a solution to insulate the NeuronCons. Mitigation strategies will be incorporated as the project progresses, the project plan follows.

8 Project Plan

To measure the progress of this project and focus the team on common goals, clear milestones have been defined in the timeline of this project. The key milestones approaching as seen in Fig 3 and 4 include: a system demo, on-site integration demo, precision testing, communication testing, component finalization, documentation finalization, integration into the site, and "On-Sky" operation.

The system demonstration will show the fundamental components of the solution performing the fundamental tasks required to execute the solution. This includes sending signals from a central computer to a NeuronCon controller over Ethernet, to ultimately drive a stepper motor. More importantly this demo will show the team's ability to use the chosen tools to execute a working demonstration of the components that will make up the final solution.

The on-site integration demo expands on the initial system demo by adding a layer of functionality and precision to the foundational capability established in the initial demo.

The testing phase includes precision testing and communication testing. The systems will be measured for precision to ensure the accuracy of the system has been maintained. Communication systems will be thoroughly investigated to ensure all components at each station can be reached through the network. Tests will prove communication can occur both ways across the network.

Component finalization is where the solution is deemed working, tested, and ready for integration. This milestone marks completion of all major hardware and software requirements. Documentation finalization comes shortly after.

Integration into the site is where the team moves the solution from the test rig and plugs the components into NPOI's working array. On-Sky operation is the final milestone marking the team's success in fitting their solution into NPOI's existing infrastructure. On-Sky operation confirms the solution works in its intended application.

9 Conclusion

NPOI developed an alternative to the traditional, single-aperture telescope design by utilizing an array of telescopes which combine their images together. Through this technology they achieved much higher precision than a single-aperture design, resulting in the highest angular resolution of any optical system on Earth. Some of the applications of this technology range from providing accurate positioning for GPS systems, generating precise measurements of binary star-systems, and creating star-charts. NPOI's partners are Lowell Observatory, United States Naval Observatory (USNO), and Navy Research Laboratory (NRL).

The goal of this capstone project is to update several components at NPOI, some hardware related, some software related. The current system in place at NPOI is outdated, hardware is constantly breaking down, and the existing system is overly complex. As a result, team Astraea has developed a plan to solve their problem. This solution will reduce the complexity of the system, reduce the points of failure, and be robust and modular so that NPOI can continue to operate as hardware and software change in the future.

The requirements detailed in this document cover the specifics of each piece of technology the team deems critical to the development of a working system for NPOI. These requirements consider constraints, capabilities, and risks to each piece of hardware or software to be integrated into the mirror control network. These devices and their specifications include the following:

- A centralized computer, BrainCon, which networks with an observer computer, sends commands to 3-6 NeuronCons and handles feedback loops. Specifically, BrainCon's software will host a TCP server, enabling persistent connections, modular design, and a multithreaded network. A GUI will allow for an easy user interface for configuration and individualized station control. BrainCon will relay mirror commands to a NeuronCon in the form of packets of a job queue and will receive feedback from networked devices through a dedicated thread.

- Each mirror station will host a NeuronCon, which will receive inputs from BrainCon, outputs electrical pulses to siderostat stepper motor micro stepping drives, outputs DC signals to Ultravolt piezoelectric actuator controllers, and receives feedback from optical limit switches. The NeuronCon software will need to network with BrainCon, receiving packets and relaying relevant data back to the central brain. It will also control mirror positioning, accomplished through sending pulses to microstepping drives for the Sid, or by sending DC signals to piezoelectric actuator controls for the NAT. Position of the mirrors will be tracked by utilizing limit switches, counting applied micro steps, and monitoring voltages applied. Finally, each Pi will be electronically insulated, preventing cross-talk with other electronic systems.

Each requirement has additional performance or environmental specifications in the document, which offer additional insight to the hardware and software added to the system.

Although this solution attempts to solve many of NPOI's hardware and software issues, a few risks threaten the success of Astraea's project. These include software updates, having issues with communicating over the existing network, realizing overlooked or incorrect interfacing solutions for existing hardware, and exposing the devices to fire, humidity, lightning, or other weather hazards. The team will attempt to mitigate these risks to ensure the project succeeds.

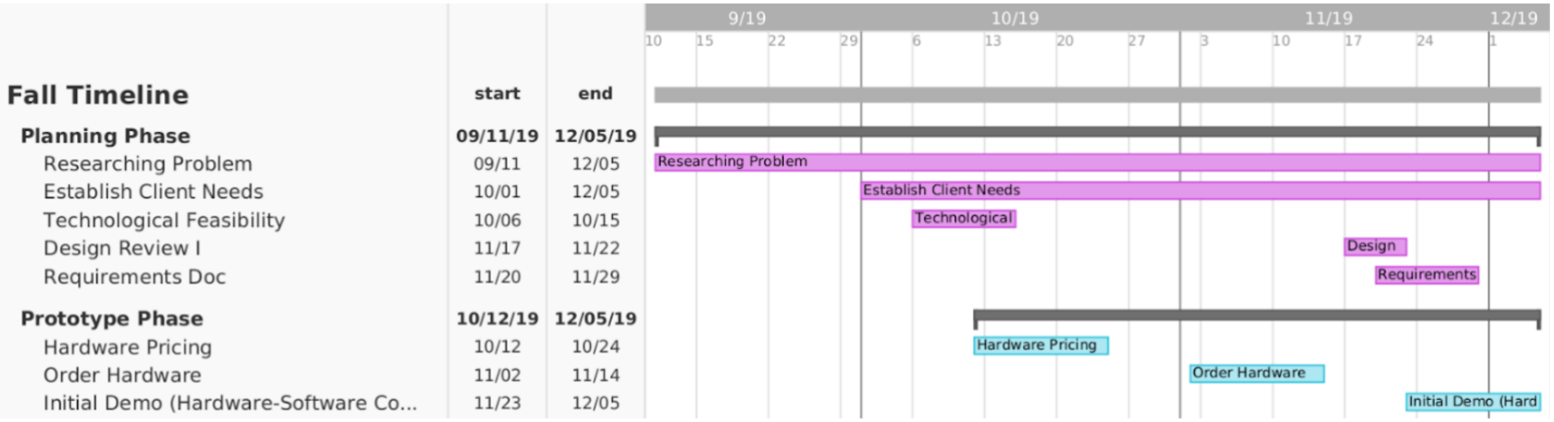
In relation to the timeline, Astraea has completed the initial requirements acquisition phase and planning phase. The demo is complete and demonstrates the ability to: communicate over the network to send and receive commands between BrainCon and NeuronCon, drive a stepper motor with NeuronCon, receive and act on limit switch feedback, and control BrainCon through a GUI. Moving forward into the Spring timeline, Astraea approaches the prototyping phase where the team will create a system that addresses the requirements outlined in this document.

To conclude, Astraea is excited to re-design the mirror control system for NPOI and would like to emphasize a desire to succeed in creating a working solution. Not only will the team save NPOI millions by replacing custom hardware and minimizing downtime, but Astraea will provide a significantly better, simplified system which reduces points of failure. Updated software and hardware will be much easier to understand, cheaper to replace, and will maintain precision of the instrument. Astraea's solution will allow NPOI to continue contributing to the field of astrometry, assisting in the exploration of the cosmos.

10 Glossary

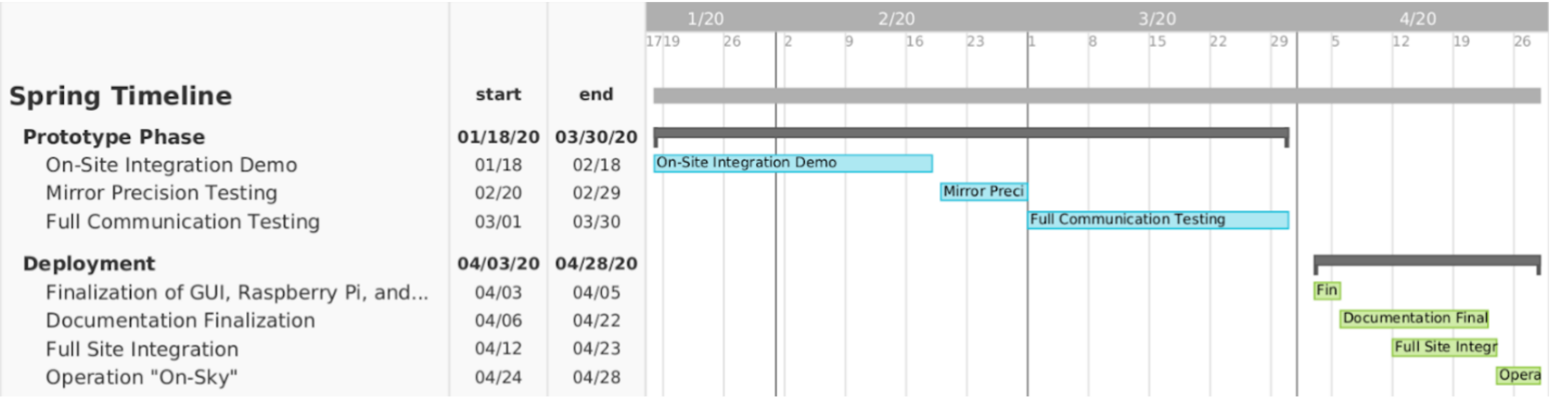
AlignCon	Internal NPOI software to help align the mirror system.
Arcsecond	$\frac{1}{3600}^{th}$ of a degree.
BrainCon	A Dell server rack running Linux which acts a central computational unit.
GitHub	An online version control management system, also used to publicize and share code.
GPIO	General-purpose input/output, uncommitted digital signal pin on an integrated circuit board whose behavior is controllable by a user at runtime.
GUI (Graphical User Interface)	A user interface that allows users to interact with a device through graphical icons.
Interferometer	Investigative tool used in engineering and science. Interferometers merge multiple light sources to make an interference pattern. This pattern can then be measured and analyzed.
Limit Switch	An optical feedback mechanism for detecting when an object crosses through the switch.
Lowell Observatory	An astronomical observatory in Flagstaff, Arizona, known as one of the oldest observatories in the United States, famous for having discovered the coordinates of Pluto.
Microstep	An angular change of a stepper motor smaller than a 1.8 degree step.
NAT (Narrow Angle Tracker)	High-precision mirror responsible for reflecting light into NPOI's vacuum tube system.

NPOI (Navy Precision Optical Interferometer)	A specialized astronomical telescope system co-owned by Lowell Observatory, Navy Research Laboratory, and United States Naval Observatory.
NRL (Navy Research Laboratory)	The corporate research laboratory for the United States Navy and United States Marine Corps. It conducts basic scientific research, applied research, technological development and prototyping.
On-Sky	An astronomer colloquialism for when an instrument is in operation and gathering data from the sky.
Piezoelectric actuator	A mechanical translator which converts electrical energy directly into linear motion.
Siderostat (Sid)	A flat mirror which is used at NPOI for reflecting light from a celestial object to the NAT.
Siderostat station	A physical building located on NPOI's array which houses a siderostat mirror, NAT, and all other devices associated with those mirrors.
Stepper motor	A type of DC motor that works in discrete steps, or angular positions.
TCP (Transmission Control Protocol)	A networking protocol that allows applications to maintain communications to each other through which to send and receive data.
Ultravolt Piezoelectric Actuator Control	A device that steps up low voltages to higher voltages and applies them to a piezoelectric actuator.



Fall Timeline	start	end
Planning Phase	**09/11/19**	**12/05/19**
Researching Problem	09/11	12/05
Establish Client Needs	10/01	12/05
Technological Feasibility	10/06	10/15
Design Review I	11/17	11/22
Requirements Doc	11/20	11/29
Prototype Phase	**10/12/19**	**12/05/19**
Hardware Pricing	10/12	10/24
Order Hardware	11/02	11/14
Initial Demo (Hardware-Software Co...	11/23	12/05

Figure 4: Fall Timeline



Spring Timeline	start	end
Prototype Phase	**01/18/20**	**03/30/20**
On-Site Integration Demo	01/18	02/18
Mirror Precision Testing	02/20	02/29
Full Communication Testing	03/01	03/30
Deployment	**04/03/20**	**04/28/20**
Finalization of GUI, Raspberry Pi, and...	04/03	04/05
Documentation Finalization	04/06	04/22
Full Site Integration	04/12	04/23
Operation "On-Sky"	04/24	04/28

Figure 5: Spring Timeline