Wellesley College
Wellesley College Digital Scholarship and Archive

Honors Thesis Collection

2018

Deformable Mirror Demonstration Mission

Jennifer Gubner
Wellesley College

Follow this and additional works at: https://repository.wellesley.edu/thesiscollection

Recommended Citation
Gubner, Jennifer, "Deformable Mirror Demonstration Mission" (2018). *Honors Thesis Collection*. 606.
https://repository.wellesley.edu/thesiscollection/606

This Dissertation/Thesis is brought to you for free and open access by Wellesley College Digital Scholarship and Archive. It has been accepted for inclusion in Honors Thesis Collection by an authorized administrator of Wellesley College Digital Scholarship and Archive. For more information, please contact ir@wellesley.edu.

Deformable Mirror Demonstration Mission

by

Jennifer Gubner

Submitted in Partial Fulfillment of the Prerequisite for

Honors in Physics

at

WELLESLEY COLLEGE

December 2018

© Jennifer Gubner 2018

Author . . . .
Department of Physics
December 13, 2018

Certified by . . . .
James Battat
Associate Professor of Physics
Thesis Supervisor

Certified by . . . .
Kerri Cahoy
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . .
Yue Hu
Chair, Department of Physics

Deformable Mirror Demonstration Mission

by

Jennifer Gubner

Submitted to the Department of Physics
on December 13, 2018, in partial fulfillment
of the prerequisite for
Honors in Physics

Abstract

The Deformable Mirror Demonstration Mission (DeMi) is a 6U cube satellite mission created to demonstrate the use of adaptive optics (AO), specifically a 140 actuator Microelectromechanical systems (MEMS) deformable mirror (DM), in space. While AO has been commonly used on ground based telescopes, it has many useful benefits in space. AO can be a critical difference in reaching the necessary contrast, of 10^{10}, to image Earth-like exoplanets. It allows for corrections of optical imperfections and thermal distortions. These correction capabilities also allow launches of cheaper optics, and have further implications for use with inter-satellite laser communication and high energy applications.

DeMi will use a closed-loop adaptive optics system, that incorporates a Shack-Hartmann wavefront sensor (SHWFS), DM, and CMOS cameras, in multiple mission operations to demonstrate the capabilities of this adaptive optics technique. DeMi will launch in to a low-Earth orbit in mid 2019. During its lifetime, DeMi will complete both internal and external observations. The internal observations will use a laser to characterize the DM and test the wavefront correction. The external observations will demonstrate the wavefront correction on stars.

Thesis Supervisor: James Battat
Title: Associate Professor of Physics

Thesis Supervisor: Kerri Cahoy
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

Thank you to my incredible thesis advisors, Professor Kerri Cahoy, and Professor James Battat, without whom this thesis would not be possible. Thank you for believing in me and giving me the opportunity to participate in this incredible project. Thank you for making me a better student and a more creative thinker.

A special thank you to Dr. Ewan Douglas, who has been a very important mentor and advisor to me for these last two years. I have learned so much from you and am forever grateful for everything you have done for me. Thank you for answering all of my questions and clarifying things for me as many times as I need.

Thank you to all of the students in STAR Lab, especially the DeMi team, for helping me and teaching me important and useful engineering skills. Also, thank you to the physics and astronomy departments at Wellesley for shaping me in to the student I am today. Without the support and encouragement of all of my professors and fellow students, I would not be where I am today.

Thank you to my friends, family, and teammates for their unwavering support and words of encouragement.

Finally, thank you to the Defense Advanced Research Projects Agency (DARPA) for the project funding and Aurora Flight Sciences, A Boeing Company for working with us.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Astronomy is a field in which new technology allows for new advances in exploration. Because astronomy relies on observations of celestial bodies as a means for data collection, science and discovery is limited by the capabilities of the observation tools. To put it plainly, astronomy is greatly dependent on the capabilities of the telescopes and optical sensors used to collect the data. Improvements in the observational capabilities of these instruments comes with advancements in materials, manufacturing processes, and technology.

Adaptive optics (AO) is one such recent technological advancement that has impacted the field of astronomy. Today, many observatories use some variation of AO systems to improve their imaging capabilities. AO has pushed the boundaries of what we can observe and has allowed astronomers to gain a better understanding of the workings of the universe.

While AO has been commonly used in ground-based imaging facilities, its applications in space have not been thoroughly explored. The Deformable Mirror Demonstration Mission (DeMi), the subject of this thesis, aims to demonstrate AO technology for use in space with the ultimate focus of using AO for direct imaging of Earth-like exoplanets from space.

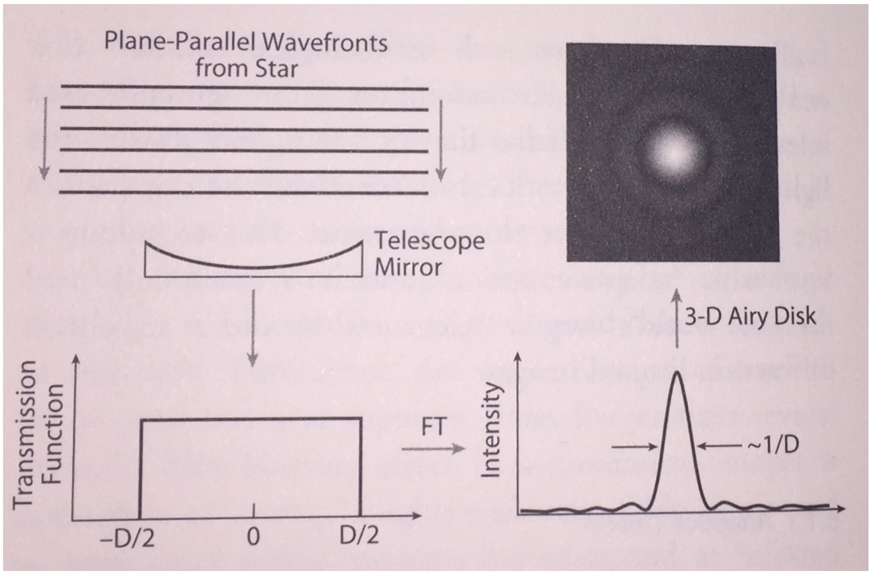


Figure 1-1: Representation of the operation of a telescope. The plane-parallel wavefronts come into the optical system through the telescope's aperture (top left) which acts as a transmission function (bottom left) on the wavefronts due to its limited aperture size. The limit causes diffraction, and influences the measured intensity of the signal (bottom right). The intensity can be plotted in two dimensions and is known as an Airy function (top right). As the image is incident on an imaging sensor, the sensor [7].

1.2 Limits of Astronomical Observation

Stars, planets, and other celestial bodies generally give off light isotropically, meaning in all directions. The light from these isotropic objects can be modeled as concentric spherical shells that travel from the object to the observer. These shells represent individual wavefronts that are collected by the observer. At very large distances from the source the spherical wavefronts become approximately planar wavefronts, which are incident on the telescope's aperture. These wavefronts are transformed by the telescope optics into a point spread function (PSF) of the object which is collected on an optical sensor (charge-coupled device (CCD), complementary metal-oxide semiconductor (CMOS), etc). The process of wavefront collection by a telescope is shown in Figure 1-1.

Optical systems, like any observational system, have many limitations that can affect the quality and quantity of data collection. For astronomy, these limitations mainly manifest from the optical design and layout, optical quality/surface finish, and size of the aperture on the telescope. The biggest limitations relevant to direct imaging of Earth-like exoplanets are angular resolution and contrast.

1.2.1 Angular Resolution

The angular separation between two orbiting objects that are very far away ($d \gg a$),

$$\theta = \frac{a(1+e)}{d}, \tag{1.1}$$

where θ is the angular separation, a is the semi-major axis of the orbit of the planet around its host star, e is the eccentricity of the orbit, and d is the distance from the observer to the star, is the measure of how far apart two objects appear to the observer [17]. For the case of exoplanet direct imaging, this is the measure of the separation between the host star and the exoplanet. The larger the angular separation, the easier it is to distinguish the distinct objects. For the case of a typical Earth-like exoplanet in the habitable zone of a host star 10 parsecs away, the angular separation is approximately 0.1 arcsecs, where 3600 arcsec = 1°.

The angular separation is the smallest detectable separation between two objects. The ability to distinguish the two objects with a telescope is limited by the telescope aperture and the optical system. The smallest possible angular separation that can be discerned by a telescope is calculated by the diffraction limit,

$$\theta_{min} = \frac{1.22\lambda}{D}, \tag{1.2}$$

where θ_{min} is the minimum angular separation discernible by the telescope, λ is the wavelength of the light being observed, and D is the diameter of the telescope. In a perfect optical system, θ_{min} would correspond to the location of the first null, or dark spot, in the Airy function, as shown in the top right of Figure 1-1.

As an example, the diffraction limit on the Keck 10-meter telescope for observations with a wavelength of 635 nm is approximately 0.015 arcsecs.

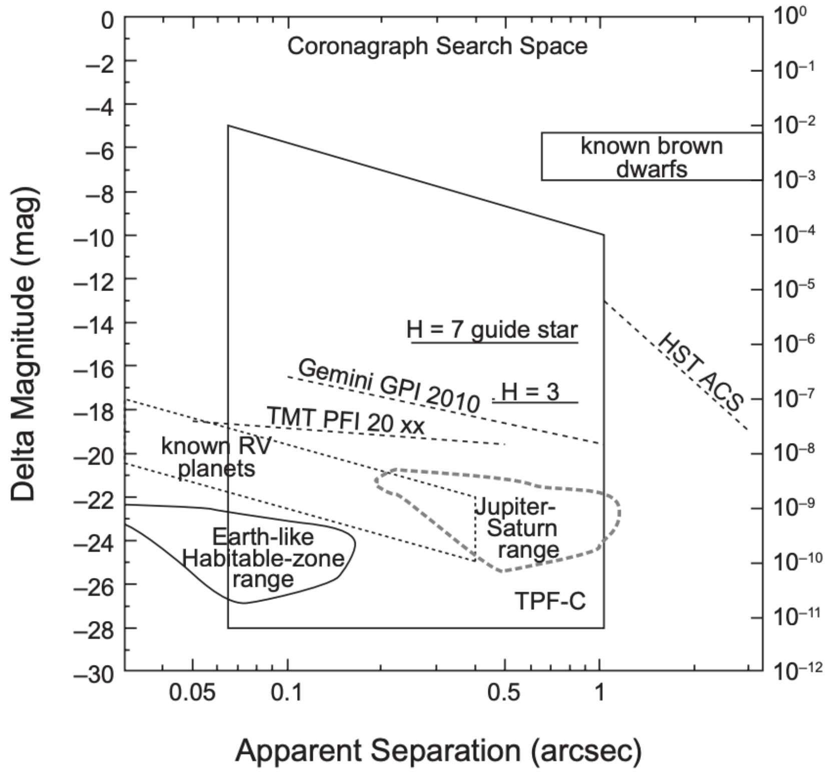


Figure 1-2: Figure showing the various ranges of angular separation (bottom axis) and contrast ratios (right axis) for different types of exoplanets. It also shows regions where current technologies are able to observe. Earth-like exoplanets in the habitable-zone are clumped around angular separations of $\leq$0.2 arcsec and contrast ratios around 10^{-10} [11].

While ground-based telescopes have small diffraction limits, the observations are severely inhibited by atmospheric turbulence, known as seeing. Seeing can affect the angular separation of two objects by as much as 0.4 arcsec [7], rendering the diffraction limit of the telescope as insignificant. In order to fix this issue, observatories either need to use sophisticated AO systems, or observe from space. However, space telescopes also come with their own limitations that I discuss further in the section 1.4.

1.2.2 Contrast

Another important concept when discussing astronomical observation is contrast. Contrast is the ratio of the flux of two objects, and is especially important when

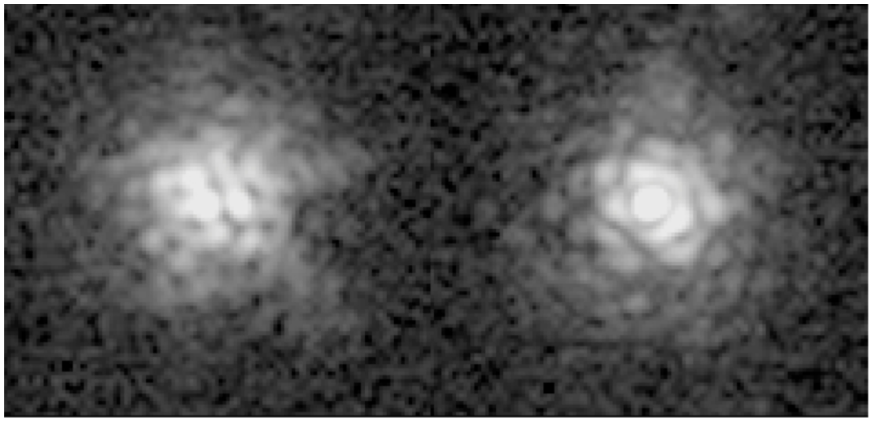


Figure 1-3: Comparison of stellar PSF captured with the Canada-France-Hawaii Telescope (CFHT) without speckle suppression (left) and with speckle suppression (right). The stellar PSF on the right has much better contrast than the PSF on the right with the help of AO [15].

imaging faint objects, such as exoplanets. In the case of exoplanets, contrast is defined as the ratio of flux of the planet and the host star. Earth-like exoplanets in the habitable zone will have contrast ratios on the order of $< 10^{-10}$, as seen in Figure 1-2, and ground-based instrumentation is limited to discerning contrast ratios of 10^{-8} at best [2]. In order to directly image Earth-like exoplanets, we either need to greatly improve ground-based AO, or image from space with AO.

A problem encountered in astronomical imaging that is related to contrast is speckles. Speckles are a source of noise caused by aberrations of the wavefront. They manifest themselves in the stellar PSF and inhibit contrast [11]. The effects of speckles on contrast can be seen in Figure 1-3 of a stellar PSF before and after speckle suppression.

1.3 Background on Adaptive Optics

1.3.1 How it Works

AO systems require three main components to operate. The first component is a wavefront sensor (WFS) that captures and records the incoming wavefront. The goal of the WFS is to provide a signal to the AO system that allows an estimate of the shape and a measurement of the deviations in the wavefront. There are three main kinds of WFS used in AO systems: Pyramid WFS, curvature WFS, and Shack-Hartmann WFS (SHWFS), which is the WFS used in the project presented here. The SHWFS uses an array of microlenslets and CMOS camera to measure the shape of the wavefront. I describe this WFS in more detail in the Sensors section.

The next component of an AO system is the wavefront reconstruction and correction algorithm. This is a closed-loop feedback system that reconstructs the incoming wavefront, measures the wavefront errors, and supplies a correction shape to the final component, the deformable mirror (DM).

The job of the DM is to supply physical corrections to the path length of the incoming wavefront. There are different types of DMs with different sizes, actuation methods, and configurations. To meet the correctional requirements of an optical system, the most important parameters to pay attention to are the number of actuators, actuator stroke, response time, actuator spacing, and size.

When the DM receives the correctional shape from the WFS algorithm, it actuates the various sections of the mirror to reshape itself into the corrected shape. This correctional cycle repeats itself to continually make adjustments to the wavefront correction shape as the incoming wavefront changes. Figure 1-4 shows the general schematic of an AO system that incorporates the three main components of AO.

The DM on DeMi is a microelectromechanical systems (MEMS) DM. MEMS DMs are convenient for use in space because of their compact size, low weight and power, low actuator mass, and high actuator density [1]. More information on the DM used on DeMi can be found in Section 2.2.1.

1.3.2 History and Current Applications

The concept of AO was first proposed in 1953 by the astronomer Horace Babcock. However, it was not put into use until the 1970's when the concept was first adopted for military use [9]. Later, AO techniques were introduced into the field of astronomy to mitigate the negative effects of atmospheric turbulence, also known as seeing, on astronomical imaging. When the light from a celestial object passes through Earth's

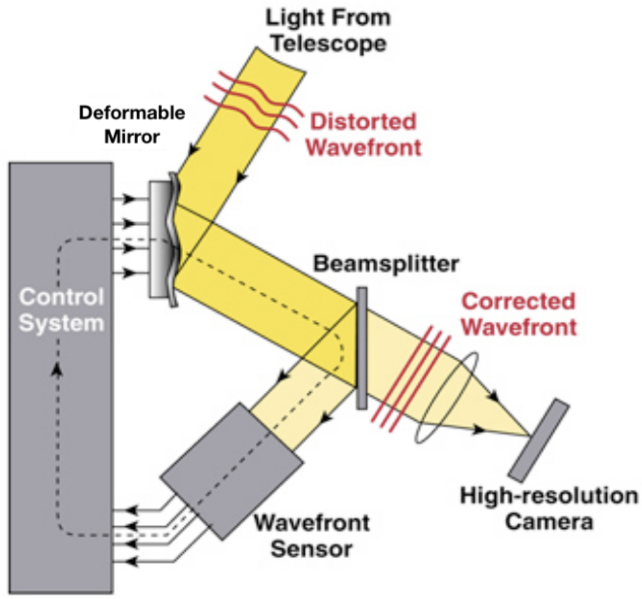


Figure 1-4: A visual representation of an AO system. A perturbed wavefront enters the optical system and is reflected through the system by a DM. A portion of the light (amount determined by the beamsplitter) is transmitted to an image sensor and the rest of the light is reflected to the WFS. The WFS measures the incoming wavefront and a closed-loop wavefront algorithm is performed on the wavefront to measure the imperfections and supply a corrective shape back to the DM [9].

atmosphere, the wavefront is distorted by the changing density and temperature of the atmosphere. These atmospheric variations cause changes to the index of refraction, which in turn introduces spatial and temporal variations to the optical path length. AO corrects these spatial and temporal variations to achieve higher angular resolutions that approach the diffraction limit of the telescope.

Today, most ground-based observing facilities use AO in their systems to mitigate aberrations due to atmospheric seeing. Ground-based facilities, like the Keck I and II telescopes [9], and the Large Binocular Telescope (LBT) [6], achieve near-diffraction limited-imaging [18], by using AO. This resolution capability is better than the imaging capabilities of most most space imagers and telescopes, due to the larger telescope apertures.

1.4 Project Motivation

While AO is commonly known to have uses on ground-based telescopes, it also has applications on space telescopes. While imaging and signal measurement in space is generally not affected by atmospheric turbulence, unless imaging Earth or transmitting signals between Earth and space, there are still several factors that affect the resolution of the signal. These factors include noise generated by the reaction wheels of the spacecraft, used for attitude control, thermal variations in the spacecraft due to eclipse in orbit, and optical misalignment or imperfections that are inherent to the system or are induced by launch [1].

AO can be used to address the wavefront errors created in space and are specifically useful in four main applications [8]. One application is to correct for atmospheric seeing while imaging the Earth. Another application is to improve resolution for ground-satellite laser communication, enabling higher resolution communication signals. Along the same lines, another application is for inter-satellite laser communication. Finally, AO has applications in high-contrast imaging of celestial objects. DeMi addresses specific applications in direct imaging of Earth-like exoplanets.

AO address the two big issues of directly imaging Earth-like exoplanets discussed earlier in this introduction: angular resolution and contrast. If we recall Equation 1.1, we can see that exoplanet systems will require very good angular resolution to be able to discern the planet from the host star because $d \gg a$. For example, for a planet with a Jupiter-like orbit around a star that is 100 parsecs away, the telescope would need an angular resolution of 0.05 arcseconds to be able to distinguish the planet from the host star [7]. AO can also make a critical difference in reaching the necessary contrast ratio, of 10^{-10}, to image Earth-like exoplanets that are typically 10^{10} times fainter than their host star [11, 17].

These correctional capabilities of AO systems enable the use of cheaper optics on the telescope because optical imperfections can be corrected. Launching cheaper optics will greatly reduce costs of missions, making space more accessible. Additionally, AO will improve imaging performance and stretch the limits of what we can observe.

1.4.1 Performance Requirements

The overall system performance is governed by the properties and abilities of the components used on DeMi. With the component limitations in mind, DeMi is expected to meet several performance standards. These standards include:

- measuring low order aberrations to $\frac{\lambda}{10}$ accuracy and $\frac{\lambda}{50}$ precision (at a nominal wavelength of 635 nm, these distances correspond to 63.5 nm and 12.7 nm respectively).
- correcting static and dynamic wavefront errors to less than 100 nm RMS.
- performing wavefront error measurements and applying corrections to the DM surface in a closed loop at at 12 Hz or faster. [1]

1.5 Project Overview

Presently, high actuator count MEMS DMs, which are a key component to an AO system, are not qualified or characterized for long-duration use in space. DeMi is a 6U (30 cm $\times$ 20 cm $\times$ 10 cm) CubeSat mission created to demonstrate the use of AO in space, as well as to qualify and characterize a 140-actuator MEMS DM, shown in Figure 1-5. DeMi will use a closed-loop AO system, made up of a SHWFS, DM, and commercial off-the-shelf (COTS) CMOS cameras, in multiple mission operations to demonstrate AO capabilities in space and characterize the DM.

DeMi is a project in the Space Telecommunications, Astronomy and Radiation (STAR) Lab at Massachusetts Institute of Technology (MIT) funded by DARPA. As a member of the DeMi team for the past year and a half, I have worked on various parts of the project including mission operations, sensor characterization, software validation, payload integration and testing, and optical alignment. This thesis is a presentation of my contributions to the DeMi mission.

DeMi is expected to launch into a circular low-Earth orbit in mid 2019. During its lifetime, DeMi will complete both internal and external observations. The internal observations will use a single-mode fiber-coupled to a 635 nm laser diode to character-

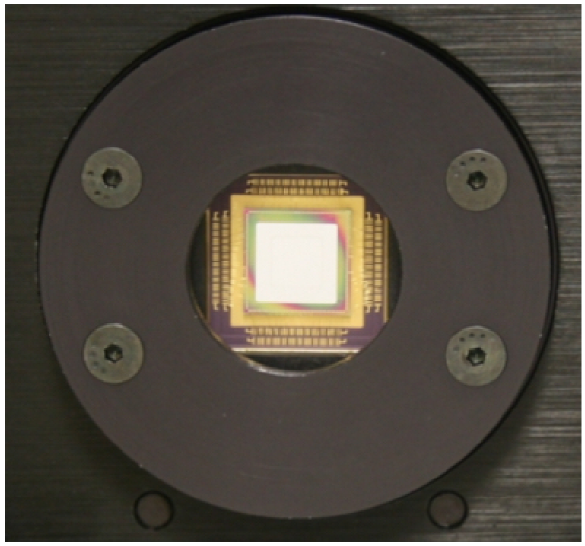


Figure 1-5: Boston Micromachines (BMC) 140-actuator MEMS DM. The DM, the square piece located in the middle of the circular aperture, is the component that will fly on DeMi. All of the housing and drivers used on the payload will be made by MIT [5].

ize the DM and test the wavefront correction system. The external observations will demonstrate the wavefront correction on astronomical point sources (distant stars).

Chapter 2

Mission Details

2.1 Overview

The DeMi payload will be flown on a Blue Canyon Technologies 6U XB6 bus and will be housed in approximately 4U of the 6U bus (1U, or one CubeSat unit, is a cube with a volume of approximately 10 cm $\times$ 10 cm $\times$ 10 cm). DeMi uses a series of off-axis parabolic mirrors (OAPs), field mirrors, complementary metal-oxide semiconductor (CMOS) cameras, a Shack-Hartmann Wavefront Sensor (SHWFS) and a Boston Micromachines Corporation (BMC) 140-actuator microelectromechanical systems (MEMS) deformable mirror (DM) to demonstrate the wavefront correction capabilities. The payload directs the light source, either internal or external, through a beamsplitter and uses one beam for the closed-loop wavefront correction and one beam for the imaging. Figures of the payload configuration are included in Section 2.3. All of the components and operations will be controlled by two Raspberry Pi Compute Module 3 flight computers.

2.2 Components

The DeMi payload uses a variety of optical components. Some are more basic, like mirrors and beamsplitters. There are a few important optical components that are more complicated than mirrors or lenses because they require power to operate. There

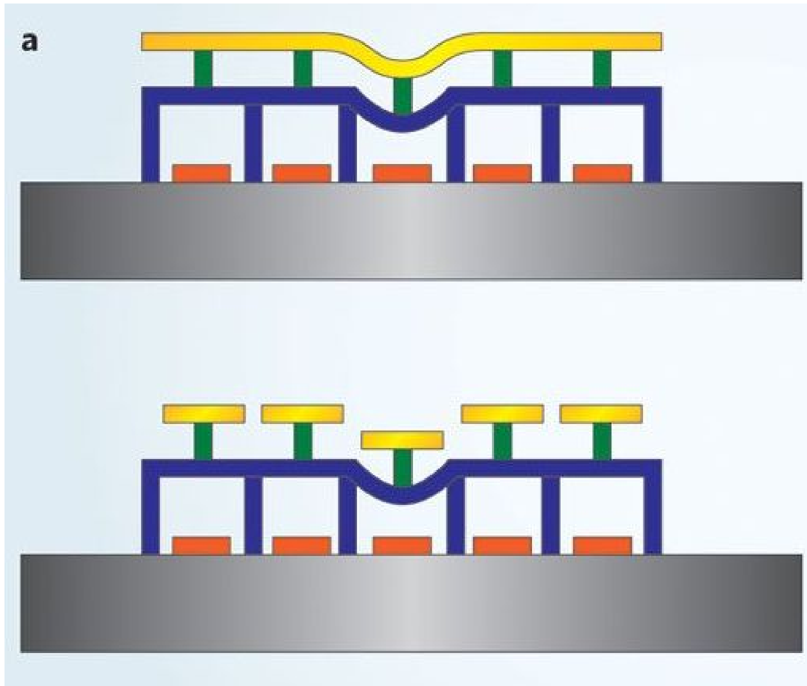


Figure 2-1: Diagram showing the difference between continuous and segmented DM. The top figure is a continuous facesheet DM. One continuous mirror sits over all of the actuators and gets deformed by the electrostatic actuation of each actuator. The bottom figure is a segmented DM. In this kind of DM, each segment of the mirror sits on a separate actuator and is moved up or down by the actuator [3].

are three essential components on DeMi that require power to operate. These include the deformable mirror, the CMOS imaging camera and the SHWFS. This section includes details on each of these components.

2.2.1 Deformable Mirror

The role of a DM in an adaptive optics (AO) system is to correct wavefront aberrations as detected by a wavefront sensor (WFS). A DM accomplishes this goal by deforming its shape into a conjugate of the detected wavefront to change the path length of the incoming section of light. There are two main kinds of DMs, segmented and continuous. Segmented mirrors have individual flat surface mirrors attached to each actuator and each piece is moved up and down to correct the wavefront. A continuous DM uses a continuous face-sheet mirror over the actuators. Instead of individual pieces of the mirror moving, parts of the mirror are pulled down or pushed up, but

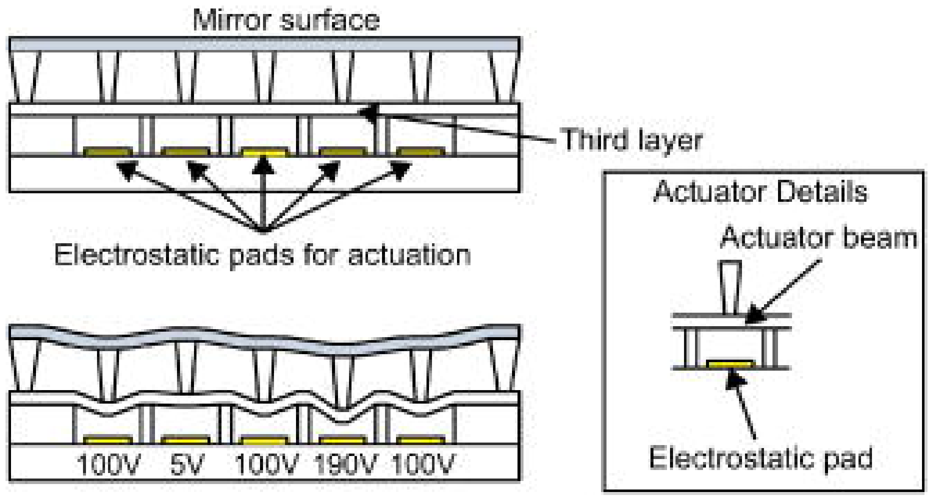


Figure 2-2: A schematic of how a MEMS DM works. Electrostatic pads underneath the actuator beams are supplied varying levels of voltages to move parts of the mirror to enable the mirror to reshape into the correctional shape calculated by the wavefront correction algorithm [14].

the mirror is continuous. Figure 2-1 shows the difference between segmented and continuous DMs.

In this project, we are specifically demonstrating the use of a MEMS DM. MEMS DMs work by using electrostatic pads to actuate the various sections of the mirror. These electrostatic pads are used on each actuator to create a parallel plate capacitor that, when charged with a voltage typically between 0 V and 250 V, attract each other and cause deformation of the mirror face-sheet. Figure 2-2 shows a schematic of a MEMS DM.

The DM on DeMi is a continuous 140-actuator BMC "multi" DM with 5.5 μm stroke and a 4.95 mm aperture, shown in Figure 2-3. The BMC MEMS DMs use electrodes and variable supplied voltages to move the actuators of the mirror. It has a mechanical response time of less than 100 μs and an inter-actuator coupling of about 22% [5]. This DM was chosen for its high actuator count, large stroke, and its CubeSat compatible size.

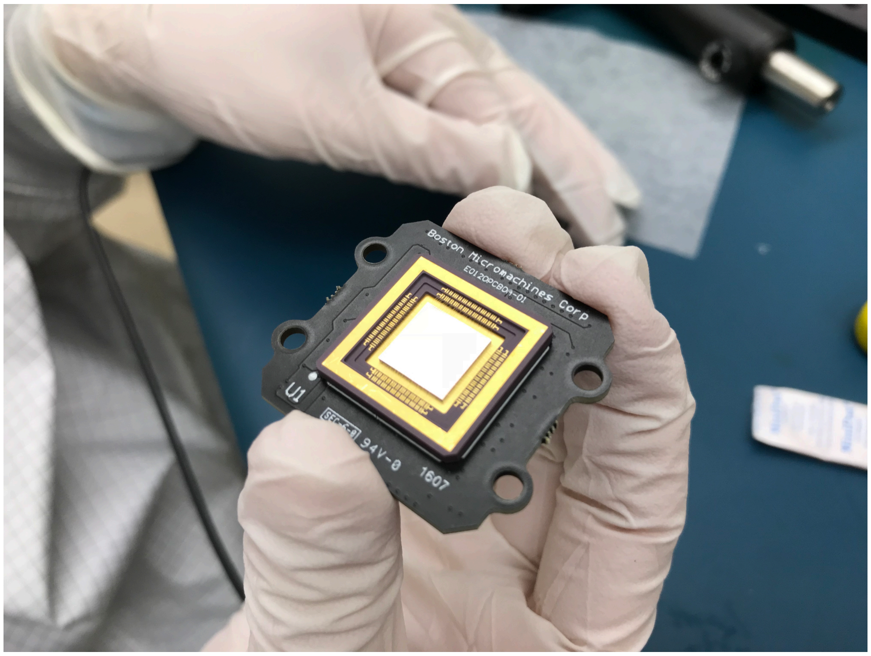


Figure 2-3: A BMC MEMS DM. The mirror has 140 working actuators and has an aperture of 4.95 mm.

2.2.2 Shack-Hartmann Wavefront Sensor

In order for an AO system to function properly, the system needs to be able to measure the aberrations in the incoming wavefront and identify the areas it needs to correct. To accomplish this measurement, AO systems use one or more WFS and wavefront reconstruction algorithms. There are many different types of WFS, including SHWFS, pyramid wavefront sensing and curvature wavefront sensing [1], however the WFS on DeMi is a SHWFS.

A SHWFS is a sensor that uses an imaging sensor, typically charge-coupled device (CCD) or CMOS, along with a microlens array. The imaging sensor is placed behind the microlens array at the focus. As the incoming wavefront is incident on the microlens array, each microlens focuses the light into a spot, called a centroid, on the imaging sensor. When the local wavefront incident on a single microlens is planar and parallel to the plane of the microlens array, the centroid location is called the reference position. When the local incident wavefront is not planar or not parallel, the centroid position on the imaging sensor is displaced from the reference location.

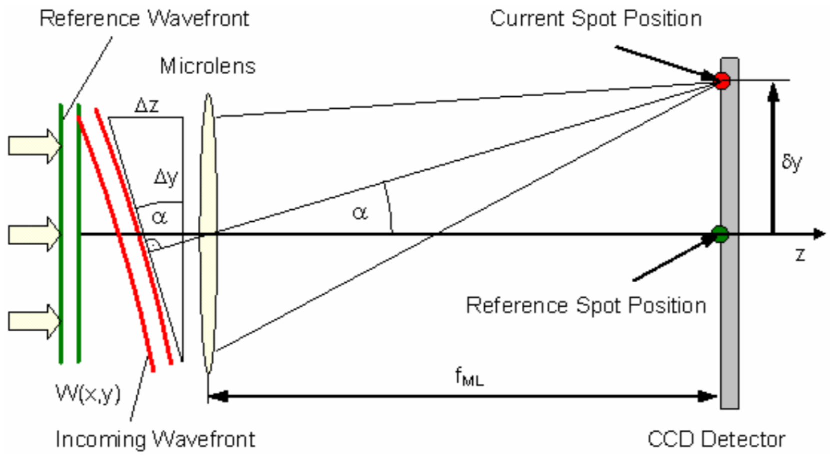


Figure 2-4: Diagram showing how each microlens in the microlens array of a SHWFS is used to measure the incoming wavefront. A parallel, planar wavefront incident on the lenslet array will create a centroid at the reference position. A non-planar, non-parallel incident wavefront will create a centroid that is displaced from the reference position. This displacement can be used to measure the angle of the local wavefront [16].

This concept is depicted in Figure 2-4.

The measurement of the displacement of each centroid can give you information about the shape of the incoming wavefront at each microlens position. For example, the angle of incidence of the local wavefront in one dimension can be determined by

$$\tan\alpha = \frac{\delta \mathrm{y}}{\mathrm{f}_{ML}} = \frac{\partial}{\partial \mathrm{y}} \text{Wavefront Shape(x,y)}, \tag{2.1}$$

where α is the angle of incidence, δy is the displacement of the centroid from the reference position, and f_{ML} is the focal length of the microlens. The partial derivative of the wavefront shape can be found in x and in y by using the above equation, and the wave shape function can be determined by performing a two-dimensional integration of the centroid displacements [16].

The array of centroids generated by a perfectly planar and parallel wavefront incident on the microlens array is called the reference spotfield. When a distorted wavefront is incident on the microlens array, the centroids from each microlens are displaced by different amounts, depending on the shape of the local wavefront incident

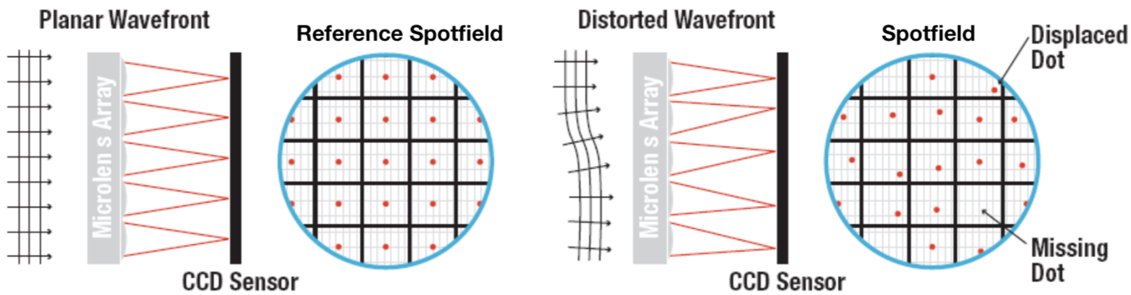


Figure 2-5: The left image shows a planar, parallel wavefront incident upon the microlens array of the SHWFS. The result is a reference spotfield created on the imaging sensor by the centroids from each microlens. The right image shows how a distorted wavefront causes the displacement of centroids and results in a different spotfield on the imaging sensor. The displacement of these centroids can be used to reconstruct the incident wavefront and to measure its shape and aberrations [16].

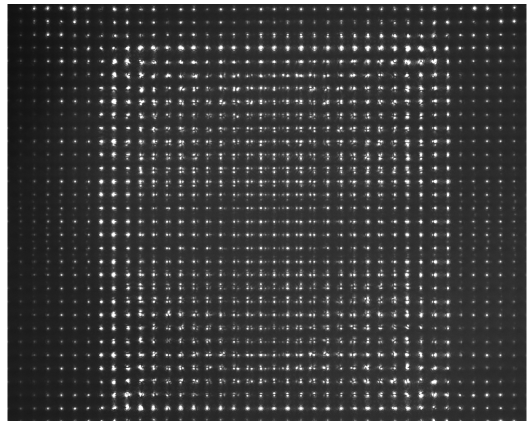
on each individual microlens. This resulting array of centroids is also known as the spotfield. A comparison of the reference spotfiled and distorted spotfield is shown in Figure 2-5.

By measuring the displacement of each centroid in the spotfiled, the total incident wavefront, and its aberrations, can be measured and reconstructed. This wavefront measurement and reconstruction can then be used to inform the corrective shape of the DM.

Initial tests of the wavefront sensing technique and capabilities were done with a Thorlabs WFS150-5C Shack-Hartmann WFS that included built-in wavefront sensing software. More details on these tests can be found in Section 4.1.

Ultimately, we decided to create our own SHWFS with a microlens array, a CMOS camera, and wavefront sensing algorithms developed in-house. The microlens array, Thorlabs MLA 150-5C-M, is a 10 mm$\times$10 mm square grid of plano-convex microlenses with a focal length, or f_{ML}, of 5.1 mm and a plate scale of 11.2°/mm. The microlens array has a 36$\times$28 array of 146 μm diameter microlenses across the sensor [1]. The CMOS camera is one of the two Pixelink CMOS cameras used on the DeMi payload. DeMi's SHWFS can capture centroid displacements of up to 75 μm [1], which, by Equation 2.1, corresponds to a wavefront angle of 0.84°. An image of DeMi's SHWFS and an example of the spotfield generated by the lenslet array is shown in Figure 2-6.

(a) The Shack-Hartmann wavefront sensor that will be used on the DeMi payload. The lenslet array, MLA150-5C-M sits in front of the CMOS image sensor and both are mounted in the aluminum housing at the end of the optical path.



(b) An example of the spotfield created by the microlens array in front of the CMOS image sensor of the SHWFS.

Figure 2-6: DeMi's Shack-Hartmann Wavefront sensor and a spotfield captured after rough alignment of the payload.

A wavefront sensing algorithm, written by MIT STAR Lab grad students, uses the captured spotfield image and measures the deviations of the centroid positions to determine the deviations in the incoming wavefront. The software will reconstruct the wavefront and send commands to the DM to correct the incoming wavefront. The algorithm has the requirement of measuring the location of each centroid to 0.01 pixels.

2.2.3 CMOS Sensors

There are two CMOS image sensors on the DeMi payload. One is used to measure the point spread function (PSF) of the target and to inform the image-plane wavefront sensing. The other is used in the SHWFS as discussed in Section 2.2.2. The CMOS cameras used on DeMi are PL-D775MU-BL COTS cameras from Pixelink, shown in Figure 2-7. These Pixelink cameras have 5 megapixel resolution and can be read out at 15 fps at full resolution [13]. The sensors come with a flex cable that moves the source heat away from the CMOS sensor and optics.

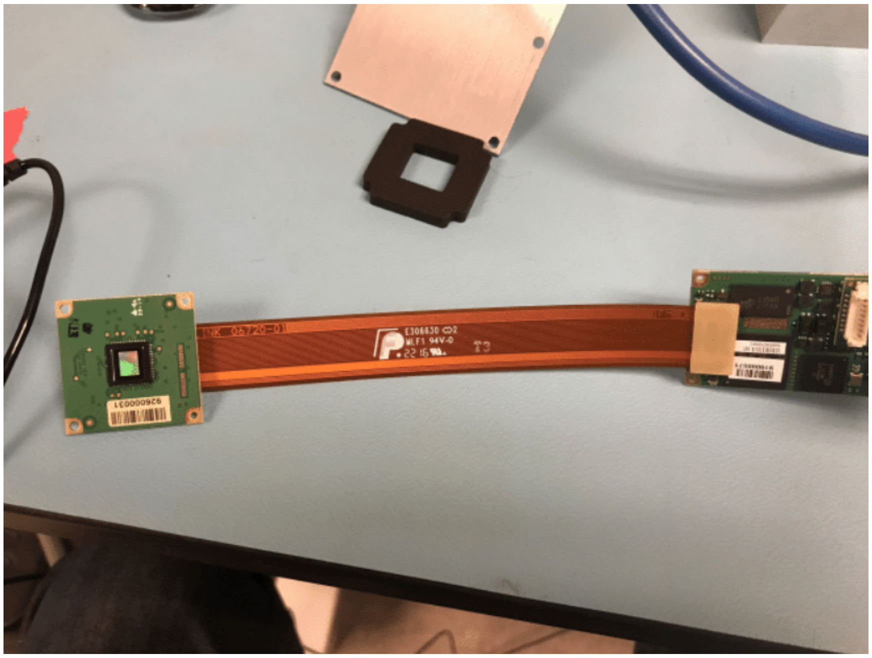
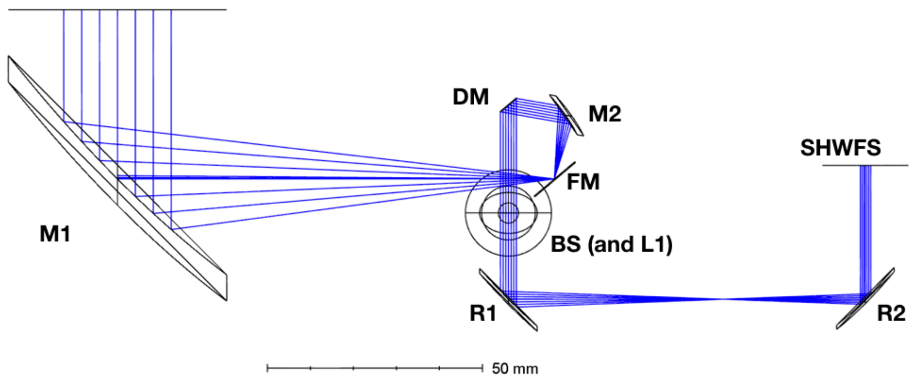


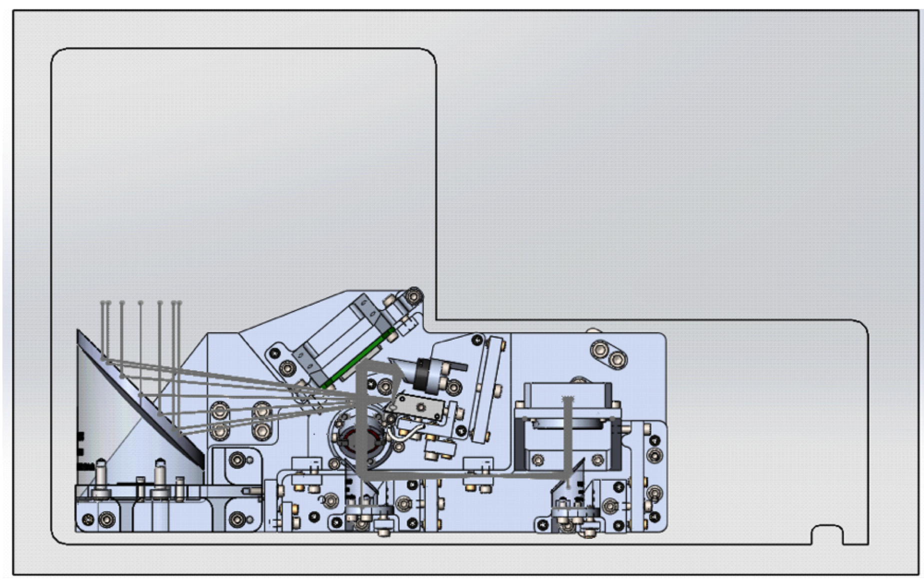
Figure 2-7: A Pixelink CMOS image sensor with its ribbons attached to the electronics. This type of sensor will be used for both the wavefront sensing and the image plane sensing. The green piece to the left is the CMOS sensor and chip. The ribbons for the electronics are the orange strip. They connect to the electronic board, which can be connected to the flight computer.

2.3 Optics Layout

The optics layout of the DeMi payload is shown in Figure 2-8(a), and Figure 2-8(b) shows the CAD model of the payload with all of its components. The first mirror, M1, is a 2” 90° Thorlabs OAP with a 4” focal length that directs light from the external target through the payload. In front of M1, a baffle blocks stray light from entering the payload, but allows the light from the target to enter. The light gets focused by M1 onto the field mirror, FM. The FM is unique because the mirror is just a polished face on the aluminum piece. The FM also has an embedded single-mode fiber, coupled to a 635 nm laser diode, that will be used for the internal observations of the mission and is shown in Figure 2-9. The fiber is injected in to a small hole in the mirror face and provides a near-diffraction limited spot. The next mirror in the configuration is M2, which is a smaller 90° OAP cut down to 8.5 mm diameter with a focal length of 15 mm. M2 collimates the light and sends it to the DM.



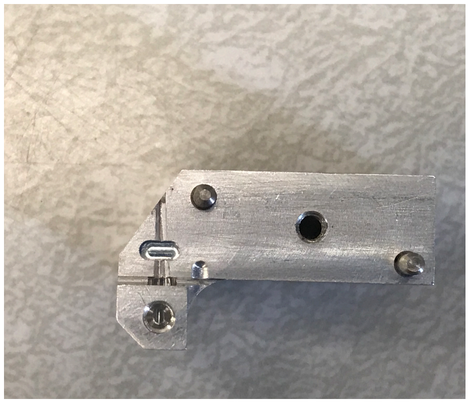
(a) Optical layout of the DeMi payload. Light from the external source enters from the top left and is reflected by the first OAP labeled M1. The light then travels through the rest of the payload. Part of the beam gets reflected down by the beamsplitter, BS, to the image sensor labeled L1. The other part of the beam is transmitted through the BS and is eventually measured by the wavefront sensor labeled SHWFS.



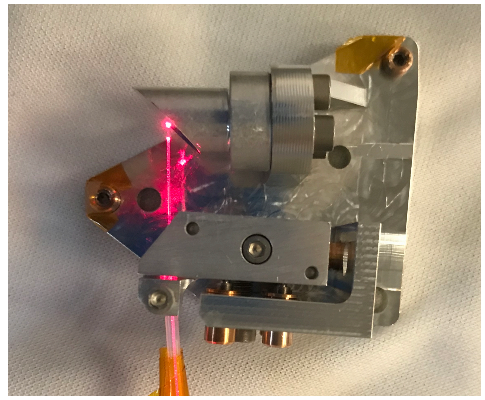
(b) Top view of the CAD model of the payload. The grey box represents the payload bus. The L-shaped black outline represents the section of the bus that will be used to house DeMi and the payload electronics. The payload is the object colored light blue and the optical path is shown in dark grey. The flight computers will sit to the right of the baffle in the top section of the payload.

Figure 2-8: Optical path and payload layout.

The DM reflects the wavefront toward a beam splitter, labeled BS, which reflects half of the wavefront down through the base of the payload and transmits half towards the first relay optic, R1. The downward reflected wavefront is captured by a CMOS camera, L1. The transmitted wavefront is directed through a set of $\frac{1}{2}$" diameter, 90°,



(a) Image of the bottom section of the field mirror. The fiber is set in the slit on the left side of the mirror and is clamped in place by the top section of the mirror. The tip of the fiber will be located right at the small opening in the mirror face.



(b) Image of the field mirror with the injected fiber clamped in place by the top piece of the field mirror. In the image, the fiber is pushed all the way through the field mirror to show the direction of the injected source. For operation, the fiber will only be injected to the surface of the field mirror.

Figure 2-9: Image showing how the fiber is injected into the field mirror. The mirror is the angled face on the left side of the piece in the image and is polished out of the aluminum. The fiber is injected in to the small hole in the face of the mirror and sits in the slot shown in the left image.

2” focal length Thorlabs OAPs, R1 and R2, to resize and redirect the beam. This OAP relay sends the wavefront to the SHWFS to inform the wavefront correction loop.

2.4 Concept of Operations

DeMi will be deployed into a low-Earth orbit and complete both internal and external observations, as described in more detail below, during its approximately one-year lifetime. The internal observations will be used to characterize the DM and to test the wavefront control loops. The external observations will demonstrate the use of the AO system on astronomical targets. During my time on the project, I have worked on defining and outlining the concept of operations, and have described the modes of operation in more detail to include more specific information about mission procedures.

Before each operation, the spacecraft will perform several checks to ensure that

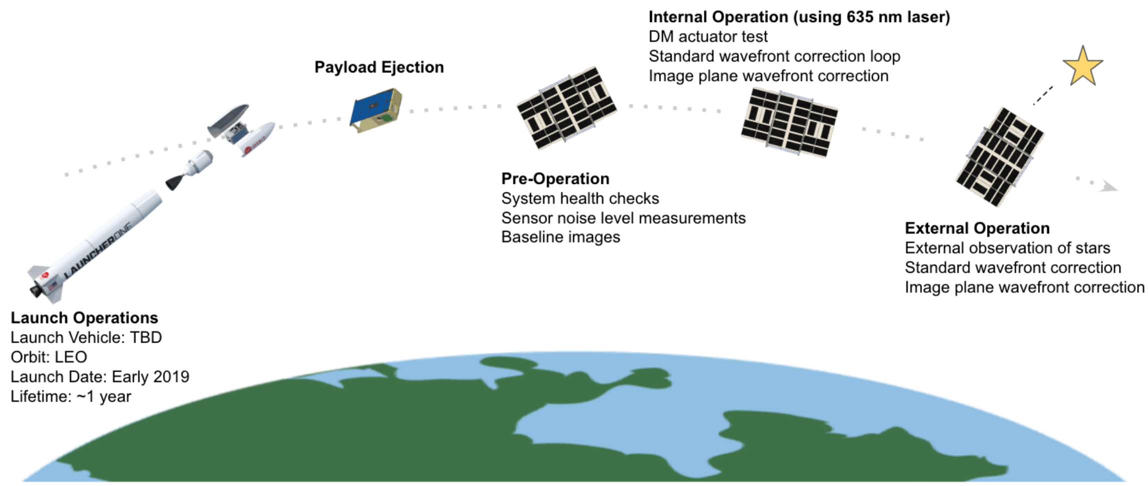


Figure 2-10: Overview of satellite operations. First, the payload will launch into a low-Earth orbit. Upon launch the satellite will de-tumble and perform pre-operation checks and baseline measurements. The payload will then perform internal and external observations over the course of its 1 year lifetime to characterize and demonstrate the deformable mirror technology.

the spacecraft can safely and correctly perform its desired functions. These checks will be different for internal and external operations, as they have different performance requirements. After the checks, the spacecraft will power on the required components for the specific mode of operation. The spacecraft will then test voltage and current to the components and finish by taking baseline measurements and image frames.

2.4.1 Internal Modes of Operation

For the internal operation, several system checks need to be done to ensure successful operation of DeMi. The spacecraft needs to ensure that the internal temperature, attitude control, data storage capacity and power supply to the payload fit the requirements. If the system checks pass, then the payload can power on the necessary components. For internal observations, the following components need to be powered on:

- Laser
- DM
- CMOS Camera

- SHWFS

During the internal operations, DeMi will perform three different demonstrations. Each of these demonstrations will use the internal laser source to illuminate the DM and take measurements. These three demonstrations are:

- Test all DM actuators to full displacement
- Run the wavefront correction loop on the internal laser
- Run the image plane wavefront correction loop on the internal laser

The first operational mode will characterize the DM by testing each individual actuator to full displacement. For each actuator, we will record a wavefront measurement and an image plane measurement.

The second operational mode tests the standard wavefront correction loop. This operational mode will use the SHWFS to measure the wavefront and will run a closed-loop correction between the SHWFS and the DM.

The third operational mode will test the image-plane wavefront sensing. This mode will use the imaging CMOS camera in a closed-loop with the DM to correct the wavefront from the internal source. The wavefront corrections will be based off of PSF measurements on the imaging sensor. This mode of operation will rely on a library of image-plane DM actuator PSF influence functions, and will not rely on the SHWFS.

2.4.2 External Modes of Operation

For external observations, the spacecraft needs to perform more system checks. In addition to the system checks needed for internal observations, the spacecraft also needs to check spacecraft pointing and stability as well as spacecraft position relative to eclipse. Once system checks are complete, the spacecraft will need to power on the DM, CMOS camera and the SHWFS.

The external mode of operation will perform astronomical observations while testing the wavefront correction loop. This mode will use the external aperture to image stars and use the closed-loop wavefront correction system to correct the PSFs. The correctional capabilities will be demonstrated using the standard wavefront correction loop as well as the image plane wavefront correction loop as described in the Section 2.4.1.

Chapter 3

Payload Integration

3.1 Overview

To test optomechanical design and optical configuration of the payload, we assembled several models of the payload. The first iteration was a full single 3D print of the payload with all optical components included in the print. This test verified the sizing and spacing of the payload components, as well as to help visualize the full payload layout. It was also useful to have a physical example of the payload for presentation purposes. The iterations that followed were more realistic integration tests of the payload components based on the existing CAD models. The 3D models are useful when making adjustments because they allow you to check and visualize the layout. 3D models also provide a means to test the integration and alignment procedures that are difficult to check just using the CAD model. 3D printing is relatively quick and cheap, so it is easier, more time efficient and more cost efficient to iterate on the payload configuration using 3D printed components for as long as possible.

The DeMi payload design is broken up in to a lot of individual sub-structures so that the positions of all optical components can be adjusted to facilitate optical alignment. There are three main decks on the payload. The primary deck is the base of the payload that holds the two other decks as well as some optics components. The wavefront sensor (WFS) deck houses the off-axis parabolic mirror (OAP) relay optics and the Shack-Hartmann wavefront sensor (SHWFS). The third deck is the mini deck, which houses the field mirror (FM) and M2. The detailed integration procedures can be found in Appendix C.

The primary goals for the integration testing were to find and correct errors in the design, test the integration procedures and make adjustments to facilitate payload integration, and test optical alignment. By going through several iterations of the payload integration, we have been able to refine the model and integration procedures in a timely and relatively inexpensive manner.

3.2 Iteration 1: White 3D Printed Payload

The first iteration of the payload assembly testing was done with a white 3D printed model of the payload. Each of the payload components (decks and optics mounts) were printed by Shapeways. This assembly was to test high-level payload configuration, spacing and tolerances.

3.2.1 Process

To test the integration of the payload components, I followed the integration procedures, detailed in Appendix C, and referenced the CAD model. I tapped each fastener hole to the correct threading and then fastened all of the components together in the correct configuration using the same fasteners we will use for the flight version of the payload. I tested each fastener hole to make sure it was the correct size, tested the bushing holes for the fine adjustment locations, and checked fastener hole alignment across the whole payload.

The integration of this version of the payload took about a week because I had to familiarize myself with the specific details of the model and the integration procedures. Since this was the first payload integration test, I encountered a large volume of errors that I needed to record and address. I also made sure to check all of the model in as much detail as possible to locate all of the errors and possible difficulties. The resulting payload configuration of the first 3D printed model is shown in Figure 3-1.

Once all of the hardware was assembled and tested, I attached some of the optical

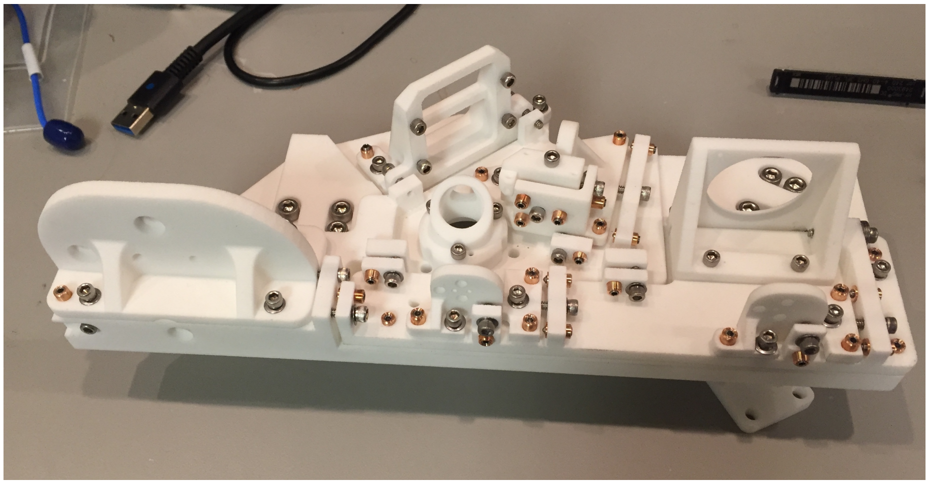


Figure 3-1: The completed assembly of the first iteration of the 3D printed DeMi payload model. All of the components are 3D printed from Shapeways. The fasteners, bushings and fine adjustment pins are from Thorlabs and are the same as what will be used for the flight version of the payload in order to test hole size and alignment. None of the optics are mounted here, just the optics mounts and the three decks.

components to the mounts (see Figure 3-2) in order to check the sizing and robustness of the mounts, as well as the optical layout of the design. During this process I found problems with the M1 mount, the FM mount, and the R2 mount. In addition to correcting bugs in the design, the DeMi team made some minor optical configuration changes, such as stretching out the relay OAP optic deck and shifting the SHWFS back. These changes were to facilitate easier payload integration optical alignment, as well as accommodate slightly larger relay OAPs.

3.2.2 Lessons Learned

The first iteration of payload assembly testing taught us a lot about the configuration of the payload. In total, I found about 10 major issues with the hardware. These issues included hole sizing, hole spacing, component spacing, clearance issues and other model imperfections. These issues were all addressed, corrected in the CAD model and then implemented in the next iteration of integration testing.

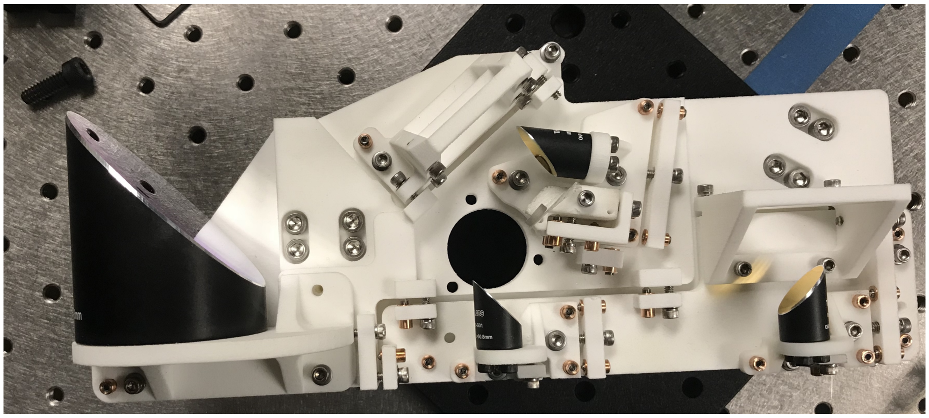


Figure 3-2: Assembly of the 3D printed components of the first iteration of the payload model with some optics mounted. The optics were mounted to check the mounts for robustness, spacing, fastener locations and to test overall optical layout of the payload.

3.3 Iteration 2: Black 3D Printed Payload

3.3.1 Process

The second iteration of the payload model incorporated all of the changes from the first iteration. All of the mechanical bugs, like hole sizing, hole spacing and tolerances, were corrected and the optical configuration was changed slightly. The main change with the configuration was the adjustment of the relay optics and the position of the SHWFS in relation to the relay optics. The WFS deck was extended slightly, moving the wavefront sensor farther away from the relay optics. This change is evident in Figure 3-3. The integration of the second iteration of the 3D printed model was almost identical to the assembly of the first model. I tapped all of the holes and followed the assembly procedures to secure all components of the payload into the correct configuration. Once again I checked hole sizes, hole spacing, and tolerances to make sure that the changes we had made to the model were implemented correctly.

After this initial phase of integration testing, I mounted the optical components to the assembly and started testing optical alignment of the payload. For the optical alignment, I used a phase shifting Zygo interferometer and followed similar techniques described in reference [4]. During this phase of optical alignment testing, I began by

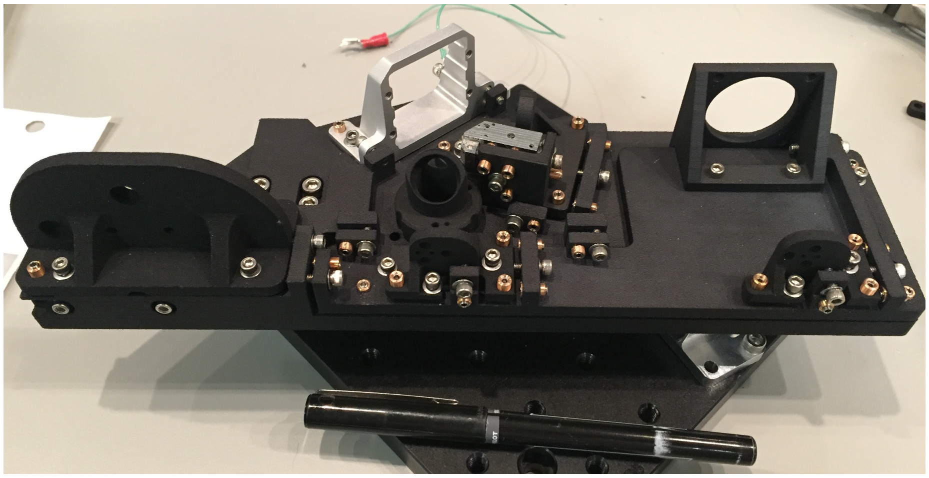
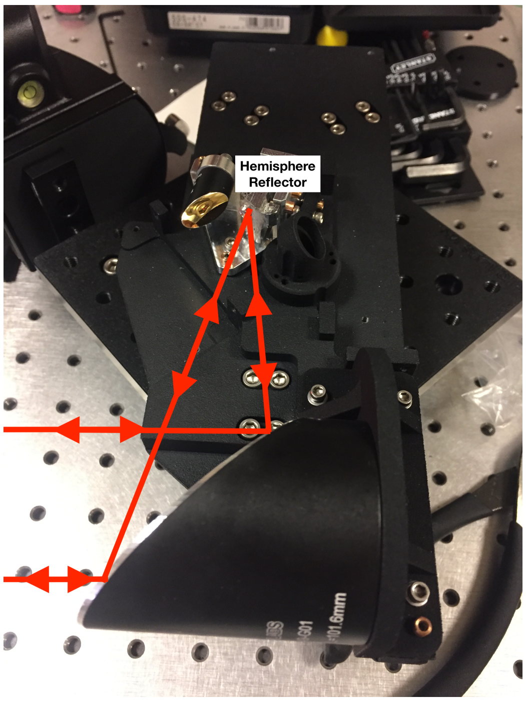


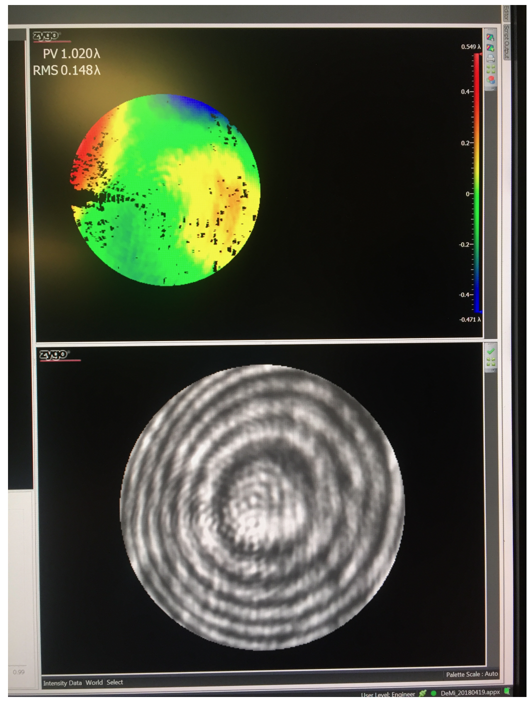
Figure 3-3: The completed assembly of the second iteration of the 3D printed DeMi payload model. All of the components, except for the aluminum DM mount, aluminum FM, and the aluminum feet shown in the image, are 3D printed from Shapeways. The fasteners, bushings and fine adjustment pins are from Thorlabs and are flight-like in order to test hole size and alignment. None of the optics are mounted here, just the optics mounts and the three decks.

testing individual optical components and pairs of optical components to get a sense for optical alignment procedures. I determined that optical alignment would be more successful if the subsystems were aligned before trying to align the whole system. In this stage, I tested the relay optical spacing and alignment by setting up the relay deck with the interferometer and a flat return mirror. I also tested the alignment of M1 with the FM by using a hemisphere attached to the FM to reflect the light back through the OAP to the interferometer. Figure 3-4 shows an example of the optical alignment testing.

The specific payload configuration used for initial optical alignment testing is shown in Figure 3-4(a). The interference fringes on the bottom of Figure 3-4(b) show the actual fringe pattern created by the interference of the wavefronts. A larger separation between the fringes means a better alignment of the optics. The figure on the top of Figure 3-4(b) shows the resulting surface error map measurement taken by the interferometer. The red and blue areas on the surface error map are areas with a high magnitude of surface error. The best optical alignment will have surface error in the green and yellow region. The difference in the highest and lowest surface



(a) The model configuration on the the interferometer. This configuration is testing the alignment of M1 at the beginning of the optical path on the payload with the FM. The light from the interferometer is reflected off of the the OAP and then off of the small hemisphere that is mounted to the FM. The wavefront then travels back to the OAP and into the interferometer to perform the alignment measurement.



(b) The resulting interferometric measurement of the configuration on the left. The top figure is the surface error map. The red areas are high and the blue areas are low. The bottom figure is the resulting interference fringes. The large circular fringes are due to focusing error and are not caused by surface alignment error, that is why the resulting surface error map is relatively flat even though there are a lot of fringes.

Figure 3-4: Optical alignment testing of the second iteration of the 3D printed payload model on the Zygo interferometer.

error measurements, or peak to valley, is denoted in the figure as PV and is about 1λ for this configuration. For our payload alignment, we are aiming for a PV of about $\frac{\lambda}{4}$. The surface error measurement for this configuration is two times worse than the actual error because the light from the interferometer is reflected back through the system. Because the light is routed twice through the optical system, the measured error is doubled.

3.3.2 Lessons Learned

While most of the egregious mechanical errors had been corrected after the first iteration, there were still some interesting conclusions drawn from this iteration of the payload design. There were a few minor details that needed to be corrected, such as an adjustment screw added to the M1 mount to enable easier alignment of the mirror. By the end of this iteration we had exploited all of the mechanical integration benefits of 3D printed models in this (nylon) material.

From the optical alignment testing done with this iteration, the most obvious lesson that I learned was that it is hard to do optical alignment testing on nylon 3D printed models. When I tried to adjust the positions of the optics with the fine adjustment screws I ended up bending the plastic that the optics were mounted to rather than moving the actual optics components. To do the more detailed optical alignment testing I would need to use an aluminum version of the payload. During this payload model integration I learned some important alignment techniques, as well as how to operate the Zygo interferometer and interpret the data. The initial alignment data from the tests I performed also indicated that there were no substantive optical alignment issues with the payload configuration, which gave me confidence that the payload could achieve the $\frac{\lambda}{4}$ PV design specification.

3.4 Iteration 3: Aluminum Payload

The third major iteration of the payload was made from aluminum instead of 3D printed material. The flight payload will be aluminum, so it is important to test all of the components mounted to the aluminum. Because most of the major bugs in the CAD model of the payload were fixed through the process of the 3D print assembly, the aluminum model was meant to check the tight-tolerance features, such as the optical alignment and overall payload operation.

3.4.1 Process

Just like with the 3D printed models, we had each individual component manufactured separately. We submitted CAD models and drawings of each part to the machine shop team at Boston University Scientific Instrument Facility machine shop, who then manufactured each piece out of 7075 aluminum. As each piece was finished, we tested the optics mounting and, on some of them, the smaller scale optical alignment. As more of the pieces were finished, we began testing integration of the hardware and components together into the whole payload

Several components went through multiple design iterations in aluminum. These components included the FM, the camera mount, the deformable mirror (DM) mount, and the SHWFS mount. These changes came about because of some tolerance and spacing issues and some optical alignment issues that were discovered only after doing the detailed integration and optical alignment testing.

With the aluminum payload we have been able to do much more thorough testing of the payload configuration. We have been able to fine-tune the optics mounting, sensor positioning, and fine adjustment procedures. We have also been able to practice using the embedded fiber in the FM, which led to the discovery of a slight misalignment of the fiber groove in the FM. We have bonded the small optical components like the beam splitter and the lenslet array to their appropriate locations so as to incorporate them into the overall payload testing.

During this integration process, the whole DeMi team has been involved and each person has specific roles in the integration. Since this integration is much more thorough and will be used for flight like testing of all components, the tasks have been divided. Currently there are four main graduate students working on the project, one post-doc, and me. We communicate through slack channels and keep all of the data,

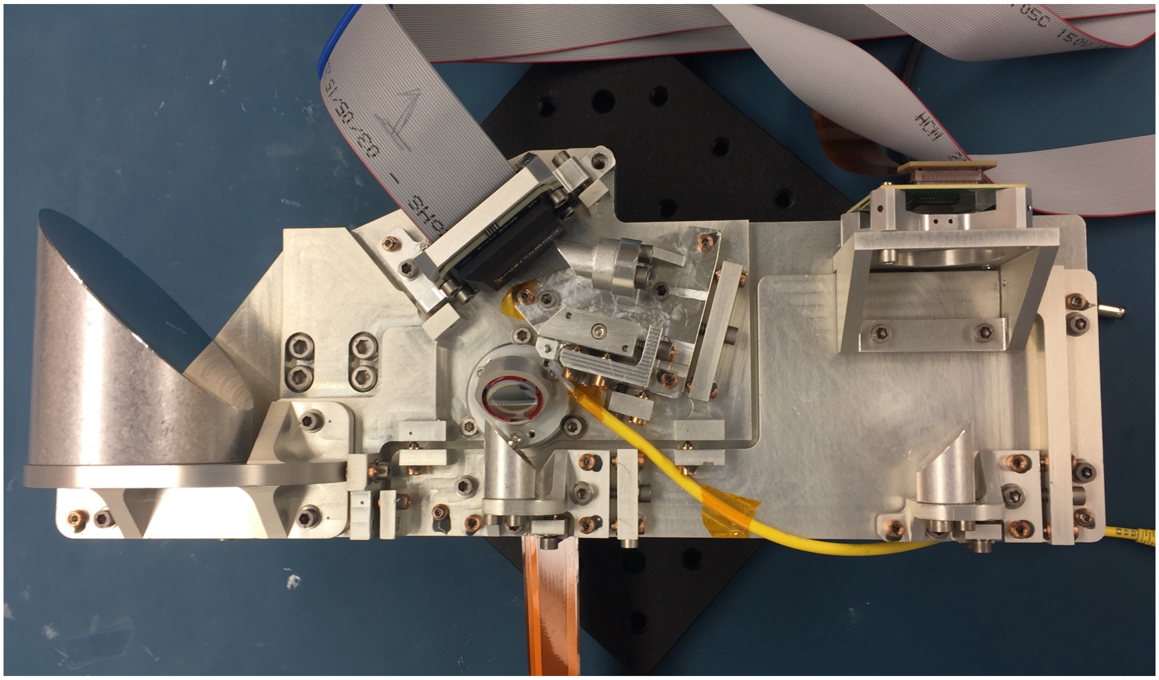


Figure 3-5: Fully integrated aluminum payload model including the DM, the WFS, the BS, and the image plane sensor. The fiber, inside the yellow tube, is embedded into the FM

code and notes in a github channel or on the dropbox.

3.4.2 Optical Alignment Testing

The more rigid aluminum payload enables more precise optical alignment tests. Some of this testing uses the Zygo interferometer, however a lot of the alignment is achievable with only the optics and the sensors.

3.4.3 Current State

Most recently I have been working on determining the correct positioning for the image plane sensor that is housed under the beam splitter. To do this, I have been using the injected FM fiber, M2, the DM and the beamsplitter (BS) to create a flight like wavefront. The camera sits below the BS and I am using point spread function (PSF) analysis code, found in Appendix D.2, to measure the PSF widths at different camera locations. Details about this testing can be found in Section 4.4.1.

While we have all of the components and sensors on the payload, as shown in Figure 3-5, some of the software has not been completely finished and the flight computers and drivers are still incomplete. For this reason, we have not yet been able to do the extensive testing that is needed, including end-to-end testing, to ensure mission success. We are currently finalizing the optical layout and making sure that all components have enough adjustment freedom to allow for near perfect alignment of the the system. Once the alignment and the algorithms are complete, we can perform end-to-end testing to understand system performance.

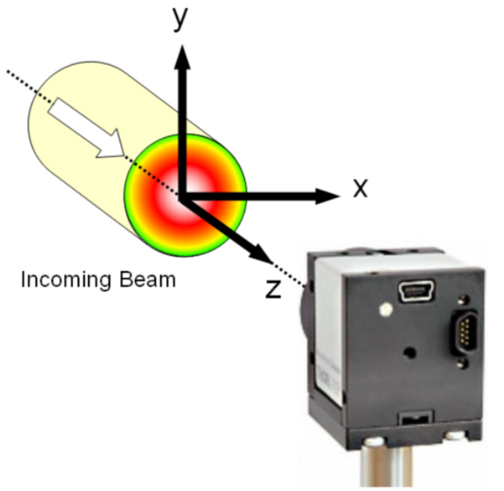
Since the payload is made out of aluminum, the same material as the flight payload, we may use some of the currently used components on the actual mission. We also might use some of the sensors and optical components on the flight payload. Because of this reason, all of the current integration and testing is being done in a clean room in the Space Systems Laboratory at MIT.

Chapter 4

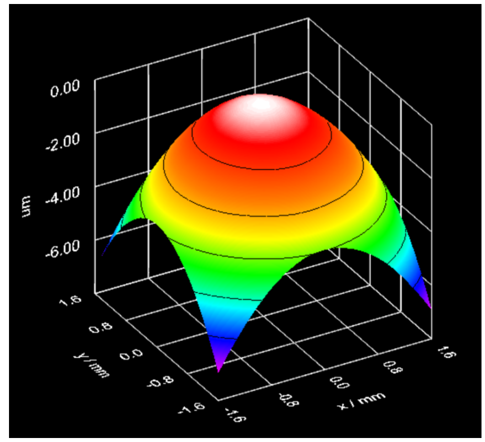
Component Testing

The success of the DeMi mission is highly dependent on component performance while in orbit. In order to ensure successful operation, we need to be able to accurately predict the behavior of each component in space. The first step to informing the performance prediction is doing a baseline characterization. The characterization is a description of the nature of a component. It identifies the inherent properties that may influence behavior. After these properties are understood, the component should be tested at the individual and system level. Testing in as many situations as possible will greatly improve predictions of in-flight behavior. Finally, all of the components should be tested together in the flight-like configuration to understand the baseline performance of the system as a whole. Once again, the system should be evaluated in as many situations as possible. Each component should be calibrated to address the specific needs of the mission and to allow for effective system performance

On DeMi there are three mission-critical components: the complementary metal-oxide semiconductor (CMOS) image plane sensor, the Shack-Hartmann wavefront sensor (SHWFS) and the deformable mirror (DM). During my time on the project I helped to perform characterization, calibration, and performance testing on these major components.



(a) Depiction of the coordinate system used by the ThorLabs WFS software. The z-axis is perpendicular to the plane of the wavefront. The x and y axis are the positions in the the plane of the wavefront.



(b) Example of a reconstructed wavefront as measured by the ThorLabs WFS software. The relative heights of the wavefront are measured in microns and are shown in the z-axis. The x and y axes show the positions on the sensor.

Figure 4-1: Representation of the coordinate system and resulting wavefront reconstruction orientation used by the ThorLabs SHWFS and its software [16].

4.1 ThorLabs Shack-Hartmann Wavefront Sensor Testing

4.1.1 ThorLabs Software Overview

The ThorLabs WFS150-5C SHWFS is a wavefront sensor (WFS) that follows the principles outlined in Section 2.2.2. It uses a spot field created by a microlens array to measure and reconstruct the incident wavefront. The ThorLabs SHWFS uses a coordinate system where the z-axis is pointing towards the sensor and the wavefront ”height” is measured along this axis. Figure 4-1(a) shows the coordinate system.

When the ThorLabs software measures the wavefront, it measures the height (or the path difference) of the wavefront relative to the center of the wavefront. Therefore, a convex incident wavefront with respect to the sensor will have negative wavefront height measurements around the center, as shown in Figure 4-1(b). The wavefront reconstruction can be visualized in 3 dimensions and can also be saved as a data set.

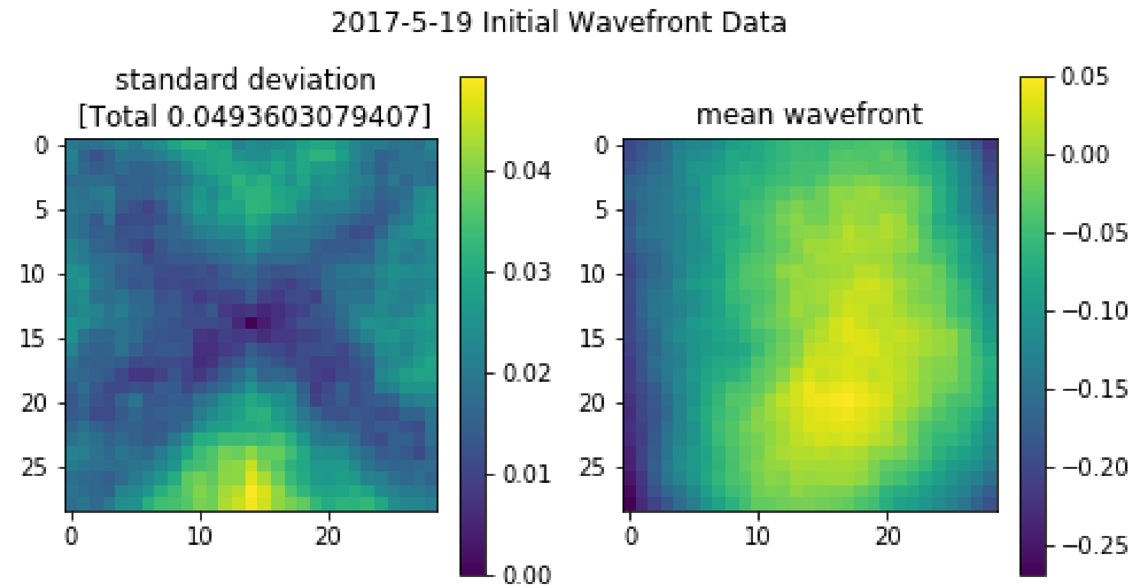


Figure 4-2: Side-by-side view of the mean wavefront reconstruction measured and calculated by the WFS and the standard deviation of the measurement. The wavefront was determined by a ThorLabs WFS150-5C SHWFS. The x and y axes are the positions on the wavefront sensor and the color map corresponds to height measurements of the wavefront in μm. The right plot shows the average of 10 wavefront reconstructions calculated by the ThorLabs WFS and its wavefront sensing algorithm. The left plot shows the standard deviation plot of each height measurement of the reconstructed wavefront taken by the ThorLabs SHWFS and its algorithm.

For my analysis, I used these wavefront reconstruction data sets to characterize the performance of the ThorLabs SHWFS.

4.1.2 Performance Testing

One of my first tasks on DeMi was studying the WFS for the adaptive optics (AO) system. Initially, I worked with a ThorLabs SHWFS that was equipped with its own wavefront reconstruction algorithm and user interface. A description of how a SHWFS works can be found in Section 2.2.2 and details about the ThorLabs wavefront sensing software are included in Section 4.1.1. I used this WFS to do initial component testing and to analyze strategies for incorporating a SHWFS into the DeMi optical system. The initial plan was to incorporate some or all of the ThorLabs wavefront sensing software into our system, so I also analyzed and characterized the performance.

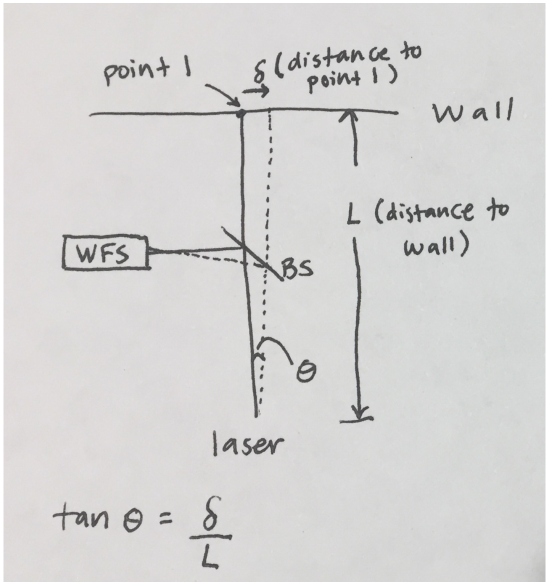
The first tests I performed were wavefront reconstruction tests with the ThorLabs software. In these tests, I set up the WFS in the path of a collimated light source, so that the incident wavefront should be parallel and planar, and ran the ThorLabs software that measured and reconstructed the wavefront. From these wavefront reconstructions, I used a Python script, included in Appendix D.3, to average 10 wavefront reconstructions and to calculate the standard deviation of the wavefront measurements, to see how well the software reconstructed the wavefront. The standard deviation,

$$\sigma = \sqrt{\langle (\mathrm{x} - \bar{\mathrm{x}})^2 \rangle}, \tag{4.1}$$

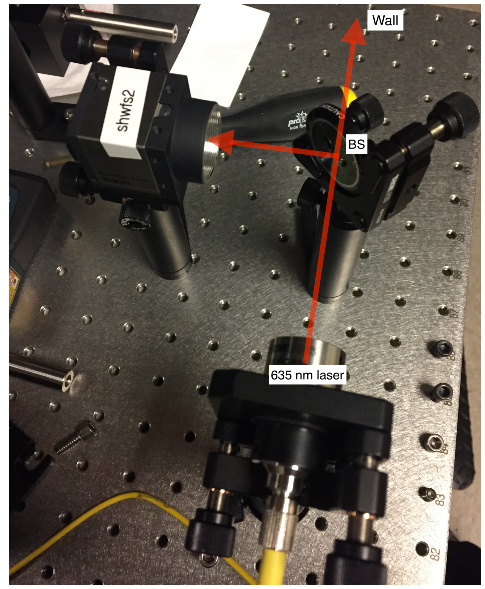
where σ is the standard deviation, x is the given value of the measurement, and $\bar{\mathrm{x}}$ is the average value of the whole array of data, measures the spread of values away from the mean in order to characterize the WFS software performance. In this case, I measured the standard deviation of the average height measurement of the wavefront at each point. An example of the average wavefront reconstruction and the standard deviation of the height measurements of the reconstructed wavefront can be seen in Figure 4-2.

Through my analysis, I discovered an interesting feature of the ThorLabs WFS and wavefront reconstruction algorithm. In the standard deviation plots, I discovered an X-shaped feature in which the standard deviations of the wavefront were zero, or close to zero. This shape can be seen in the left frame of Figure 4-2. I initially was not sure why this pattern was appearing and wondered if it was due to a characteristic in my optical setup. To test this theory, I took wavefront measurements after rotating the laser, using different filters in front of the WFS, using a different SHWFS, and changing the settings in the ThorLabs software. Ultimately, all of the tests resulted in the same X-pattern in the standard deviation plot.

To double check that the X-shaped pattern was a feature of the wavefront sensor and not of my optical set up or Python script, I contacted tech support at ThorLabs and explained the feature and my testing to them. After they ran some tests of their own, they encountered the same X-shaped pattern when taking the standard deviation of the reconstructed wavefront. This led me to believe that the wavefront



(a) Diagram of the tilt testing setup showing how I used the various measurements to calculate the angle of the wavefront and compare it to the angle measured by the ThorLabs SHWFS. I found the angle of tilt of the beam, θ, by using measurements of the distance from the laser to the wall, L, and the distance between the displaced point and the reference point 1, δ.



(b) Test setup to determine tilt measurement accuracy of the ThorLabs SHWFS. The test setup consists of a 635 nm wavelength laser mounted on a tip-tilt stage, a beam splitter, and the ThorLabs SHWFS. I used the tilt knob on the laser mount to tilt the wavefront by small amounts.

Figure 4-3: Tilt measurement setup diagram and image of the components used.

reconstruction algorithm started in the center of the wavefront and worked its way to the corners in order to reconstruct the wavefront. Because of this odd feature in the wavefront reconstruction algorithm, and the lack of customizability of the ThorLabs software, we decided to create our own wavefront sensing and reconstruction algorithms.

4.1.3 Tilt Measurements

With the WFS setup shown in Figure 4-3, I tested how well the ThorLabs SHWFS could measure the tilt (angle of incidence) of a wavefront. I used the ThorLabs SHWFS, a beam splitter, a 635 nm wavelength laser mounted to a tip/tilt stage, and the wall. First, I located the point on the wall that corresponded to no tilt of the laser, and therefore the wavefront. Then, I tilted the laser by several different small amounts and measured the displacement of the spot on the wall, as well as used the

ThorLabs SHWFS to measure the tilt angle of the wavefront. I compared the two measurements to test the accuracy of the ThorLabs SHWFS wavefront measurement and reconstruction algorithm for several different tilt amounts. Table 4.1 shows the various tilt measurements I made and the comparison of the measurements.

Point	Distance to point 1 (mm)	Distance to wall (mm)	Tilt measured from wall (deg)	Tilt measured by WFS (deg)	Difference (deg)
1	N/A	2421 mm	N/A	N/A	N/A
2	7.5 mm	2421 mm	0.1775^o	0.1766^o	0.0009^o
3	16.5 mm	2421 mm	0.3905^o	0.3531^o	0.0374^o
4	6 mm	2421 mm	0.1420^o	0.1324^o	0.0096^o
5	12.5 mm	2421 mm	0.2958^o	0.2882^o	0.0076^o

Table 4.1: Measurement table of the tilt determination of the wavefront sensor versus the actual tilt of the wavefront. Point one is the baseline point with no tilt that I used to measure the tilt from the wall. Using the distance from the laser to the wall and from the point to point 1, I was able to calculate the angle of tilt.

4.1.4 Wavefront Determination Algorithms

In addition to looking at the performance of the SHWFS, I also worked on the centroid displacement determination code, a critical piece in the wavefront sensing algorithm as described in Section 2.2.2, for MIT's custom SHWFS. To do this, I adapted a star tracking algorithm written by Julian Brown, a previous graduate student in the STAR Lab. Julian's star tracking algorithm determined the pixel locations of each star in the star-field and measured the distance between the stars to locate recognizable patterns that could help determine the relative location of the spacecraft that took the image. I adapted the algorithm to, instead of locate all of the stars in the star field, locate all of the centroids created by the microlens array, and determine their displacements from the reference positions. The full code can be found in Appendix D.4. This code was then modified and incorporated in to the wavefront sensing algorithm being developed by the DeMi software team.

4.2 Image Plane Sensor Testing

A lot of my work with the image plane sensor involved writing code to analyze the resulting point spread functions (PSF)s from the tests. I tested out some charge-coupled device (CCD) cameras as well as CMOS cameras and worked on interfacing the Pixelink CMOS camera to a Raspberry Pi, which will be DeMi's flight computer.

4.2.1 Characterization and Calibration

In the STAR Lab space at MIT I characterized and calibrated the CMOS cameras. I designed a box to hold the CMOS camera and open/cover the aperture to protect the camera and to have more control over the light entering the aperture.

To characterize the sensors, I took bias images, dark images, and flat images. Bias images are images taken with no exposure time, or the lowest possible exposure time and no light entering the aperture. This type of image can be used to measure the read noise of the sensor. Dark images are taken with no light entering the aperture, but with a finite exposure time. These images are used to measure the base level of counts on each pixel in the sensor without any light. This corrects for the hot pixels and the dark current on the sensor. Flat images are taken with an exposure time that matches that of the dark images and are images of a uniform intensity background. Flats are used to correct for any optical imperfections in the system, such as dust particles on the optics and scratches.

I used these calibration images to calculate a few of the characteristics of sensor including the dark rate, gain and the read noise. The dark rate,

$$\text{Dark Rate} = \frac{\sqrt{\langle \text{dark frame}^2 \rangle}}{\text{time}}, \tag{4.2}$$

measures the thermal noise of the sensor. The gain,

$$\text{Gain} = \frac{\langle \text{flat frame} \rangle}{\text{variance}}, \tag{4.3}$$

and,

$$\text{variance} = \frac{\sigma_{flat}^2}{2}, \tag{4.4}$$

where σ_{flat} is the standard deviation of the flat image, is a measurement of how many electrons captured by each pixel correspond to one analog to digital unit (ADU), otherwise known as a count. For example, a gain of 2 electrons/ADU means it takes 2 photons to generate a signal of 1 count on the pixel. The read noise,

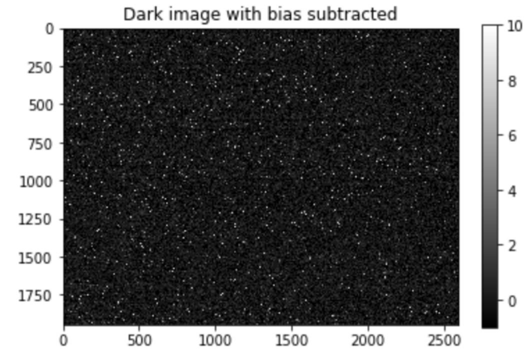
$$\text{RN} = \sigma_{bias} \times \text{Gain}, \tag{4.5}$$

where σ_{bias} is the standard deviation of the bias frame and the Gain is calculated from Equation 4.3, is the noise generated by the sensor and electronics as the data is read off of the chip [12].

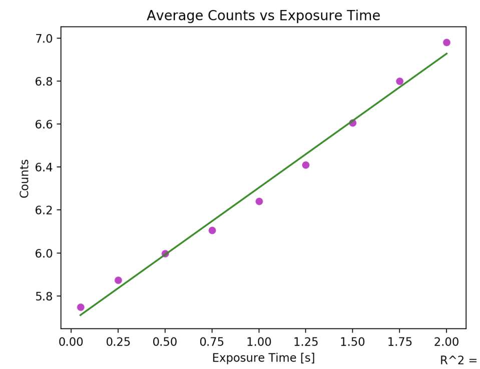
Some of the data taken from the CMOS sensor calibration and characterization is shown in Figure 4-4. From the bias frames, dark frames, and flat frames, I calculated the gain to be 28.2 electrons/ADU, and the read noise to be 12.7 electrons/pixel. While the dark rate measurements are ongoing, the most recent data at room temperature gave a dark rate of approximately 17 electrons/second, which is the slope of Figure 4-4(b) multiplied by the gain.

4.2.2 Raspberry Pi Interfacing

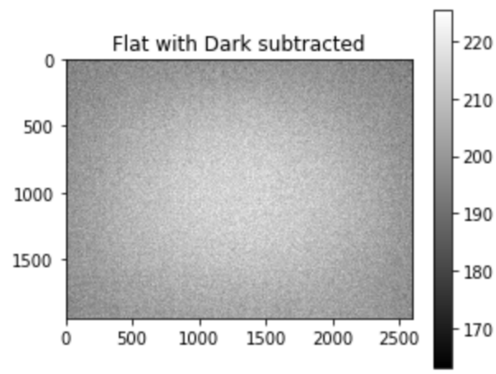
A large amount of my time working with the CMOS camera was spent learning how to interface and control the camera with the flight computer, a Raspberry Pi. First, I located Pixelink software that could work on a Linux system, what the Raspberry Pi uses, and worked on connecting the camera, installing the software, and learning how to control the camera from the Pi. The only software available for the Raspberry Pi operation was an SDK package that required a lot of configuration. I worked through configuring the software and documented the process so that when the actual flight sensors are interfaced with the real flight computer, the Raspberry Pi Compute Module 3s, the initial setup is clear and easy. The initial configuration enabled



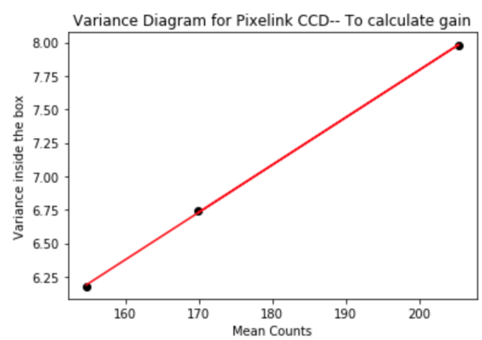
(a) An example of a dark frame, with the bias subtracted, taken with an exposure time of 2000 ms and at a temperature of 20.9 degrees Celsius.



(b) Plot of the mean counts versus exposure time from several dark frames. The green line is the linear fit used to calculate the dark rate. (figure credits: Abby Stein)



(c) Example of a flat field image taken on the CMOS camera.



(d) A plot of variance of the flat field image versus mean counts of the flat field image. The slope of the fit line is approximately 1/Gain.

Figure 4-4: Some data from the CMOS sensor characterization and calibration I performed.

control of the camera, such as setting exposure times and taking images, by use of the Raspberry Pi command line or executable scripts.

More extensive work needs to be done to set up the operating procedures and scripts for the CMOS cameras, both imager and SHWFS. The Raspberry Pis will also need to be interfaced with the other componenets.

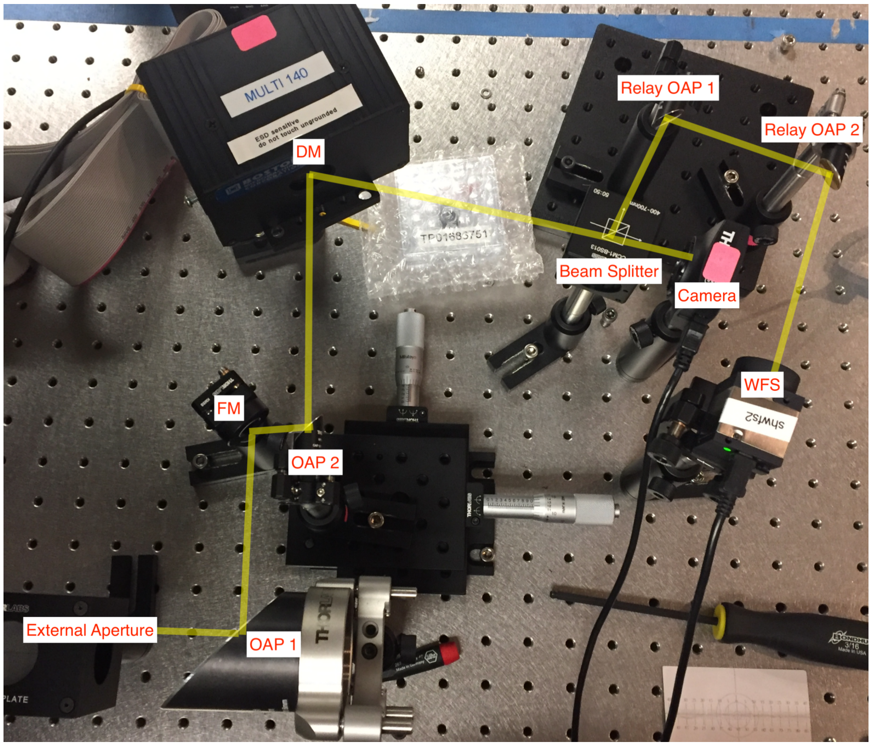


Figure 4-5: Complete bench-top optical system layout of the payload. This setup was produced to perform system testing and individual testing. It was used to verify the WFS operation, the PSF generation on the camera, and for DM actuation testing. This layout is very similar to the final design of the payload optics. We tested several optical layouts before deciding on this one.

4.3 Lab Bench and Deformable Mirror Actuator Testing

After the individual sensor testing was performed, I began working with combinations of the sensors and optical components to test more applicable performance of the sensors and test system performance between the various components. I performed these tests in the STAR laboratory space on the optical bench. I started with the smaller scale systems, like using a series of off-axis parabolic mirrors (OAP)s to generate a PSF on a camera. Then, I began incorporating other components like flat mirrors, beam splitters and the wavefront sensor to test tilt measurements and compare PSFs to the reconstructed wavefront generated by the wavefront sensor. Finally, I integrated the DM and did full layout testing with various optical configurations.

4.4 Integrated Testing

The most recent testing I performed involved the CMOS camera that is mounted to the bottom of the payload bench in the clean room. While I have already done a substantial amount of sensor testing in the optics lab, there is still a lot of testing that needs to be done on the integrated payload to check the system operation, sensor control, and sensor performance in the flight configuration.

4.4.1 CMOS Focus Testing

One of my first tasks for sensor testing on the integrated payload in the clean room was to focus the CMOS camera used for the image plane sensor. This camera sits below the beamsplitter and is used to measure the PSF of the incoming wavefront. The CAD model of the mount is shown in Figure 4-6. By loosening the screw on the camera mount, the camera can slide up and down in the mount and can be set to the correct height.

Initially, I worked on determining a rough focus location for the camera. I needed to run some DM actuator tests with the current setup, and a roughly focused image plane sensor was good enough. For this configuration, I loosened the camera clamp, powered the injected laser fiber, and used the Pixelink GUI on the cleanroom laptop to see a live view of the PSF on the sensor. I moved the camera up and down until it was at approximately the position with the smallest PSF. The resulting PSF from the roughly focused camera position is shown in Figure 4-7. Because the payload has not gone through all of its optical alignment, and there are a few outstanding issues with some of the components, such as the fiber injection slot, the PSF is not perfect. The unusual artefacts in the PSF are possibly due to incorrect fiber injection, stray light being reflected off of other optical components and being captured by the sensor, or other system imperfections. Once the system is more precisely aligned, these errors should disappear.

To achieve higher precision focus of the camera, we will use small shims with known thicknesses to set the camera at several known locations that straddle the

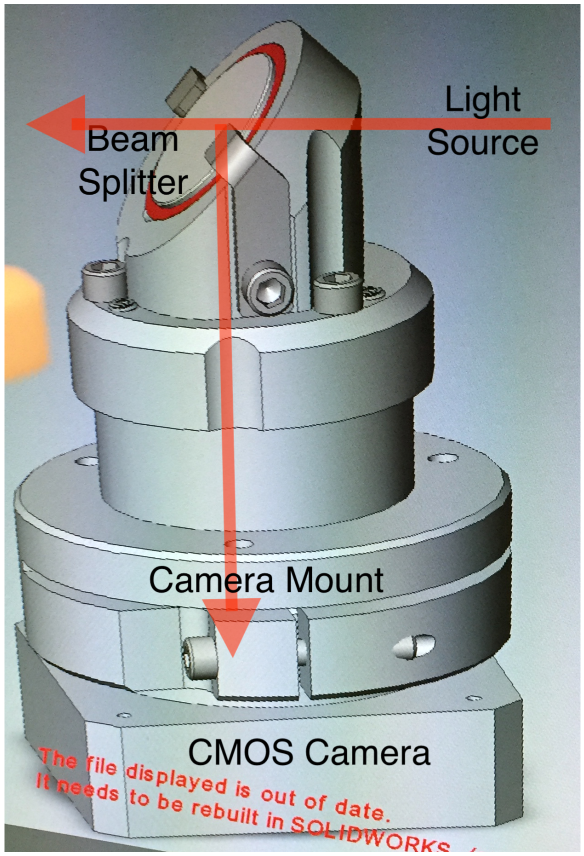


Figure 4-6: The CAD model of the Pixelink CMOS camera mount and beam splitter assembly. This assembly straddles the payload optical bench. The beam splitter peice sits above the bench and is used to send the incoming beam to the CMOS image plane sensor and to the wavefront sensor. The camera mount and camera sit below the payload optical bench. The position of the camera relative to the assembly can be adjusted by loosening the camera mount clamp and sliding the camera up or down in the mount.

estimated focus location. At each location, we will capture the PSF and measure its width using the same Gaussian PSF analysis code. The location versus PSF width data can then be plotted and fit to determine the location of best focus for the camera.

To verify that the camera is in the position of best focus we will check the focus of both the internal source and the external source on the sensor. This cross-check will confirm that there are no alignment or positioning errors in the first section of the optical path. To simulate the external source, we will use a collimated beam, made out of lasers and a series of lenses, placed in front of M1.

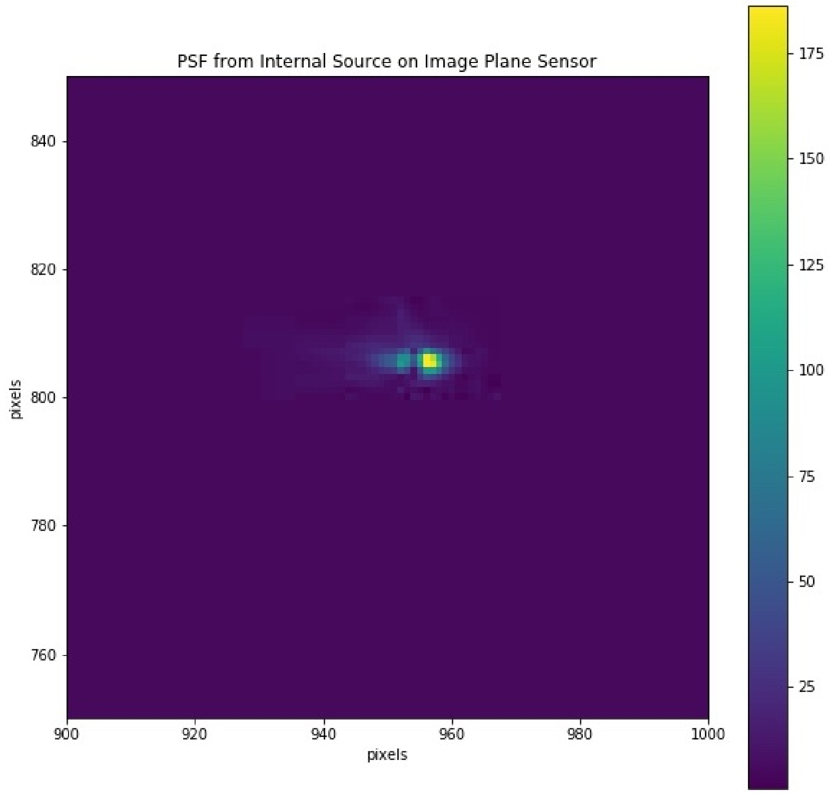
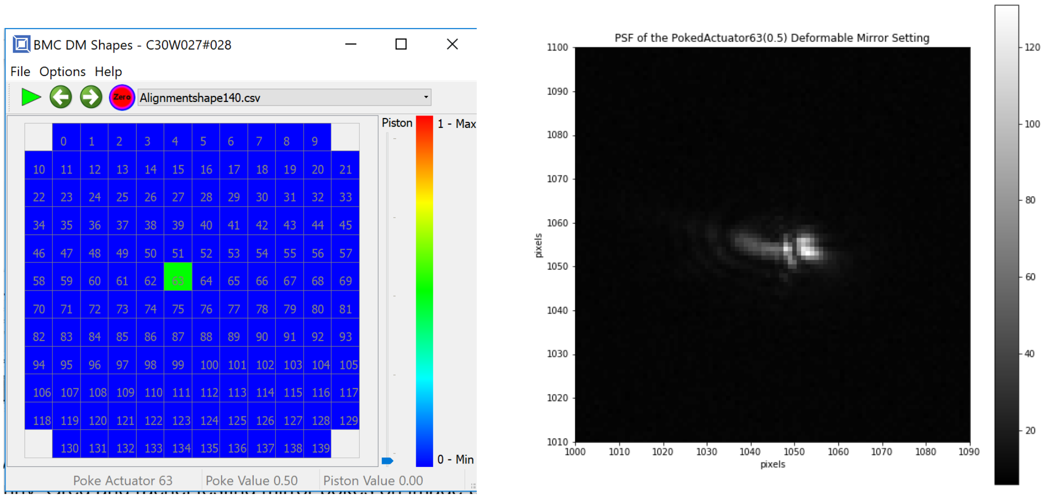


Figure 4-7: The PSF on the image plane sensor located under the beamsplitter. This is the PSF created by the 635 nm wavelength fiber that is injected into the field mirror. The wavefront travels from the fiber to the M2, off of the DM and then through the beamsplitter to the camera. Using a Python script I will measure the PSF characteristics at multiple image plane sensor positions to determine the optimal location for the camera.

4.4.2 CMOS PSF with DM Actuation

With the image plane sensor in a generally focused position, I was able to run tests of DM actuation with the internal source to see how the actuation affected the PSF. In my Gaussian fit Python notebook (mentioned in Section 4.4.1), I created a block of code that allowed me to access the Raspberry Pi and CMOS camera through SSH and measure the resulting PSFs. The DM was connected to the clean room laptop and controlled by the DM software on the laptop. As explained in section 2.2.1, the DM has 140 working actuators. The DM software allows you to move individual actuators, a combination of actuators, or a pre-programmed shape on the mirror. Figure 4-8(a) shows the DM actuator map that you can use to specify which actuators you would



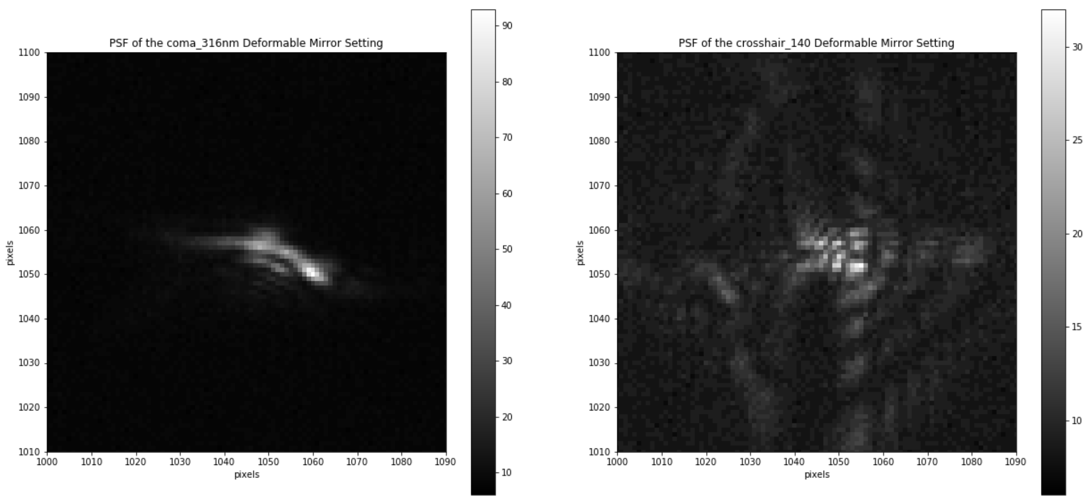
(a) Map of the BMC DM actuators on the DM controller software. In this configuration, actuator number 63 is poked to 50% of its maximum displacement.

(b) Image of the internal source at 0.03 mW after DM actuator 63 was poked to 50% of its maximum displacement. This figure is zoomed in and the maximum value of the image was 131 counts. The the Gaussian fit has a FWHM of about 2.6 pixels.

Figure 4-8: PSF of the internal laser source after the DM actuator in the actuator map is poked.

like to move. The figure of the map shows that actuator number 63 is poked and Figure 4-8(b) shows the resulting image on the CMOS image plane sensor.

In addition to moving the individual actuators, the DM software has the ability to poke shapes in the mirror. Figures 4-9(a) and 4-9(b) show how different DM patterns affect the PSF from the internal source. These are just fun demonstrations of how the DM can influence the image from the light source.



(a) Image of the internal source at 0.03 mW after the DM was actuated with the coma setting. The image has a maximum value of 93 counts and the Gaussian fit has a FWHM of approximately 9.2 pixels.

(b) Image of the internal source at 0.03 mW after the DM was actuated with the crosshair setting. The image has a maximum value of 32 counts and the Gaussian fit has a FWHM of approximately 9.1 pixels.

Figure 4-9: Resulting images of the internal source on the CMOS image plane sensor after various patterns were supplied to the DM.

Chapter 5

Conclusion

5.1 Contributions Overview

During my time working on the DeMi mission, I worked on an array of tasks. One of my largest contributions to the project was the integration testing of the payload. During this phase of the project, I assembled two iterations of 3D printed models and an aluminum model of the payload. These assemblies were used to check payload configuration and were helpful in visualizing changes that needed to be made to the design. Throughout this process, I identified a long list of issues that were subsequently addressed and corrected.

In addition to payload integration, I performed component testing of the complementary metal-oxide semiconductor (CMOS) image plane sensor, the Shack-Hartmann wavefront sensor (SHWFS), and the deformable mirror (DM). This testing included calibration and characterization of the CMOS sensors used in both the image plane sensor and the SHWFS, performance characterization of a SHWFS, and DM actuator performance evaluation.

Parallel to the component testing, I created Python scripts to analyze the data. These scripts included point spread function (PSF) measurement algorithms, Gaussian fit scripts, and other image processing scripts to measure read noise and gain of the CMOS cameras. I also worked on an algorithm to measure the positions of the centroids created by the SHWFS to enable wavefront reconstruction.

Furthermore, I worked on interfacing the image plane sensor and SHWFS with the Raspberry Pi flight computers. This included installing and configuring the software developer's kit (SDK) for the Pixelink CMOS cameras on the Pi and learning how to control the sensors' operation from the Pi.

Finally, I created a detailed concept of operations (ConOps) for the mission. The ConOps includes a comprehensive list of all operational modes and outlines the progression of payload states and functions as it performs each task.

5.2 Future Work

Before the payload is delivered for integration in to the spacecraft bus, it needs to go through more testing. While we have done a lot of work on the optical alignment, we need to improve the accuracy of this alignment and do comprehensive testing to ensure that the wavefronts from both the internal and external source will travel in the correct path. These small adjustments can be done both with the Zygo interferometer and with the PSF measurements from the CMOS cameras.

Additionally, the flight software and electronics are in progress, but have not been completed. Once all driver boards and software are ready, we need to test the performance of the software with the various payload components they control. We need to verify the operation and control to ensure that the system can function autonomously.

Once all sub-system testing is complete, we will do end-to-end testing of the payload with all components operational. This end-to-end test will be a demonstration of the corrective capabilities of the system and will incorporate all components and software. The plan is to check the corrective capabilities for both the internal and external source.

If time and resources allow, we would like to do environmental testing in a thermal-vacuum chamber (TVAC) to show that the payload will operate nominally in the space environment.

5.3 Applications

Microelectromechanical systems (MEMS) DMs have a wide variety of applications in space. The specific target application of the DeMi mission is for coronagraphic direct imaging of Earth-like exoplanets. The correctional capabilities of MEMS DMs enable the necessary contrast ratio of 10^{-10} to image the faint exoplanets next to their host stars.

Another application for this technology is use in laser communication systems. This includes both in space-to-ground communication and inter-satellite communication. The correctional abilities of MEMS DMs will improve signal transmission and pointing.

Because of its precise wavefront control, MEMS DMs also have possible uses in Earth imaging from space, and high-power laser space systems.

Appendix A

List of Acronyms and Abbreviations

AO: Adaptive optics

BMC: Boston Micromachines Corporation

BS (component label): Beam splitter

CAD: Computer-aided design

CCD: Charge-coupled device

CMOS: Complementary metal-oxide semiconductor

ConOps: Concept of operations

COTS: Commercial off-the-shelf

DeMi: Deformable Mirror Demonstration Mission

DM: Deformable mirror

FM (component label): Field mirror

L1 (component label): Image plane sensor

M1 (component label): The first, and largest, off-axis parabolic mirror

MEMS: Microelectromechanical systems

MIT: Massachusetts Institute of Technology

OAP: Off-axis parabolic mirror

PSF: Point spread function

PV: Peak to valley

R1 (component label): The first relay off-axis parabolic mirror

R2 (component label): The second relay off-axis parabolic mirror

SDK: Software developer's kit

SHWFS: Shack-Hartmann wavefront sensor

SSH: Secure shell

STAR Lab: Space Telecommunications, Astronomy and Radiation Laboratory

TVAC: Thermal vacuum

WFS: Wavefront sensor

Appendix B

OAP Alignment Procedures

Aligning two OAPs can be a long and tedious process. I figured out that it is best to start from the source and work towards the last OAP. Only introduce the next piece of the layout once all previous components are aligned.
Here are some useful steps to help with the alignment of the OAPs:

1. Orient the collimated beam source

    (a) Use a shear plate to look at the collimated beam. Set up the shear plate perpendicular to the desired direction of propagation of the beam

    (b) Adjust the direction and positioning of the beam until the fringes on the shear plate line up parallel with the reference line and are as low density as possible

        i. Adjust the position of the fiber relative to the collimator (left, right, forward, backward)

        ii. Adjust the position and direction of the beam until it travels in the desired direction (try to have the beam travel parallel with the optics table– irises are a good tool for this)

2. Adjust the heights of all of the optical components (OAPs/cameras/sensors/shear plate) to match the height of the collimated beam

3. Align the large OAP

   (a) Set up the large OAP at the desired distance from the collimator/aperture with the back plate of the OAP in the same orientation as the shear plate (perpendicular to the direction of propagation)

   (b) Adjust the rotation of the OAP using a spanner (make sure the reflected beam travels at the same height as the incident beam and continues parallel to the optics table)

   (c) Use a camera (I used a ThorCam) to adjust the OAP to focus the beam as best as possible

      i. Adjust the tip and tilt of the OAP using the knobs on the mount

      ii. You may also need to adjust the position of the camera to make sure it is placed at the focus (check this after a lot of tip/tilt changes)

      iii. You may have to iterate through the large OAP alignment process several times until you get a focused beam that travels parallel to the table

4. Align the small OAP

   (a) Place the small OAP on a y-direction moveable stage and at a distance equal to the specified reflected focal length (RFL) of the small OAP from the focal point

      i. Make sure the back plates of the OAPs are parallel– the OAPs only work to collimate or focus the beam in one orientation

   (b) Adjust the rotation of the OAP using a spanner and make sure the beam travels parallel to the table

   (c) Adjust the tip and tilt of the OAP using the knobs on the mount to get the beam to travel parallel to the source beam

   (d) Adjust the y direction, moving the small OAP towards or away from the large OAP, using the moveable stage

(e) Watch the pattern on the shear plate and make adjustments ii-iv until you get parallel fringes of low density that align with the reference line

To find the relevant focal length information for ThorLabs OAPs, go to the ThorLabs website. For more help aligning OAPs, check out the article in reference [10].

Appendix C

Payload Integration Procedures

Unless otherwise specified, each adjustable part should initially be position of at middle of adjustment. Do not torque any fasteners until preliminary alignment is complete.

Flight assembly checklist *Doesn't apply to 3D printed model*

1. appropriate strength/alloy fasteners
2. parts are clean
3. you are wearing cleanroom gloves
4. very small quantities of braycote 601EF have been applied to threads of each fastener (this will minimize the risk of galling and maximize the number of possible fastener cycles. *REMEMBER to account for lubrication in all torques*
5. belleville washers are installed on each fastener
6. torque wrench(es) are preset or fixes to desired torque and can't be used on the wrong fastener please don't get finger prints on engineering optics either.

Fastener selection

Mark Egan's fastener analysis indicated >100,000 psi fasteners are required. A286 stainless allow was selected since it is affordable compared to cadmium plated grade 8 or PH-4 fastener it, meets this requirement, and of course because "A286 is the best fastener material for aerospace usage." (NASA Fastener Design Manual, Barrett 1990). DO NOT USE ANY FASTENER IF YOU CAN NOT CONFIRM IT IS A286.

Three decks

1. primary deck: has telescope (OAP1+DM+ minideck)
2. WFS deck: has relay lens assembly and SHWFS
3. mini deck: has field mirror and OAP2

Payload assembly *change gloves before you handle/install every new optic*

1. fasten feet to primary deck
2. fasten OAP1 mount to mirror deck
3. fasten "mini" field/OAP2 deck
4. fasten assembled beamsplitter assembly to deck
5. fasten (appropriate grade grade) DM on DM mount
6. fasten field mirror to minideck without fiber
7. torque OAP1 to mount (torque these fasteners now so mirror doesn't move after adjustment)
8. torque OAP2 to mount (torque these fasteners now so mirror doesn't move after adjustment)
9. Install mounted DM
10. install camera + lens assembly
11. install relay OAP *mounts* on WFS deck
12. install reference flat on OAP3 mount (AKA R1)
13. install WFS Deck on primary deck
14. align telescope
15. focus imaging camera
16. remove WFS deck, align OAP3/OAP4 independently
17. insert fiber and and adjust position until focused on camera
18. reinstall WFS deck
19. align telescope to WFS

Appendix D

Python Scripts

D.1 PSF Analyzer

The code below was written in a Python notebook in Jupyter

```
import numpy as np
from PIL import Image
import astropy
from astropy.io import fits
import poppy
import matplotlib.pyplot as plt
import os
%matplotlib inline
from os.path import expanduser
home = expanduser("~")

# Specify which file you want to open (each file contains full set of
    images)
img1 = Image.open(home + "filepath/file.tif")
img2 = Image.open(home + "filepath/file.tif")

# Create a 3D cube to import all data
cube1= np.zeros((1024,1280,100), dtype=np.float)
cube2= np.zeros((1024,1280,100), dtype=np.float)

```

```
# Add the individual images to the cube
for i in range(100):
    try:
        cube1[:,:,i] += np.array(list(img1.getdata()), dtype=np.int).
    reshape((1024, 1280))
    except EOFError:
        break

# Add the individual images to the cube
for i in range(100):
    try:
        cube2[:,:,i] += np.array(list(img2.getdata()), dtype=np.int).
    reshape((1024, 1280))
    except EOFError:
        break

# Calculate the average
average1= cube1.mean(axis=2)
average2= cube2.mean(axis=2)

max1=np.max(average1)
max2=np.max(average2)

print('maximum pixel value image 1=', max1)
print('maximum pixel value image 2=', max2)

# Power information for the two images
power1=0.00001 # W
power2=0.00014 # W

expt1=0.00039 # s
expt2=0.01792 # s

# Want to take the ratio of flux/time to scale up saturated image to
    match ratio of unsaturated image
energy1 = power1*expt1
```

```
print('energy 1=',energy1)
energy2 = power2*expt2
print('energy 2=',energy2)

# Ratio of energy
ratio=energy2/energy1# units of counts per Joule
print(ratio)

# Scale unsaturated image to an array of values between 0 and 1
adjave1=average1/max1

# Scale image 2 by the same amount as image 1 to put the adjusted image 1 and 2 on the same count scale
adjave2=average2/max1

# Adjust the adjusted image 2 to account for the difference in energy
scaledave2=adjave2/ratio

# Max pixel count for pixel and energy adjusted saturated image
print(np.max(scaledave2))

# Specify which arrays to use
raw2=astropy.io.fits.HDUList([astropy.io.fits.PrimaryHDU(scaledave2)])
raw1=astropy.io.fits.HDUList([astropy.io.fits.PrimaryHDU(adjave1)])

# Find the centroid of the images
center2=poppy.measure_centroid(HDUlist_or_filename=raw2,boxsize=10)[::-1]
center1=poppy.measure_centroid(HDUlist_or_filename=raw1,boxsize=10)[::-1]

print('center1', center1)
print('center2', center2)

np.indices(scaledave2.shape)

```

```python
# The radial profile function taken from poppy documentation with edits for my purposes
# I had to edit the function to maunually state the pixel scale because the poppy function was having a hard time
# finding a value and I couldn't figure out another way to define the value

def radial_profile(HDUlist_or_filename=None, ext=0, EE=False, center=None, stddev=False, binsize=None, maxradius=None):
    if isinstance(HDUlist_or_filename, str):
        HDUlist = fits.open(HDUlist_or_filename)
    elif isinstance(HDUlist_or_filename, fits.HDUList):
        HDUlist = HDUlist_or_filename
    else: raise ValueError("input must be a filename or HDUlist")

    image = HDUlist[ext].data
    pixelscale = 42.173 # HDUlist[ext].header['PIXELSCL']

    if maxradius is not None:
        raise NotImplemented("add max radius")

    if binsize is None:
        binsize=pixelscale

    y,x = np.indices(image.shape)
    if center is None:
        # get exact center of image
        #center = (image.shape[1]/2, image.shape[0]/2)
        center = tuple((a-1)/2.0 for a in image.shape[::-1])

    r = np.sqrt((x-center[0])**2 + (y-center[1])**2) *pixelscale / binsize # radius in bin size steps
    ind = np.argsort(r.flat)


    sr = r.flat[ind]
```

```
    sim = image.flat[ind]
    ri = sr.astype(int)
    deltar = ri[1:]-ri[:-1] # assume all radii represented (more work if not)
    rind = np.where(deltar)[0]
    nr = rind[1:] - rind[:-1] # number in radius bin
    csim = np.cumsum(sim, dtype=float) # cumulative sum to figure out sums for each bin
    tbin = csim[rind[1:]] - csim[rind[:-1]] # sum for image values in radius bins
    radialprofile=tbin/nr

    # Pre-pend the initial element that the above code misses.
    radialprofile2 = np.empty(len(radialprofile)+1)
    if rind[0] != 0:
        radialprofile2[0] =  csim[rind[0]] / (rind[0]+1)  # if there are multiple elements in the center bin, average them
    else:
        radialprofile2[0] = csim[0]                       # otherwise if there's just one then just take it.
    radialprofile2[1:] = radialprofile
    rr = np.arange(len(radialprofile2))#*binsize + binsize*0.5 (I got rid of the binsize command because I wasnt sure what it was doing. I think without it, it will set the x axis to pixel, which is easier to follow)  # these should be centered in the bins, so add a half.

    if stddev:
        stddevs = np.zeros_like(radialprofile2)
        r_pix = r * binsize
        for i, radius in enumerate(rr):
            if i == 0: wg = np.where(r < radius+ binsize/2)
            else:
                wg = np.where( (r_pix >= (radius-binsize/2)) &  (r_pix < (radius+binsize/2)))
                #wg = np.where( (r >= rr[i-1]) &  (r <rr[i] )))
            stddevs[i] = image[wg].std()
```

```
        return (rr, stddevs)

    if not EE:
        return (rr, radialprofile2)
    else:
        #weighted_profile = radialprofile2*2*np.pi*(rr/rr[1])
        #EE = np.cumsum(weighted_profile)
        EE = csim[rind]
        return (rr, radialprofile2, EE)

# Find the radial profiles for both images
radial_mean2 = radial_profile(raw2,center=center2)
radial_mean1 = radial_profile(raw1,center=center1)

print(np.max(radial_mean2[1]))

# Plot the radial profiles on top of each other
# NOTE: the scaled overexposed image has much smaller values so it is hard to see the small scale structure on the large scale axes

plt.plot(radial_mean2[0][:], radial_mean2[1][:],radial_mean1[0][:], radial_mean1[1][:],'r')
#plt.xlim([0,600])
#plt.xlim([0,400])
plt.ylim([0,.2])
plt.show()

lam = 635 # nm

xset1= radial_mean1[0]
xset2= radial_mean2[0]
x1 = xset1[1:]
x2 = xset2[1:]

lamratio1 = lam /(pixelsize*2)
lamratio2 = lam /(pixelsize*2)
```

```
lam1 = lamratio1/x1
lam2 = lamratio2/x2

yset1=radial_mean1[1]
yset2=radial_mean2[1]

y1= yset1[1:]
y2= yset2[1:]
```

D.2 PSF Gaussian Fit

The code below was written in a Python notebook in Jupyter

```
import numpy as np
from PIL import Image
import astropy
from astropy.io import fits
import poppy
import matplotlib.pyplot as plt
import os
%matplotlib inline
from os.path import expanduser
home = expanduser("~")

# Access the flight computer through ssh and command to take an image
    with the PixeLink camera
%%bash
ssh pi@IPADDRESS "source ~/.pixelink_env_vars; ./PixeLink-interface/
    camera_commands.sh"

# Access the image that was just taken by the camera
%%bash
scp pi@IPADDRESS:/home/pi/snapshot.jpg .

# Open the image in the Python notebook
```

```python
pic = Image.open("image.jpg")

# Get the data from the image and put it in an array
new = np.array(list(pic.getdata()), dtype=np.int).reshape((1944, 2592)) # This reshape value may change depending on your image size
picture = new[790:820,940:970] #The size will depend on the PSF location in the image

# Import the gauss fitting function
import gaussfitter
params =gaussfitter.gaussfit(picture,err=np.sqrt(picture),return_error=True)
height, amplitude, x, y, width_x, width_y, rota= params[0]
print(params)
shape=picture.shape
fit = gaussfitter.twodgaussian(params[0], shape=shape)

print(params)
shape=picture.shape
fit = gaussfitter.twodgaussian(params[0], shape=shape)

# Find the width of the PSF based on the Gaussian fit
width = np.sqrt(((width_x)**2) + ((width_y)**2))
print("width = ",width)
```

D.3 Wavefront Sensor Characterization

D.3.1 ParseWFS.py

```python
import numpy as np
import os

def load_wavefront(file, header=True):
    """
    Parameters
```

```
    ----------
    File: string
        The file name path to the Thorlabs Wave front CSV
    header : bool
        If True a 100 line header is assumed and returned as a dictionary.

    Returns
    -------
    wavefront measurement array and optional header dictionary
    """

    f = open(file,'rb')
    header_dict={}
    if header:
        for i,line in enumerate(f):
            if i <99:
                line=line.decode("Windows-1252") #guessed this encoding
                splitline=line.split(",") #break apart the header lines
                if  len(splitline)==2:
                    key, value= line.split(",")
                    if key[:2]== '  ':
                        key="RMS WAVEFRONT VARIATIONS "+key
                    value = value.replace("\n","").replace("\r","").strip()
                elif len(splitline)==4:
                    key="Zernike "+splitline[0]
                    value=splitline[1:]
                else:
                    continue
                header_dict.update({key.strip():value})
            else:
                break
    # Convert wavefront to a numpy array
    wavefront=np.genfromtxt(f.readlines(),delimiter=",")[1:,1:-1]
    if header:
```

```
        return header_dict,wavefront
    else:
        return wavefront

def load_centroids(file, header=True):
    """
    Parameters
    ----------
    File: string
        The file name path to the Thorlabs centroid CSV
    header : bool
        If True a 41 line header is assumed and returned as a dictionary
    .

    Returns
    --------
     centroid measurement array and optional header dictionary
    """

    f = open(file,'rb')
    header_dict={}
    if header:
        for i,line in enumerate(f):
            if i <40:
                line=line.decode("Windows-1252") #guessed this encoding
                splitline=line.split(",") #break apart the header lines
                if  len(splitline)==2:
                    key, value= line.split(",")
                    value = value.replace("\n","").replace("\r","")
                else:
                    continue
                header_dict.update({key:value})
            else:
                break
    #convert wavefront to a numpy array
    wavefront=np.genfromtxt(f,delimiter=",")[1:,1:-1]
```

```
    if header:
        return header_dict,wavefront
    else:
        return wavefront

def load_intensity(file, header=True):
    f = open(file,'rb')
    header_dict={}
    if header:
        for i,line in enumerate(f):
            if i <40:
                line=line.decode("Windows-1252") #guessed this encoding
                splitline=line.split(",") #break apart the header lines
                if  len(splitline)==2:
                    key, value= line.split(",")
                    value = value.replace("\n","").replace("\r","")
                else:
                    continue
                header_dict.update({key:value})
            else:
                break
    #convert wavefront to a numpy array
    spotfield=np.genfromtxt(f,delimiter=",")[1:,1:-1]
    if header:
        return header_dict,spotfield
    else:
        return spotfield
```

D.3.2 Mean and Standard Deviation Plots of WFS Data.ipynb

```
import numpy as np
import matplotlib.pyplot as plt
import os
%matplotlib inline

```

```
import parseWFS

### Baseline WFS Measurements
# Set up 3D array to combine the WFS measurements (x,y,# of files)
cube = np.zeros((29,29,10), dtype=np.float)

# Specify the folder from which you want to retrieve the data files
directory = "filepath"

# For each file in the specified folder load the wavefront data into the
    cube
for i, filename in enumerate(os.listdir(directory)):
        if filename.endswith(".csv"):
            header,wf=parseWFS.load_wavefront(directory+filename)
            cube[:,:,i] += wf[:,:]
            continue
        else:
            continue

# Calculate the standard deviation and the mean along the 2nd axis (for
   each point in all of the data files)
stdev = cube.std(axis=2)
meanWF = cube.mean(axis=2)
print(np.nanmax(stdev))

### Using Spot Centroid Data
# Set up 3D array to combine the WFS measurements (x,y,# of files)
cubex = np.zeros((28,28,10), dtype=np.float)
cubey = np.zeros((28,28,10), dtype=np.float)

# Specify the folder from which you want to retrieve the data files
directory = "path/"

# For each file in the specified folder load the wavefront data into the
    cube
for i, filename in enumerate(os.listdir(directory)):
```

```python
        if filename.endswith(".csv"):
            header,wf=parseWFS.load_centroids(directory+filename)
            #print(wf.shape)
            cubex[:,:,i] += wf[:,:-1:2]
            cubey[:,:,i] += wf[:,1::2]
            continue
        else:
            continue
print(cubex.shape)
print(cubey.shape)

# Calculate the standard deviation and the mean along the 2nd axis (for each point in all of the data files)
stdevx = cubex.std(axis=2)
meanWFx = cubex.mean(axis=2)

stdevy = cubey.std(axis=2)
meanWFy = cubey.mean(axis=2)
```

D.4 Centroiding Algorithm

```python
import numpy as np
import itertools
from PIL import Image
import scipy.ndimage
import scipy.optimize
import scipy.stats
import glob
import matplotlib.pyplot as plt
import astropy
from astropy.convolution import convolve, convolve_fft
import csv
import parseWFS
%matplotlib inline

```

```python
# Code developed from Julian Brown's Tetra program that can be found here: https://github.com/brownj4/Tetra/

import os
from os.path import expanduser
home = expanduser("~")
### Using .tiff files from a CCD
## Raw Image

# minimum number of pixels in a group of bright pixels
# needed to classify the group as a star
min_pixels_in_group = 5

# centroiding window radius around a star's center pixel
# does not count the center pixel
window_radius = 3

# ~70% of max counts in an image
counts=220

# run the tetra star tracking algorithm on the given image
def centroid(image_file_name):

  # For .tif files with multiple frames
  #open up the image (this is for a tif file with 100 images taken--will need to change if using a different file type or number of images)
  im=Image.open(image_file_name)
  cube= np.zeros((1024,1280,100), dtype=np.float)
  for i in range(100):
    try:
        cube[:,:,i] += np.array(list(im.getdata()), dtype=np.float).reshape((1024, 1280))
    except EOFError:
        break

```

```
# take the mean value of the pixels along axis 2 to get an average image of all slices of the file
image=cube.mean(axis=2)
# extract height (y) and width (x) of image
height, width = image.shape

# find all groups of pixels brighter than 150 counts
bright_pixels = zip(*np.where(image > counts))
# group adjacent bright pixels together
# create a dictionary mapping pixels to their group
pixel_to_group = {}
# iterate over the pixels from upper left to lower right
for pixel in bright_pixels:
  # check whether the pixels above or to the left are part of
  # an existing group, which the current pixel will be added to
  left_pixel = (pixel[0]  , pixel[1]-1)
  up_pixel   = (pixel[0]-1, pixel[1]  )
  in_left_group = left_pixel in pixel_to_group
  in_up_group = up_pixel in pixel_to_group
  # if both are part of existing, disjoint groups, add the current pixel and combine the groups
  if in_left_group and in_up_group and id(pixel_to_group[left_pixel]) != id(pixel_to_group[up_pixel]):
    # add the current pixel to the upper pixel's group
    pixel_to_group[up_pixel].append(pixel)
    # append the upper pixel group onto the left pixel group
    pixel_to_group[left_pixel].extend(pixel_to_group[up_pixel])
    # replace all of the upper pixel group's dictionary entries
    # with references to the left pixel group
    for up_group_pixel in pixel_to_group[up_pixel]:
      pixel_to_group[up_group_pixel] = pixel_to_group[left_pixel]
  # if exactly one of the left pixel or upper pixels is part of an existing group,
  # add the current pixel to that group and add the current pixel to the dictionary
  elif in_left_group:
```

```
      pixel_to_group[left_pixel].append(pixel)
      pixel_to_group[pixel] = pixel_to_group[left_pixel]
    elif in_up_group:
      pixel_to_group[up_pixel].append(pixel)
      pixel_to_group[pixel] = pixel_to_group[up_pixel]
    # if neither of the left pixel or upper pixel are in an existing group,
    # add the current pixel to its own group and store it in the dictionary
    else:
      pixel_to_group[pixel] = [pixel]
  # iterate over the dictionary to extract all of the unique groups
  seen = set()
  groups = [seen.add(id(group)) or group for group in pixel_to_group.values() if id(group) not in seen]
  #print(groups)
  # find the brightest pixel for each group containing at least
  # the minimum number of pixels required to be classified as a star
  star_center_pixels = [max(group, key=lambda pixel: image[pixel]) for group in groups if len(group) > min_pixels_in_group]
  #print('center pixels:', star_center_pixels)
  #print(max(groups, key=lambda pixel: image[pixel]))
  # find the centroid, or center of mass, of each star
  window_size = window_radius * 2 + 1
  # pixel values are weighted by their distances from the left (x) and top (y) of the window
  x_weights = np.fromfunction(lambda y,x:x+.5,(window_size, window_size))
  y_weights = np.fromfunction(lambda y,x:y+.5,(window_size, window_size))
  star_centroids = []
  for (y,x) in star_center_pixels:
    # throw out star if it's too close to the edge of the image
    if y < window_radius or y >= height - window_radius or \
       x < window_radius or x >= width  - window_radius:
      continue
```

```
        # extract the window around the star center from the image
        star_window = image[y-window_radius:y+window_radius+1, x-window_radius:x+window_radius+1]
        #print(star_window)
        # find the total mass, or brightness, of the window
        mass = np.sum(star_window)
        # calculate the center of mass of the window in the x and y dimensions separately
        x_center = np.sum(star_window * x_weights) / mass - window_radius
        y_center = np.sum(star_window * y_weights) / mass - window_radius
        # Make the centroid values an integer
        #x_center = np.int(x_cen)
        #y_center = np.int(y_cen)
        # correct the star center position using the calculated center of mass to create a centroid
        star_centroids.append((y + y_center, x + x_center))
    #sort star centroids from brightest to dimmest by comparing the total masses of their window pixels
    #star_centroids.sort(key=lambda yx:-np.sum(image[yx[0]-window_radius:yx[0]+window_radius+1, yx[1]-window_radius:yx[1]+window_radius+1]))
    print('centroid coordinates [(y, x)] = ', star_centroids)
    #print(image.shape)
    plt.imshow(image[700:800, 400:500])
    plt.plot(55.0436, 31.2512,'or')
    plt.colorbar()

# Run the tetra program to find the centroid(s) for the specified file
centroid(home + "filepath/filename.tiff")
```

Appendix E

SmallSat Conference Paper

This is the paper I submitted in June, 2018 to the Frank J. Redd Student Competition at the Small Satellite Conference hosted by AIAA and Utah State University. I presented the paper at the conference in Logan, Utah on August 8, 2018. Sections of this conference paper have been adapted and included in this thesis.

Payload Configuration, Integration and Testing of the Deformable Mirror Demonstration Mission (DeMi) CubeSat

Jennifer Gubner
Wellesley College, Massachusetts Institute of Technology
21 Wellesley College Road, Wellesley, MA
jgubner@wellesley.edu

Faculty Advisor: Kerri Cahoy
Massachusetts Institute of Technology

ABSTRACT

Adaptive optics is an imaging technique that has been used on many ground based telescopes to improve image resolution and reduce the effects of atmospheric turbulence. While adaptive optics has known uses on the ground, applying this technique to space telescopes has major advantages for exoplanet imaging, inter-satellite laser communication, high energy systems, and other military applications.

The Deformable Mirror Demonstration Mission (DeMi) is a 6U CubeSat, that will demonstrate the use of adaptive optics, specifically a microelectromechanical system (MEMS) deformable mirror, in space. Not only will the DeMi mission characterize the deformable mirror on-orbit, the mission will also demonstrate deformable mirror control using closed loop image plane sensing and wavefront sensing on internal and external light sources. DeMi uses COTS components like Thorlabs mirrors, Pixelink complementary metal-oxide-semiconductor cameras, and a Boston Micromachines Corporation "multi" deformable mirror.

DeMi is currently in the optical integration and testing stage. The payload design and assembly is being tested by assembling 3D printed payload components. Optical alignment and configuration is being tested by mounting the optical components to the 3D printed payload assembly. Current and future testing will inform payload design and payload assembly plan changes. DeMi is expected to launch winter of 2019.

INTRODUCTION

Adaptive optics (AO) have been commonly used on ground based telescopes, such as the Keck I and II telescopes[1], to correct for the negative impacts that atmospheric turbulence has on astronomical imaging. While AO is commonly known to have uses on ground based telescopes, it also has applications on space telescopes. AO can be a critical difference in reaching the necessary contrast, of 10^{-10}, to image Earth-like exoplanets.[2] It allows for corrections of wavefront error caused by optical imperfections and thermal distortions. These correction capabilities allow launches of cheaper optics, improve imaging resolution, and have applications for optical amplification in intersatellite communication and various military projects.

Mission Overview

The Deformable Mirror Demonstration Mission (DeMi) is a 6U CubeSat mission created to demonstrate the use of AO, specifically a Microelectromechanical System (MEMS) deformable mirror (DM), in space. MEMS DMs have not previously been demonstrated on long duration space missions. One of DeMi's main objectives is to demonstrate the AO capabilities in space using a closed loop wavefront control system.[3]

DeMi's other mission objectives are to characterize the MEMS DM on-orbit and to image a star, or other astronomical object, using the DM to improve the point spread function.

Optical Design Overview

The payload, which will be flown on a Blue Canyon Technologies 6U XB6 bus, uses a series of off-axis parabolic mirrors (OAPs), field mirrors, complementary metal-oxide semiconductor (CMOS) cameras, a Shack-Hartmann Wavefront Sensor (SHWFS) and a Boston Micromachines Corporation 140 actuator MEMS DM, to demonstrate the wavefront correction capabilities.

Software Overview

The DeMi mission will demonstrate the correctional capabilities of AO in space by employing the use of closed loop wavefront correcting software. The software will work by taking the data from the SHWFS or the CMOS image sensor and using it to correct the shape of the deformable mirror. The new mirror shape will then be fed into the system and the loop will process again with the new image or wavefront. This wavefront correction software will run on payload computers that are independent from the XB6 bus.

Paper Organization

In this paper, I begin by with the Optical Layout section to discuss, in more detail, the optical configuration of DeMi. I have included subsections to provide more detailed information on the important optical components. These subsections include a Deformable Mirror section, a Complementary Metal-oxide Semiconductor (CMOS) Camera section, and a Shack-Hartmann Wavefront Sensor section. Following the Optical Layout section, I discuss the concept of operations for the mission. The section, Concept of Operations, includes general information about the mission as well as details about the internal and external observations of the mission. In the next section I discuss payload integration and testing, followed by plans for future work and testing.

OPTICAL LAYOUT

The payload for DeMi works by directing the light source, either internal or external, through a beamsplitter and using one beam for the closed loop wavefront correction and one beam for the imaging. Figure 1 shows the configuration of the optics in the payload.

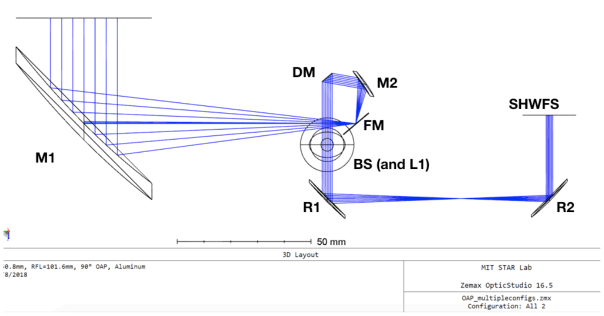


Figure 1: Optical configuration of DeMi

The first mirror, M1, is a 2” 90^o Thorlabs OAP with a 4” focal length that takes the light from the external observation and directs it in to the payload. The light gets focused by the OAP and reflects off of the field mirror, labeled FM in the diagram. The field mirror has an embedded single mode fiber, coupled with a 635 nm laser diode, that will be used for the internal observations of the mission. The fiber is off axis by about 0.2 degrees, and provides a near diffraction-limited spot. The next mirror in the configuration is M2, which is a smaller 90^o OAP cut down to 8.5 mm diameter with a focal length of 15 mm. This OAP collimates the light and sends it to the DM. The DM will correct the incoming wavefront and send the corrected wavefront through the rest of the payload.

The DM reflects the wavefront through a beam splitter, labeled BS, which sends one wavefront down through the base of the payload and one through to the back of the payload. The downward reflected wavefront is captured by a CMOS camera, labeled L1.

The back-moving wavefront is directed through a set of ½” diameter, 90^o, 2” focal length Thorlabs OAPs, R1 and R2, to resize and redirect the beam. This OAP relay sends the wavefront to the SHWFS to inform the wavefront correction loop.

Deformable Mirror

The role of a DM in an AO system is to correct the wavefront for any aberrations or imperfections detected by the wavefront sensor. A DM accomplishes this goal by deforming its shape into a conjugate of the detected wavefront.

There are two main kinds of DMs, segmented and continuous. Segmented mirrors have individual flat surface mirrors attached to each actuator. A continuous DM uses a continuous face-sheet mirror over the actuators.

The DM on DeMi is a 140 actuator BMC multi with 5.5 µm stroke and a 4.95 mm aperture. The MEMS DMs made by BMC use electrodes and variable supplied voltages to move the actuators of the mirror.[4] The 140 actuator multi DM was chosen for its high actuator count, large stroke, and good correctional capabilities, all within a reasonable cost.

Complementary Metal-oxide Semiconductor (CMOS) Camera

There are two CMOS cameras used on the DeMi payload, and both play major roles. One is used as a camera to capture the image from the observation and to direct the image plane wavefront sensing. The other is used in the SHWFS. The CMOS cameras used on DeMi are PL-D775MU-BL COTS cameras from Pixelink, shown in Figure 2. These Pixelink cameras have 5 megapixel resolution and can perform at 15 fps at full resolution.[5] The cameras come with a flex cable that directs the heat generated by the readout electronics away from the optics.

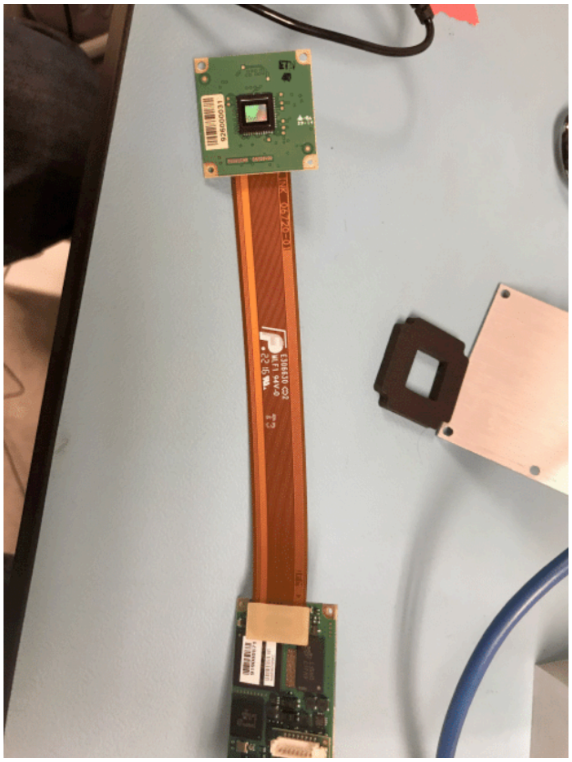


Figure 2: Pixelink CMOS camera

Shack-Hartmann Wavefront Sensor

Shack-Hartmann wavefront sensors use a lenslet array to divide up the incoming beam of light and focus the divided beams on to a CMOS camera. The direction and shape of the incoming beam can be determined based on the displacement of the centroids from each beam in the divided array. An example of the SHWFS results for a distorted wavefront is shown in Figure 3.

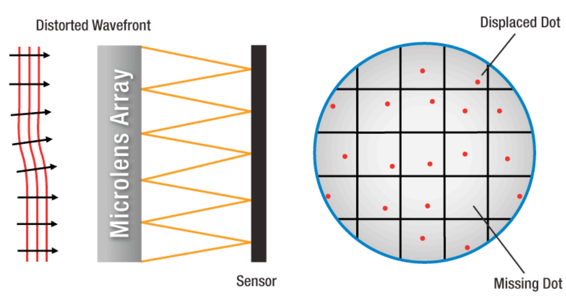


Figure 3: SHWFS simulation[6]

The SHWFS used for DeMi has a lenslet array, Thorlabs MLA 150-5C(-M), that has approximately 4 lenslets per DM actuator spacing, and uses one of the Pixelink CMOS cameras mentioned above to capture the centroids from the lenslets.

The SHWFS will use MIT written software to determine centroid positioning and send appropriate commands to the DM.

For preliminary testing of the optics configuration, we used a Thorlabs WFS150-5C.

CONCEPT OF OPERATIONS

DeMi will be deployed in to a ~500 km, mid-latitude inclination, circular low Earth orbit and complete both internal and external observations, as described in more detail below, during its approximately one year lifetime. The internal observations will be used to characterize the DM and to test the control loops. The external observations will demonstrate the use of the AO system astronomical targets. I have worked significantly on defining and outlining the concept of operations, and have broken the modes down to include more details about mission procedures.

Before each operation, the spacecraft will perform several checks to ensure that the spacecraft can safely and correctly perform its desired functions. These checks will be different for internal and external operations, as they have different performance requirements. After the checks, the spacecraft will power on the required components for the specific mode of operation. The spacecraft will then test voltage and current to the components and finish by taking baseline measurements and image frames.

Internal Modes of Operation

For the internal operation, several system checks need to be done to ensure successful operation of DeMi. The spacecraft needs to ensure that the internal temperature, attitude control, data storage capacity and power supply to the payload fit the requirements. If the system checks pass, then the payload can power on the necessary components. For internal observations, the following components need to be powered on:

- Laser
- DM
- CMOS Camera
- SHWFS

During the internal operations, DeMi will perform three different demonstrations. Each of these demonstrations will use the internal laser source to illuminate the DM and take measurements. These three demonstrations are:

- Test all DM actuators to full displacement

- Run the wavefront correction loop on the internal laser
- Run the image plane wavefront correction loop on the internal laser

The first operational mode will characterize the DM by testing each individual actuator to full displacement. For each actuator, we will record a wavefront measurement and an image plane measurement.

The second operational mode tests the standard wavefront correction loop. This operational mode will use the SHWFS to measure the wavefront and will run a closed loop correction between the SHWFS and the DM.

The third operational mode will test the image plane wavefront sensing. This mode will use the CMOS camera in a closed loop with the DM to correct the wavefront from the internal source. The DM corrections will be based on a built up library of image plane DM actuator influence functions.

Figure 4 outlines the steps required to complete the internal observations.

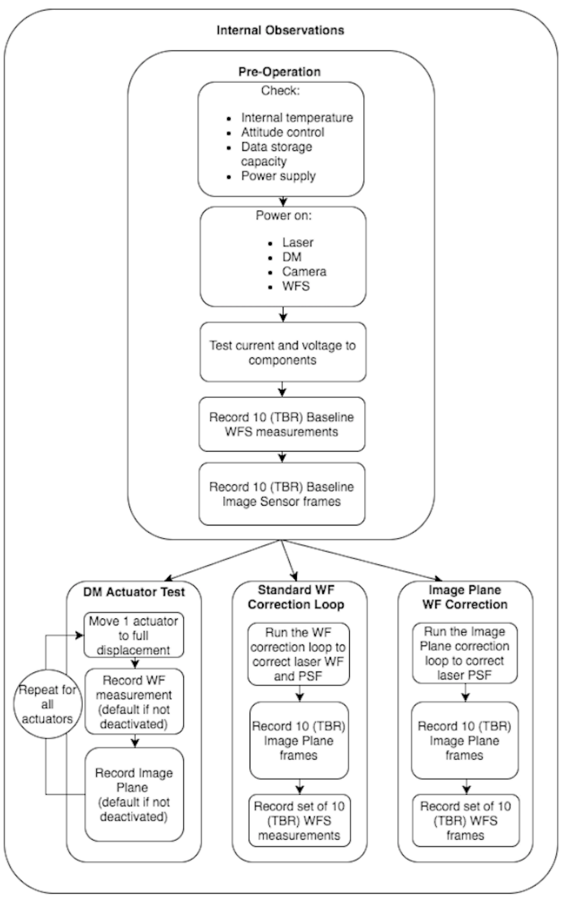


Figure 4: Internal modes of operation

External Mode of Operation

For external observations, the spacecraft needs to perform more system checks. In addition to the system checks needed for internal observations, the spacecraft also needs to check spacecraft pointing and stability as well as spacecraft position relative to eclipse. Once system checks are complete, the spacecraft will need to power on the DM, CMOS camera and the SHWFS.

The external mode of operation will perform astronomical observations while testing the wavefront correction loop. This mode will use the external aperture to look at the light source from stars and use the closed loop wavefront correction system to demonstrate the correctional capabilities.

Figure 5 outlines the steps required to complete external observation successfully.

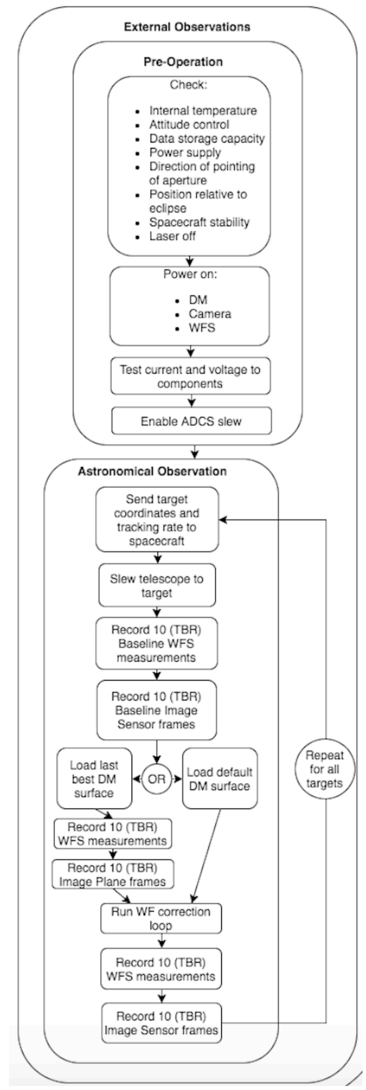


Figure 5: External operational mode

PAYLOAD INTEGRATION AND TESTING

To test optomechanical design and optical configuration of the payload, I have put together 3D printed models of the payload. The initial test was a full print of the payload with all components already included in the print. This test was to check for sizing and spacing of the payload components.

For the next test, we individually 3D printed all of the components for payload assembly. Using fasteners and fine adjusters I practiced assembling the payload and checked for any design flaws. Figure 6 shows the assembled 3D printed payload without mounted optics. The model is approximately 30 cm in length and 10 cm in width.

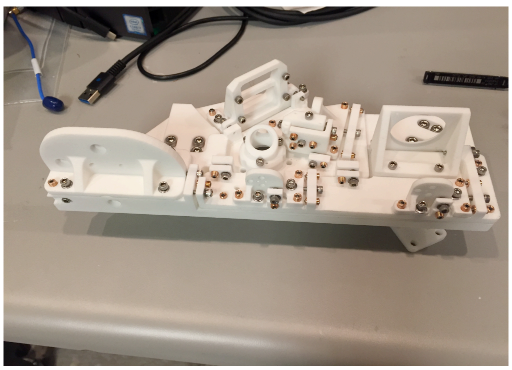


Figure 6: 3D printed payload assembly

After checking the payload assembly, I mounted the optics to the model to check clearances and spacing. Figure 7 shows the payload assembly with some of the mounted optics.

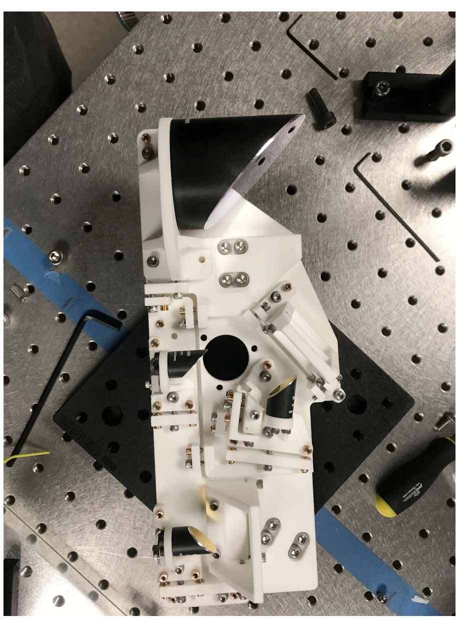


Figure 7: 3D printed payload assembly with optics

FUTURE WORK

After making modifications to improve and refine the design based on lessons learned during the practice with the 3D printed model and optics, we sent out the corrected design to be 3D printed. These new 3D printed components have just arrived to the lab and I will begin testing of the assembly procedures and optics spacing. After the new model is assembled, I will then test complete optical alignment of the payload and revise the assembly plan to ensure smooth flight payload assembly. The complete assembly plan and optical alignment will then be tested on an aluminum model of the payload before final payload integration.

Before the payload delivery, we will complete additional testing and characterization of the individual optics components. These tests will include characterization of the DM using an interferometer, calibration and thermal testing of the CMOS cameras, and SHWFS performance testing.

We also need to finalize wavefront correction software and make refinements to the mission operation plans before launch in the winter of 2019.

ACKNOWLEDGEMENTS

I would like to thank the Defense Advanced Research Projects Agency (DARPA) for funding DeMi and acknowledge Annie Marinan for the original DeMi mission idea proposed in her PhD thesis. I would like to thank Bobby Holden, Gregory Allan, and Warren Grunwald for their large contributions to the development of DeMi and would like to acknowledge the rest of MIT STAR Lab for their contributions to the project. I would also like to extend my sincere appreciation to Professor Kerri Cahoy and Dr. Ewan Douglas for their continued support throughout this project.

REFERENCES

1. Max, C., “Introduction to Adaptive Optics and Its History.” *UC Observatories,* NSF Center for Adaptive Optics, Jan. 2001, www.ucolick.org/~max/History_AO_Max.htm.
2. Traub, W. A., and Oppenheimer, B. R. 2010. “Direct Imaging of Exoplanets.” In Exoplanets, 111–156. Tucson, AZ, USA: University of Arizona Press.
3. Cahoy, K., Marinan, A., Novak, B., Kerr, C., Webber, M., "Wavefront control in space with MEMS deformable mirrors", Proc. SPIE 8617, MEMS Adaptive Optics VII, 861708 (5 March 2013);

doi:10.1117/12.2005685; https://doi.org/10.1117/12.2005685.

4. "AO Tutorial." Thorlabs, www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=3208. Accesses April 16, 2018

5. "PL-D775." Pixelink, http://files.pixelink.com/datasheets/PL-D700/Datasheet_PL-D775.pdf. Accesses April 16, 2018.

6. "SH Tutorial." Thorlabs, www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2946.

Bibliography

[1] Gregory W. Allan. Simulation and testing of wavefront reconstruction algorithms for the deformable mirror (demi) cubesat. Master's thesis, Massachusetts Institute of Technology, 9 2018.

[2] Vanessa P. Bailey. Key technologies for the wide field infrared survey telescope coronagraph instrument. Technical report, Jet Propulsion Laboratory, California Institute of Technology, 2018.

[3] Thomas Bifano. Mems deformable mirrors. *Nature Photonics*, page 21, 2011.

[4] Jan Burke. Null test of an off-axis parabolic mirror. ii. configuration with planar reference wave and spherical return surface. *Optics Express*, 17:3242–3254, 2009.

[5] Boston Micromachines Corporation. Low actuator count deformable mirrors. http://bostonmicromachines.com/low-actuator-count.html. Accessed on 2018-9-19.

[6] R. Davies and M. Kasper. Adaptive optics for astronomy. *Annual Review of Astronomy and Astrophysics*, 50:305–351, 2013.

[7] John Asher Johnson. *How Do You Find Exoplanets?*, chapter Directly Imaging Exoplanets. Princeton University Press, 2016.

[8] Anne Marinan, Kerri Cahoy, Matthew Webber, Ruslan Belikov, and Eduardo Bendek. Payload characterization for cubesat demonstration of mems deformable mirrors. *SPIE*, 2014.

[9] Claire Max. Introduction to adaptive optics and its history. *American Astronomical Society 197th Meeting*.

[10] Kevin Newman. An introduction to off-axis parabolic mirrors. https://wp.optics.arizona.edu/optomech/wp-content/uploads/sites/53/2016/10/521_Tutorial_Newman_Kevin.pdf, 2013.

[11] Ben R. Oppenheimer and Sasha Hinkley. High-contrast observations in optical and infrared astronomy. *Annual Review of Astronomy and Astrophysics*, 47:253–289, 2009.

[12] Photometrics. Tech notes and whitepapers. https://www.photometrics.com/resources/imaging-topics-tools. Accessed November, 2018.

[13] Pixelink. *PL-D775*. Accessed April, 2018.

[14] Benjamin M. Potsaid and John Ting-Yung Wen. Adaptive scanning optical microscope: large field of view and high-resolution imaging using a mems deformable mirror. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 7, 2008.

[15] Rene Racine, Gordon A. H. Walker, Daniel Nadeau, Rene Doyon, and Christian Marios. Speckle noise and the detection of faint companions. *The Astronomical Society of the Pacific*, 111:587–594, 1999.

[16] ThorLabs. *WFS Series Operation Manual: Optical Wavefront Sensors (Shack-Hartmann)*, 4.6 edition, 10 2015.

[17] W. A. Traub and B. R. Oppenheimer. *Exoplanets*, chapter Direct Imaging of Exoplanets, pages 111–156. University of Arizona Press, 2010.

[18] P. Wizinowich, D. S. Acton, C. Shelton, P. Stomski, J. Gathright, K. Ho, W. Lupton, K. Tsubota, O. Lai, C. Max, J. Brase, J. An, K. Avicola, S. Oliver, D. Gavel, B. Macintosh, A. Ghez, and J. Larkin. First light adaptive optics images from the keck ii telescope: A new era of high angular resolution imagery. *Publications of the Astronomical Society of the Pacific*, 112:315, 2000.