Investigating Proposed Movement Methods and Collision Avoidance Algorithms for the Fiber Positioners for the Dark Energy Spectroscopic Instrument

Efrain Segarra
March 2016

Advisor:
Dr. Gregory Tarlé
Professor of Physics

Department of Physics
University of Michigan

Abstract

The Dark Energy Spectroscopic Instrument (DESI) will probe dark energy through baryon-acoustic oscillations and the growth of structure through redshift-space distortions. Utilizing 5,000 fiber optic positioning robots, closely packed on the telescope's focal plane, DESI will collect spectra over five years. It will observe 14,000 deg^2 through a wide-area galaxy and quasar redshift survey, and it will target approximately 50 million galaxies [2]. To maximize telescope observation time, we must construct an "efficient" movement-method to maneuver all of the fiber-optic positioning robots (positioners) to their respective target locations. The efficiency of any method is characterized by how quickly all 5,000 positioners can reach target locations. Also, this scheme must be subject to the mechanical constraints of the positioners and the DESI science requirements. To achieve the best repositioning time, the movement method should have a low number of collisions to resolve and the complexity of these collisions should also be low. This paper will investigate and present various movement schemes and their respective collision avoidance algorithms to aid future large-scale spectroscopic survey experiments that utilize fiber-optic positioning robots with eccentric parallel axis θ-ϕ kinematics. The Retract-Rotate-Extend (RRE) method, proposed by Joe Silber at LBNL, was implemented and outfitted with collision avoidance software by me, Efrain Segarra at the University of Michigan. My software ensures that 99% of positioners reach new target locations within 10 seconds after a previous exposure has finished. My collision avoidance software effectively masks collision avoidance steps within the three-step movement method itself, and exceeds DESI science requirements on the focal plane. In this paper, I will characterize the performance and offer DESI a fully developed collision avoidance algorithm and movement method for testing and deployment in the future telescope.

Contents

Acknowledgements

I would like to extend the greatest thanks to Professor Gregory Tarlé, my research advisor, for the humbling opportunity to contribute to an international collaboration project as an undergraduate. Professor Tarlé has been the most active mentor in my education - teaching, guiding, and permitting space to recognize my own mistakes. Furthermore, I would like to thank Dr. Michael Schubnell for his support and complete commitment to the DESI collaboration, and to my software research project. Also, I would like to thank Curtis Weaverdyck for leading the manufacturing of the DESI positioning robots, and for the future opportunity to test my algorithmic schemes with real positioners, not just in software simulations. Lastly, I would like to thank Anthony Kremin for advice on software design and his integration of my movement method and collision avoidance scheme into the DESI software mainframe.

Chapter 1

Science Motivation and Background

1.1 The Accelerating Expansion

More than half of a century after Hubble discovered an expanding universe in 1929 by observing galaxies, in 1998, the observation of Type Ia supernovae (SNe) surprised the scientific community by suggesting the expansion of the universe is accelerating. Even today, the cause for the accelerating, expanding universe is still widely debated.

If General Relativity is correct, the dynamics of the universe are described by Friedmann's equations. Without any modifications, the acceleration of the universe follows:

$$\frac{\ddot{a}}{a} = \frac{-4\pi G}{3}(\rho + \frac{3p}{c^2}) \tag{1.1}$$

where $a(t)$ is the scaling factor of the universe (today $a = 1$), ρ is the energy density, and p is the pressure. We assume an equation of state that relates the pressure and energy density to be of the form $p = w\rho c^2$. However, if we take ρ and p to be positive-definite, then the universe must always be contracting by Equation 1.1. Einstein originally sought a static solution for the universe, so he was motivated to add the infamous cosmological constant, Λ. Thus, Equation 1.1 becomes (taking $c = 1$):

$$\frac{\ddot{a}}{a} = \frac{-4\pi G}{3}(\rho + 3p) + \frac{\Lambda}{3} \tag{1.2}$$

Although his efforts for a static solution were foiled by Hubble's expansion discovery, the cosmological constant has since been revived to help explain the acceleration of the universe. For a solution that permits an accelerating universe, either Λ must be positive, or, for some component of the energy density, w must be less than $-1/3$. For the cosmological constant specifically, $w = -1$.

The critical energy density, $\rho_{crit} = \frac{3{H_0}^2}{8\pi G}$, is defined as the density to slow the expansion exactly to 0 after an infinite time. The density parameter is defined as $\Omega = \rho/\rho_{crit}$. If we split up ρ into the different energy components (relativistic matter, non-relativistic matter, curvature density, dark energy density), and use the previous definitions, the first Friedmann equation becomes:

$$\frac{H^2}{{H_0}^2} = \Omega_{rel}a^{-4} + \Omega_m a^{-3} + \Omega_k a^{-2} + \Omega_{DE}F(a) \tag{1.3}$$

where $H_0 \approx 70$ km/s/Mpc, the Hubble constant today. The energy density ρ has been decomposed into relativistic matter ($w = 1/3$), non-relativistic matter ($w = 0$), curvature density, and dark energy components. More generally, w may vary with the scale factor, so then $\rho(a) = \rho_0 F(a)$ with $F(a) = \exp\left(3\int_a^1 \frac{da'}{a'}(1 + w(a'))\right)$. For $w = -1$, $F(a) = 1$ and we write $\Omega_{DE} = \Omega_\Lambda$ since this is a constant energy density component, independent of the scale factor - hence the name, cosmological "constant". For Ω_{DE} more generally, we allow the parametrization of w to vary with the scale factor, taking $w(a) = w_0 + (1 - a)w_a$.

Inflation predicts that the curvature of the universe, k, is null, and thus the universe is flat. Cosmic Microwave Background (CMB) measurements confirms this prediction with 99.6% confidence. Since the universe is flat and $k = 0$, the Friedmann equations demand that $\Omega_{total} = 1$. Furthermore, the relativistic energy density component is negligible today due to cooling during the expansion of the Universe. Then, we are left with trying to characterize two parameters:

$$\Omega_{total} = \Omega_{DE} + \Omega_m = 1 \tag{1.4}$$

The Dark Energy Spectroscopy Instrument (DESI) seeks to tightly constrain the distribution between Ω_m and Ω_{DE} with respect to Equation 1.4. More over, recall that there are three possibilities for the accelerating universe:

1. Positive cosmological constant with $w = -1$
2. Dark energy equation of state with $-1 < w(a) < -1/3$
3. Failure of General Relativity (GR), so then a GR modification is required

Observations of SNe and the baryon acoustic oscillations (BAO) have, together, proposed a tight constraint on $w \approx -1 \pm .03$. DESI seeks to constrain the dark energy equation of state (and w) even further, and probe whether the acceleration is due to a modification to GR.

1.2 Baryon Acoustic Oscillation

In the early primordial universe, electrons and baryons formed a dense quantum plasma. In a given region, the gravitational attraction between baryons tends collapse this region and form over-densities. However photon pressure tends to erase these over-densities, combating gravitational attraction. As these two forces compete, acoustic oscillations of the baryons arise, as they fall in and out of these over-dense regions at the speed of sound in the quantum plasma, $c/\sqrt{3}$. However once the universe cools enough, baryons are decoupled from photons and the balancing of the two forces end, reducing the speed of sound in the region from $c/\sqrt{3}$ to zero. These regions of over-densities, which previously had baryons oscillating in and out, now begin to collapse to form neighboring galaxies.

Thus, there is a characteristic scale imprinted on the universe. During the quantum plasma, there are many regions undergoing these acoustic oscillations, however these regions are not infinitely large. There is a specific scale at which the gravitational force and pressure force can compete and keep baryons oscillating - this is called the acoustic scale. Once the photons decouple from the baryons, the pressure force becomes null and baryons are free to collapse

towards the center of this region, and neighboring galaxies are slowly created from two previously oscillating regions. This means that there is a scale, even today, where we are more likely to see two galaxies at this certain distant apart. Thus, we have a standard ruler for the universe, and by correlating the distance between two galaxies at any redshift, we can measure how the universe's expansion rate has evolved over time. Today that characteristic distance is about 500 million light years between neighboring galaxies.

Once cool enough, the acoustic baryon wave will stop oscillating, creating quantum density perturbations of about $\rho \approx 10^{-5}$, which can be viewed as an angle subtended:

$$s = (1+z)D_A(z)\theta = \int_0^z \frac{cdz'}{H(z')}\theta \tag{1.5}$$

If we survey a region of space along the line of sight in some range Δz, calculate the two-point correlation function, which relates the distance between galaxy pairs, there should be a peak in that function when:

$$\frac{c\Delta z}{H(z)} \approx s \tag{1.6}$$

A representation of this imprint on the two-point correlation function is shown in Figure 1.1., taken from the Baryon Oscillation Spectroscopic Survey (BOSS) pre-published results [1].

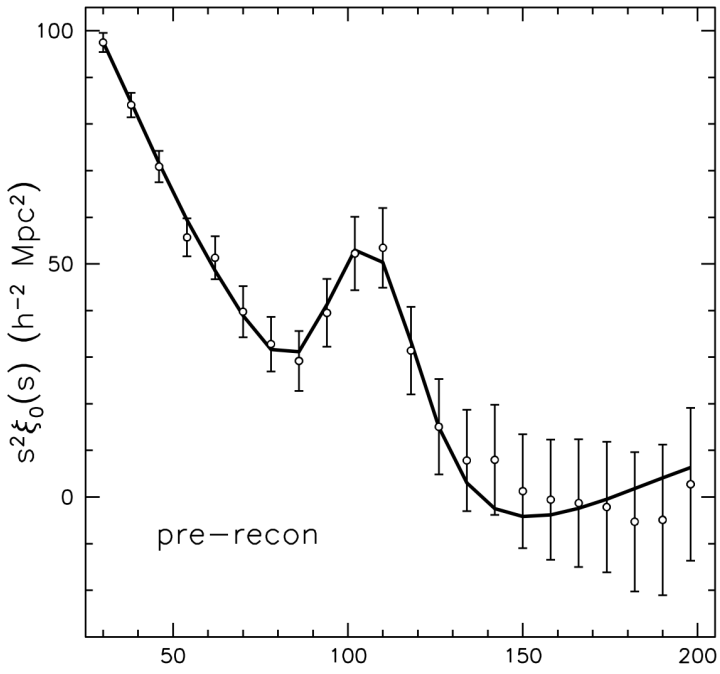


Figure 1.1: Two Point Correlation function of DR11 CMASS galaxies [1]. These data points are before nonlinear evolution reconstruction has been performed. The solid curve is a best-fit model.

Since at higher redshifts ($z > 2$), galaxy redshift observation becomes more difficult, dark energy is then probed through Ly-α forest BAO. The "Ly-α forest" is a collection of absorption lines in the spectra of quasars, based on the Ly-α electron transition of neutral

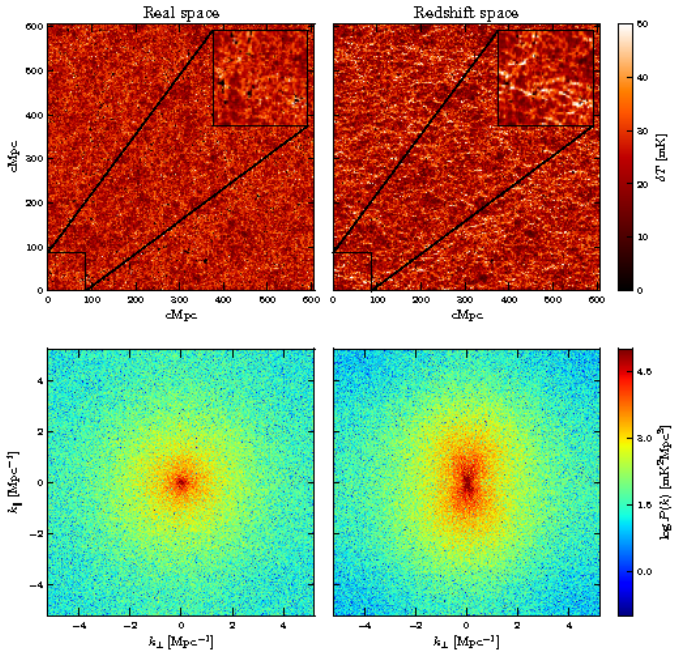


Figure 1.2: Illustration of anisotropy in redshift space distribution [3]. The top panels are slices at z=9.5, 607 Mpc wide, and the bottom panels are slices of 3D power spectra of the data cubes.

hydrogen. As the quasar continues to get further away, it's light is redshifted constantly. Neutral hydrogen in the intergalactic medium absorbs the light, but it absorbs the light at different wavelengths depending on the redshift of the medium. The density of absorption is a measure of the density of neutral hydrogen, and thus can outline the dark matter distribution on large enough scales to detect the BAO imprint.

1.3 Redshift-Space Distortion

The BAO approach will probe dark energy under our current understandings of GR (investigate possibilities 1 and 2 for the accelerated expansion). However, it is possible that our current theory of GR is incorrect, and a modification of GR is needed to explain the accelerating universe. DESI utilizes redshift space distortions to test the theory of Relativity.

As the universe expands at an accelerated rate, galaxies will move further and further away. Redshifts of these galaxies reflect the Hubble flow velocity, however galaxies may be moving locally due to gravitational attraction by large scale structures - an effect known as peculiar velocity. When observing the galaxy distribution in redshift-space, instead actual distance-space, anisotropies arise because distance depends on the Hubble flow and the peculiar velocity. This causes a "pancaking" of galaxies in the redshift space distribution of galaxies, as seen in Figure 1.2 [3]. This pancaking is known as one example of a redshift space distortion.

Density perturbations with respect to the uniform density describe the observed large-scale structure of the universe. These perturbations in redshift space manifest as:

$$\delta_s(\vec{k}) = \delta(\vec{k})(1 + \beta\mu^2) \tag{1.7}$$

with $\beta = f/b$, where b is the galaxy bias and f correlates to the linear growth function $D(z)$ by:

$$f = \frac{d \ln D(a)}{d \ln a} \tag{1.8}$$

The power spectrum in redshift space relates to the linear mass theory power spectrum P_m by:

$$P_s(\vec{k}) = (\beta + f\mu^2)^2 P_m(k) \tag{1.9}$$

By taking a Fourier transform of the two-point correlation function of galaxy distribution, we can measure the power spectrum in redshift-space (and its anisotropy due to the RSD), and investigate the form of f and the growth of gravitational structures. If GR is correct without modifications, the form of f is specified as $f \approx \Omega_m(z)^{6/11}$, and thus, DESI will be able to test the validity of GR through these redshift space distortions.

1.4 Beyond Accelerated Expansion

Apart from probing the nature of dark energy (or the failure of GR) through BAO and RSD, DESI measurements will probe other cosmological and physical phenomena. Through broad-band power spectrum data, DESI will test inflation, primordial non-Gaussianity distribution of density perturbations, the damping of structure from neutrinos, and possibly the mass hierarchy of neutrinos.

Chapter 2

The Dark Energy Spectroscopic Instrument

In order to sufficiently constrain dark energy parameters and precisely test General Relativity, DESI must meet the following science requirements [2]:

- 9,000 deg^2 BAO/RSD redshift survey, with possible extension to 14,000 deg^2
- Target 50 million galaxies comprised of luminous red galaxies (LRGs), emission-galaxy lines (ELGs), quasar (QSO) tracers, and Ly-α forest QSOs spanning, $0.5 < z < 3.5$
- The S/N for ELGs detection greater than 7 for a flux of 8 x 10^{-17} erg/s/cm^2
- Spectral range of 360-980 nm
- Spectroscopic resolution sufficient for redshift error (precision and accuracy) $< 0.0005(1+z)$ (> 1500 for $\lambda > 360$ nm and longer, > 3000 for $\lambda > 555$nm, and > 4000 from $\lambda > 656$ nm)
- Fiber density ~ 700 per square degree
- Field of view not less than 7.9 square degrees
- Survey duration of 4 years for 9,000 deg^2 survey, including 6 months commissioning and validation

In this paper, I will focus on describing the instrument components relevant to the fiber optic positioning robots: (1) the focal plane, (2) the fiber view camera, and (3) real-time control and data acquisition systems. In order to achieve the ambitious science goals, certain critical instrument parameters must be met. The focal plane system will contain 5,000 fiber-optic positioners, that can be repositioned for the next exposure in less than 120 seconds, with a repositioning accuracy of 5 μm [2]. This repositioning time will overlap spectrometer readouts and telescope slew. In addition to the 5,000 positioners the focal plate must contain a certain number of Guide, Focus, and Alignment (GFA) cameras to measure telescope pointing and the tip/tilt of focal plane, as well as Field Fiducials that act as point sources for fiber view camera repositioning correction. I will leave discussion of the fiber optic positioning robots for the next chapter.

2.1 Focal Plane System

The focal plane system has three fundamental functions:

1. Each fiber positioner moves a single fiber to a unique target position to make observation for science throughput
2. Fiducials provide feedback for the fiber view camera on accuracy and positioning of the fiber optics

3. GFA sensors measure telescope pointing, focus, and tip/tilt of surface, needed for calibration and correction

These functions are supported in a focal plate assembly, which is insulated and thermally controlled to remove heat from the system. Some of the focal plane system requirements are listed in Table 2.1, as flowing down from the conceptual design and targeting science of DESI.

The focal plate that supports the fiber positioners, GFAs, and field-illuminated fiducials is a radially symmetric dome, built in 10 symmetric sectors - such that each sector is 36 degrees of the total dome. These sectors, or "petals", are wedged-shaped, and contain 500 positioners, 6-7 field-illuminated fiducials (FIF), and 1 GFA each. The 500 positioners and FIFs are hexagonally closed-packed on the wedge to optimize spacing (further discussion on overlapping and reaching-envelopes will be discussed in the next chapter). The GFA sensors lie on the perimeter of the focal plate. See Figure 2.1 for sample of focal plane structure [2]. The 5,000 positioners will be fed into 10-50 spectrographs for observation capture.

Fiber positioners are mounted in a hole on the focal plate, with a bore and spotface for tip/tilt, lateral, and transverse positioning. Each positioner has a power budget of 1.2 W when operating during re-positioning moves, however controlling electronics have been designed such that positioners not moving are in a sleep mode with effectively zero power usage (0.023 W to be precise). To keep the goal thermal budget of 293.6 kJ as specified in the science requirements, the duty cycle of all 5,000 positioners while moving should be no longer than 10 seconds in 15 minute periods. The thermal budget is set such that the temperature of the focal plate does not exceed air temperature by 1°C to ensure seeing stability [2]. We will revisit this constraint when discussing the proposed DESI movement method in Chapter 5.

The GFAs are critical to the focal plane system assembly, as they:

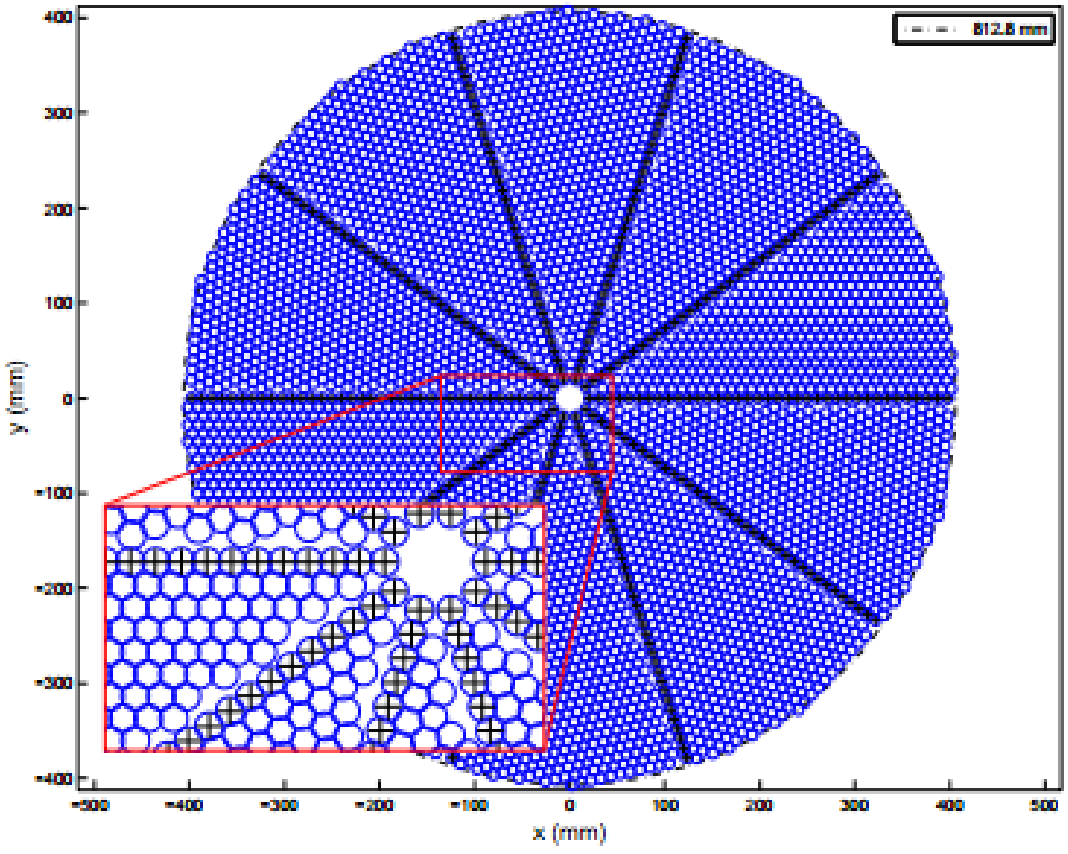
- Determine the current telescope pointing orientation within 20 seconds
- Determine focal plate scale, rotation, and astrometric solution
- Monitor the intensity of starts during observation
- Provide guide signals to the telescope
- Determine wavefront errors in focus, decenter, tip and tilt and provide it to the hexapod

The 10 GFAs are split into two subgroups: 3-4 focus and alignment sensors and 6-7 guide sensors. The guide sensors guide the telescope to target positions on the sky and maintain that pointing throughout the observation. During observation, intensity centroid data from stars will help determine the pointing conditions. The focus and alignment sensors will provide intra- and extra-focal images of stars, used to generate wavefront error maps.

Table 2.1: Focal Plane System - Science Requirements [2]

Item	Value	Rationale	Current Design
Number of fiber positioners supported within field of view	5,000	Science throughput	5,000
Number of GFAs supported within field of view	10	Provides sufficient sensor area	10
Number of illuminated fiducials within field of view	≥60	Kinematics of fitting the field when viewed through fiber view camera	61
Number of fiber management units supported	10	Logical distribution to 10 spectrographs	10
Heat load per observation	≤500 kJ	Ensure practical sizing of cooling system	293.6 kJ

Figure 2.1: Hexagonal closed-packed focal plate arrangement with 10 petals, each having 500 positioners, 6-7 FIFs and a GFA [2].



2.2 Fiber View Camera

While the computer controlled robot positioners position the 5,000 fiber optic cables to their target location, feedback is required to ensure precision in placing the fibers. The required precision due to science throughput requirements is a total of $5\mu m$. A re-position from an initial location to a final location will be partitioned into three separate moves. The $5\mu m$ accuracy is not required in each move, but after all three moves, an $5\mu m$ accuracy to final location must be met.

A first "blind" move will be made from initial location towards the final target location. The purpose of the fiber view camera (FVC) then comes into play. The FVC will take an image of the fibers on the plane, calculate the offset from their true location to their target location, and then feed the correction into the controlling electronics to adjust the positioners towards the $5\mu m$ accuracy. Only two iterations of the FVC feedback is required to get all of the positioners to the required precision.

For the FVC feedback, the fibers will be back illuminated by monochromatic LEDs, providing 25,000 electrons per fiber image on the CCD. The images from the FVC must be made through the telescope corrector optics, however there are 60-70 fixed FIFs on the fiber plane with measured positions to adequately correct for distortions [2]. The FVC cannot be run while the telescope is slewing to a new position, however the first blind move can be executed.

2.3 Instrument Readout and Control System

To guarantee maximum efficiency, the DESI instrument will complete all activities between exposures, on average, in less than 120 seconds. However the design goal is to achieve this in 60 seconds. Immediately following an exposure, the following sequence, in correct time order, will occur before the next exposure takes place (with a goal of executing within 60 seconds):

1. Shutter in the spectrographs will close after previous exposure ends
2. Telescope slewing will begin, spectrometer readout will begin, focal plane fans will turn on, positioners will execute first blind move
3. After slewing is finished, focus and alignment sensors will acquire data and tweak the focal plane if necessary
4. Guiding sensors will activate, aquire, and guide the telescope's pointing
5. Fiducial lamps will activate, fibers will be backlighted, and FVC will measure a first correction move
6. First FVC correction move will be executed
7. FVC will acquire a second correction move
8. Second FVC correction move will be executed
9. FVC will check to ensure all fibers reach precision tolerance

10. Focal plane fans and fiducial lamps will turn off
11. Spectrometer readout will complete and clear
12. Shutter will open, new exposure will begin

For the development of movement methods for positioners, we must keep in mind the following critical processes:

- Slewing and initial blind move will be initiated simultaneously, and ideally the blind move of the positioners will finish before slewing is done
- FVC corrections must occur after focal plane alignment and tweaking has completed
- FVC will take two measurements and each will require a correction move, with a final FVC check
- Total length of all the focal plane processes must complete within 120 seconds, and ideally within 60 seconds

See Figure 2.2 for a visual representation of the exposure sequence execution [2] – note that more steps are outlined in this illustration than described above. The sequence describe above just focuses on parameters most relevant to fiber positioner movement.

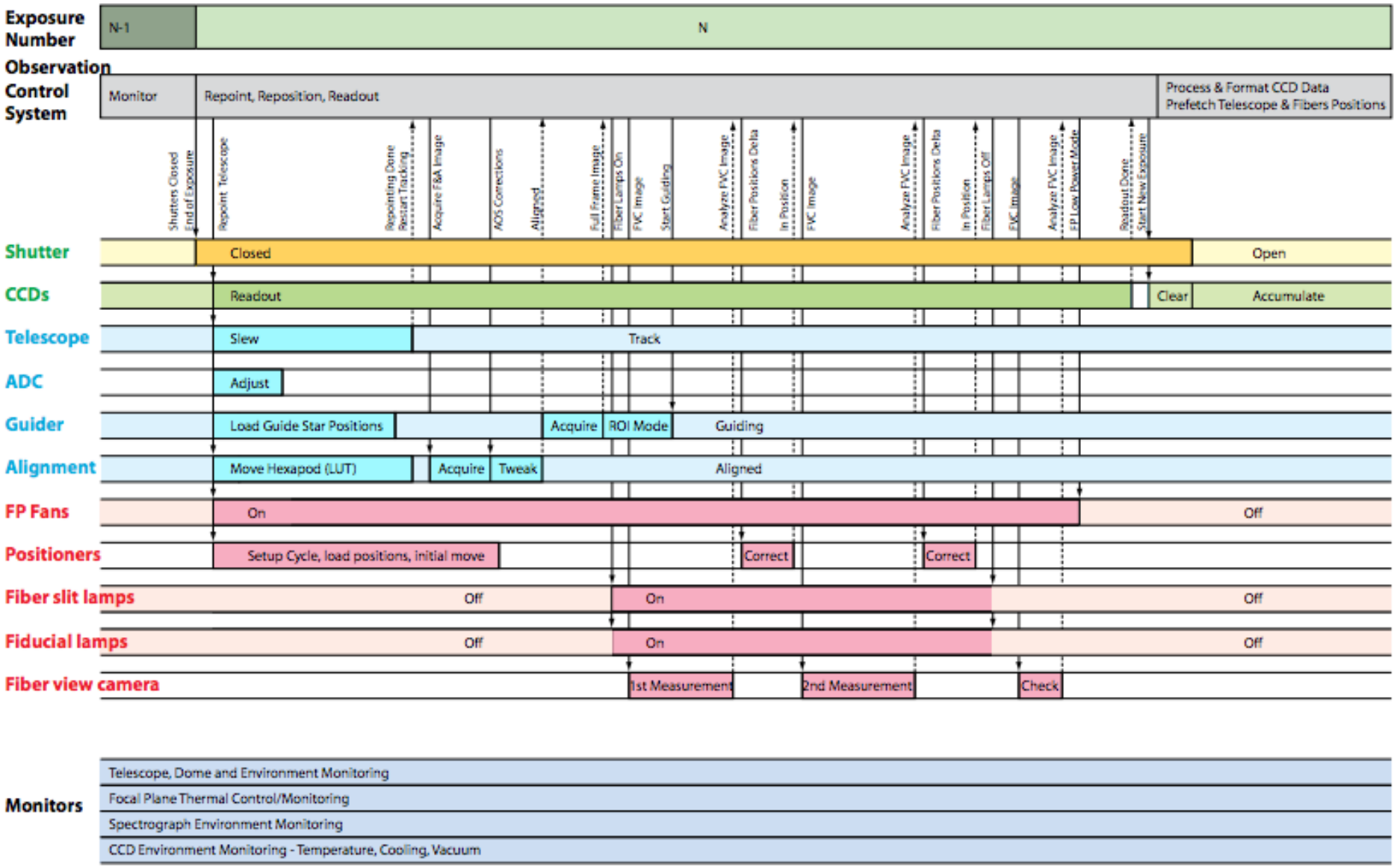


Figure 2.2: Illustration of inter-exposure sequence, as outlined in the DESI Conceptual Design [2].

Chapter 3

Fiber-Optic Positioners

3.1 Design Specifications

The primary requirements for the fiber positioners are: (1) they must consistently locate their targets with an accuracy of $< 5\mu m$, (2) the positioners must reach their targets within 120 seconds - ideally 60 seconds - including FVC feedback and correction moves, (3) each positioner must have mass less than 50 grams, (4) have power consumption of less than 1.2 W during movement and less than 0.001 W during rest, (5) must operate for the lifetime of the project ($\approx$ 4 yrs), and (6) have small mechanical envelopes to fit all 5,000 on the focal plane [2].

The fiber optic positioners were initially designed by LBNL and modified by UM. They have an eccentric (θ,ϕ) parallel axis scheme [5] – see Figure 3.1 for the coordinate system definition of the eccentric axis scheme. There are two components to this scheme: the theta body and the phi arm. From a top-down view, the body of the positioner takes up a circular envelope of diameter ϕE_0, with the center of the circular envelope at (x_c, y_c). From the center, there is a first arm, R1, that connects to the theta body, and from the theta body, there is another arm, R2, that connects to the ferrule holder (where the tip of the fiber optic cable is held to observe target galaxies) – the arm R2 is also known as the phi arm. It is much like a double pendulum scheme where we transform global coordinates (x, y) to local positioner coordinates (θ, ϕ).

ϕE_0 defines the "safety" envelope for a positioner. These envelopes will be hexagonally closed packed on the focal plane, maximizing sky coverage while keeping target density about 1 per positioner in each exposure, so that positioners are not idle during an exposure, nor overloaded with targets. If all positioners are within this envelope, they cannot hit each other. However, just because a single positioner is within its own safety envelope, does not mean other positioners cannot reach it. Taking a look at Figure 3.1, we see that positioners can reach outside of their own respective safety envelope.

Figure 3.1 shows R1 and R2 leading to single point rotation axes, however, the real positioner is an extended body. A more accurate picture is shown in Figure 3.2 with R1 and R2 superimposed to demonstrate the relation. The theta body can travel around the safety envelope by increasing or decreasing θ. As the theta body rotates, the phi arm will also travel, as it is housed on the theta body. The phi arm can go outside of the safety envelope, or into it, by increasing or decreasing ϕ.

Figure 3.1: Illustration of coordinate system for the eccentric axis scheme positioner [5].

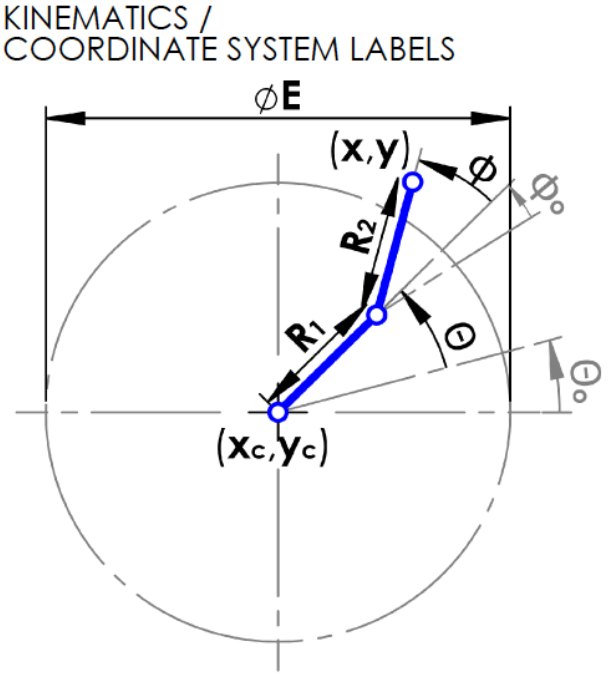
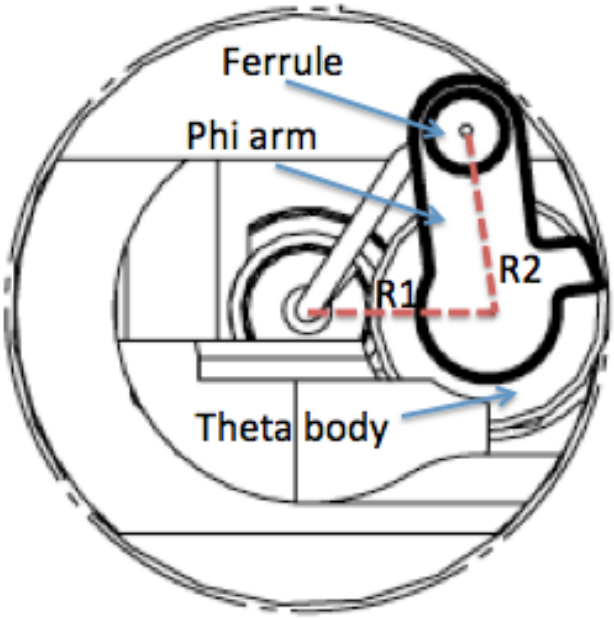


Figure 3.2: Illustration of top-down positioner schematic [5].



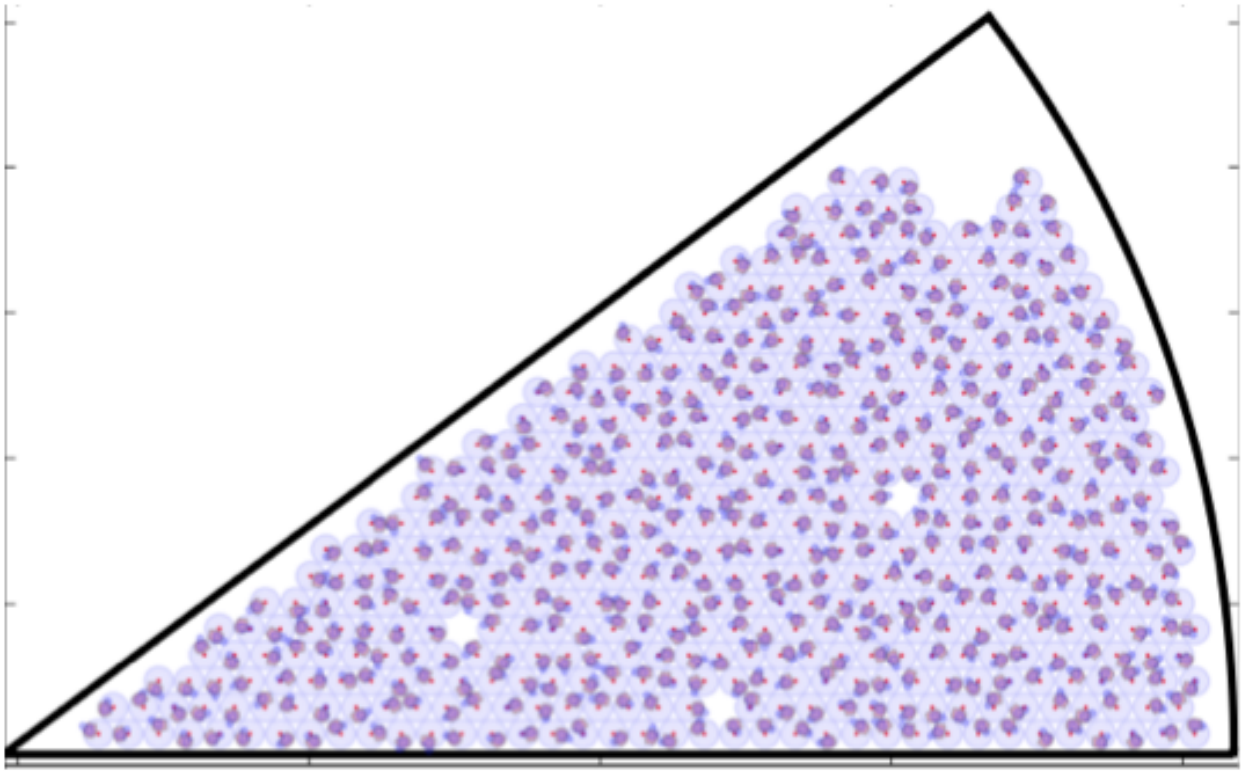
DESI will hexagonally pack 500 safety envelopes onto each of the 10 petals, meaning that each positioner will have a total of 6 neighboring positioners, with exceptions of those positioners near the edge of each petal on the focal plane. A simulated petal structure with hexagonally close packed positioners is demonstrated in Figure 3.3. Each fiber optic positioner has a 9.9mm safety envelope ϕE_0 and a 12mm total patrol diameter equal to $2(R1 + R2)$. The pitch from one positioner center to any neighboring positioner center is 10.4mm – thus positioners have overlapping patrol disks. Figures 3.4-3.5 demonstrate overlapping of patrol disks and safety envelopes of positioners.

Although this paper will focus on the movement of all positioners (the movement in the θ, ϕ) the fiber optic positioners are much more than just these bodies. The entire positioner is a long cylindrical rod, electronically plugged into the focal plane. The theta body rotates about the cylindrical axis, however it can only rotate from $-190 < \theta < 190$ degrees before hitting a "theta stop" on either side. The phi body rotates about the theta body axis, yet it can only rotate from $-5 < \phi < 185$ degrees, with 0 defined as fully extended towards a neighbor and 185 defined as fully tucked inside of the safety envelope. Figures 3.6 shows the entire positioner as it will be installed into the focal plane [2].

Key terms to remember:

- Safety envelope - 9.9 mm envelope surrounding the positioner center
- Patrol disk - 12 mm envelope defining where a positioner can navigate its ferrule holder
- Theta body - extended body that rotates about the positioner's central axis
- Phi arm - extended body that rotates about the theta body's central axis

Figure 3.3: Simulation of hexagonally close packed petal with 500 positioners, the GFAs, and FIFs.



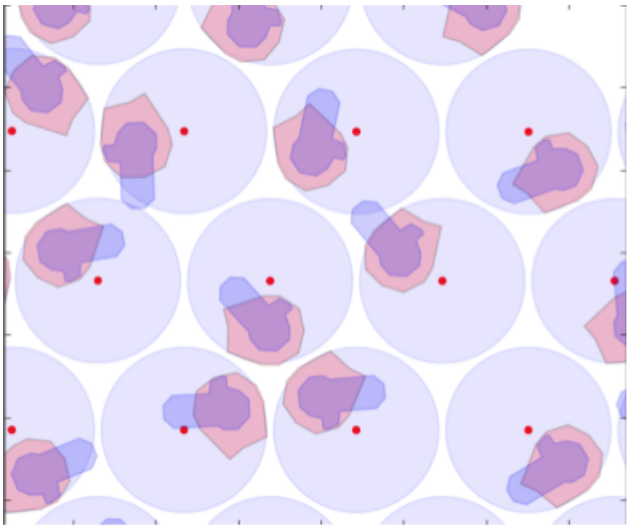


Figure 3.4: Zoomed in to show hexagonally packed positioners. Blue disk is the safety envelope of 9.9mm, red solid the is theta body, blue solid on the theta body is the phi arm, and the red dots signify the centers of positioners.

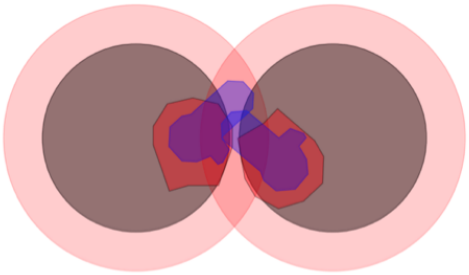
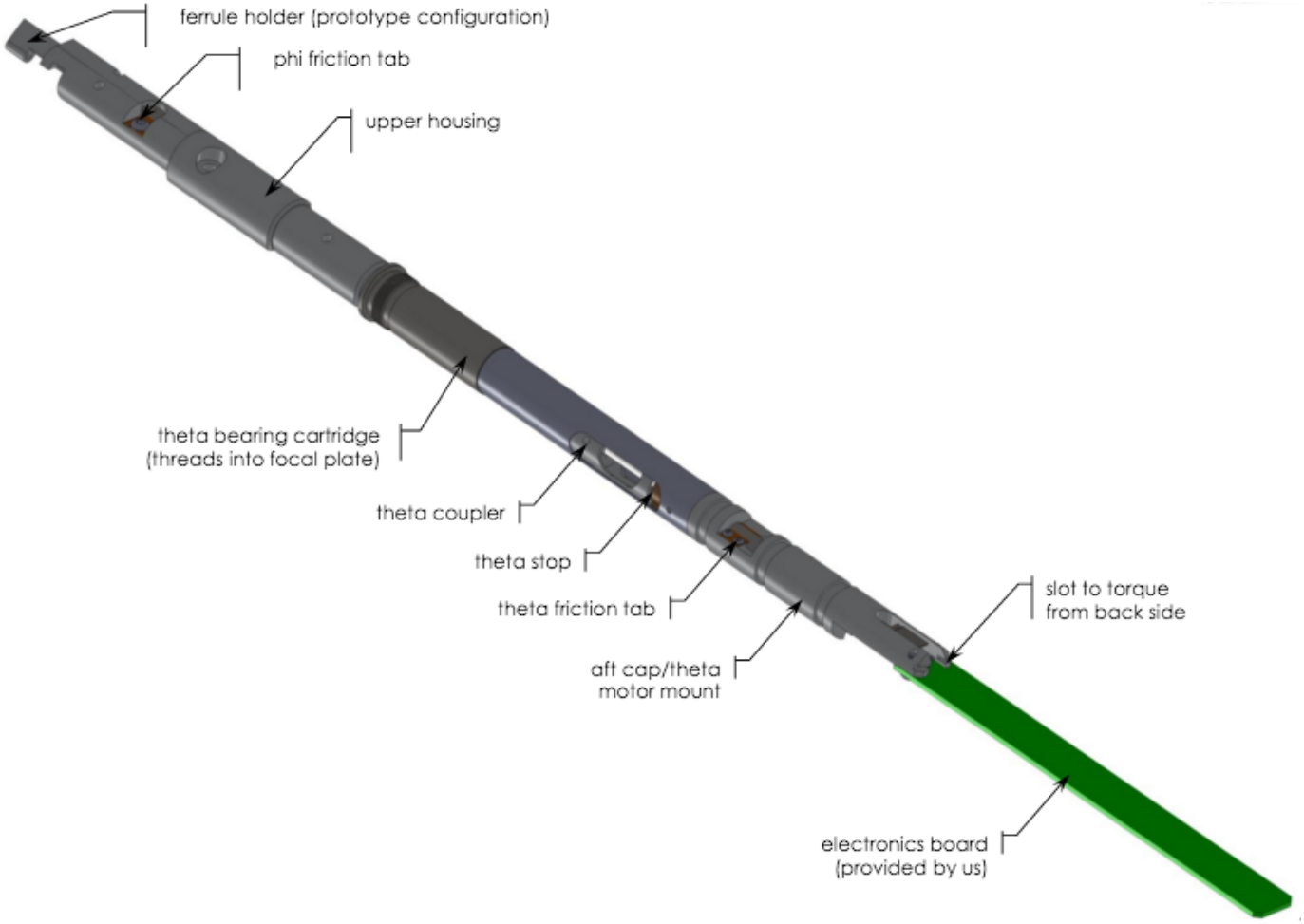


Figure 3.5: The dark red and dark blue bodies are the theta and phi bodies, respectively. The dark grey, smaller circle is the safety envelope, and the light red, larger circle is the full patrol disk of each positioner. The distance between neighbors is the pitch, 10.4mm, and patrol disk of positioners reach 12mm. Safety envelopes do not overlap, but a positioner can patrol into a neighbor's safety envelope.

Figure 3.6: Positioner device prototype, designed and developed by UM for DESI [2].



3.2 Interactions of Positioners

Each positioner has a physical center of (x_c, y_c) and during each re-positioning, must go from the previous target location of (θ_0, ϕ_0) to a new target location (θ_f, ϕ_f). 500 safety envelopes of diameter ϕE_0=9.9mm centered around (x_c, y_c) are hexagonally close packed in each petal of the focal plane, with a pitch of 10.4 mm between neighbors, while the patrol disk of each positioner has diameter of 12mm. The overlap between the patrol disk of a single positioner and the neighboring safety envelopes create instances where possible positioner-to-positioner collisions may occur.

As seen in Figure 3.5, the overlap of patrol disks between two positioners create "lens" zones (in the figure, the overlap looks like a convex lens) where a collision may occur. For a single positioner, there are 6 lens zones, one for each neighbor. Before characterizing types of collision interactions, we can easily estimate the probability for a collision to even occur.

The probability of a collision is roughly equal to the probability that two positioners will occupy the same lens zone (but this is an estimate as positioners may still be in a lens zone, but the extended bodies will not actually overlap). The area between two overlapping circles with the same radius r (r is the patrol radius) separated by distance d (the pitch) is given by (i.e. the area of the lens):

$$A = 2r^2 \cos^{-1}\left(\frac{d}{2r}\right) - \frac{d}{2}\sqrt{4r^2 - d^2} \tag{3.1}$$

The probability that any positioner target is in any zone where it could have a collision (any of the 6 lens zones), P_{lens}, is simply ratio of the lens area, multiplied by 6, to the total area. The probability that a second positioner is in the same lens zone is just P_{lens}, because it must choose the same lens zone yet does not have 6 total options. So the total probability that two positioners are in the same lens zone, i.e. roughly colliding, is ($r = 6$mm, $d = 10.4$mm):

$$P_{col} = 6P_{lens}^2 = 6\left(\frac{2r^2 \cos^{-1}\left(\frac{d}{2r}\right) - \frac{d}{2}\sqrt{4r^2 - d^2}}{\pi r^2}\right)^2 \approx 2\% \tag{3.2}$$

Thus, the probability that any two neighbors will have a collision is approximately 2% (technically this is the chance that two targets are in the same lens area, however since the positioners are extended bodies, it is likely this will cause a collision anyway). Indeed, we will see this order of magnitude estimate return once we simulate movement methods.

While collisions between two positioners will not mechanically harm them, the collision will throw off position tracking, as the movements of the positioners are run in an open-loop. Positioners don't receive feedback on where they truly are until the FVC feedback occurs, but they are expected to be within tenths of millimeters of target locations. To avoid collisions from actually occurring at all, we define a recommended "keep-out" zone surrounding the phi and theta body of a positioner. This keep-out zone is an extra 0.1mm padding zone, not actually part of the extended body. However, in simulation, we assume this padding is part of the extended body, so that even if a collision occurs in a simulation, a collision is not actually occurring physically. Figures 3.7 - 3.10 show nominal and keep-out envelope definitions for phi and theta bodies [5].

Figure 3.7: Upper phi arm, physical actual dimensions and then recommended keep out zone of 0.1mm padding [5].

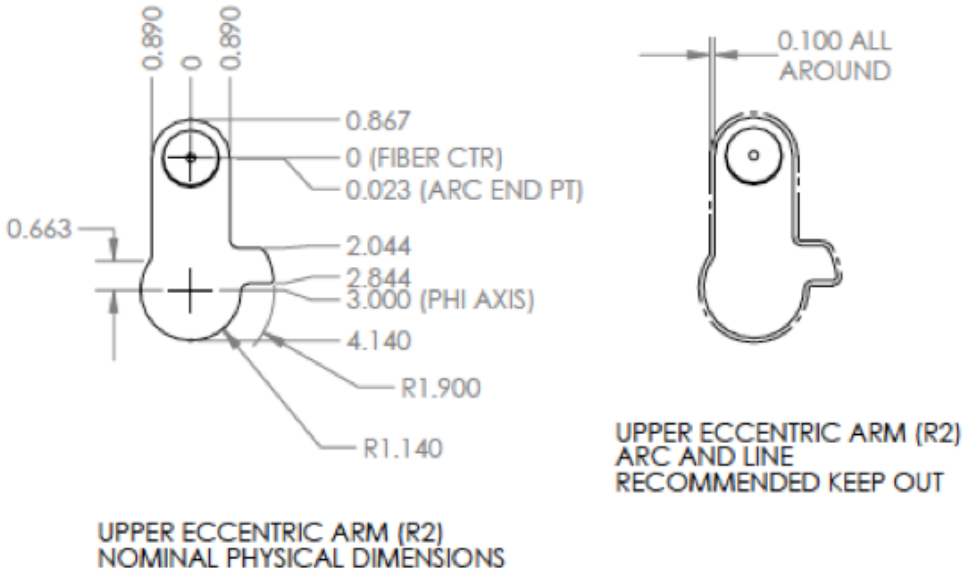


Figure 3.8: Lower phi arm, physical actual dimensions and then recommended keep out zone of 0.1mm padding [5].

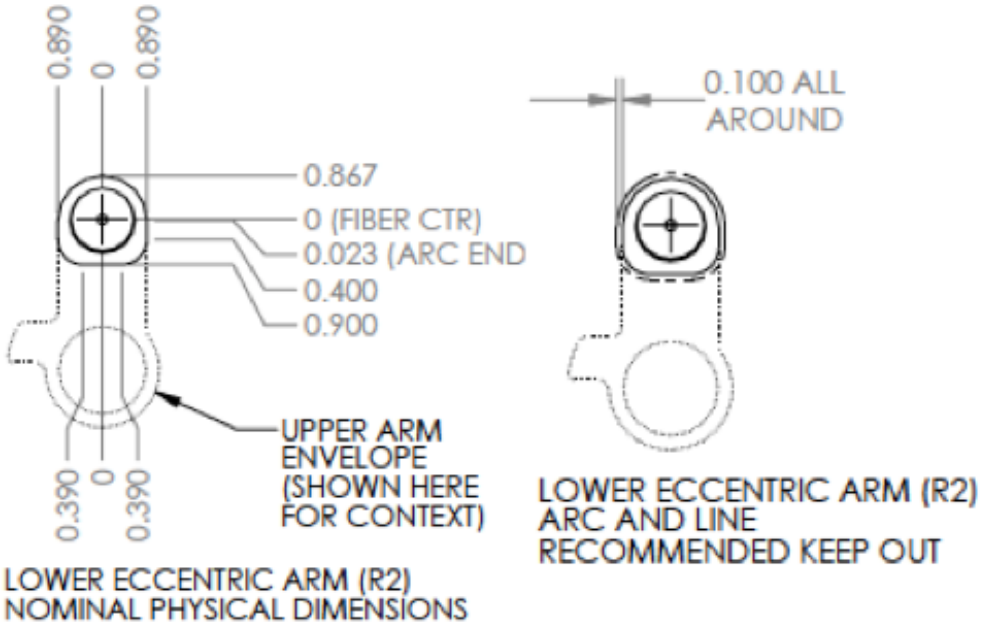


Figure 3.9: Theta body definitions - potential collisions can occur with shaded areas [5].

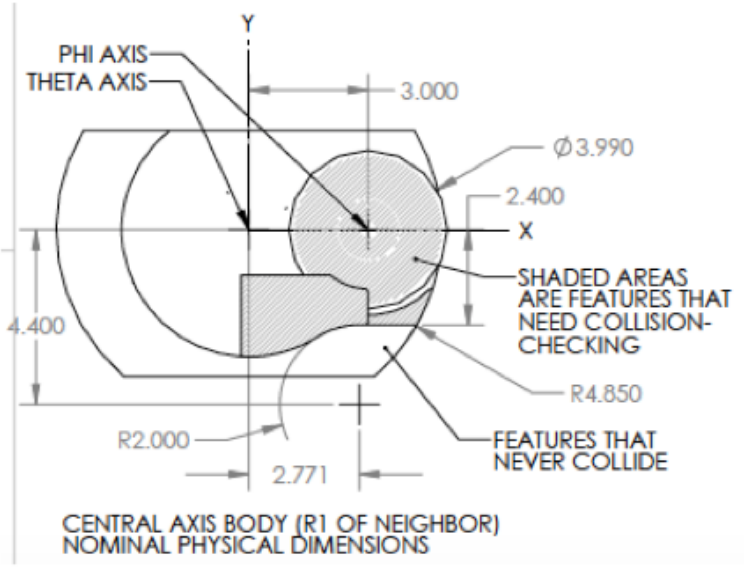


Figure 3.10: Theta body polygon, physical and recommended keep out zone definitions [5]

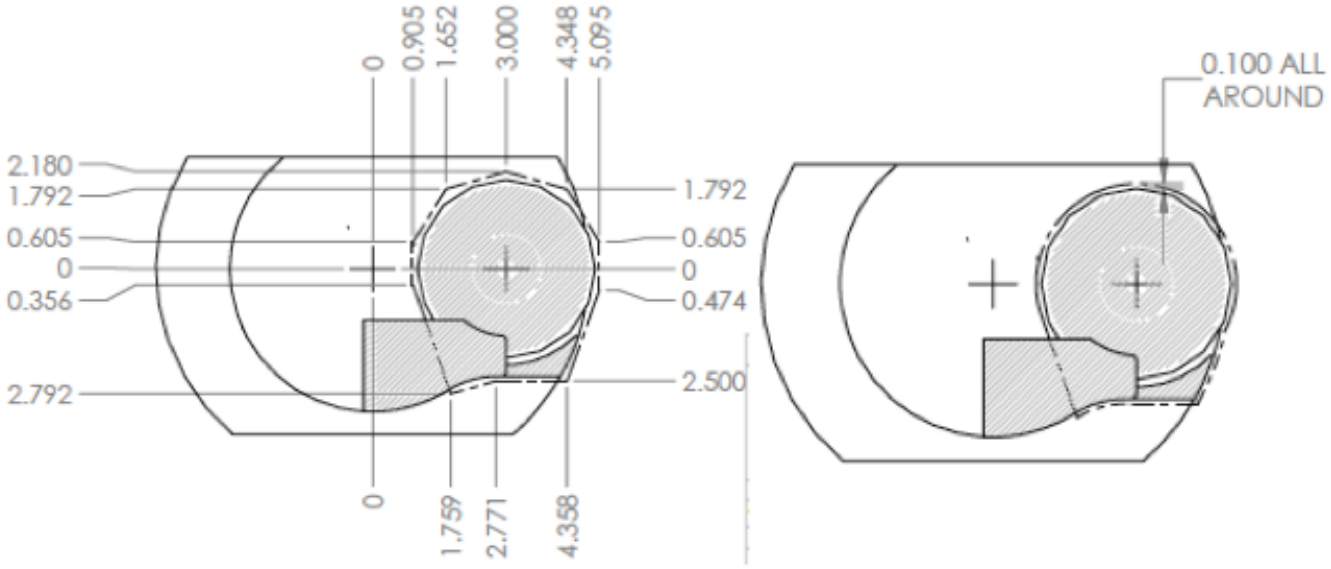
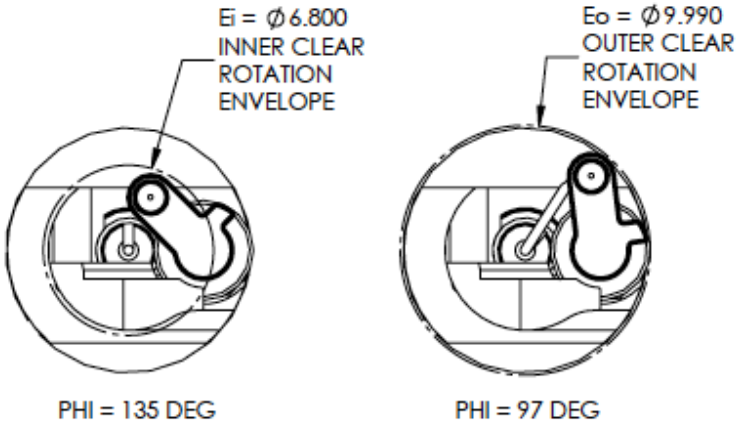


Figure 3.11: Definition of safety envelope (on the right), ϕE_0, and inner safety envelope (on the left), ϕE_i [5].



When positioners are within zones of collisional potentiality (i.e. the lens zones), there are two types of collisions that can occur between positioners A and B:

1. The Phi arm of A can collide with the phi arm of B (the two recommended keep out zones collide) - and vice versa.
2. The Phi arm of A can collide with the theta body of B (the two recommended keep out zones collide) - and vice versa

Note from now on a collision in simulation software is treated as collision of recommended keep out zones. This is a "fail-safe" design feature so that an actual collision will never occur. Also recall that the phi and theta body have mechanical hard-stops, limiting the range of travel. θ ranges from [-190,190] and ϕ ranges from [-5,185].

In positioner interactions, there are two envelopes of interest. The first is the safety envelope, which we have already mentioned ($\phi E_0 = 9.9$mm). When all positioners remain completely within this envelope, no collisions are ever possible. There is a second envelope, called inner-safety envelope, $\phi E_i = 6.8$mm. The advantage to the inner-safety envelope is that if positioner A is within ϕE_i, positioner B can only reach the theta body of positioner A, and not reach the phi arm. Figure 3.11 demonstrates the difference and also highlights what phi angles correspond to entering ϕE_0 and ϕE_i [5].

The two possible collision types (Phi-Phi and Phi-Theta) can occur under the conditions outlined in Table 3.1. Figures 3.12 - 3.14 visualize collision types [5].

Table 3.1: Given two positioners A and B, types of collisions that can occur [5].

State of Arm A	State of Arm B	Collision Type
Inside E_0	Inside E_0	NONE
Outside E_0	Inside E_i	A Phi - B Theta
Outside E_0	Outside E_i	A Phi - B Phi A Phi - B Theta A Theta - B Phi

Figure 3.12: When both positioners are inside ϕE_0, no collision is possible [5].

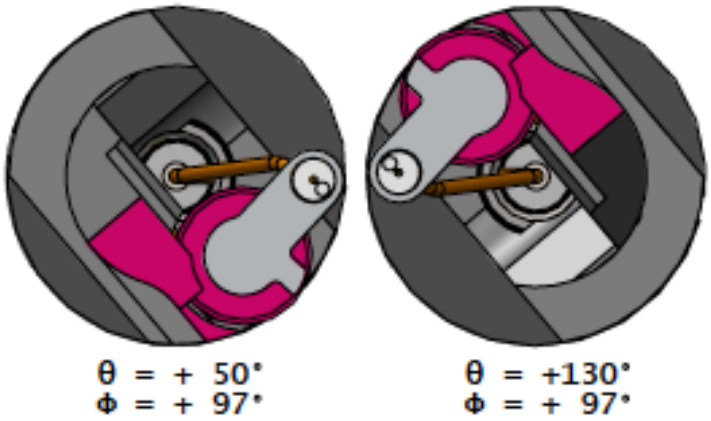


Figure 3.13: Even though neighbor (right) is inside ϕE_i, a phi-theta collision is possible. But neighbor phi is tucked out of reach [5].

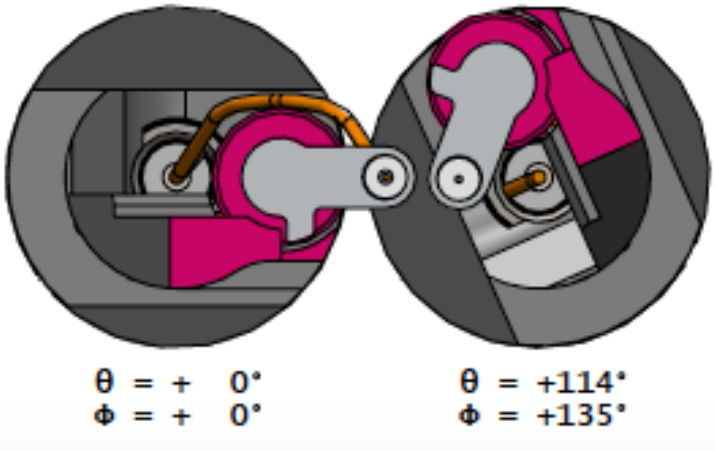
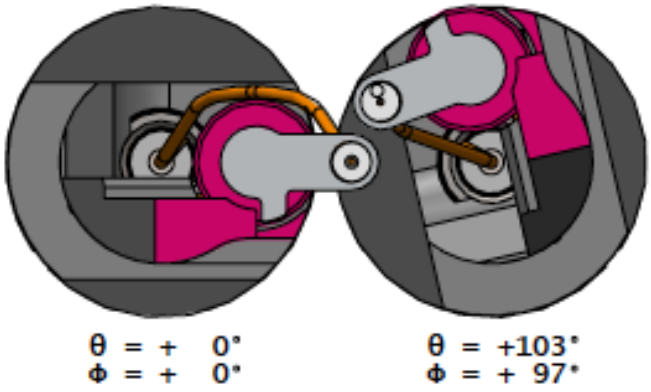


Figure 3.14: Even though neighbor (right) is inside ϕE_0, a phi-phi collision is still possible because neighbor it outside of ϕE_i [5].



3.3 Kinematics of Positioners

In order to accurately simulate positioner movement and thus accurately detect when a collision will occur, the kinematics of the positioners must be discussed and programmed.

The positioners have four phases of movement when moving from (θ_0, ϕ_0) to (θ_f, ϕ_f), **provided that $\Delta\theta$ and $\Delta\phi$ are** $> 6°$. Note that θ and ϕ can be moved simultaneously or separately, and these rules apply to θ and ϕ separately. A "move" can be just θ, just ϕ, or both. The gear ratio of the motors for DESI is currently 337:1. The four phases of movement are:

1. Spin Up - Acceleration from 0 RPM to 9900 RPM
2. Cruise - Constant 9900 RPM
3. Spin Down - Deceleration from 9900 RPM to 0 RPM
4. Creep - Constant 60 RPM

If $\Delta\theta$ and $\Delta\phi$ are $\leq 6°$, positioners will **only** creep for the entire move. The threshold is set at 6° because a positioner takes about 4° of rotation to completely spin up and spin down and we must creep for at least 2°.

3.3.1 Spin Up (and Spin Down)

$\Delta\theta$ and $\Delta\phi$ are $> 6°$, spin up occurs within an angular displacement of 628.2° of shaft rotation. Recall, the gear ratio is 337:1, so spin up occurs for approximately 2° of θ or ϕ. Since the displacement is fixed, and we know the final and initial RPM of the motor, we can calculate the time it takes for spin up to occur:

$$\theta_f - \theta_0 = \frac{1}{2}(\omega_f - \omega_i)t \Longrightarrow t = \frac{2\Delta\theta}{\Delta\omega} \tag{3.3}$$

With $\Delta\theta = 628.2°/337$ and $\Delta\omega = 9900$ RPM$/337$, the time to execute a spin-up move is ≈ 0.021 seconds with current DESI parameters. **Spin down occurs in the same amount of time and within the same angular displacement.** If $\Delta\theta$ and $\Delta\phi$ are $\leq 6°$, spin up and spin down does not occur, only creep does.

3.3.2 Creep

If $\Delta\theta$ and $\Delta\phi$ are $> 6°$, creep occurs within the last two full rotations of the shaft - 720° of shaft rotation, or $720°/337$ ϕ or θ displacement. Creep is slow movement adjustment for two rotations until the final location is reached. This is desired to minimize motor backlash and ensure that we do not overshoot / undershoot the target location, and keep the required $5\mu m$ target accuracy.

Currently, creep occurs at a constant 60RPM, but this may change for the final DESI parameters as the testing of the accuracy with current motors has yet to be completed. At a

constant speed with defined angular displacement, we can also similarly calculate the time it takes for the creep phase to occur:

$$\theta_f - \theta_0 = \omega_0 t \Longrightarrow t = \frac{\Delta\theta}{\omega_0} \tag{3.4}$$

Now with $\omega_0 = 60$ RPM/337 and $\Delta\theta = 720°/337$, the time to execute a creep move is ≈ 2 seconds.

$\Delta\theta$ and $\Delta\phi$ are $\leq 6°$, the the time to execute the creep move is the same as the time to execute the entire move, and will depend on the total displacement of the move. However, we can put bounds on this movement from 0° angular displacement to 6° angular displacement, using again Equation 3.4 but allowing $0° \leq \Delta\theta \leq 6°$ of θ / ϕ rotation. The time to execute a creep move if $\Delta\theta$ and $\Delta\phi$ are $\leq 6°$ is: $0 \leq t \leq 5.61\overline{6}$.

3.3.3 Cruise

If $\Delta\theta$ and $\Delta\phi$ are $> 6°$, the angular displacement for creep varies depending on the total angular displacement of the move:

$$\Delta\theta_{tot} = \Delta\theta_{spin-up} + \Delta\theta_{cruise} + \Delta\theta_{spin-down} + \Delta\theta_{creep} \tag{3.5}$$

However since $\Delta\theta_{spin-down}$, $\Delta\theta_{creep}$, $\Delta\theta_{spin-up}$ all are executed with definite displacements, we can put limits on $\Delta\theta_{cruise}$ since θ and ϕ have mechanical hard-stops and can only rotate so far. Furthermore, the cruise move is executed at a constant 9900 RPM, so we can use Equation 3.4 again to put a range on the total cruise time.

The largest angular rotation possible is when the theta body rotates from $-190°$ to $+190°$ (hard-stop to hard-stop), so $\Delta\theta_{tot,max} = 380°$. Since $\Delta\theta_{spin-up}+\Delta\theta_{spin-down}+\Delta\theta_{creep} \approx 5.86°$, $0 \leq \Delta\theta_{cruise} \leq 374.14°$. So if $\Delta\theta$ and $\Delta\phi$ are $> 6°$, the time to execute a cruise move is: $0 \leq t \leq 2.155$ seconds. If $\Delta\theta$ and $\Delta\phi$ are $\leq 6°$, cruise does not occur, only creep does.

3.3.4 Total Movement Time

Using these characterizations for each phase in the movement of a phi arm or theta body, we can look at the total time it takes for a positioner to move from (θ_0, ϕ_0) to (θ_f, ϕ_f).

If $\Delta\theta$ and $\Delta\phi$ are $\leq 6°$, $\Delta\theta_{tot} = \Delta\theta_{creep}$, so $0 \leq t_{tot} \leq 5.61\overline{6}$ seconds.

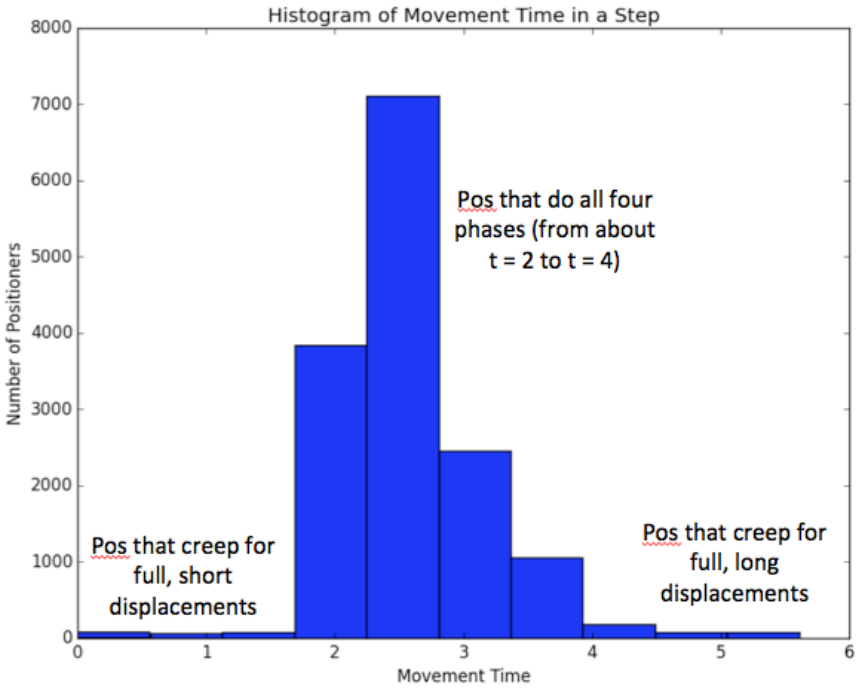
If $\Delta\theta$ and $\Delta\phi$ are $> 6°$, $\Delta\theta_{tot} = \Delta\theta_{spin-up} + \Delta\theta_{cruise} + \Delta\theta_{spin-down} + \Delta\theta_{creep}$, so $t_{tot} = t_{spin-up} + t_{cruise} + t_{spin-down} + t_{creep}$. Where $t_{spin-up} = t_{spin-down} \approx 0.021$ and $t_{creep} = 2$ with $0 \leq t_{cruise} \leq 2.155$. Thus $2.04 \leq t_{tot} \leq 4.198$ seconds. Note that the lower bound is not 0 due to the fact that spin up, spin down, and creep must occur, while cruise can have 0 displacement.

The largest conclusion we see is that creeping a positioner for a full 6° is the longest time consumer. Since all 5,000 positioners are moved simultaneously, the total time to move all

5,000 positioners only depends on the positioner that takes the longest time. If 4,999 positioners have times $2.04 \leq t_{tot} \leq 4.198$ seconds, yet that last positioner must creep for 6°, before executing the next move, all of the rest of the 4,999 positioners must wait for the last one to finish. Statistically, it is likely that at least one positioner will have to undergo creeping at the full 6° (or at least very close to it), and thus we can comfortably say that the time for all 5,000 positioners to move from (θ_0, ϕ_0) to (θ_f, ϕ_f) using the spin-up, cruise, spin-down, creep phasing, will take ≈ 5.617 seconds. See Figure 3.15 for a histogram of randomly sampling moves for 15,000 positioners. Even though only a few positioners have total times of about 5.5 seconds, these positioners will completely dominate how long all the other positioners must wait before starting a next move.

Even more, if we partition a move from (θ_0, ϕ_0) to (θ_f, ϕ_f) into three intermediate "steps", then the total time to complete that move would be around 16.8 seconds. We must move all 5,000 positioners simultaneously in each of the intermediate steps and it is likely that there will be at least one positioner that must creep for a very long displacement. Those initial 16.8 seconds are only for the first blind move of a re-positioning - after that, the FVC will guide two more small adjustment re-positionings, which will have all 5,000 positioners creeping for various small displacements. Currently, this is a topic of design discussion for the UM DESI team, as the time to re-position the entire focal plane system, ideally, needs to be constrained under 60 seconds.

Figure 3.15: Histogram of total movement time, randomly sampling total angular displacements from 0 to 390°, for full phasing of spin up, cruise, spin down, creep.



Pending investigation into the accuracy of these positioners, I propose that if a full move from (θ_0, ϕ_0) to (θ_f, ϕ_f) is broken up into any number of intermediate steps, that between steps, the positioners do not creep until the last step to (θ_f, ϕ_f). This will save a large amount of time since a move without creep would take a maximum of 2.197 seconds per iterative step, until the last one. So for a three-iterative-step move, the total time for the first blind re-positioning would decrease to about 10 seconds. If testing shows that the positioners do not accumulate significant error during each iterative step, and only one creep per entire move (regardless of number of intermediate steps) will ensure an accuracy of $5\mu m$, then a large amount of time can be saved by not creeping at each intermediate step.

Furthermore, I would also propose - pending investigation of accuracy - that the creep speed (or total angular displacement) is adjusted so that the time to execute maximum displacement of creeping is equal to the time to execute the maximum displacement of full phasing. In other words, we note that a move of spin up + cruise + spin down + creep takes a maximum of 4.198 seconds, yet a maximum creep displacement move takes up to 5.616 seconds. Adjusting the creep speed (or creep total displacement) to close the gap between these two maximums would reduce total time for a three-iterative-step move down to about 12.6 seconds (or if we also did not creep between iterative steps and equated the two maximum times, we could reduce a three-iterative-step move down to 8.6 seconds). Every second counts while re-positioning.

Chapter 4

Movement Methods

In this chapter I will explore how positioners move and the most efficient movement method.

4.1 Movement Command Sequence

To understand how best to move a positioner, we must first understand the physical and mechanical constraints on how positioners actually execute moves.

Before taking an exposure, the DESI targeting software will choose targets from the patch of sky for each positioner, so that each of the 5,000 positioners has a unique target location, (θ_f,ϕ_f) to move to. These targets are fed into the collision avoidance software to ensure all 5,000 positioners will reach their targets, free of collision. The collision avoidance software creates a "move table" for each positioner - a schedule detailing how each one should move. The move table is then uploaded into the CAN bus and each positioner accesses its unique move table. Once a sync signal is sent to all 5,000 positioners, they all begin reading the move table and executing each detailed step.

Each positioner receives a unique move table, where each move table is a 2D array. Each row in the array is an intermediate step that contributes to the entire move. For example, a complete move is from (θ_i,ϕ_i) to (θ_f,ϕ_f), but this move may be completed in any number of intermediate steps $\Delta\theta_s$, $\Delta\phi_s$ so long as $\theta_f = \theta_i + \sum_s \Delta\theta_s$, $\phi_f = \phi_i + \sum_s \Delta\phi_s$. Each step k of the move schedule has the following information (subject to change in final DESI design):

- A pre-pause, τ_{k1}, in which the positioner delays before executing the motion $(\Delta\theta_k,\Delta\phi_k)$
- The step displacement motion $(\Delta\theta_k,\Delta\phi_k)$, which will be executed following the spin up, cruise, spin down, and creep movement scheme.
- A post-pause, τ_{k2}, a padding-time where the positioner waits for all other positioners to complete their own k^{th} moves before this positioner continues to its next step sequence.

From the movement command sequence for a positioner, there are three main constraints on our movement method:

1. The move table for each positioner must be tabulated before the commands are sent to positioners, and cannot be updated once movement begins.

2. A full move from (θ_i,ϕ_i) to (θ_f,ϕ_f) must be discretizable into k intermediate steps.

3. The number of discretized steps, k, for a move should not exceed 10, otherwise the data transmission of the move tables to positioners will be slow, cutting into observation time.

4.2 Characterization of a Movement Method

Keeping in mind the mechanical constraints of our movement method, the best movement method for positioners should (a) move them to their target locations in the fewest amount of intermediate steps, but while (b) not creating a large number of collisions to avoid. More collisions to resolve means adding intermediate steps in a move table, which in turn means more time for the entire focal plate to re-position.

A movement method is different from a collision avoidance scheme. The movement method just details how positioners move from (θ_i,ϕ_i) to (θ_f,ϕ_f) - how many intermediate steps, what do those steps look like, etc.. As a result of a specific type of method, positioners will collide, and collision avoidance will be a necessary addendum to complete the movement method. However, to efficiently run the software, a collision avoidance scheme should be adapted to the movement method. A movement method is relatively easy to program and think of, however a collision avoidance scheme adapted to that method takes much more time investment. As such, each movement method idea does not have a corresponding collision avoidance scheme, and thus, each movement method is not fully complete.

Yet, we can still characterize the efficiency and quality of each movement method through three merits: (1) the total time, t_{tot}, to reposition all 5,000 positioners from (θ_i,ϕ_i) to (θ_f,ϕ_f), (2) the number of collisions the method must resolve, w, and (3) the complexity of the collisions. Since we do not solve the collisions in each movement method, the time to reposition only details how long it would take to reposition the focal plate if no collisions were resolved, giving a lower time bound on the method. The complexity of the collisions that need to be resolved can give us an idea of how many intermediate steps, k, must be added to the move table. The number of positioners determines statistics for a time histogram on each of the intermediate-step-added-for-collision-resolution.

Combining all three merits can give us a very good estimate on the quality of the movement method. Assume a movement scheme causes only $w = 5$ low-complexity collisions to resolve, collisions that only need $k = 1$ intermediate step added to resolve. Since there are only 5 collisions, statistically, the movement to resolve these collisions will almost never be in the upper bound of Figure 3.15, and thus the extra k steps to the move table will not take a lot of time to execute. Yet, if t_{tot} is enormously large to begin with, then this algorithm isn't worthwhile, even though the extra step would take a very short time.

So the best movement algorithm will have all 5,000 positioners get from (θ_i,ϕ_i) to (θ_f,ϕ_f) quickly (t_{tot} small), while not causing highly-complex collisions so that only a few number of steps need to be added to the move table (k small), and only causing a small number of collisions (w small) so these steps will statistically be short in time to execute.

4.3 Software Structure of Methods

The investigation into each movement method is done through software simulations. Coming up with movement methods is usually more difficult than investigating the three merits for that method because movement methods only change how a positioner goes from (θ_i,ϕ_i) to (θ_f,ϕ_f), which is relatively easy to program. The complete software structure includes:

- A function that creates random (θ_i,ϕ_i) and (θ_f,ϕ_f) positions for all 5,000 positioners
- A function that creates the 10-petal structure of the focal plate with 500 hexagonally packed positioners with a GFA and accurate FIF locations per petal.
- A function takes each positioner and creates an array of neighboring positioner IDs to check against in the future for collisions once movement takes place
- A function that takes in initial locations and final locations and moves positioners following the spin-up, cruise, spin-down, and creep scheme
- A function that checks whether two polygons overlap, checking if a collision has occurred. If a collision occurs, a text file is created that saves the parameters of that collision and it saves a screenshot of the collision.
- A function that tells how to partition the movement from (θ_i,ϕ_i) to (θ_f,ϕ_f) - **the movement method in question**

So for each newly born movement method, the last function must be edited, but then we use the structure in place to count the total time for repositioning (but not counting time it would take to resolve collisions), count the total number of collisions, and qualitatively observe the complexity of the collisions.

4.4 Simple Movement

The quickest way to move from (θ_i,ϕ_i) to (θ_f,ϕ_f) is to simply move from (θ_i,ϕ_i) to (θ_f,ϕ_f), in one step. This "simple-movement" method (SMM) is the baseline total time, the best time we can achieve. From the discussion earlier looking at movement-time histograms, we immediately know the total time to reposition all 5,000 positioners (excluding collision-avoidance steps) is about 5.6 seconds. Although this is certainly a very fast repositioning time, the algorithm has its fallbacks when it comes to the number of collisions needing to be resolved.

On average, the SMM requires us to resolve 1500 collisions (or 3000 positioners) per repositioning. Requiring the resolution of 30% of the positioners is large, however this would okay if the collisions were trivial to solve - recall that the time is only about 5.6 seconds, so adding a single avoidance-move or two would be within the time constraint. Yet, in SMM, the collisions are highly complex.

To detect collisions on my software, I first let positioners go to their final locations and draw the area that a positioner has covered during its movement - a "band". Then, I simply check if bands overlap or not, telling me if there is a potential collisions. Finally, I re-run

only the positioners that potentially collide, from start to finish, step by step, to see if they actually collide. Typically, a band in SMM will intersect with many other positioner bands, since the positioner will travel through many potential-collision-zones (the overlapping lens in Figure 3.5) on its way to the target location. This means that a single positioner will have collisions with many neighbors in a single move, requiring many adjustment moves to resolve all these collisions. This complexity makes it extremely difficult to generalize a solution to every single collision that a positioner has, since a positioner could have up to 6 collisions it needs to avoid in a single move, each one of a different nature - see Figure 4.1.

Summarizing, the SMM offers a baseline of best achievable time **if we ignore collisions**: 5.6 seconds for repositioning. However, for this method DESI cannot throw away the positioners that will collide in the respositioning since there are 3000 of them, and thus that 5.6 seconds will dramatically increase. SMM is unrealistic because out of these 3000, statistically at least one positioner will travel its full range of motion (from hard stop to hard stop), meaning it will cross 6 zones of collision-potentiality, and possibly require 6 collision-avoidance moves. That could require almost 40 seconds to do the first blind move - far too long for DESI focal plane constraints.

Our motivation for future methods is to try to get as close to this best achievable time for respositioning while dramatically reducing the number of collisions to avoid. If we get the number of collisions to about 1% (5 positioners per petal), DESI could actually tell those positioners to just not go to their location and DESI will still achieve its science goals. Even better, that means that there does not have to be any additional collision-avoidance steps added into the movement, saving total time (provided the original repositioning time is relatively small).

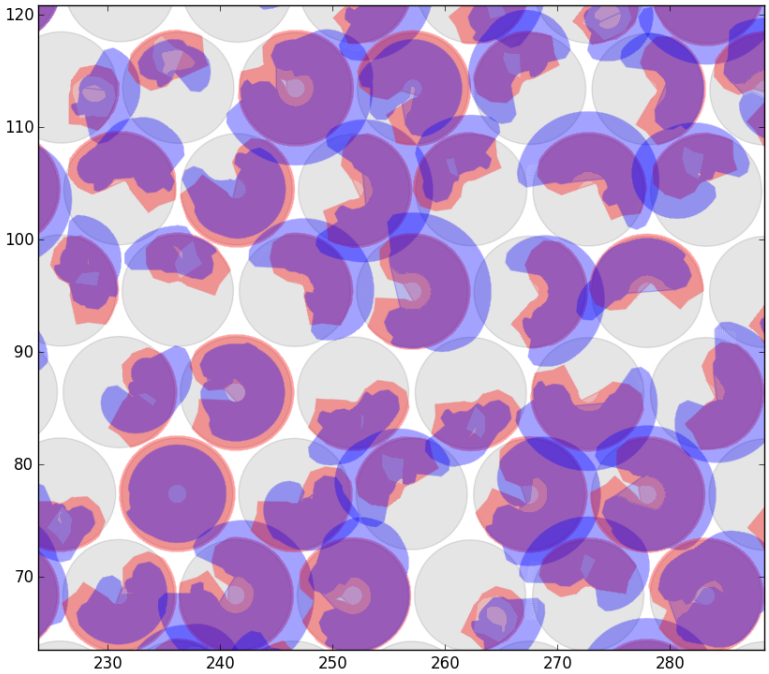


Figure 4.1: A snapshot of how positioners sweep out their movement areas during SMM. Note that positioners sweep out huge areas and cover a lot of area outside their safety envelopes ϕE_0, and a significant number of bands overlap, requiring a more detailed look to see if a collision will occur. Usually as the theta body rotates, the phi arm will swing out, sweeping into many of the potential collision lenses seen in Figure 3.5.

4.5 Straight-Line Movement

An improvement on SMM is changing how the positioners navigate to (θ_f,ϕ_f) so that the time a positioner spends outside of its safety envelope ϕE_0 is reduced. Although this will increase the time it takes to reach (θ_f,ϕ_f), it will, hopefully, dramatically reduce the number of collisions.

Even though the quickest way from (θ_i,ϕ_i) to (θ_f,ϕ_f) in (θ,ϕ) space is to allow (θ,ϕ) to increment until their final values, this leads to the swinging out of the phi arm that we saw in Figure 4.1. If we think in (x,y) space, the quickest way from point A to B is a straight line, and this type of movement would reduce the area outside of the safety envelope ϕE_0 of the positioner. This means that instead of taking values of (θ,ϕ) as input to our movement method, we will take the target location (x,y) of the ferrule holder (the mechanical object that holds our fiber optic cable) and determine the discretization in (θ,ϕ) space that will get the positioners there in a straight-line from the previous location.

Using this Straight-Line Movement (SLM) has the advantage of drastically reducing the area spent outside of the safety envelope ϕE_0, as illustrated when comparing it to SMM in Figure 4.2. However working in (x,y) space has its own fallbacks for DESI. First and foremost, the collision avoidance software will receive the input in terms of (θ,ϕ), not in terms of (x,y), meaning our software would have to translate back to the (x,y) space. Although this seems trivial, there is an ambiguity in (θ,ϕ) space, as each (x,y) point is non-unique - there are always two (θ,ϕ) orientations possible for each (x,y) point. As such, our method would have to pick the most optimal (θ,ϕ) with respect to the geometry of the phi and theta bodies of the positioners (which is precisely why the collision software is given (θ,ϕ) and not (x,y) points, since this is already optimized and calculated in target assignment software). For instance, one of these orientations may not be possible due to the mechanical hard-stops. Furthermore, requiring the ferrule holder to travel a straight line would require **continuous** changes in (θ,ϕ), or a large number of discretizations if we let the ferrule holder travel a straight-"ish" line. Remember that while the ferrule holder is traveling a straight line, the theta body and phi arm must manuver simultaneously and constantly change to allow the ferrule holder to keep on the straight line - see Figure 4.2 and try to imagine how the two bodies must move to keep along the straight line! Even worse, a straight-line path might not even be possible if one of the hard stops are reached. For long straight lines, the number of intermediate discretizations of the straight-line-path can grow very large, and thus simply take too long to reposition the focal plane for DESI constraints. Simply put, there are too many variables to uniquely define for each path and too many discretizations to be made for SLM to be efficient enough for DESI.

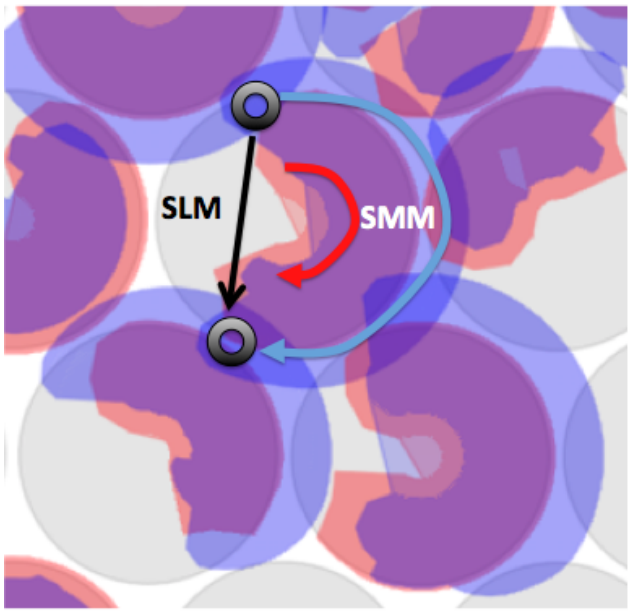


Figure 4.2: Comparing the paths taken to a target location for SMM and SLM. Note that in SLM, if we tell the ferrule holder to travel alone a straight line from initial location to target, the time that the positioner spends outside of its safety envelope ϕE_0 drastically reduces.

Although SLM is unrealistic for DESI, SLM's motivation should be captured in DESI's selected movement method: reducing the time spent outside the safety envelope ϕE_0. SLM cannot achieve this for DESI, as it does not meet mechanical constraints - the algorithm is simply not built for the mechanics of the project. A straight line path is not always, nor easily, discretizable for (θ,ϕ) space, and the number of discretizations can grow too large for practical implementation. A possible solution to these issues could be to tell the movement to limit the number of discretizations of a straight-line-path, however this opens up an entire new issue of trying to optimize what the best number of discretizations is, and what those discretizations should be. SLM does not adapt well to the setup of DESI's positioners. Rather than trying to manipulate the positioners to find the best path that keeps a positioner inside the safety envelope ϕE_0 the longest, our method should try to utilize the geometry of the problem.

4.6 Retract-Rotate-Extend

Utilization of the geometry is where the Retract-Rotate-Extend Method (RRE) surpasses all movement methods. To minimize collisions, our method should maximize time spent inside safety envelopes ϕE_0. Furthermore, we do not want to deal with translating between (θ,ϕ) space and (x,y) space, since the ambiguities have already been solved optimally by target assignment software. The best approach for our parameters is to partition the path from (θ_i,ϕ_i) to (θ_f,ϕ_f) by first requiring the phi arm to **retract** into safety envelope ϕE_0 - sending ϕ_i to some $\phi_{intermediate} = 97°$ (97° is when the phi arm completely pulls into the safety envelope ϕE_0). Now that all positioners are inside their outer safety envelopes, collisions are impossible, so we **rotate** θ_i to θ_f. Finally, we send $\phi_{intermediate} = 97°$ to its ϕ_f position, and we have reached the target location!

The benefits of this movement method are vast. First, we reach the target location in 3 simple moves (discounting any intermediate collision-avoidance steps), repositioning the entire focal plate within about 17 seconds for the first blind move (10 seconds if we do not execute creep adjustment during the retract and rotate phases). Second, collisions can **only** occur during the retraction and extension phases. Furthermore, for a given positioner undergoing a retraction / extension, theta is fixed so it can only encounter collisions with a maximum of two neighbors. The types of collisions that can happen in RRE are almost entirely phi arm - theta body collisions (the probability that two phi arms will be in overlapping lens-zones is low, as a phi arm is smaller than a theta body) and these types of collisions are very easy to resolve since the theta body simply has to "get-out-of-the-way" for the phi arm to pass it by. Phi arm - phi arm collisions are rare, but can be solved too by adding a time delay to one of the positioners or adjusting the theta orientation to move it out of the way. Complete details on the collision avoidance software for RRE is described in the next chapter. Finally, if we save the current collision avoidance solution for an extension, we can reverse those steps to use it in the next exposure's retraction move without having to worry about resolving any collisions! Then in each exposure (besides the very first one), RRE method only has to calculate collision avoidance for the current extension move, since the retraction is the reverse of a previous extension!

For RRE, the types of collisions are usually very simple to resolve. Even for the most complex collisions, maximally there will only be 2 adjustment-moves necessary to resolve these (explained further in the next chapter). At most then, there are 5 intermediate steps for a focal plate readjustment, and so the total time for the first blind move (before the FVC activates) will take about 28 seconds - if we creep at each step - or 16 seconds - if we save the creep for the last step. Well within time constraints, this method seems to be the best movement method for DESI. The last big benefit for RRE is the number of collisions this movement method causes.

After implementing RRE in software simulation and running samples multiple times, the average number of collisions per move (retract-rotate-extend but neglecting retract since it is a reflected-previous-extend) is 160.94 ± 23.43 from over 1,000 simulated sample runs of 5,000 positioners. This means a bit more than 15 positioners per petal encounter collisions that need to be resolved in the extension move. Even better, this means that after performing the retraction move (which is the previous step's reflected-extension), if the collision is complicated and requires more than one additional step to resolve, we can simply throw that target out and leave the positioner where it is! This would save DESI more time during the focal plate repositioning, and worse-case if we throw out all 150 colliding positioners, the telescope can still reach almost 97% of all the targets!

The RRE Method offers the most efficient and adaptable movement method for DESI. During each repositioning, the algorithm allows the possibility to simple ignore trying to resolve a complex collision while still maintaining DESI science goals. Lastly, the RRE Method keeps the first blind move focal plate repositioning between 16 to 28 seconds, within DESI time constraints. **This method was first proposed by Joe Silber at LBNL and I developed the first Python implementation that is being used, tested, and optimized by UM DESI team. My code will be used on the first positioners undergoing testing at UM. Complete characterization of this algorithm and implementation will be described in the next chapter.**

4.7 Other Methods

Beyond these 3 movement methods, SMM, SLM, and RRE, there are two other movement methods for positioners that have merit, but simply not mechanically feasible for DESI.

4.7.1 Right-of-Way Method

In this method, we allow all positioners to just travel towards their final target locations and when a collision is about to occur, yielding is assigned. A positioner that is closer to its target must yield to a positioner further away from its target, and it yields by moving out of the way for the right-of-way positioner. The yielding positioner moves out of the way much like it is following a repulsive magnetic field, being pushed out of the way by the right-of-way positioner. After the right-of-way positioner clears the yielding positioner and

continues towards its target, the yielding positioner can try again to move towards its own target location. The drawbacks of such a method are: (1) convergence of all positioners to their targets is not guaranteed, and (2) this method requires continuous change of direction between colliding positioners - even instantaneously - and thus impractical for DESI.

4.7.2 Potential Field Method

A close relative of the Right-of-Way Method is the Potential Field Method, a direct version of the Navigation Function proposed by a team of scientists in collaboration in Spain, [4]. A positioner feels an attractive force of its target and a repulsive force from neighbors that get close. Although beautifully imagined and guaranteed to converge to targets, this method would again require continuous and instantaneous change of motion for positioner, which is not feasible for DESI. This method - and the Right-of-Way Method - could be possibly implemented if these continuous paths are discretized with a maximum number of steps. However convergence can no longer be guaranteed and this discretization can lead to an excessive number of steps for a single focal plate repositioning if a positioner has to get out of the way many times.

Chapter 5

DESI Proposed Movement Method

In this chapter I will fully characterize and describe my implementation and deployment of the RRE Method for the DESI collision avoidance system. Since this movement method was chosen as the best one moving forward, it has been completed with anticollision software, and so the method is fully complete and ready to be deployed in real fiber optic positioners for testing.

5.1 The RRE Method

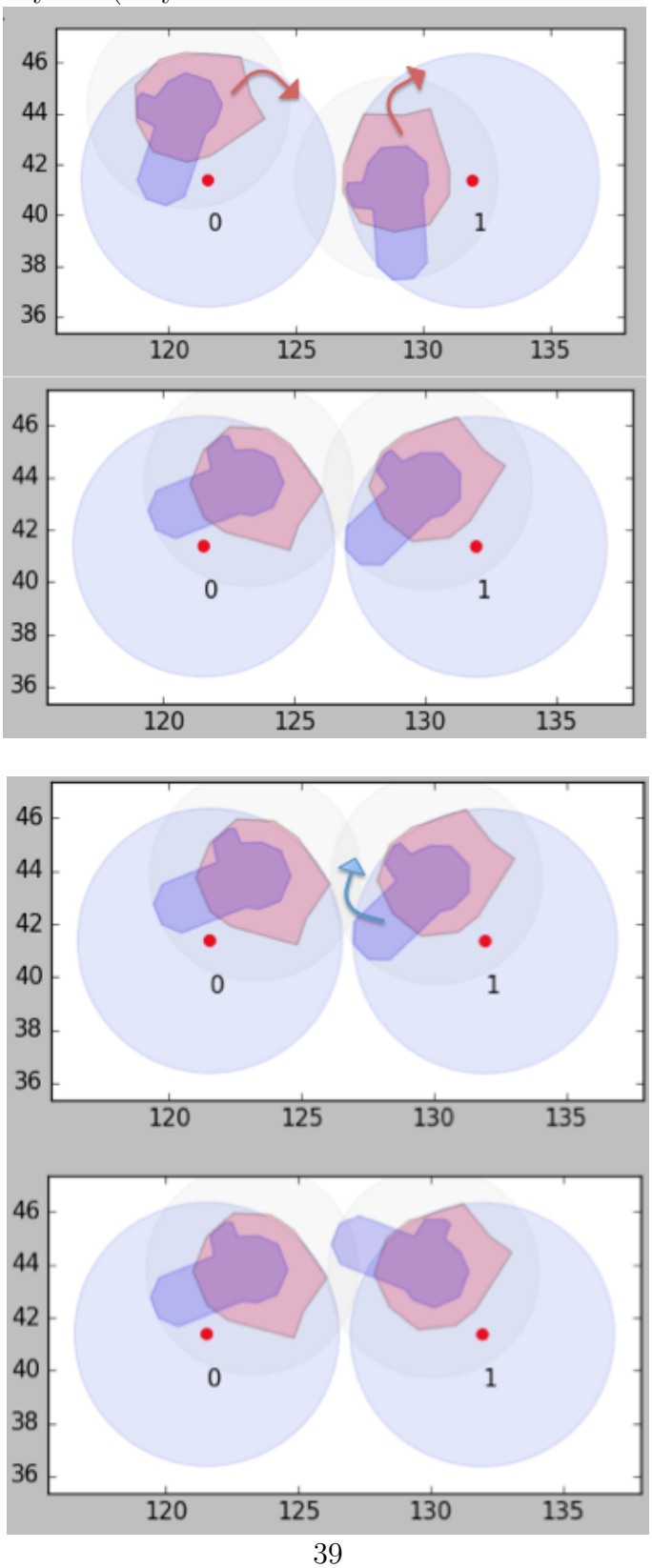
The RRE Method was proposed by Joe Silber at LBNL and implemented on Python for Linux deployment by myself, Efrain Segarra at the University of Michigan under the guidance of Dr. Gregory Tarlé and Dr. Michael Schubnell. My deployment was completed after investigating other possible movement methods, demonstrating how powerful the RRE method is.

The RRE Method centers around three steps for positioner movement:

1. **Retract** all positioners to the ϕE_0 safety envelope, only moving ϕ to 97° - collisions would be possible here but we just simply reverse the previous repositioning extension move that was already resolved of collisions.
2. **Rotate** all positioners in θ to their final theta target value.
3. **Extend** all positioners out of the ϕE_0 safety envelope, moving ϕ to final phi target value - this step requires collision avoidance

This movement exploits the geometry of the fiber optic positioners, utilizing the safety envelope ϕE_0 where no collision can occur if all positioners are inside of this envelope. Although the move is partitioned into moving θ and ϕ independently and not simultaneously, this partitioning reduces the complexity of collisions and reduces the overall number of collisions to resolve. Figure 5.1 demonstrates the look of the RRE movement method.

Figure 5.1: Top: Rotation part of RRE Movement - Positioner 0 and 1 move CW by 50°. Note that both positioners have phi arms inside of ϕE_0 safety envelope (light blue circle); grey light circle is range of phi arm. Bottom: Extension part of RRE Movement - Positioner 1 swings out phi arm by 43° (only extension is shown since retraction is reverse extension)



The type of collisions that can occur in the RRE Method happen during the extension move, and involve theta-to-phi collisions and phi-to-phi collisions. The former type of collision occurs much more often, as the probability that two phi arms will hit during the extension move is lower compared to a theta-to-phi hit. By running many samples of the RRE movement software and tracking types of collisions, Table 5.1 was populated.

Table 5.1: Collision Statistics for the RRE Method based on sample runs of 500 positioners

Total Number of Sample Runs	1238
Total Number of Positioners Sampled (500 per run)	619000
Total Number of Positioners that Collided	19924
Average Number of Collisions per Run	16.094
Percentage of Positioners in Collision per Run	3.22%
Percentage of Phi-Phi Type Collisions	33.43%
Percentage of Theta-Phi Type Collisions	66.57%

With this movement method, as we see in Table 5.1, only about 3% of all 5,000 positioners will have a collision to be resolved during the extension move! Furthermore, most of these collisions (67%) are theta-phi collisions, which are very easy to resolve. The remaining 33% of the collisions are phi-phi type, which are more difficult to resolve. The beauty of this movement method is that due to the low number of collisions we must resolve in each move, we can throw out any positioners that have highly complex collisions and still maintain DESI science requirements. More over, the RRE method has 3 raw steps, and then an extra two steps to resolve collisions (to be discussed in the next section). Thus for a total of 5 steps, without creeping in between steps, the total time to reposition the focal plate during the first blind move is about **16 seconds – a quarter of the ideal DESI repositioning time**. This movement method leaves plenty of time for the FVC feedback adjustments. Now that we have a complete characterization of this movement method, I will describe how my implementation resolves the collisions that occur in the extension phase of RRE.

5.2 Collision Avoidance Algorithm

The software implementation of collision avoidance is difficult to generalize so that it functions for any type of collision. As such, I used a Test-Driven Development (TDD) scheme when constructing my collision avoidance algorithm. Using the TDD model, I first run the RRE movement method and look at positioners that have collided, and I save all the parameters of a collision. Then, I take a **test case** from those collisions and write up code to resolve that one collision. Following that, I take another test case and apply my code, adding any fixes that are required to resolve that collision (while making sure the code still works for the previous test cases). As I check against more and more test cases, the algorithm becomes more and more general. However with a TDD scheme, one must implement a fail-safe once the algorithm is deployed. This is because there could be a very complicated collision that the test cases did not capture, and thus the algorithm cannot resolve that collision. In this case, my algorithm simply tells the positioner to not execute the move that would cause the collision - I tell the positioner to not try to go to its target, throwing that positioner

out. Yet with only 3% of positioners colliding initially, if I can resolve at least most of the collisions, we are still well within the DESI science goals. Furthermore, with around 66% of the collisions being theta-phi collisions, and these being very easy to resolve, resolving most of the collisions should not be an issue once enough test cases are passed.

5.2.1 Theta-to-Phi Collision Type

A theta-to-phi collision is when the theta body of one positioner is hit by the phi arm of another positioner - as seen in Figure 5.2. Since the theta bodies of both positioners are static - recall that these collisions occur during the extension move, where only ϕ is moving - the phi arm must get past the neighbor's theta body is some manner. The phi arm cannot rotate the other way due to hard stops, so the easiest way to resolve a theta-to-phi collision is to insert a collision avoidance step between the Rotate and Extend move in RRE Method.

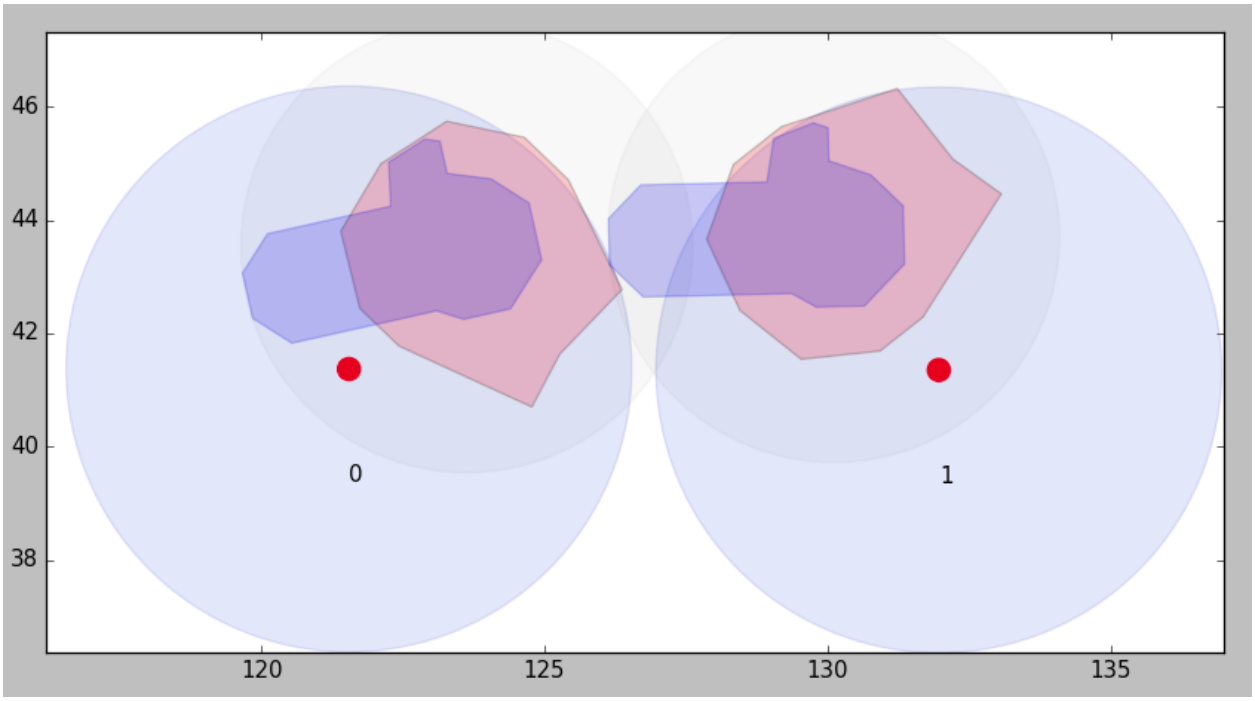


Figure 5.2: Example of Theta-to-Phi Collision Type: Theta body (red) of Positioner 0 is hit by the Phi arm (dark blue) of Positioner 1 during a retraction (phi arm pulling into envelope). Although this is technically a **retraction** collision, the same situation can follow from an extension move.

This **preparatory step** must be executed **before** the extension of the phi arm is executed. Looking at Figure 5.2, there is no way that the phi arm can retract inside of its envelope (or by the same logic, extend out of its envelope) while the theta body of Positioner 0 is in its way. As such, the theta body of Positioner 0 must move out of the way before the phi arm of Positioner 1 can move - this is the preparatory step, or the collision-avoidance step - that must be added into the RRE sequence.

The question that the collision avoidance algorithm must answer then becomes: how does the theta body of positioner 0 need to move to resolve this collision? And the answer is very simple: the theta body of positioner 0 must exit any overlap it has with patrol disk of the neighboring phi arm. In Figure 5.2, the light grey circles centered on the phi arm are the patrol disks of the phi arms, identifying the range of motion of the phi arms. In this Figure, there is a slight overlap between the theta body of 0 and the phi patrol disk of positioner 1. So to resolve the collision, positioner 0 just needs to adjust its theta body by some $\Delta\theta$ until it exits the grey circle of positioner 1! **However**, the algorithm must be careful in telling which way the theta body must move.

To resolve a theta-to-phi collision we will move the theta body out of the way, however there are two directions available, clockwise or counterclockwise. Remember, that these collisions must be resolved during an extension move. So to resolve a collision, instead of actually adding a preparatory step between the rotation of the theta bodies and the extension of the phi arms, we will just adjust the move tables during the theta rotation step. Instead of sending the theta body of positioner 0 to its final target location, we will **stop the rotation prematurely** or **overshoot the theta target** - whatever is most efficient. So the software structure will function as follows:

1. Run RRE movement and build a move table, assuming we let positioners collide
2. Select the positioners that will collide and isolate the theta-to-phi collisions
3. Select a pair of positioners where the phi arm of Positioner A hits the theta body of Positioner B
4. Run collision avoidance algorithm on the pair of positioners that collide. This determines what $\Delta\theta$ adjustment will resolve the collision for Positioner B.
5. Update move table such that Positioner B either overshoots its original target location or stops short of its original target location by $\Delta\theta$. This means that there is no current preparatory step added between the Rotate-Extend phase of RRE movement - there are still only 3 steps to the movement.

Stopping or overshooting the theta body will resolve the theta-to-phi collision, however, it means that the positioner did not reach its true final target location - it is off by $\Delta\theta$. To correct this, we will utilize the pre-pause functionality of the positioner move table. Allow the phi arm of Positioner A to be in collision with the theta body of Positioner B. The collision is avoided while stopping or overshooting the theta body of Positioner B during the rotation phase. Now during the extension phase, we only allow the phi arm of Positioner A to move - **we do not allow the phi arm of Positioner B to extend out**. While the

phi arm of Positioner A extends, we will allow Positioner B to reach its final theta location, traversing the $\Delta\theta$ it either overshot or stopped early of reaching. However, this too will cause a collision, unless we allow the phi arm of Positioner A to pass before we move the theta body of Positioner B. In short, we tell Positioner B it will travel $\Delta\theta$ back to it's target location, but it will **pause** before it begins moving, to allow Positioner A to extend its first phi arm, so there is no collision.

Yet this still means that Positioner B has not reached the complete target location. Since we changed the move table for Positioner B to do a rotation while all of the other positioners are extending the phi arms, Positioner B still must execute its extension move to the final ϕ position. Now we take advantage of simultaneity in the move table: instead of adding an additional step at the end of the move table, we will just tell Positioner B to **simultaneously** extend its phi arm to the final ϕ location, and rotate its theta body by $\Delta\theta$ to the final θ location, but only after it has executed its pre-pause such that the phi arm of Positioner A can reach its target too. And so, **the theta-to-phi collision is resolved, without adding any additional steps into the RRE movement method, but by just adjusting the move table!**

However we must keep in mind that once we add this adjustment to the move table, we must rerun a collision check as this simultaneous rotation/extension will cause its own collision! As of now, if another collision were to occur, my algorithm will simple tell Positioner B to not undergo that extension, and in its final move, it will **only** rotate by $\Delta\theta$ after the pre-pause. In other words, to avoid trying to resolve this secondary collision, we will just tell Positioner B to not execute the motion that would cause the secondary collision. But we will keep all of the previous updates to the move table we just made, because those updates resolved the primary collision we were after. Only during the extension move of Positioner B could another collision occur, and if it does, we simply tell Positioner B not to extend.

Our motivation for not resolving the secondary collision is mostly due to the probability of these events even happening. There are only about 3% of the positioners who collide in the first place and 66% of those collisions are theta-to-phi collisions. Most of these collisions can be resolved with the approach described above, and if another secondary collision were to occur, at least one of the two positioners in the initial collision will be reach the target location. Even 3% is good enough to throw away all positioner targets and still meet DESI science requirements, and so not resolving secondary collisions is well within our limits. The second reason to not resolve these secondary collisions is because additional steps in the move table would likely have to be added, which means more time for a repositioning. The benefit-to-cost ratio trying to resolve secondary collisions is just too low to rationalize.

As a final note, the decision on whether to overshoot or undershoot the theta rotation must be made. Generally, it does not truly matter since the $\Delta\theta$ difference will not affect the time to execute the entire step because, statistically, other positioners will have to travel longer anyway. In any case, my algorithm will choose to generally undershoot the theta rotation, just in case that $\Delta\theta$ does cause it to move the longest out of all theta rotations, and will save overall time. Yet, in some cases there is not an ambiguity to choose from. If the $\Delta\theta$

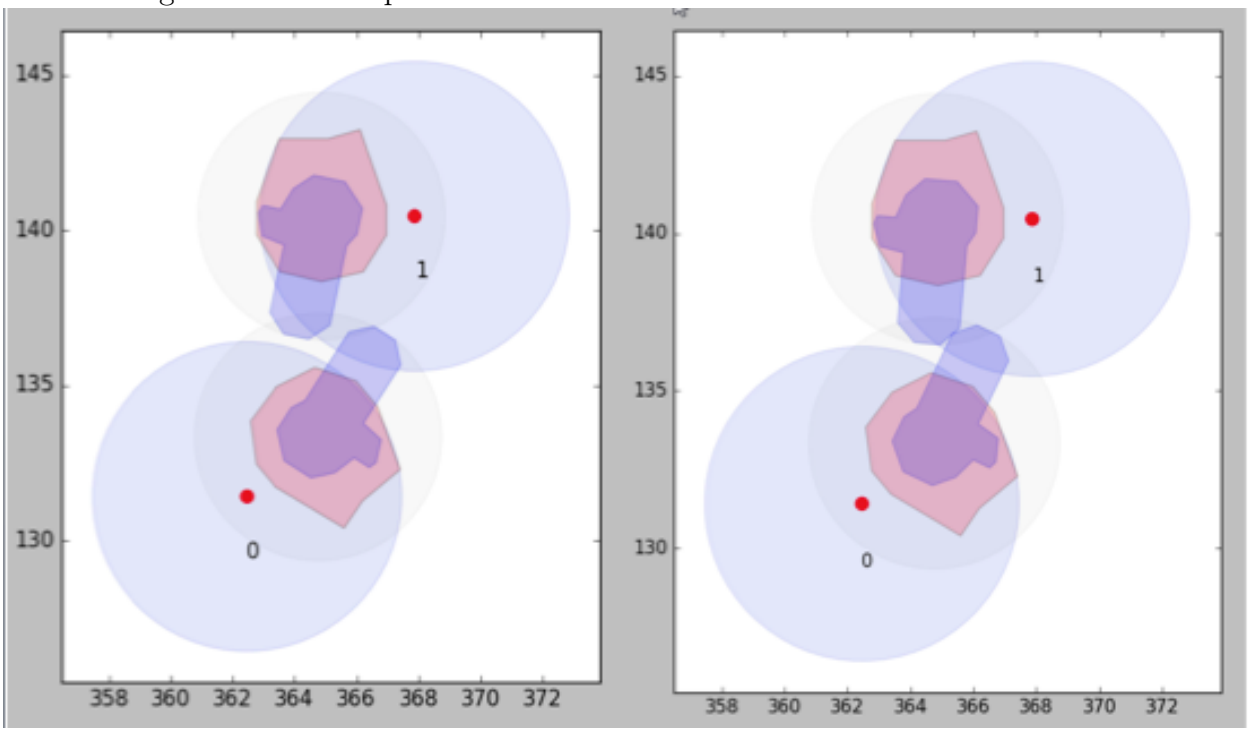
overshoot would cause a violation of the hard-stop, then we must undershoot - this is also why my implementation will always choose an undershoot.

To summarize, the theta-to-phi collisions are resolved by the following collision avoidance implementation:

1. Move positioners following the RRE method, create the move table while assuming collisions will occur
2. Identify and select a single theta-to-phi collision where Positioner A phi is hitting Positioner B theta
3. Determine $\Delta\theta$ and pre-pause, τ_1, for Positioner B such that the collision is resolved
4. Update the move table: (i) During rotation phase, undershoot θ_f by $\Delta\theta$ for Positioner B, (ii) During extension phase, set a pre-pause τ_1 for Positioner B to pause the positioner before moving
5. Run collision check to see if simultaneous movement of θ,ϕ towards final target locations θ_f,ϕ_f for Positioner B causes another secondary collision
6. If secondary collision does not occur: Update move table to allow simultaneous movement of θ,ϕ towards final target locations θ_f,ϕ_f for Positioner B after the pre-pause has executed. If secondary collision occurs: Update move table to keep Positioner B static after pre-pause has finished.
7. Repeat 3-6 until all theta-to-phi collisions are resolved
8. Failsafe: Simulate the final move table and recheck for theta-to-phi collisions. In case the TDD scheme for building out this collision avoidance algorithm did not catch a more complex case, do not allow any movement of Positioner A or B and save the collision parameters for troubleshooting.

This collision resolution ensures that the RRE movement method stays as a 3-step movement method (no additional steps have been added for collision avoidance, only adjustments to the move table). **Furthermore, this algorithm for collision avoidance will maintain science requirements of DESI** by rarely throwing out a positioner due to complex collisions. This is only possible because of how much RRE exploits from the geometry of the positioners and the safety envelopes. **Thus far, the RRE method and my collision avoidance implementation maintains a repositioning time of about 10 seconds** (well below the ideal repositioning time set forth by DESI), **while having close to 99% of all positioners reach their targets** (3% will collide, but of those, 66% are theta-to-phi collisions that most often can be solved).

Figure 5.3: Example of Phi-to-Phi Collision Type: Both Positioner 1 and 0 have phi arms attempting to retract towards blue safety envelope, however they are in the way of each other, and so a phi-to-phi collision occurs. Although this is technically a retraction collision, the same logic follows for a possible extension collision.



5.2.2 Phi-to-Phi Collision Type

The other type of collision that can occur is a phi-to-phi collision, when the phi arm of one positioner hits the phi arm of another positioner - as seen in Figure 5.3.

In phi-to-phi collisions, there isn't a theta body that needs to be moved out of the way, but instead, both phi arms run into each other. In this case, the easiest way to resolve this type of collision is to utilize the pre-pause again. If we tell one of the phi arms to wait before moving for a certain amount of time, this allows the other phi arm to clear the collision, and then the waiting arm starts moving, and the collision is avoided! Figure 5.4 demonstrates this pre-pausing solution to the collision, using the same collision that occurred in Figure 5.3.

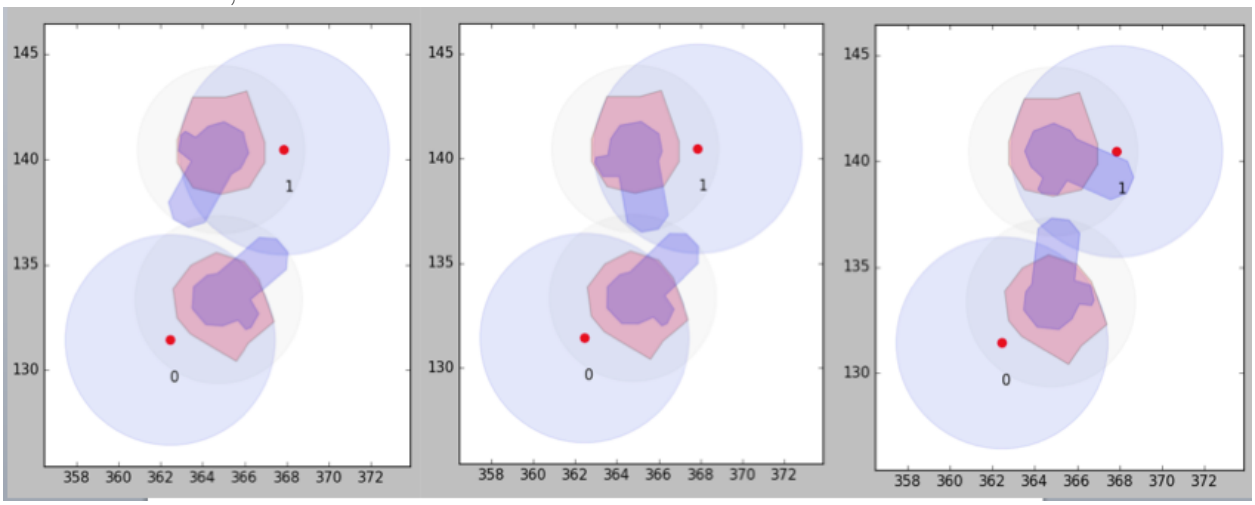
In this scheme for the collision resolution, there is a positioner that yields (pauses before moving) and a positioner that has the right-of-way (moves right away). We classify the pair of positioners based on the following rule: during an extension, the positioner that has the smallest final phi value (the one that extends out the most) will have the right-of-way and the positioner with the largest final phi value will have to pause. During a retraction, that logic is simply reversed. This is to ensure that the two positioners can reach their targets without one stopped in the other's way. As soon as we assign that the longer displacement positioner must yield and pause, my collision algorithm tries different values of τ_1 until the collision is

resolved. First, we try to pause the yielder by 0.05 seconds before it begins motion. If that still causes a phi-to-phi collision, then we increase τ_1 until the collision is resolved.

This simple pre-pausing method will resolve about 74% of the phi-to-phi collisions, but not all of them because there is a complication that often arises. It usually happens that the two positioners are close enough that although a phi-to-phi collision occurred first, a theta-to-phi collision would likely happen if the phi arm wasn't in the way. Essentially, the two positoners are too close that the phi arm cannot extend all the way out, like in a normal theta-to-phi collision, but before that happened, the theta body's phi arm just happened to be in the way first. These collisions can be identified by seeing the phi arm patrol disk (the light grey circles) will overlap the neighbor phi arm and the neighbor theta body.

26% of the time, a primary phi-to-phi collision will cause a secondary theta-to-phi collision. Although this is a very small number of positioners in a secondary collision - recall that only about 33% of collisions are primary phi-to-phi - the solution to these follows the previous section. We can also understand why secondary collisions are much more likely with phi-to-phi collisions: positioners must be fairly close for a phi-to-phi collision to happen, and if one of the phi arms weren't there, it would likely be a theta-to-phi collision anyway. Secondary collisions after a theta-to-phi collision rarely occur because having a phi arm in the way is much less likely than a theta body.

Figure 5.4: Example of Phi-to-Phi Collision Resolution with Pre-Pausing: Positioner 1 and 0 need to retract their phi arms inside of the safety envelope, but will run into each other. We tell positioner 0 to pause its movement for τ_1 so that Positioner 1 can retract inside without collision, and now Positioner 0 can retract inside.



In any case, my algorithm must resolve the secondary theta-to-phi collisions that arise after primary phi-to-phi collisions. To do so, another preparatory step must be made to resolve this collision. Since this is a theta-to-phi collision, the same process outlined in the previous section is followed with some modifications. If a phi-to-phi collision can be resolved with a pre-pause, then we only do that. If there must be another theta body adjustment due to secondary collisions, we follow the previous section's rules. We can do so by redefining $\Delta\theta$ as the displacement needed for the theta body and phi arm of Positioner A (who must yield) to clear the phi patrol disk of Positioner B. If there are any tertiary collisions caused, we simply do not execute the last simultaneous move of Positioner A as before.

1. Move positioners following the RRE method, create the move table while assuming collisions will occur
2. Identify and select a single phi-to-phi collision where Positioner A phi is hitting Positioner B phi
3. Determine that Positioner A will yield while Positioner B will begin motion
4. Determine pre-pause, τ_1, for Positioner A such that phi-to-phi collision is resolved
5. Keep running movement to check if a theta-to-phi collision will occur. If not, update the move table such that during the extension phase, set a pre-pause τ_1 for Positioner A to pause before the positioner begins moving; then iterate to the next pair in collision. If there is a secondary theta-to-phi collision, do not update move table but instead follow steps 6 through 10:
6. Identify that, in this case, the phi arm of Positioner A to hit the theta body of Positioner B.
7. Determine $\Delta\theta$ and pre-pause, τ_1, for positioner B such that collision is resolved
8. Update move table: (i) During rotation phase, undershoot θ_f by $\Delta\theta$ for Positioner B, (ii) During extension phase, set a pre-pause τ_1 for Positioner B to pause the positioner before moving
9. Run collision check to see if simultaneous movement of θ,ϕ towards final target locations θ_f,ϕ_f for Positioner B causes a tertiary collision.
10. If tertiary collision does not occur: Update move table to allow simultaneous movement of θ,ϕ towards final target locations θ_f,ϕ_f for Positioner B after the pre-pause has finished. If secondary collision occurs: Update move table to keep Positioner B static after pre-pause has finished. Iterate to next phi-to-phi collision pair
11. Failsafe: Run the final move table and recheck for any collisions. In case the TDD scheme for building out this collision avoidance algorithm did not catch a more complex case, do not allow any movement of Positioner A or B and save the collision parameters for troubleshooting.

Again, **this collision resolution ensures that the RRE movement method stays as a 3-step movement method and will maintain science requirements put forth by DESI. It repositions the focal plate in about 10 seconds (if we do not creep during intermediate steps) and has close to 99% of positioners reach their targets.** As a final note, I previously stated that RRE method has up to 5 steps - 3 raw steps and 2 for

collision avoidance, and the total time would be increased to 16 seconds. To resolve primary and (some) secondary collisions, I masked collision avoidance steps to overlap into the RRE 3-step move table. However I will maintain that to resolve **all** collisions, then 2 additional steps would have to be added to the RRE method. Although I have not implemented these steps into my collision avoidance algorithm, for the collision avoidance to be complete, it is likely mores steps would have to be added to the move table. I currently do not execute these additional steps because 99% of positioners reaching targets is well within limits for DESI science requirements and saves overall repositioning time.

5.2.3 Reversed Extensions as Retractions

To complete the collision avoidance algorithm, a word must be said on how we approach the retraction phase of the RRE method. A retraction is readily collision free by reversing the previous extension schedule. This can be easily done for all exposures, except for the very first exposure when there is no previous extension schedule. To resolve this, I have slightly adapted collision avoidance software for retraction collisions that follow the similar steps as before except now the $\Delta\theta$ will not be masked into the previous rotation move (there is no previous rotation move). Now, this is truly an additional step that must be added to the move table, before any positioners can move, so our collision avoidance scheme resembles the following:

1. Move positioners following the RRE method, create the move table while assuming collisions will occur
2. Identify collision types that are theta-to-phi and select one pair
3. Determine the $\Delta\theta$ needed to resolve the collision and update the move table such that there is a step before retraction that allows the positioner to move a $\Delta\theta$ while its neighbor is static during this adjustment move. No pre-pause is now necessary and the retraction phase can proceed freely. No secondary collisions can occur now either.
4. Identify collision types that are phi-to-phi and select one pair
5. Determine τ_1 pre-pause needed to resolve the phi-to-phi collision. If no secondary theta-to-phi collision occurs, update the move table such that **during the retraction phase** this positioner has a pre-pauses before retraction. If a secondary theta-to-phi occurs, follow step 3, adding a new step before retraction occurs. No tertiary collisions can occur now either.

Thus, the first retraction move will have four steps instead of the three steps as before, and will take an average of 12.6 seconds to reposition - a bit more time than following exposures. However now, all 100% of the positioners will reach their inner safety envelope without collision. Previously, if a positioner could not make it to the final target location, we left it in the safety envelope so it can at least try to make it to the next target location. Thus, we must ensure - and we do - that all positioners at least make it the their safety envelopes.

5.3 Summary of Performance

The RRE method, proposed by Joe Silber at LBNL, was implemented and fully equipped with collision avoidance software by Efrain Segarra. My software ensures that 99% of positioners reach new target locations within 10 seconds after a previous exposure has finished. If

one wishes 100% of positioners reach target locations, an additional 2 moves must be added to allow the solution of complex collisions. My implementation does not follow this due to the cost-benefit ratio when balancing the overall time to reposition. My collision avoidance software effectively masks the avoidance steps within the three-step movement method itself, and exceeds DESI science requirements on the focal plane. Theta-to-phi collisions and phi-to-phi collisions are almost always completely resolved. Since my collision avoidance software has a TDD approach, I also implemented a failsafe in case there is a collision that was not resolved, and in this case, positioners are left within their safety envelopes to readily reach for the next target location. For data analysis, my software simulation currently tracks power usage of the 5,000 positioners, collects movement time data for each positioner, and exports all collision parameter data (for troubleshooting and for statistical analysis).

Chapter 6

Next Steps

It is clear from software simulation that the RRE method is a superior movement method for the DESI positioners. Currently, the DESI team at UM is planning to adopt my collision avoidance algorithm and my implementation of Joe Silber's RRE movement method for DESI deployment. First however, real testing must be done on physical positioners, not just code simulation. In the next few weeks, the team at UM will have completed enough final-design positioners to deploy them in a hexagonal pod for collision avoidance testing. At that point, the code may be adapted or changed to resolve any issues. As the timeline for DESI progresses, so will my collision avoidance algorithm adapt and receive input from future DESI scientists.

Bibliography

[1] Anderson et al., "The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: Baryon Acoustic Oscillations in the Data Release 10 and 11 galaxy samples." In: arXiv: 1312.4877 [astro-ph.CO] (2013).

[2] "DESI Conceptual Design Report", August 26, 2014.
http://desi.lbl.gov/wp-content/uploads/2014/04/DESI_CDR_20140827_1135.pdf

[3] Jensen, Hannes et al., "Probing reionization with LOFAR using 21-cm redshift space distortions." In: Mon. Not. Roy. Astron. Soc. 435 (2013) 460, arXiv:1303.5627 [astro-ph.CO].

[4] Makarem et al., "Collision-free motion planning for fiber positioner robots: discretization of velocity profiles". In: Astronomy & Astrophysics 566, A84 (2014), arXiv:1410.1181 [astro-ph.CO].

[5] Silber, Joe. "Fiber Positioner Kinematics and Collision Envelopes", Ernest Orlando Lawrence Berkeley National Laboratory.