LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Final Report	**Caltech LIGO SURF**	2025/11/04

Filter Cavity Angular Control and Sensing System for Green Auxiliary Beam

Tomris Ilkim Dogan

Co-Mentors
Adam Mullavey
Begum Kabagoz

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW22-295
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

Contents

1 Introduction

The Laser Interferometer Gravitational-Wave Observatory (LIGO) is the most advanced gravitational wave detector in operation. It measures extremely small changes in spacetime caused by gravitational waves. Quantum noise is one of the main limiting factors for LIGO sensitivity. It has two main components: shot noise at high frequencies and radiation pressure noise at low frequencies.

To reduce quantum noise, LIGO uses squeezed light. In O3 observing run, frequency-independent squeezing was applied. This reduced shot noise but increased radiation pressure noise. To overcome this trade-off, frequency-dependent squeezing is employed using a filter cavity.

For stable squeezing performance, the filter cavity is controlled with an infrared auxiliary field in length, in position and the filter cavity must remain well aligned for stable operation. It is first Pound-Drever-Hall (PDH) [1] locked with an auxiliary green beam at 532 nm. Since the green beam has lower finesse than the infrared (IR) main beam, it is easier to lock. After this initial acquisition, control is transferred to the IR main beam for frequency-dependent squeezing. This transition is delicate and may fail in the presence of misalignments or a large differential phase noise.

This project aims to improve the reliability of this process by developing an automatic alignment scheme for the green beam. The approach is based on a dither-based angular alignment control method that reuses the existing filter cavity alignment sensing control (ASC) infrastructure.

2 Background

2.1 Quantum Noise in LIGO

Quantum noise in LIGO comes from the fundamental fluctuations of light due to the Heisenberg uncertainty principle, limiting the detector's sensitivity. At low frequencies, it shows up as radiation pressure noise because the photon number pushing the mirror is fluctuating. At high frequencies, it shows up as shot noise, which comes from random photon arrival times. In LIGO, quantum noise is one of the most significant limitations, and squeezing is used to reduce this noise [2].

2.2 Squeezed Light

Frequency-independent squeezing was employed during the O3 observing run. While it did reduce the shot-noise at higher frequencies, it came at the cost of higher radiation pressure at low frequencies, and limited the level of squeezed light that could be injected. Frequency-dependent squeezing solves this problem by rotating the squeezing ellipse in a frequency-dependent manner. It decreases both shot noise and radiation pressure noise, reducing quantum noise across the detection band.

2.3 Filter Cavity

To achieve the rotation, the squeezed vacuum field is reflected off a detuned optical resonator, known as the filter cavity. The cavity's finesse and detuning determine the rotation bandwidth and thus the frequency dependence of the squeezing phase. In LIGO, the filter cavity operates at a detuning of approximately 35 Hz from resonance with a linewidth of about 74 Hz [3]. This linewidth is achieved by choosing the cavity length as 300 m, according to

$$\frac{\mathrm{FSR}}{\mathcal{F}} = \frac{c}{2L\mathcal{F}},$$

where FSR is the free spectral range, $\mathcal{F}$ the finesse, c the speed of light, and L the cavity length.

The cavity mirrors are suspended via triple pendulums and mounted on seismically isolated platforms to suppress environmental disturbances. While these parameters are chosen to reduce quantum noise across the detection band, the high finesse makes the filter cavity particularly sensitive to angular misalignment and length fluctuations [4].

2.4 Green Locking and IR Hand-off

Accurate alignment of the filter cavity is crucial for efficient coupling of the squeezed field and for maintaining stable lock. The initial lock is achieved using a 532 nm auxiliary green beam via the Pound–Drever–Hall technique. The green beam is generated via second-harmonic generation from the main laser. A nested feedback system adjusts both the green laser frequency and the position of the cavity mirrors to establish resonance. Once the green lock is achieved, control is transferred to the IR main beam to enable squeezing operation. During this transition, maintaining alignment between the incoming beam and the cavity mode is critical [3].

2.5 Suspension System in LIGO

Mirror suspension systems are essential in interferometric gravitational wave detectors. They hold the mirrors in place and reduce the effect of external disturbances, which allows precise gravitational wave measurements. The main role of these systems is to isolate the mirrors from seismic vibrations To achieve this, multiple suspension stages are used, each one targeting a different frequency range of noise.

The mirrors are hung by low-loss silica fibers to reduce thermal noise. With the help of the suspension systems, the mirrors behave like free test masses over a wide frequency band. Both passive isolation (pendulums, blades) and active isolation (sensors and actuators that apply corrections in real time) are used together.

The masses are suspended either in triple or quadruple pendulum systems. For a simple pendulum, the transfer function from horizontal ground motion to the pendulum mass decreases as $1/f^2$ above the pendulum resonance. For each extra suspension stage, it decreases faster, as $1/f^4, 1/f^6, \ldots$ [5].

Advanced detectors also need very precise mirror control and alignment, because even small deviations can degrade the signals. For this reason, feedback control systems are applied to keep the mirror positions stable.

2.6 Control Loops

Feedback control systems keep a system close to a chosen operating point, even when there are disturbances. They watch the output, compare it to the desired value, and use this difference (the error signal) to correct the system. In optical experiments, analog and linear loops with negative feedback are most common. They are used to stabilize things like position, frequency, or cavity length [6, 7].

A feedback loop has four main parts:

- **Plant (G):** the system that we want to control.
- **Sensor:** measures how far the plant is from the operating point and gives the error signal.
- **Servo (H):** a filter that changes the error signal into a useful correction.
- **Actuator:** applies the correction to the plant.

In many cases, sensor and actuator are counted as part of the plant, since their signals cannot be measured directly.

Any linear system can be described by a transfer function

$$G(\omega) = \frac{\mathrm{Out}(\omega)}{\mathrm{In}(\omega)}.$$

With feedback, if a disturbance x_N enters, the signals become

$$x_O(\omega) = \frac{x_N(\omega)}{1 - G(\omega)H(\omega)}, \quad x_{FB}(\omega) = \frac{G(\omega)H(\omega)}{1 - G(\omega)H(\omega)}\, x_N(\omega).$$

As explained by Freise (Appendix B, [6]), if $|GH| \gg 1$, the disturbance x_O is reduced by about $1/|GH|$. The factor $\frac{1}{1-GH}$ is called the closed-loop gain. Negative feedback means the correction has the opposite sign, which helps to stabilize the plant. Positive feedback does the opposite and usually leads to oscillations. For stability, when the open-loop gain is close to one, the phase must stay above $-360°$; otherwise the system can become unstable.

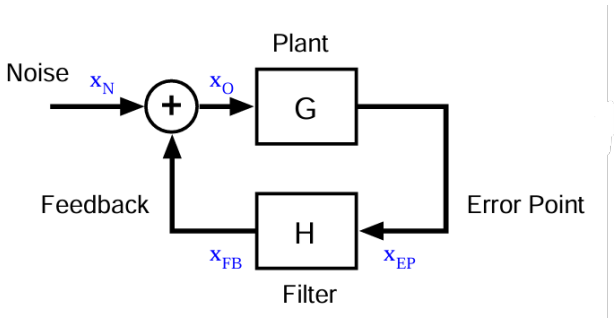

Figure 1: Basic closed-loop system

2.7 Initial Alignment Procedure

The filter cavity length control is primarily performed with the 1064 nm field, since this wavelength provides a better witness of the squeezed field and exhibits lower phase noise. Nevertheless, monitoring and controlling the cavity at 532 nm is highly useful for diagnostics, alignment, and for stabilizing the cavity during lock acquisition and commissioning. In this process, the 532 nm Filter Cavity Green Sensing (FCGS) field is combined with the squeezed field and sent into the cavity. To establish the initial alignment, a mode-matched 532 nm or 1064 nm beam must be prepared, and the FC1 optic must be placed in position. The ZM1–ZM3 mirrors are then adjusted so that the beam is centered on FC1, as verified from the transmitted beam profile of FC1. After centering, the ZM mirrors are tuned to maintain this condition while steering the beam down the 300 m cavity tube. Once the beam reaches HAM8, the retro-reflection from FC1 is observed. FC1 is then adjusted so that this retro-reflection overlaps with the incoming beam, thereby aligning FC1. Because FC1 introduces a strong AR lens effect, this step requires iteration: moving FC1 changes the beam deflection along the cavity tube, so multiple adjustments are needed. FC2 is aligned by tuning it such that the reflected beam is directed back toward ZM1 [8].

2.8 Ray Transfer Matrix

Ray transfer matrix analysis (also known as ABCD matrix analysis) is a mathematical method for ray tracing in systems where only paraxial rays are considered. Each optical element is represented by a 2×2 matrix that transforms the ray position and angle from an input plane (x_1, θ_1) to an output plane (x_2, θ_2). Multiplying the matrices of successive elements gives the overall matrix of the system, allowing calculation of focal planes, magnification, and other first-order optical properties under the paraxial approximation where (x_1, θ_1) and (x_2, θ_2) are the ray position and angle at the input and output planes [9].

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix},$$

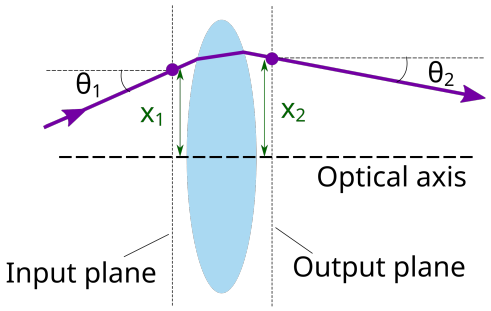

Figure 2: In ray transfer matrix analysis, an optical element (here, a thick lens) gives a transformation.

The optical paths from the ZM1 and ZM3 mirrors to FC1 were modeled using the ray transfer (ABCD) matrix method under the paraxial approximation. For each segment, the corresponding matrix was chosen according to the optical element:

- **Flat mirrors (ZM1, ZM3):** Represented by the identity matrix, since they do not change the ray's position or slope in the paraxial regime:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

- **Curved mirror (ZM2):** Modeled using

$$\mathbf{L}_{\text{ZM2}} = \begin{bmatrix} 1 & 0 \\ -\dfrac{2}{R_e} & 1 \end{bmatrix},$$

  where $R_e = R\cos\theta$ in the tangential plane (horizontal direction) and $R_e = R/\cos\theta$ in the sagittal plane (vertical direction). Here, R is the radius of curvature ($R > 0$ for concave) and θ is the incidence angle in the horizontal plane.

$$\mathbf{L}_{\text{ZM2}} = \begin{bmatrix} 1 & 0 \\ -\dfrac{2}{0.84} & 1 \end{bmatrix}.$$

- **Thick lens (FC1):**

$$\mathbf{L}_{\text{thick}} = \begin{bmatrix} 1 & 0 \\ \frac{n_2 - n_1}{R_2 n_1} & \frac{n_2}{n_1} \end{bmatrix} \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{n_1 - n_2}{R_1 n_2} & \frac{n_1}{n_2} \end{bmatrix},$$

  where n_1 is the refractive index outside the lens, n_2 inside the lens, R_1, R_2 are the radii of curvature of the first and second surfaces, and t is the center thickness.

For FC1, the parameters were:

$$n_1 = 1, \quad n_2 = 1.46, \quad R_1 = 1.0 \text{ m}, \quad R_2 \to \infty \text{ (flat surface)}, \quad t = 0.07 \text{ m}.$$

Substituting these into the formula:

$$\mathbf{L}_{\text{FC1}} = \begin{bmatrix} 1 & 0 \\ 0 & 1.46 \end{bmatrix} \begin{bmatrix} 1 & 0.07 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\dfrac{0.46}{1.46} & \dfrac{1}{1.46} \end{bmatrix} = \begin{bmatrix} 0.98 & 0.05 \\ -0.46 & 1 \end{bmatrix}.$$

- **Free-space propagation:** For each distance between elements, the translation matrix

$$\mathbf{S}(d) = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix},$$

was applied, where d is the separation along the optical axis.

$$\mathbf{S}_{\text{ZM3}\to\text{FC1}} = \begin{bmatrix} 1 & 0.99 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{S}_{\text{ZM2}\to\text{ZM3}} = \begin{bmatrix} 1 & 1.84 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{S}_{\text{ZM1}\to\text{ZM2}} = \begin{bmatrix} 1 & 1.5 \\ 0 & 1 \end{bmatrix}.$$

By multiplying the matrices in sequence, the total ray transfer matrices from ZM1 to FC1 and from ZM3 to FC1 were obtained. These final matrices describe how an input ray's position and slope transform along each optical path, allowing evaluation of beam direction and angle behavior at the FC1 input plane.

The resulting ray transfer matrices for the two optical paths are:

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM1}\to\text{FC1}} = \mathbf{L}_{\text{FC1}}\, \mathbf{S}_{\text{ZM3}\to\text{FC1}}\, \mathbf{S}_{\text{ZM2}\to\text{ZM3}}\, \mathbf{L}_{\text{ZM2}}\, \mathbf{S}_{\text{ZM1}\to\text{ZM2}} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix} \approx \begin{bmatrix} -5.72 & -5.80 \\ 0.25 & 0.08 \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix},$$

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM3}\to\text{FC1}} = \mathbf{L}_{\text{FC1}}\, \mathbf{S}_{\text{ZM3}\to\text{FC1}} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix} \approx \begin{bmatrix} 0.98 & 1.02 \\ -0.46 & 0.54 \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix},$$

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM1}\to\text{FC2}} = \mathbf{S}_{\text{FC1}\to\text{FC2}} \begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM1}\to\text{FC1}} \approx \begin{bmatrix} 70.1 & 19 \\ 0.25 & 0.08 \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix},$$

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM3}\to\text{FC2}} = \mathbf{S}_{\text{FC1}\to\text{FC2}} \begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM3}\to\text{FC1}} \approx \begin{bmatrix} -136 & 163.12 \\ -0.46 & 0.54 \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix}.$$

For pitch and yaw rotations, only angular deviation is applied; thus, x_1 is set to zero.

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM1}\to\text{FC1}} \approx \begin{bmatrix} -5.80\,\theta_1 \\ 0.08\,\theta_1 \end{bmatrix} \quad \begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM3}\to\text{FC1}} \approx \begin{bmatrix} 1.02\,\theta_1 \\ 0.54\,\theta_1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM1}\to\text{FC2}} \approx \begin{bmatrix} 19\,\theta_1 \\ 0.08\,\theta_1 \end{bmatrix} \quad \begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix}_{\text{ZM3}\to\text{FC2}} \approx \begin{bmatrix} 163.12\,\theta_1 \\ 0.54\,\theta_1 \end{bmatrix}$$

These results show the effect of a pure angular input on the beam position at the filter cavity input and output planes.

$$\begin{bmatrix} -5.80 & 19 \\ 1.02 & 163.12 \end{bmatrix} \begin{bmatrix} \theta_{\mathrm{ZM1}} \\ \theta_{\mathrm{ZM3}} \end{bmatrix} = \begin{bmatrix} x_{\mathrm{FC1}} \\ x_{\mathrm{FC2}} \end{bmatrix}$$

2.9 Dithering

Dithering is an angular excitation of a suspension of a defined frequency [10]. First, we apply a small dither to a mirror in pitch or yaw. Dithering in this context is intentionally inducing noise for a degree of freedom (DoF) (ZM1 pitch for example), to determine the noise coupling from that DoF to our signal of interest (filter cavity green transmission/reflection signal for example), and using the noise coupling to determine the "best" configuration for the DoF. Dither-based alignment techniques are widely and successfully used in LIGO control systems [10], space communications [11], fiber applications [12] and other areas.

3 Objective

3.1 Problem Statement

Since the filter cavity is a suspended cavity, we should consider the input alignment of the beam to the cavity, as well as the alignment of the cavity mirrors making up the optical axis of the cavity. There are already alignment control schemes in place for the IR locked configuration; however, the auxiliary green locking has no such alignment aid currently.

The purpose of the auxiliary green beam is to pre-lock the filter cavity length, then tune the length of the cavity to be sufficiently close to the IR resonance frequency, and keep the cavity locked while the IR beam is in the process of acquiring lock. If at any time during this process the green cavity loses lock, IR locking cannot be achieved. As reported in alogs [13, 14, 15, 16], cavity misalignment is a common reason for the green cavity to lose lock during the hand-off and as such, we propose making a dither alignment scheme for the green auxiliary beam.

3.2 Proposed Solution

The auxiliary green beam used for filter cavity (FC) locking is tapped off from the SHG (Second Harmonic Generation). The existing FC alignment sensing control (ASC) infrared beam path is used for this purpose. Before entering the filter cavity, the green beam is directed through mirrors ZM1, ZM2, and ZM3 as explained in 2.7.

In this configuration, the FC mirrors are responsible for controlling the cavity axis, while the ZM mirrors determine the input beam spot position with respect to the cavity. A total of two degrees of freedom (DoFs) were assigned to the FC mirrors and two to the ZM mirrors. Considering both pitch and yaw, this corresponds to a total of eight angular degrees of freedom in the control system.

In this project, two types of alignment control loops were implemented:

- **ZM loops:** These loops use the transmitted green signal from the filter cavity as the sensor input. The error signals extracted from this channel are processed through the filter banks, and the resulting control signals are applied to the electromagnetic actuators on the ZM mirrors. In this way, the ZM loops steer the input beam onto the filter cavity.
- **FC loops:** These loops use the filter cavity length control signal as the sensor input. After being processed in the filter banks, the corresponding control signals are applied to the electromagnetic actuators on the FC mirrors. The FC loops thus stabilize the cavity axis relative to the incoming beam.

Together, the ZM and FC loops provide cooperative control: the ZM mirrors adjust alignment of the input beam relative to the cavity axis, whereas the FC mirror angles will adjust the cavity axis relative to the center of the FC mirrors, thereby maintaining stable coupling of the green beam to the filter cavity.

We defined the control loops by assigning specific actuation channels to the relevant mirrors based on the front-end C code, compiled from Simulink model (see Figure 3).

First, the demodulated error signals obtained from the sensors are passed through an input matrix. The ZM and FC mirrors are dithered at some higher frequency and the transmitted power and cavity length are demodulated. The input matrix assigns each error signal to the corresponding controller. In this configuration, ZM1 pitch and yaw are driven through the `INJ_POS` channels, ZM3 pitch and yaw are controlled via the `INJ_ANG` channels, and FC1 pitch as well as FC2 pitch are controlled through the `CAV_POS` and `CAV_ANG` channels, respectively.

Subsequently, the signals are processed by the servo controller, which incorporates the filter bank. The controller outputs are then passed through an output matrix that maps them to the appropriate degrees of freedom. Finally, the signals are applied to the actuators corresponding to FC1 pitch, FC2 pitch, ZM1 pitch, and ZM3 pitch.

It should be noted that the figure presented here reflects only the pitch configuration. An equivalent version of the control loop exists for yaw, which follows the same structure.

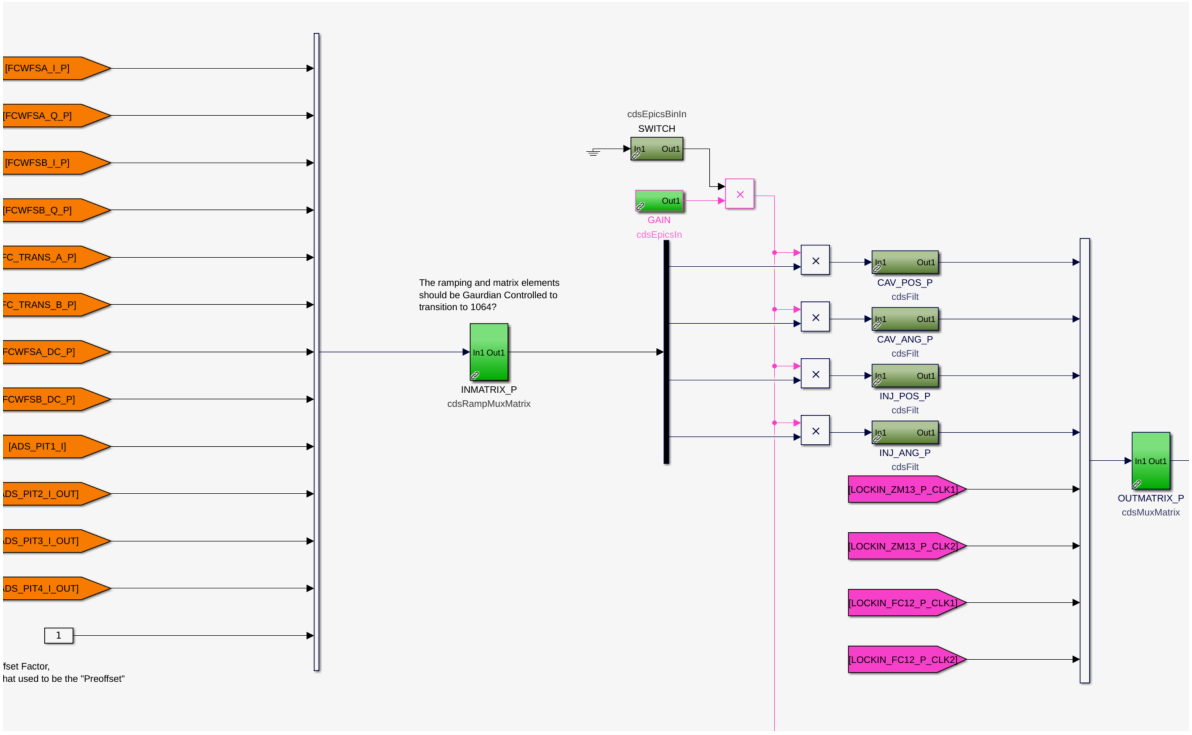

Figure 3: Simulink model of the filter cavity alignment sensing control.

4 Progress

4.1 Obtaining Transfer Functions of ZMs

The first step in the green beam auto alignment system was to select appropriate dither frequencies for each angular degree of freedom (pitch and yaw) of the filter cavity input path. To achieve this, we measured the actuator to sensor transfer functions of the ZM1 and ZM3 mirrors, specifically their bottom-stage (M2) actuators.

A swept-sine excitation was applied in the 5–2500 Hz range to the `L1:SUS-{ZM1, ZM3}_M2_DITHER_{P,Y}_OUT` channels. While exciting each degree of freedom (DoF) individually, we monitored the filter cavity's transmitted green power signal at `L1:SQZ-FC_TRANS_A_NSUM_OUTPUT`.

In order to enhance the sensitivity of the cavity transmission to each individual alignment DoF, we pre-misaligned the mirror under test in the corresponding DoF. This enhances the sensitivity of the transmitted signal to angular motion in the selected DoF, thereby improving the observability of the system response and allowing for reliable transfer function estimation.

The transfer function plot corresponding to ZM1 Pitch is shown below. It shows the magnitude and coherence of the response from angular drive to filter cavity (FC) green transmission.

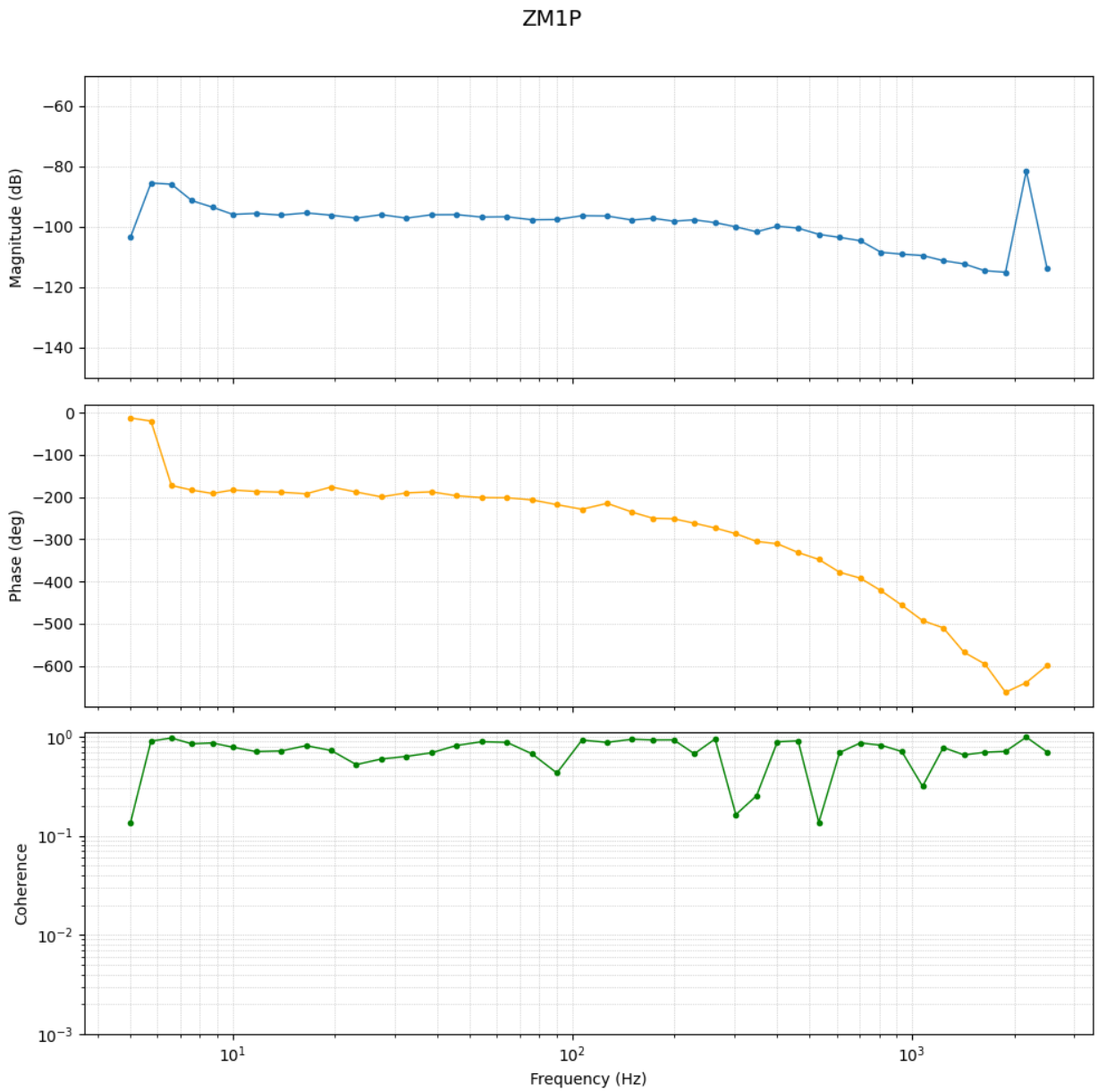

Figure 4: ZM1 Pitch

The observed responses are broadly flat in the frequency range of interest, consistent with prior actuator transfer function measurements performed at Hanford [17].

4.2 ZM Dither/Demodulation

In the next phase of the FC green beam angular alignment scheme, we defined the control loops by assigning specific actuation channels to the relevant mirrors. Based on the Simulink control model (see Figure 3), we used:

- `INJ_POS_P` and `INJ_POS_Y` actuators to control pitch and yaw of ZM1,
- `INJ_ANG_P` and `INJ_ANG_Y` actuators to control pitch and yaw of ZM3.

In all four cases, the alignment error signals are obtained by demodulating the `TRANS_A_SUM` channel at the applied dither frequencies. The resulting signals (e.g., ADS1, ADS2) correspond the response of the cavity transmission to angular misalignments, and serve as inputs to the alignment control loop.

To define how the sensed alignment signals are mapped into actuation commands, `SQZ-FC_ASC_INMATRIX` is constructed. This matrix multiplies the demodulated alignment sensing signals (e.g., ADS1, ADS2) and distributes the result to the available pitch and yaw control channels. The output of this matrix serves as the input to the servo filters, which process the signals before sending them to the corresponding angular actuators.

The demodulation parameters, including dither frequency, gain, and phase, were selected based on measured mirror transfer functions as shown in Part 1. :

DoF	**Frequency (Hz)**	**CLK Gain**	**Demod Phase (deg)**
ZM1 Pitch	101.1	2000	-40
ZM1 Yaw	126.1	2000	140
ZM3 Pitch	151.1	200	120
ZM3 Yaw	176.1	200	105

Although four filters were originally preloaded in the FC ASC control chain for the IR alignment path, we enabled only the FM9, whose bode plot is given in Figure 5 :

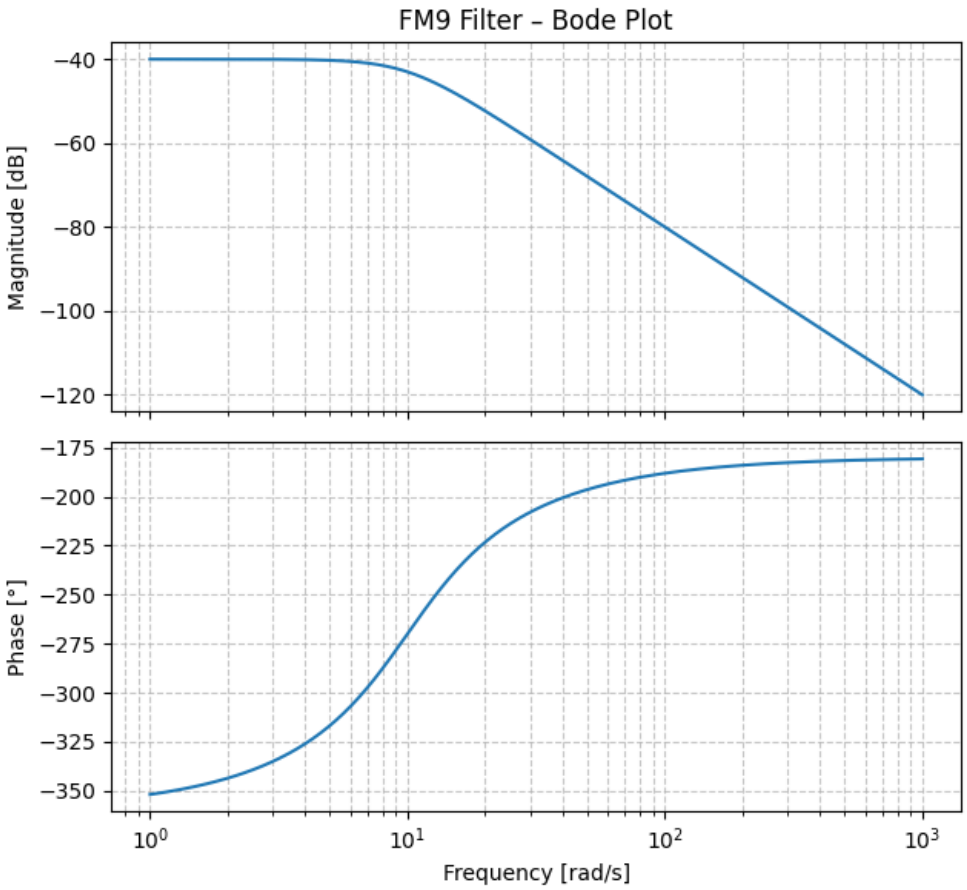

Figure 5: FM9 Bode Plot

After enabling the dithers and closing the loops, we observed a clear increase in the `TRANS_A_SUM` signal, confirming improved alignment.

The alignment loops are relatively slow on the order of several minutes. To improve speed and control bandwidth, we planned to implement inverted plant filters based on measured mirror responses.

4.3 Implementing Plant Inversion Filters for ZMs

As mentioned in Part 2, the alignment control loops were slower than desired. We suspected that this limited performance might be due to the pendulum dynamics of the ZM1 and ZM3 suspensions. To obtain a flatter overall response and improve control bandwidth, we decided to measure the plant transfer functions from the top stages (M1) of ZM1 and ZM3, and design corresponding inverse filters.

We began by generating expected models for pitch and yaw using `DoubleModel_Production tool` in Matlab* [18] (see Figure 6 and Figure 7)

* Located at
`/ligo/svncommon/SusSVN/sus/trunk/Common/MatlabTools/DoubleModel_Production`

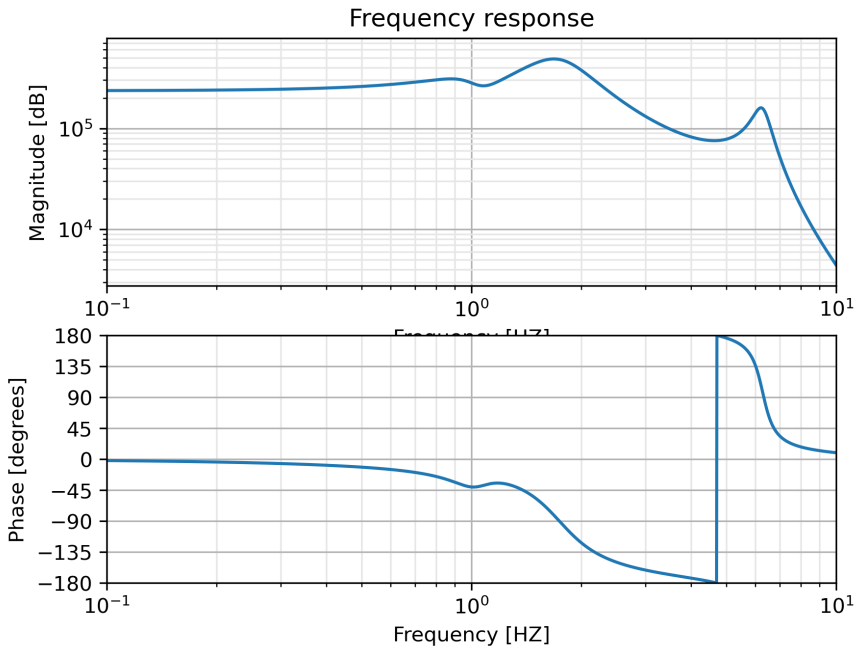

Figure 6: Expected transfer function model for pitch

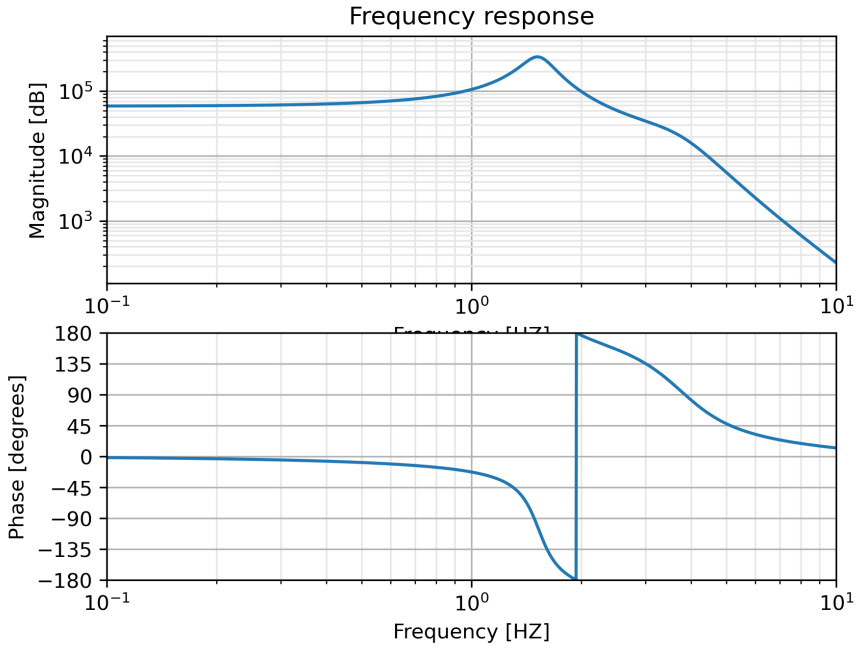

Figure 7: Expected transfer function model for yaw

These models served as a reference to verify the shape of our measured transfer functions. We then applied low-frequency swept sine excitations to the M1 stages from the `L1:SUS-{ZM1, ZM3}_M1_DITHER_{P,Y}_EXC` channel. We also measured the resulting angular response (see Figure 8). A similar technique had previously been applied at the M2 stage to characterize high-frequency behavior (see ALOG [19]).

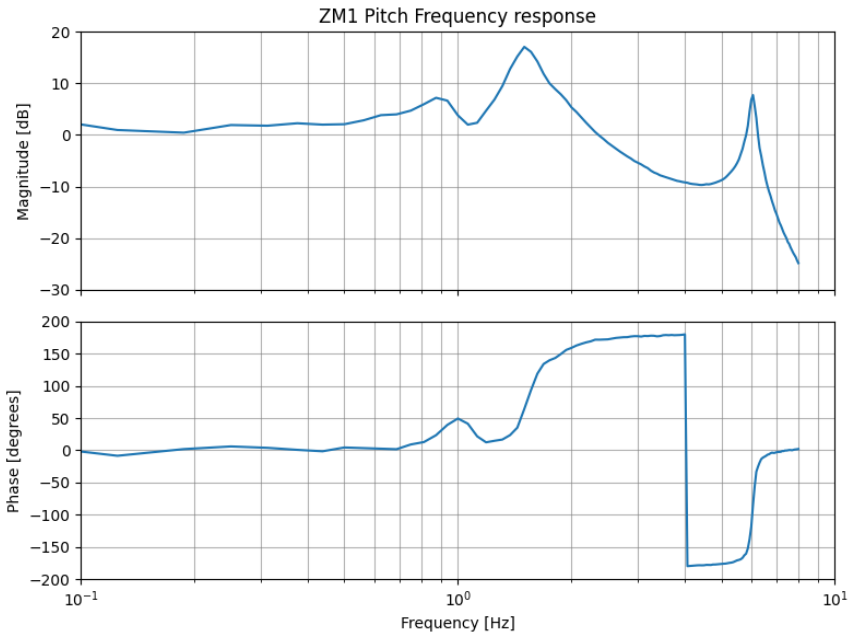

Figure 8: Measured transfer function for ZM1 Pitch

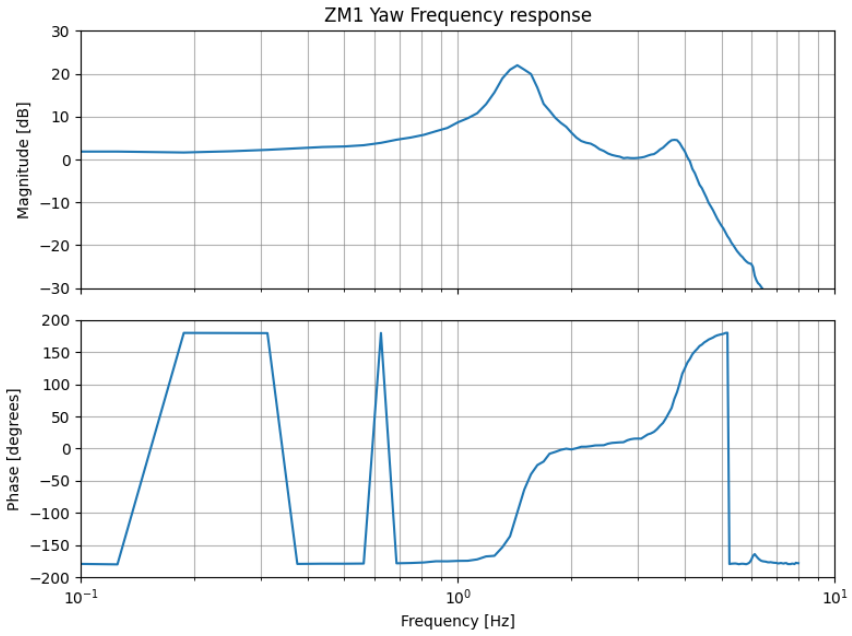

Figure 9: Measured transfer function for ZM1 Yaw

After obtaining the transfer function data, we used the chebyChef tool [20] to perform zero-pole-gain (ZPK) fitting. This tool provides an accurate representation of complex frequency responses. The raw transfer function data was extracted using the dttxml [21] utility. Once the ZPK models were obtained, we implemented their inverses to use as pre-filters in the alignment control loops. The resulting fits and parameters are shown in the accompanying

figure for ZM1 pitch (see Figure 10).

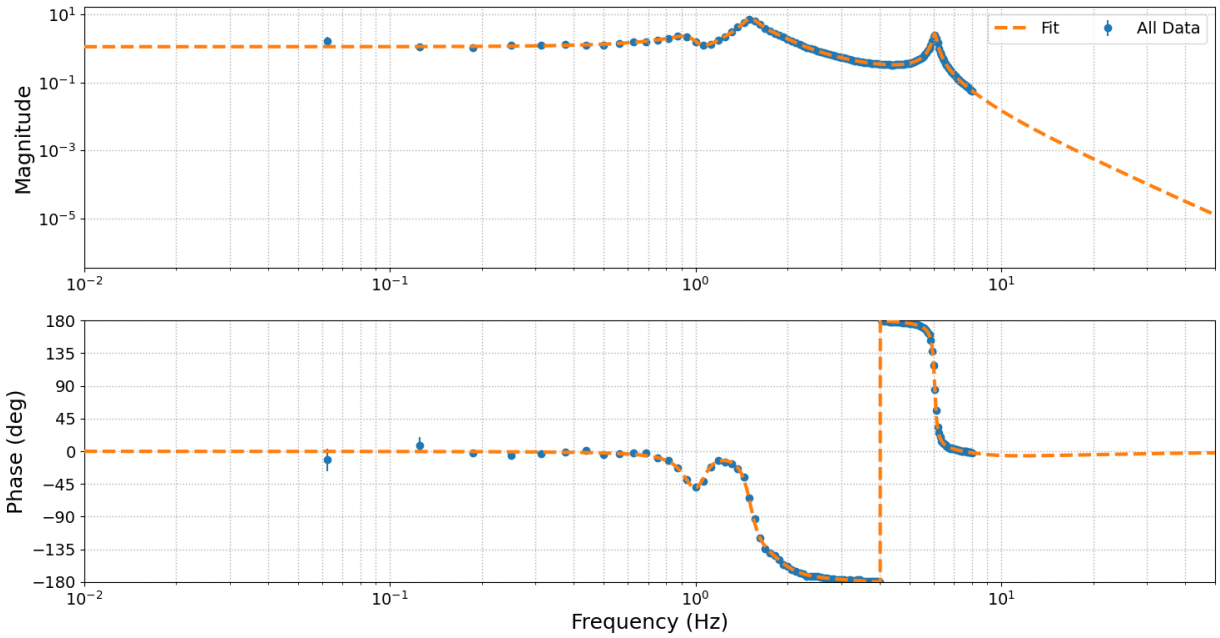

Figure 10: Measured and fitted transfer functions for ZM1 Pitch

Each loop was initially tested individually. After enabling the new Guardian state `GR_ALIGN`, we first misaligned ZM1 in pitch and closed only the `FC-ASC-INJ-POS-P` loop. Upon loop closure, we observed a clear increase in the `FC_TRANS_A_SUM_OUT` signal, indicating successful correction of misalignment. Simultaneously, the demodulated alignment error signals decayed toward zero as the system converged.

After validating the stability of each individual loop, all four alignment loops were then closed simultaneously. We subsequently increased the corresponding input matrix gains to improve performance further.

Following the application of the inverse filters, the alignment loops showed significantly faster convergence compared to previous measurements. However, we observed some oscillatory behavior in the error signals. The source of these oscillations is under investigation. The power spectral density (PSD) of the error signals is shown in Figure 11. As seen in the power spectral density plot, two unexpected peaks appear near 3.9 Hz and 6 Hz. These features are present in the yaw error signals, suggesting that the observed oscillations may originate from the yaw control loops.

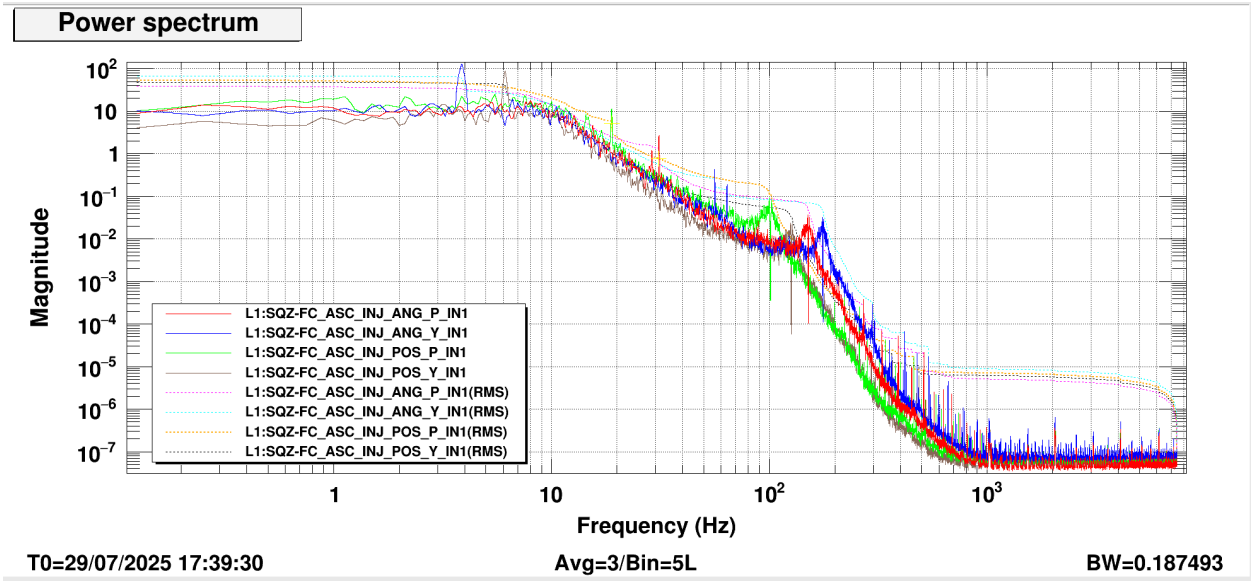

Figure 11: Power spectral density of the error signals

4.4 Determining Inmatrix Values for ZMs

After closing all ZM loops and confirming their performance for correcting ZM1/3, we decided to test their reaction to misaligned FC mirrors. However, we saw the ZM loops reacting too slowly to FC misalignments, so we decided to reevaluate the input matrix values for the ZMs.

We wanted to create a cavity-axis-based alignment system, so we constructed the degrees of freedom in that way. For this reason, we gave excitation to the controller outputs of the FC loops, looking at the ZM error signals (Figure 12). At each excitation frequency, we compared the responses of ZM1 and ZM3 in both phase and magnitude, and from these measurements, we constructed a coupling matrix. Then, we inverted this measured matrix and used the result as the input matrix.

Excitation channel: `L1:SQZ-FC_ASC_{CAV,POS}_{PIT,YAW}_OUT`

Comparison channels: `L1:SQZ-FC-ASC_ADS_{PIT,YAW}{1,2}_DEMOD_I_OUT`

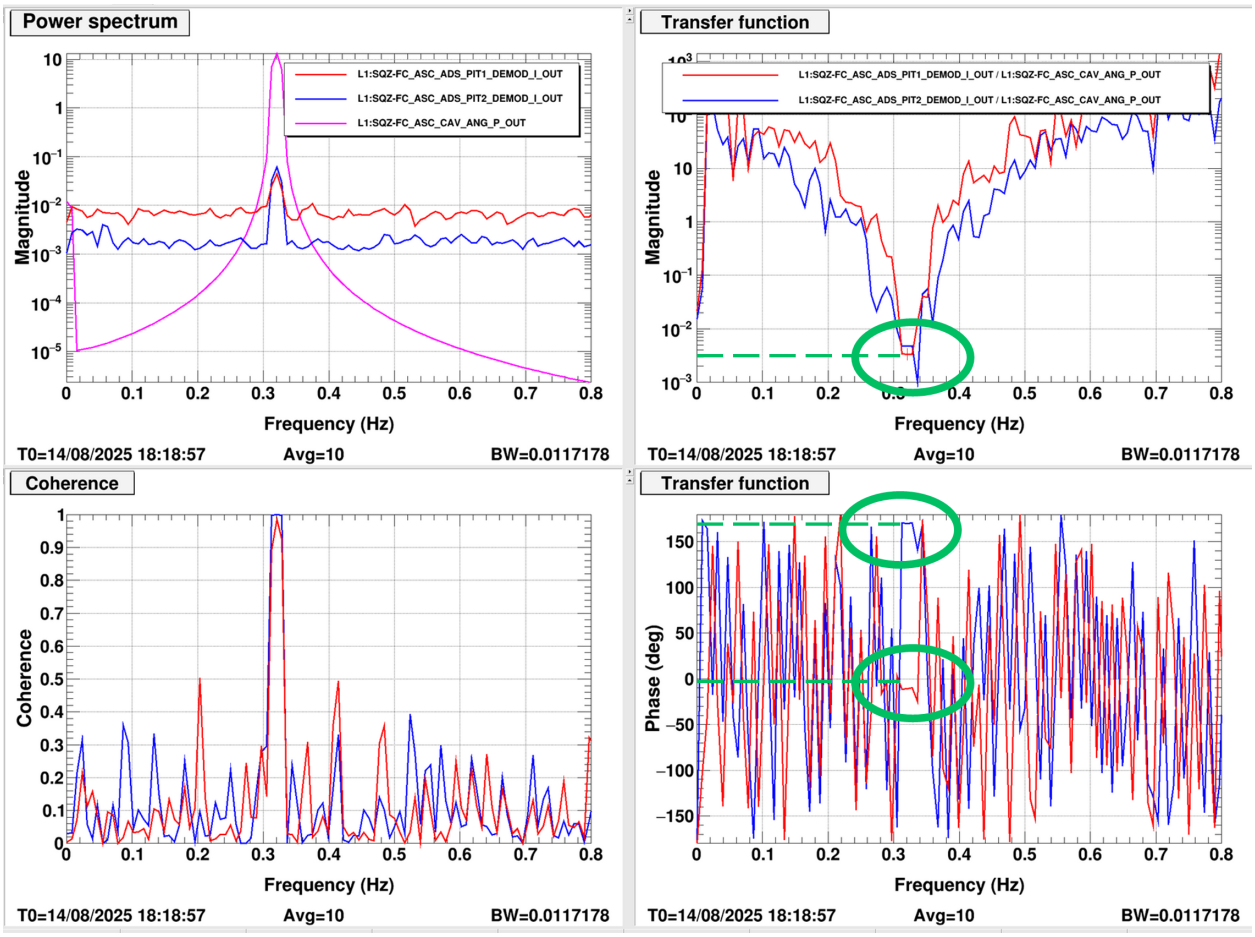

Figure 12: Measured transfer functions of ZM error signals in response to FC excitations.

$$\begin{bmatrix} \mathrm{ZM1}_{\mathrm{Pitch}} \\ \mathrm{ZM3}_{\mathrm{Pitch}} \end{bmatrix} = \begin{bmatrix} -0.004 & 0.003 \\ 0.003 & -0.004 \end{bmatrix} \begin{bmatrix} x_{\mathrm{FC1}} \\ x_{\mathrm{FC2}} \end{bmatrix} \Rightarrow \begin{bmatrix} -571.4 & -428.6 \\ -428.6 & -571.4 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{ZM1}_{\mathrm{Yaw}} \\ \mathrm{ZM3}_{\mathrm{Yaw}} \end{bmatrix} = \begin{bmatrix} 0.005 & -0.004 \\ 0.003 & -0.004 \end{bmatrix} \begin{bmatrix} x_{\mathrm{FC1}} \\ x_{\mathrm{FC2}} \end{bmatrix} \Rightarrow \begin{bmatrix} 500.0 & -500.0 \\ 375.0 & -625.0 \end{bmatrix}$$

4.5 Closing the FC Loops

After the arrangement with the ZM loops was completed, we moved on to the FC loops. The same dither frequencies as those used in the IR loops were applied, namely:

- PIT3: 10 Hz (FC1)
- PIT4: 12 Hz (FC2)
- YAW3: 6 Hz (FC1)

- YAW4: 8 Hz (FC2)

The control logic was similar to that of the ZM loops, however in this case we drove (dithered) the FC mirrors and demodulated the FC length signal. Angle-to-length coupling is naturally introduced if the beam is off-center on the mirror, or more accurately not co-located with the rotation point. As such, demodulating the length signal at the dither frequency provides a measurement of this angle-to-length coupling and an indirect estimate of how far the beam spot was displaced from the mirror center. Prior to closing the FC loops, gain factors were obtained by applying offsets and monitoring the differences in the demodulated signals. The corresponding gain filters were then implemented. All required gain filters were integrated into the filter banks on the servo side and included in the GUARDIAN code. Additionally, the servo consists of a Butterworth low-pass filter, an integrator, and a gain stage.

Finally, to determine the output matrix for the FC mirrors, the beam spot location matrix was calculated from the cavity geometry using the g-parameters.

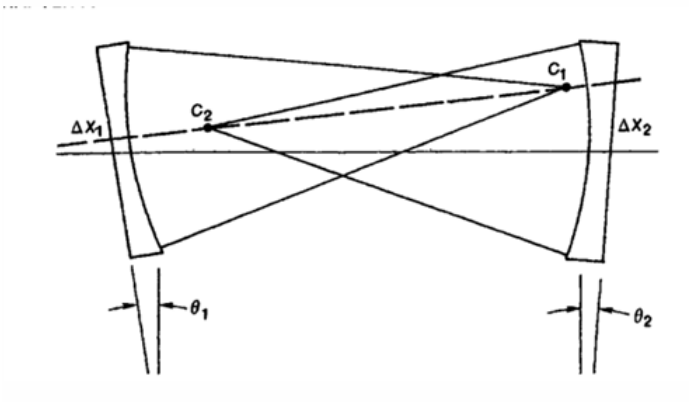

Figure 13: Beam spot positions

The relation between mirror tilt angles (θ_1, θ_2) and the corresponding beam spot displacements (x_1, x_2) on the mirrors can be expressed as [22]:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{L}{1 - g_1 g_2} \begin{bmatrix} g_2 & 1 \\ 1 & g_1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \tag{1}$$

with the stability parameters defined as

$$g_i = 1 - \frac{L}{R_i}, \qquad i = 1, 2, \tag{2}$$

where L is the cavity length and R_i are the mirror radii of curvature. In our case, the cavity length is $L = 300\,\mathrm{m}$, corresponding to the filter cavity length. The radii of curvature of the mirrors are $R_1 = \infty$ (flat mirror) and $R_2 = 531\,\mathrm{m}$. Accordingly, the stability parameters become

$$L = 300\,\mathrm{m}, \quad g_1 = 1, \quad g_2 = \tfrac{231}{531}$$

$$\begin{bmatrix} x_{\mathrm{FC1}} \\ x_{\mathrm{FC2}} \end{bmatrix} = \begin{bmatrix} 231 & 531 \\ 531 & 531 \end{bmatrix} \begin{bmatrix} \theta_{\mathrm{FC1}} \\ \theta_{\mathrm{FC2}} \end{bmatrix}$$

$$\begin{bmatrix} \theta_{\mathrm{FC1}} \\ \theta_{\mathrm{FC2}} \end{bmatrix} = \begin{bmatrix} -3 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} x_{\mathrm{FC1}} \\ x_{\mathrm{FC2}} \end{bmatrix} \quad \text{(Pitch)}$$

$$\begin{bmatrix} \theta_{\mathrm{FC1}} \\ \theta_{\mathrm{FC2}} \end{bmatrix} = \begin{bmatrix} -3 & 3 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathrm{FC1}} \\ x_{\mathrm{FC2}} \end{bmatrix} \quad \text{(Yaw)}$$

This matrix was inverted and implemented in the control system OUTMATRIX so that the controller signals are mapped correctly to the FC1 and FC2 actuators.

5 Results

After intentionally misaligning the ZM mirrors, we observe that the error signals are gradually driven toward zero over time (Figure 14), demonstrating that the ZM loops act to correct the misalignment. Initially, all signals show a disturbance, but as the loops engage, they converge toward zero. In the plot, the blue trace corresponds to ZM1 pitch, the orange trace to ZM1 yaw, the green trace to ZM3 yaw, and the red trace to ZM3 pitch. This confirms that each loop works to bring its associated degree of freedom back into alignment.

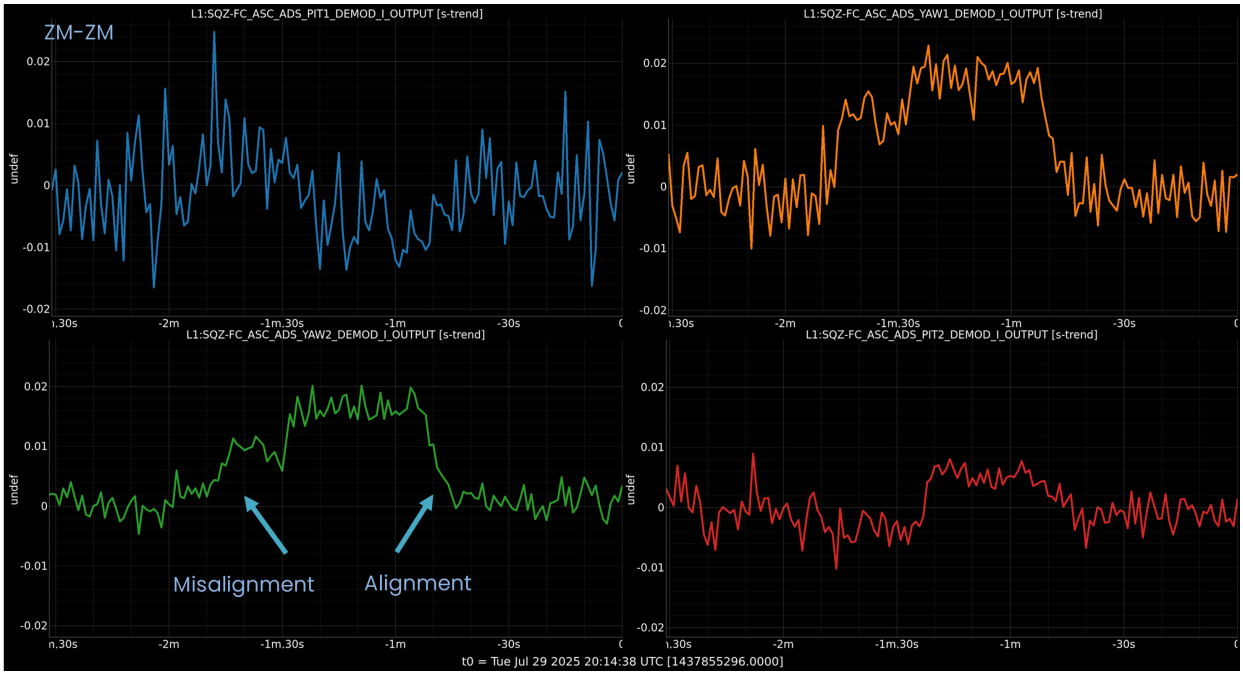

Figure 14: ZM Loops

After intentionally misaligning the FC loops, we observe the response of the loops to this disturbance in Figure 15. During the misalignment, we introduced a yaw-to-length coupling, which can be clearly seen in the top plot. In the middle plot, the corresponding control signal is shown reacting to the misalignment. Finally, the bottom plot displays the error signal. It is gradually driven toward zero over time, showing that the loops act to re-align the system.

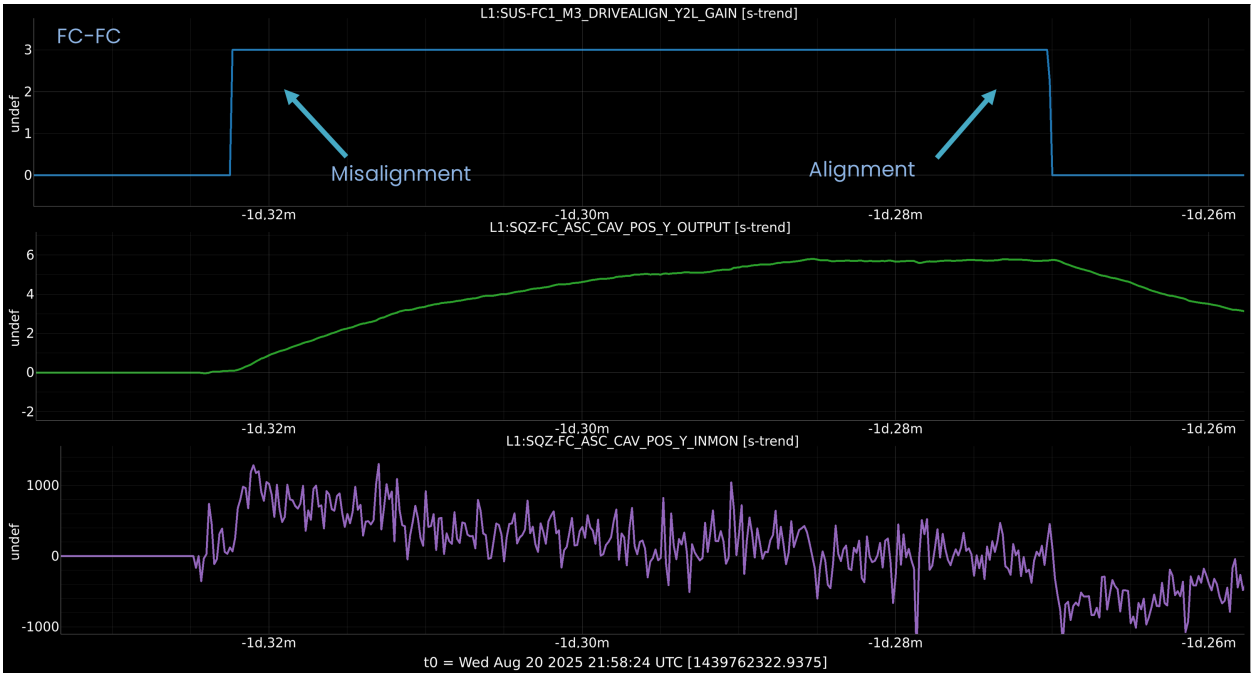

Figure 15: FC Loops

After purposely misaligning the pitch and yaw of the FC and ZM mirrors, all control loops were engaged. As shown below (Figure 16 and Figure 17), the loops act cooperatively to recover the alignment. Initially, the transmitted green power decreased due to misalignment, but once the loops are active, the alignment is restored. The blue curves show the transmitted green power. The orange, purple, red, and green traces represent the error signals, which can be seen approaching zero over time as the loops act to correct the misalignment.

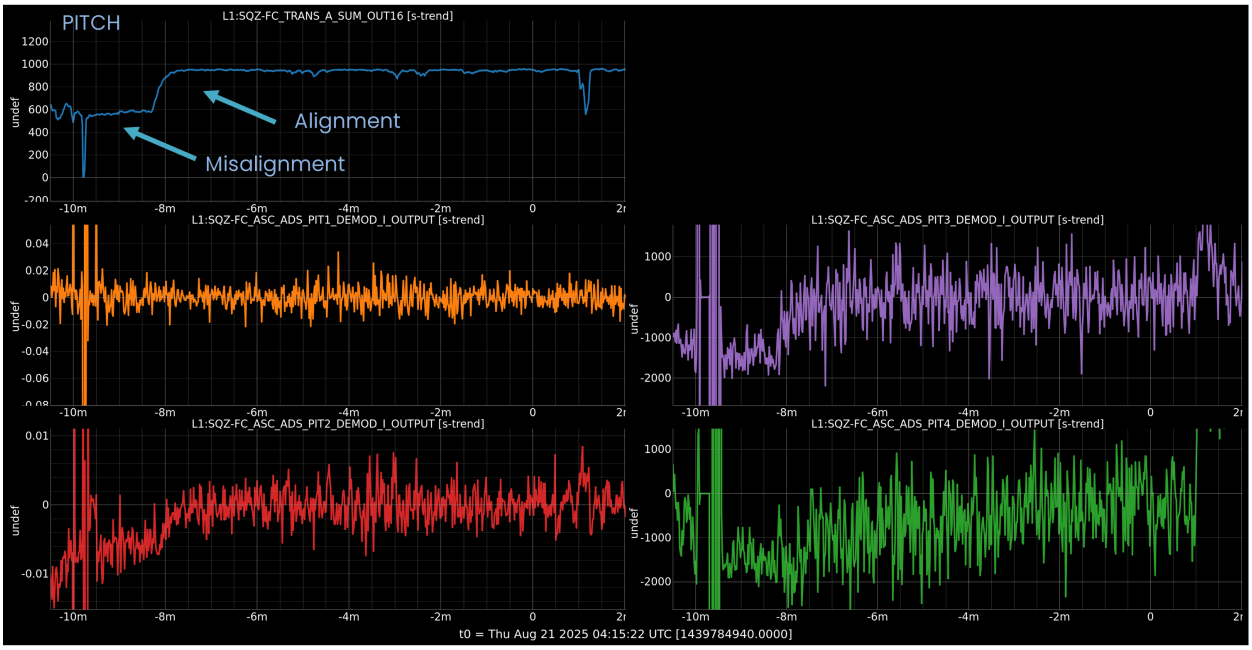

Figure 16: Pitch

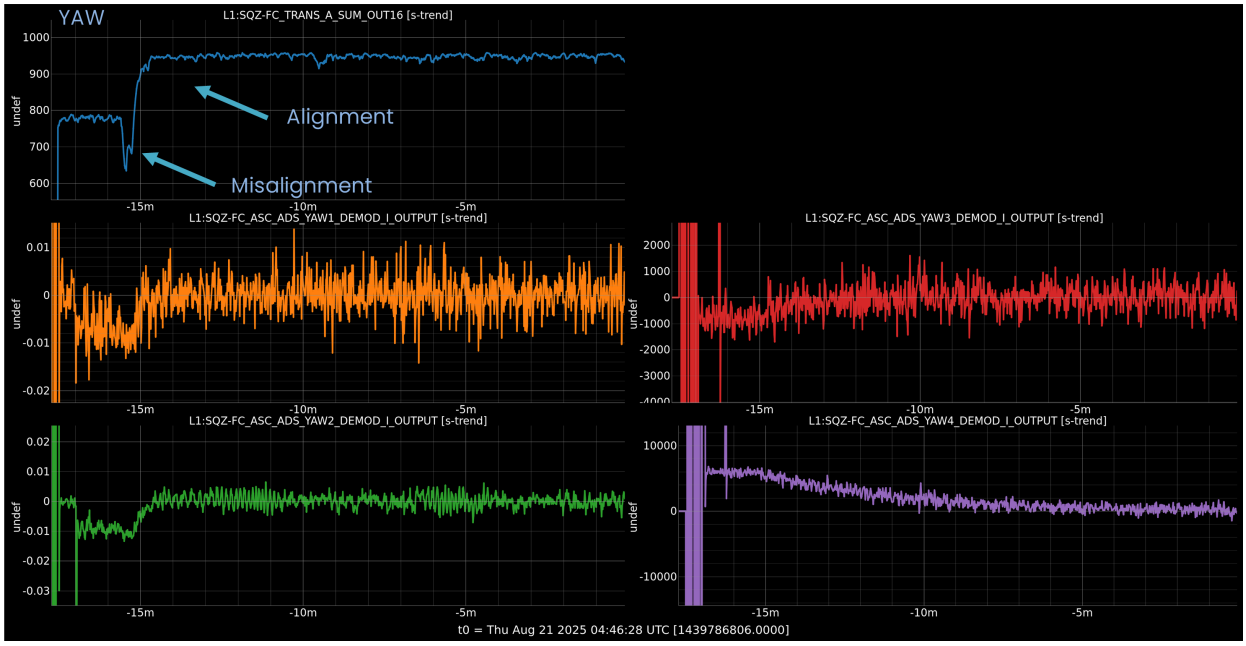

Figure 17: Yaw

6 Conclusion

The Filter Cavity Green Alignment Sensing Control loops have been successfully designed and commissioned. These loops are employed to correct the alignment of the green beam to the cavity (via ZM1/ZM3) as well as the alignment of the filter cavity to the input beam (via FC1/FC2). Due to their fast response, they are considered suitable for integration into the filter cavity locking sequence, with the aim of improving the reliability of the green to IR hand-off. All Guardian codes required for this procedure have already been prepared and thoroughly tested during the course of this project. The state is currently designated as `GR_ALIGN`, and the corresponding Guardian code is provided in the Appendix A.

7 Future Work

At this stage, the only remaining task is integration. One possible approach is to enable the FC Green ASC loops immediately after green lock is established and to maintain them until just before the filter cavity begins sweeping its length to acquire IR resonance. Also, the resonance peaks (Figure 11) at 3.9 Hz and 6 Hz should be suppressed.

8 Acknowledgments

I would like to express my sincere gratitude to my advisors, Adam Mullavey and Begum Kabagoz, for their continuous guidance, insightful feedback, and patience throughout this study. Their support played a crucial role in shaping this work both technically and conceptually.

I am also thankful to all the LIGO Livingston Observatory people, whose encouragement and collaboration made this process more manageable, even enjoyable. The shared challenges, discussions, and small moments of relief truly made a difference.

I would also like to thank the National Science Foundation for supporting this research through the Research Experience for Undergraduates (REU) program. Additionally, I am grateful to the Caltech Student-Faculty Programs and the LIGO Laboratory for providing such a unique and inspiring research opportunity.

References

[1] E. D. Black, "An introduction to pound–drever–hall laser frequency stabilization," *American Journal of Physics*, vol. 69, no. 1, pp. 79–87, 2001. [Online]. Available: https://doi.org/10.1119/1.1286663

[2] M. Evans, L. Barsotti, J. Harms, P. Kwee, and H. Miao, "Realistic filter cavities for advanced gravitational wave detectors," 2013. [Online]. Available: https://arxiv.org/abs/1305.1599

[3] D. Ganapathy, "Expanding the reach of quantum enhanced gravitational-wave detectors," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2024.

[4] D. Ganapathy, W. Jia, M. Nakano, V. Xu, N. Aritomi, T. Cullen, N. Kijbunchoo, S. E. Dwyer, A. Mullavey, L. McCuller, L. Barsotti, and et al., "Broadband quantum enhancement of the ligo detectors with frequency-dependent squeezing," *Phys. Rev. X*, vol. 13, no. 4, p. 041021, 2023.

[5] A. Staley, "Arm length stabilization for advanced ligo," Ph.D. dissertation, California Institute of Technology, Pasadena, California, USA, 2015, lIGO Document P1500273. [Online]. Available: https://dcc-backup.ligo.org/public/0123/P1500273/001/StaleyThesis.pdf

[6] A. Freise, "The next generation of interferometry: Multi-frequency optical modelling, control concepts and implementation," Ph.D. dissertation, Universität Hannover, Hannover, Germany, 2003, doctoral dissertation. [Online]. Available: https://edocs.tib.eu/files/e01dh03/361006918.pdf

[7] A. Abramovici and J. Chapsky, *Feedback Control Systems: A Fast-Track Guide for Scientists and Engineers*. New York: Springer, 2000.

[8] LIGO Scientific Collaboration, "Filter cavity optical layout and initial mode matching procedures," LIGO Laboratory, Tech. Rep. LIGO-T1900649-v8, 2021, lIGO Technical Document. [Online]. Available: https://dcc.ligo.org/public/0163/T1900649/008/FCLayout_T1900649-v8.pdf

[9] Wikipedia contributors, "Ray transfer matrix analysis," 2025. [Online]. Available: https://en.wikipedia.org/wiki/Ray_transfer_matrix_analysis

[10] M. Kasprzack, "Angular sensing and control in advanced ligo," 2020.

[11] G. Drougakis, K. G. Mavrakis, S. Pandey, G. Vasilakis, K. Poulios, D. G. Papazoglou, and W. von Klitzing, "Precise and robust optical beam steering for space optical instrumentation," *arXiv preprint arXiv:1907.10425*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1907.10425

[12] PI USA, "Advances in fiber alignment / photonics alignment automation," 2025. [Online]. Available: https://www.pi-usa.us/en/tech-blog/history-and-future-of-photonics-alignment-automation-test-assembly-of-sip-components

[13] LIGO Livingston Logbook, "Alog 77575: Fc green alignment + polarization into the fibers," 2025. [Online]. Available: https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=77575

[14] ——, "Alog 77470: Sqz fc green alignment adjusted," 2025. [Online]. Available: https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=77470

[15] ——, "Alog 74935: Sqz fc realignment due to cold," 2025. [Online]. Available: https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=74935

[16] ——, "Alog 74190: Lockloss at 01:59 utc - unknown," 2025. [Online]. Available: https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=74190

[17] LIGO Hanford, "A+ ham double suspension hxds (hsds, hdds, hpds, htds, hrds, hds) actuator (actuation) control range (ranges)," LIGO Document Control Center, Tech. Rep. T2100275, 2021. [Online]. Available: https://dcc.ligo.org/LIGO-T2100275

[18] E. Bonilla, "Matlab suspension model review (for low frequency workshop 2021)," 2021. [Online]. Available: https://dcc.ligo.org/LIGO-G2100766

[19] LIGO Livingston Logbook, "Alog 77707: Fc green angular control part 1 - high frequency transfer functions of zms," 2025. [Online]. Available: https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=77707

[20] V. Bossilkov, "chebychef," 2025. [Online]. Available: https://git.ligo.org/vladimir.bossilkov/chebychef

[21] LIGO Scientific Collaboration, "dttxml," 2025. [Online]. Available: https://git.ligo.org/cds/software/dttxml

[22] A. E. Siegman, *Lasers*. Mill Valley, CA: University Science Books, 1986.

A Guardian State

```
#TID 07/22/25
class GR_ALIGN(GuardState):
    index = 24
    request = False

    @GR_lock_checker
    def main(self):

        self.timer['wait'] = 0
        self.counter = 0

        # Turn Off the dithering for top stage and turn on bottom stage for ZM's
        for optic in ['ZM1','ZM3']:
            for dof in ['P','Y']:
                ezca.switch('SUS-{}_M1_DITHER_{}'.format(optic,dof),'OUTPUT','OFF')
                ezca.switch('SUS-{}_M2_DITHER_{}'.format(optic,dof),'OUTPUT','ON')

        # Turn Off the dithering for top stage and turn on bottom stage for FC's
        for optic in ['FC1','FC2']:
            for dof in ['P','Y']:
                ezca.switch('SUS-{}_M1_DITHER_{}'.format(optic,dof),'OUTPUT','OFF')
                ezca.switch('SUS-{}_M2_DITHER_{}'.format(optic,dof),'OUTPUT','OFF')
                ezca.switch('SUS-{}_M3_DITHER_{}'.format(optic,dof),'OUTPUT','ON')


        #TURN OFF THE CAV SWITCHES:
        for m in ['POS', 'ANG']:
            for n in ['P','Y']:
                ezca.switch("SQZ-FC_ASC_CAV_{}_{}".format(m,n),'INPUT','OFF')

        #CLEAR ALL HISTORYs
        for i in ['CAV','INJ']:
            for m in ['POS', 'ANG']:
                for n in ['P','Y']:
                    ezca["SQZ-FC_ASC_{}_{}_{}_RSET".format(i,m,n)]=2


        #Set ADS senmatrix
        for dof1 in ['3','4']:
           matrixSQZ.FC_ASC_SENMATRIX_P.zero(row= f'ADS_{dof1}_I')
           matrixSQZ.FC_ASC_SENMATRIX_Y.zero(row= f'ADS_{dof1}_I')


        matrixSQZ.FC_ASC_SENMATRIX_P['ADS_3_I','FCL_FB'] = 1
```

```
matrixSQZ.FC_ASC_SENMATRIX_P['ADS_4_I','FCL_FB'] = 1
matrixSQZ.FC_ASC_SENMATRIX_Y['ADS_3_I','FCL_FB'] = 1
matrixSQZ.FC_ASC_SENMATRIX_Y['ADS_4_I','FCL_FB'] = 1

#set the input matrices for ZM's
matrixSQZ.FC_ASC_INMATRIX_P.zero(row= 'INJ_POS')
matrixSQZ.FC_ASC_INMATRIX_P.zero(row= 'INJ_ANG')
matrixSQZ.FC_ASC_INMATRIX_P['INJ_POS','ADS_1_I']= -571.4*1e-1
matrixSQZ.FC_ASC_INMATRIX_P['INJ_ANG','ADS_2_I']= -571.4*1e-1
matrixSQZ.FC_ASC_INMATRIX_P['INJ_POS','ADS_2_I']= -428.6*1e-1
matrixSQZ.FC_ASC_INMATRIX_P['INJ_ANG','ADS_1_I']= -428.6*1e-1

matrixSQZ.FC_ASC_INMATRIX_Y.zero(row= 'INJ_POS')
matrixSQZ.FC_ASC_INMATRIX_Y.zero(row= 'INJ_ANG')
matrixSQZ.FC_ASC_INMATRIX_Y['INJ_POS','ADS_1_I']= 125*0.5
matrixSQZ.FC_ASC_INMATRIX_Y['INJ_ANG','ADS_2_I']= -156.25*0.5
matrixSQZ.FC_ASC_INMATRIX_Y['INJ_POS','ADS_2_I']= -125*0.5
matrixSQZ.FC_ASC_INMATRIX_Y['INJ_ANG','ADS_1_I']= 93.75*0.5

#set the output matrices for ZM's
matrixSQZ.FC_ASC_OUTMATRIX_P.zero(col= 'INJ_POS')
matrixSQZ.FC_ASC_OUTMATRIX_P.zero(col= 'INJ_ANG')
matrixSQZ.FC_ASC_OUTMATRIX_P['ZM1','INJ_POS'] = 10
matrixSQZ.FC_ASC_OUTMATRIX_P['ZM3','INJ_ANG'] = 1
matrixSQZ.FC_ASC_OUTMATRIX_P['ZM3','INJ_POS'] =  0

matrixSQZ.FC_ASC_OUTMATRIX_Y.zero(col= 'INJ_POS')
matrixSQZ.FC_ASC_OUTMATRIX_Y.zero(col= 'INJ_ANG')
matrixSQZ.FC_ASC_OUTMATRIX_Y['ZM1','INJ_POS'] = 10
matrixSQZ.FC_ASC_OUTMATRIX_Y['ZM3','INJ_ANG'] = 1
matrixSQZ.FC_ASC_OUTMATRIX_Y['ZM3','INJ_POS'] = 0


#set the input matrices for FC's
matrixSQZ.FC_ASC_INMATRIX_P.zero(row= 'CAV_POS')
matrixSQZ.FC_ASC_INMATRIX_P.zero(row= 'CAV_ANG')
matrixSQZ.FC_ASC_INMATRIX_P['CAV_POS','ADS_3_I']= -0.0017
matrixSQZ.FC_ASC_INMATRIX_P['CAV_ANG','ADS_4_I']= 1

matrixSQZ.FC_ASC_INMATRIX_Y.zero(row= 'CAV_POS')
matrixSQZ.FC_ASC_INMATRIX_Y.zero(row= 'CAV_ANG')
matrixSQZ.FC_ASC_INMATRIX_Y['CAV_POS','ADS_3_I']= 1
matrixSQZ.FC_ASC_INMATRIX_Y['CAV_ANG','ADS_4_I']= 1

#set the output matrices for FC's
matrixSQZ.FC_ASC_OUTMATRIX_P.zero(col= 'CAV_POS')
```

```
matrixSQZ.FC_ASC_OUTMATRIX_P.zero(col= 'CAV_ANG')
matrixSQZ.FC_ASC_OUTMATRIX_P['FC1','CAV_POS'] = -3
matrixSQZ.FC_ASC_OUTMATRIX_P['FC1','CAV_ANG'] = 3
matrixSQZ.FC_ASC_OUTMATRIX_P['FC2','CAV_POS'] =  3
matrixSQZ.FC_ASC_OUTMATRIX_P['FC2','CAV_ANG'] =  -1

matrixSQZ.FC_ASC_OUTMATRIX_Y.zero(col= 'CAV_POS')
matrixSQZ.FC_ASC_OUTMATRIX_Y.zero(col= 'CAV_ANG')
matrixSQZ.FC_ASC_OUTMATRIX_Y['FC1','CAV_POS'] = -3
matrixSQZ.FC_ASC_OUTMATRIX_Y['FC1','CAV_ANG'] = 3
matrixSQZ.FC_ASC_OUTMATRIX_Y['FC2','CAV_POS'] = -3
matrixSQZ.FC_ASC_OUTMATRIX_Y['FC2','CAV_ANG'] = 1


matrixSQZ.FC_ASC_INMATRIX_P.load()
matrixSQZ.FC_ASC_INMATRIX_Y.load()

#set the oscillator for dithering ZM's
turn_on_osc('PIT1',101.1,2000,-40)
turn_on_osc('PIT2',151.1,200,120)
turn_on_osc('YAW1',126.1,2000,140)
turn_on_osc('YAW2',176.1,200,105)

#set the oscillator for dithering FC's
turn_on_osc('PIT3',10,1000,0)
turn_on_osc('PIT4',12,3000,-5)
turn_on_osc('YAW3',6,1000,16)
turn_on_osc('YAW4',8,3000,0)


#demodualtion filters for ZM's
for i in ['PIT','YAW']:
    for m in ['1', '2']:
        for n in ['I']:
            ezca.switch(f'SQZ-FC_ASC_ADS_{i}{m}_DEMOD_{n}','FMALL','OFF')
            ezca.switch(f'SQZ-FC_ASC_ADS_{i}{m}_DEMOD_{n}','FM6','ON')

#demodualtion filters for FC's
for i in ['PIT','YAW']:
    for m in ['3', '4']:
        for n in ['I']:
            ezca.switch(f'SQZ-FC_ASC_ADS_{i}{m}_DEMOD_{n}','FMALL','OFF')
            ezca.switch(f'SQZ-FC_ASC_ADS_{i}{m}_DEMOD_{n}','FM1','ON')
            ezca.switch(f'SQZ-FC_ASC_ADS_{i}{m}_DEMOD_{n}','FM3','ON')
            ezca.switch(f'SQZ-FC_ASC_ADS_{i}{m}_DEMOD_{n}','FM10','ON')
```

```
    # turn ON the servo FM9 filter for ZM's
    for m in ['POS', 'ANG']:
        for n in ['P','Y']:
            ezca.switch('SQZ-FC_ASC_INJ_{}_{}'.format(m,n),'FMALL','OFF')
            ezca.switch('SQZ-FC_ASC_INJ_{}_{}'.format(m,n),'FM9','ON')



    for m in ['POS', 'ANG']:
        for n in ['P','Y']:
            ezca.switch(f'SQZ-FC_ASC_CAV_{m}_{n}','FMALL','OFF')
            ezca.switch(f'SQZ-FC_ASC_CAV_{m}_{n}','FM1','FM8','FM9','ON')

    # turn ON the servo filters for FC's
    ezca.switch('SQZ-FC_ASC_CAV_ANG_P','FM2','ON')
    ezca.switch('SQZ-FC_ASC_CAV_POS_Y','FM2','FM3','ON')
    ezca.switch('SQZ-FC_ASC_CAV_ANG_Y','FM2','ON')


    ezca.switch('SQZ-FC_ASC_INJ_POS_Y','FM3','ON')
    ezca.switch('SQZ-FC_ASC_INJ_POS_P','FM3','ON')


    #TURN ON THE ZM SWITCHES:
    log('Turning on ZM angular servos')
    for m in ['POS', 'ANG']:
        for n in ['P','Y']:
            ezca.switch("SQZ-FC_ASC_INJ_{}_{}".format(m,n),'INPUT','ON')

    #TURN ON THE FC SWITCHES:
    log('Turning on FC  servos')
    for m in ['POS', 'ANG']:
        for n in ['P','Y']:
            ezca.switch("SQZ-FC_ASC_CAV_{}_{}".format(m,n),'INPUT','ON')



    #close the loop
    ezca['SQZ-FC_ASC_GAIN'] = 0.5


@GR_lock_checker

def run(self):
    return True
```