

Studies towards improved gravitational wave detector thermodynamics

by

Daniel Vander-Hyde

B.S. California State University, Fullerton, CA, 2015

M.S. Syracuse University, NY, 2016

Submitted in partial fulfillment of the requirements

For the degree of

Doctor of Philosophy in physics

at

SYRACUSE UNIVERSITY

March 2024

Abstract

Abstract goes here

Dedication

In memory of Michael (“Mikey”) Vander-Hyde (1990-2017)

Beloved brother, son, and friend.



Declaration

I declare that

Acknowledgements

Contents

1	Introduction	1
1.1	Gravitational waves	1
1.2	Detector configurations	3
1.2.1	Interferometry with a Michelson configuration	3
1.2.2	Fabry-Pérot Michelson (FPMI)	8
1.2.3	Dual-Recycled Fabry-Pérot Michelson (DRFPMI)	15
1.3	ALIGO	19
1.3.1	Thermodynamic considerations	19
2	Commissioning Adaptive Optics for O3a	23
2.1	Motivation	23
2.1.1	Thermal Compensation System	24
2.1.2	Thermo-optic transients	25
2.2	Dynamic Thermal Compensation	27
2.2.1	Reducing Parametric Instabilities	31
2.2.2	Limitations	32
2.3	A priori TCS pre-load methodology for O3a	33
2.4	A posteriori thermal compensation for O3a	34
2.4.1	Point absorption in O3a	35
2.5	Results	38

2.5.1	Custom CO ₂ mask	38
2.5.2	Beam position offset	39
3	Electro-optic study of a GaAs/Al_{0.92}Ga_{0.08}As coated mirror	40
3.1	Motivation	40
3.1.1	Coating Thermal Noise	40
3.1.2	Coating Electro-optic Noise	45
3.2	Birefringence in zincblende materials	45
3.2.1	The Indicatrix	45
3.2.2	GaAs and Al _{0.92} Ga _{0.08} As crystal classification	47
3.2.3	Linear electro-optic effect (Pockel's effect)	48
3.2.4	New principal (electro-optic) dielectric axis for zincblende structures .	49
3.2.5	The photoelastic effect	50
3.2.6	The generalized indicatrix	50
3.2.7	EO Modulation (Application)	50
3.3	Electro-optic noise of a GaAs / Al _{0.92} Ga _{0.08} As stack	52
3.3.1	Static Birefringence / Miller indices from a HR GaAs / Al _{0.92} Ga _{0.08} As coating	52
3.3.2	Electro-optic coupling to the reflected phase of a HR mirror coating .	54
3.3.3	Initial projected DARM coupling	57
3.4	Electro-optic measurement apparatus	58
3.4.1	PDH servo	59
3.4.2	Servo Parameters	62
3.4.3	Longitudinal Pockels Cell mirror mount assembly	64
3.4.4	Servo Overview	74
3.4.5	Calibration	74
3.4.6	Assembly Mount Solution	75
3.4.7	Measured birefringence from HR GaAs/Al _{0.92} Ga _{0.08} As mirror	77

3.5 Results	79
3.5.1 Acousto-optical coupling	80
4 Conclusion	82
4.1 Adaptive optics comissioning	82
4.2 GaAs/Al _{0.92} Ga _{0.08} As Electro-optic noise	83
5 Appendix	84
5.1 Interferometer Configurations (code)	84
5.1.1 ifo_configs.py	84
5.1.2 MICH	87
5.1.3 FPMI	91
5.1.4 DRFPMI	97
5.2 Paraxial equation	106
5.3 Cavity stability criteria ($G(g_1, g_2)$)	107
5.4 The Equipartition theorem and the Fluctuation dissipation theorem	107
5.5 RH control pre-filter	108
5.5.1 recipe	108
5.5.2 code	109
5.6 Misc. thermo-optic filters	125
5.6.1 COMSOL self heating filter	125
5.6.2 CO ₂ filter	126
5.7 Thermo-optic Path Distortion (analytical)	126
5.7.1 Thermorefractive aberration	126
5.7.2 Thermoelastic aberration	127
5.7.3 Ring Heater actuation	127
5.8 CO ₂ mask	128
5.9 Anisotropic media	128

5.9.1	Monochromatic plane wave propagation	129
5.9.2	The Dielectric tensor	130
5.9.3	Miller indices for highly reflective GaAs/Al _{0.92} Ga _{0.08} As coatings . . .	131
5.10	Mode matching data for Electro-optic sample cavity	132
5.10.1	Pre MMT beam scan	132
5.10.2	“Just another mode matching tool” (JAMMT) solution	133
5.10.3	Post MMT beam scan	133
5.11	Laser PZT sweep	134
5.12	Voltage drive electronics	135
5.13	Alternate Mounting Solutions (Assemblies 1 → 3)	136
5.13.1	Assembly 1	136
5.13.2	Assembly 2	139
5.14	Assembly blueprints and alternative views	142
5.14.1	Assembly 1	142
5.14.2	Assembly 2	145
5.14.3	Assembly 3 [MACOR] (blueprint)	148
5.15	LaplacE code	149
5.15.1	laplace.py	149
5.15.2	set_params.py	159
5.15.3	run.py	166
5.16	Calibration code	173
5.17	HVA	185
5.18	FSS	185
5.19	Measuring OLG [H]	185
References		186

Chapter 1

Introduction

1.1 Gravitational waves

“Space-time tells mass how to move; mass tells space-time how to curve” can provide a concise and sufficient summarization of Einstein’s theory of general relativity (GR). While providing the most complete theory of gravity to date, GR provides tools that allow considerations of high energy astrophysical phenomena (highly massive binary coalescences, spherically assymetric compact objects, etc.) whose fractional mass/energy output generate distortions in space-time known as gravitational waves (GW). This is represented as a perturbation ($|h_{\mu\nu}| \ll 1$) in the Minkowski metric tensor defining a local linearized space-time:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}$$

The wave-like behavior for $h_{\mu\nu}$ is realized after imposing the Lorentz gauge; producing 10 harmonic wave amplitudes from the Einstein field equations. Imposing a wave vector (k_j) onto one of three linearly independent spatial coordinates ($h^{ij}k_j$), the non-trivial amplitudes from the equations imply a transverse and traceless (h_i^i) gauge [1]:

$$\nabla^2 h_+ - \frac{1}{c^2} \frac{\partial}{\partial t} h_+ = 0$$

$$\nabla^2 h_\times - \frac{1}{c^2} \frac{\partial}{\partial t} h_\times = 0$$

In other words, there exists a wave solution with two separate transverse polarizations h_+ and h_\times with a 45° separation between them.

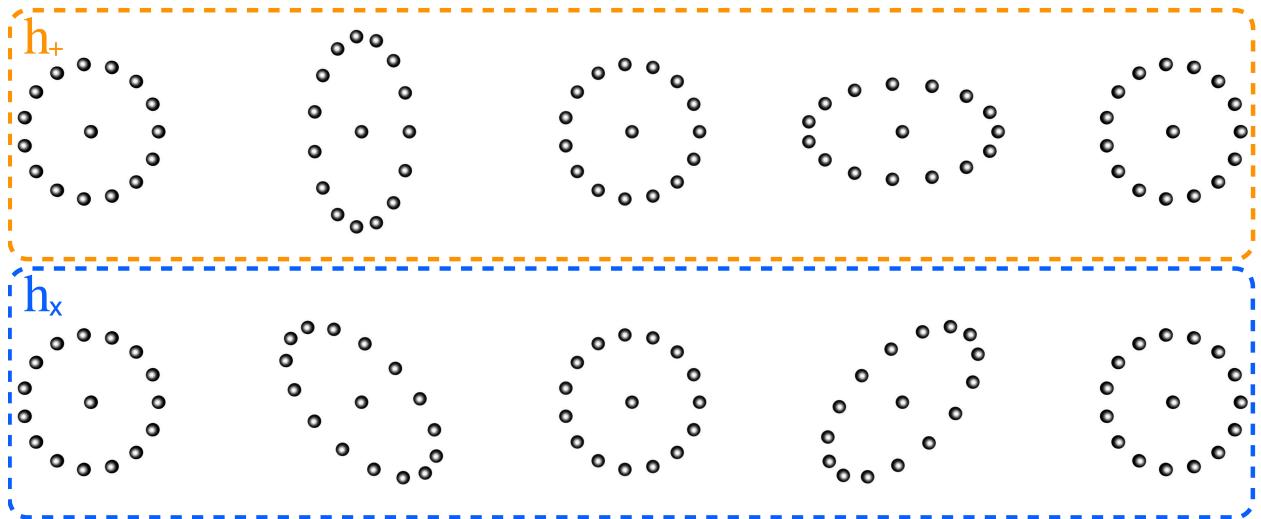


Figure 1.1: A stop motion pictograph displaying the influence of a single gravitational wave period from the two polarizations on a ring of particles. The top row shows the influence of the ‘+’ polarization while the bottom row demonstrates that of the ‘×’ polarization

When measured and analyzed, these waves allow astronomers to extract novel information from their progenitors through the testing of various hypotheses pertinent to the violent dynamics of these systems. On September 14th, 2015 the Laser Interferometric Gravitational Wave Observatories made the first direct gravitational wave detection from a pair of coalescing black holes 1.3 billion light years away, and since then other gravitational wave detectors (i.e. Virgo, KAGRA) have joined the search; continuing the search for novel events. The current GW detector network has developed a track record with an ever increasing list and rate of detections including the first multimessenger event and a surprising population of compact binary coalescences [2, 3].

A more experienced reader may be familiar with the following primer to gravitational wave instrumentation, but it is all done with the hope of providing context of novel contributions within the body of this work while also demonstrating reverence to those whose work prior made this dissertation possible.

1.2 Detector configurations

The current gravitational wave detector network primarily uses terrestrial bound Dual-Recycled Fabry-Pérot Michelson interferometers; though to configure them into a state of observing, fundamental modes of operation are necessary to acquire first. A quick review of these modes provides some of the basic “whats” and “hows” of detector operation with the intention of developing a holistic view of the LIGO detection schema especially as it pertains to the studies to be discussed. Most introductory detector configuration discussions start with the Michelson interferometer and end at the dual-recycled Fabry-Pérot Michelson interferometer; this section follows in kind. Alongside the discussion are citations providing exceptional alternative and more detailed explanations of topics discussed.

1.2.1 Interferometry with a Michelson configuration

The Michelson interferometric detection schema (aka “The Michelson”), used by Michelson and Morley to test the existence of luminiferous aether, demonstrates inherent potential for measuring gravitational wave amplitudes generated by time varying quadrupole moments with high energy astrophysical progenitors; making it a prime candidate as a gravitational wave detector / observatory. The interferometry begins with a beam of coherent laser light split at a 50/50 beamsplitter (BS) along two perpendicular beam paths with respective lengths L_x and L_y , set by highly reflective end mirrors (ETMX, ETMY). Upon arrival at the length terminating mirrors, the respective beams are back-reflected towards the beam splitter where they are made to interfere. The fringe power from this interference is measured

at the anti-symmetric port photodiode:

$$P_{\text{out}} = \frac{P_{\text{in}}}{2} \left[1 + \cos\left(\frac{4\pi}{\lambda}(L_x - L_y)\right) \right] \quad (1.1)$$

The Michelson detects microscopic differential length changes on the order of a fractional wavelength of the light used and are more aptly discussed as differential phase ($\Delta\phi(t)$) between the returning perpendicular phasefronts ($\phi_x(t), \phi_y(t)$). Understanding this inherent method of detection, a time-varying metric perturbation ($h(t)$), like that generated from a gravitational wave, is tested on a Michelson interferometer with a nominal arm length of L and a laser with optical angular frequency of Ω :

$$\Delta\phi(t) = \phi_x(t) - \phi_y(t) = \int_{t-2L/c}^t \Omega \left[1 + \frac{1}{2}h(t) \right] dt - \int_{t-2L/c}^t \Omega \left[1 - \frac{1}{2}h(t) \right] dt \quad (1.2)$$

Evaluating the above as a function of frequency yields:

$$\Delta\phi(\omega) = h_0 \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c} = h_0 \cdot H(\omega, \phi_0) \quad (1.3)$$

With the wave amplitude h_0 , angular frequency ω , nominal interferometer arm length L , and speed of light c . The differential phase Equation 1.3 combined with the power at the anti-symmetric port Equation 1.1 provides a function of optical gain, dependent on frequency and a differential offset phase (ϕ_0):

$$\Delta P(\omega, \phi_0) = h_0 \frac{P_{\text{in}}}{2} \Delta\phi(\omega) \cdot \sin(\phi_0) \quad (1.4)$$

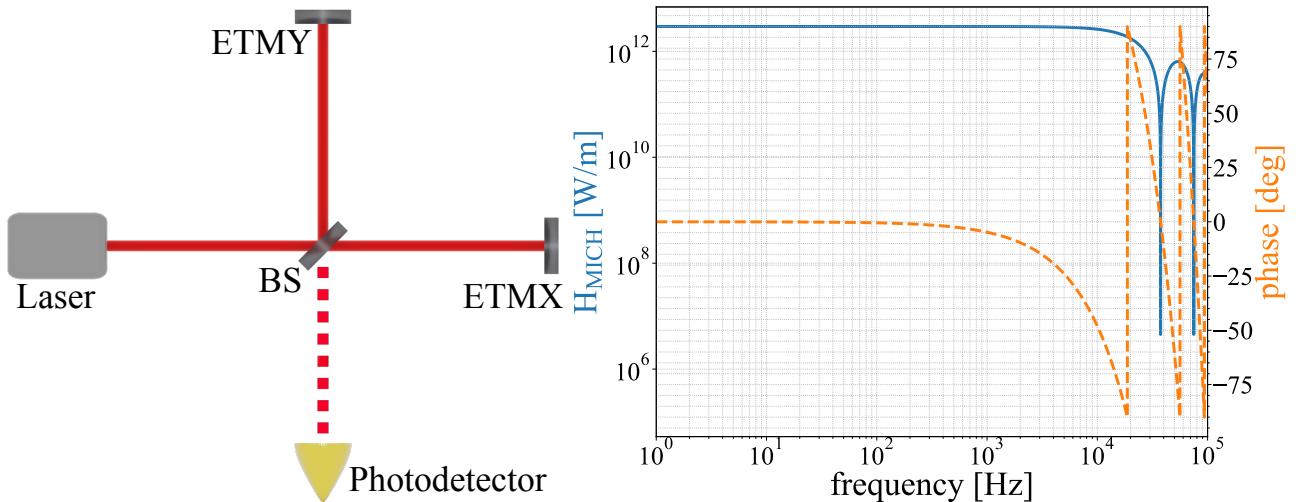


Figure 1.2: [Left] A simplified schematic of a Michelson interferometer. [Right] The associated optical transfer function with $H(\omega, \pi/2)$ defining the optical gain of a Michelson interferometer with 4 km long arms and an input power of 25 [W]

Assuming a 4km arm configuration with 25 Watts input power as indicated in figure Equation 1.3 the differential arm response provides a reasonable optical gain with the notches coorelating to an integer number of gravitational wave half periods ($n\lambda_{gw}/2$) to the interferometer arm length in such a way that the response is null for cooresponding frequencies. Though with sights set on optimizing detection bandwidth for neutron star (NS) binaries @ 100 Hz, the basic Michelson optical gain remains insufficient, with enhancements required. This is better visualized by computing the shot noise limited Michelson sensitivity which does not reach he requirement to confirm a NS-NS coalescence ($\approx 10^{-21} [\frac{1}{\sqrt{\text{Hz}}}]$) but an astonishing start nonetheless [4].

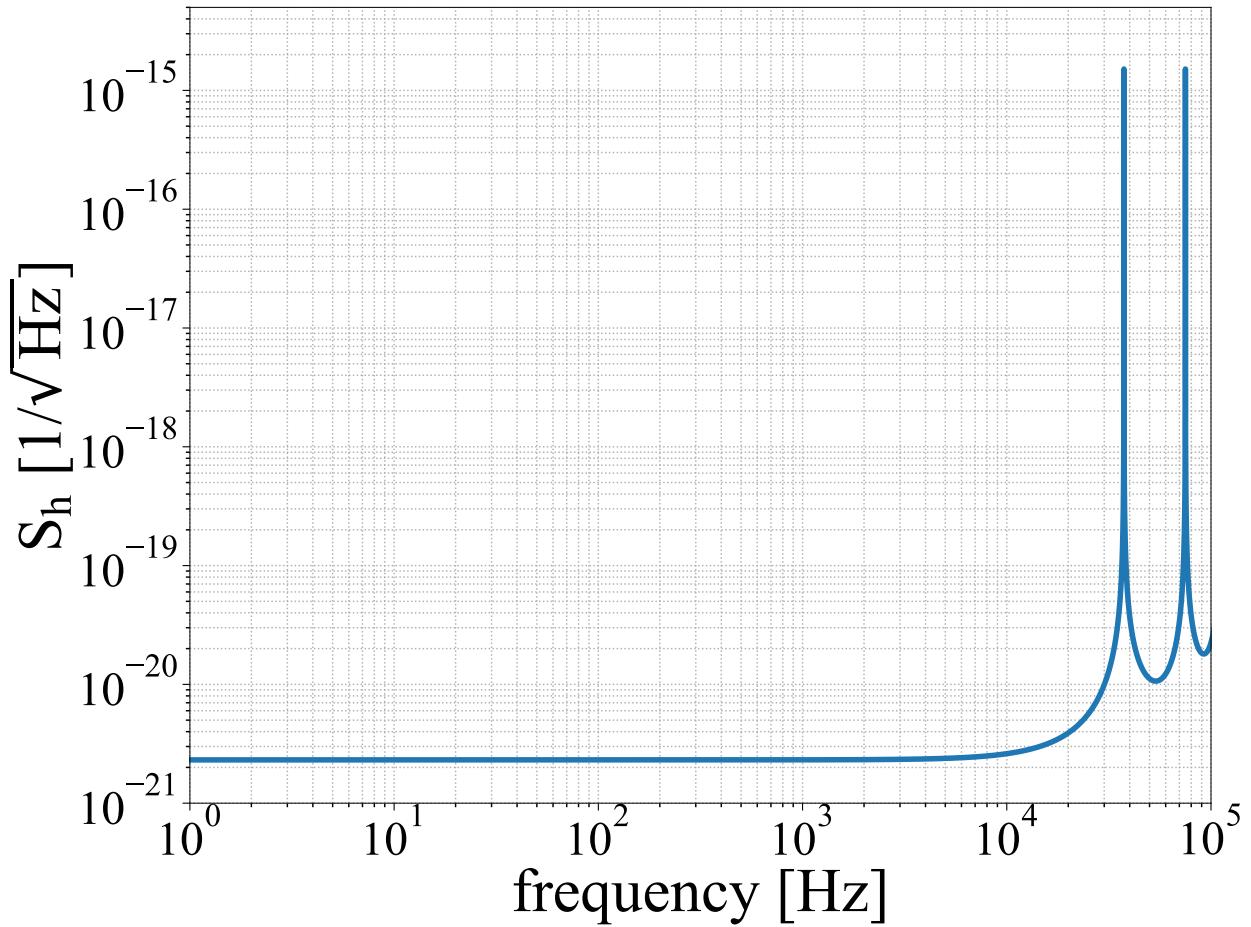


Figure 1.3: Shot noise limited sensitivity ($\sqrt{\hbar\Omega P_{\text{in}}}$) of a Michelson with 4 km long arms and an input power of 25 [W]. Compared to the apriori estimate of $10^{-21} [\frac{1}{\sqrt{\text{Hz}}}]$ the signal to noise (SNR) comes to be unity. The desired confidence is set at a much higher standard with more unquestionable measurements set at SNR = 5.

1.2.1.1 Contrast (Mode Matching Pt. 1)

As presented, the functional behavior of the simple Michelson is to perform optical autocorrelation; though overly simplified depictions of interferometry suggest operation by periodic planar phasefronts and omit the full reality of Gaussian beam propagation. A standard laser carrier beam mode is represented by the Gaussian beam (TEM00 mode) with wavelength λ , and propagation axis (z):

$$E(r) = E_o \frac{\sqrt{[\lambda z_o]/\pi}}{W(z)} e^{-r^2/W^2(z)} e^{-ikz - ik[r^2/(2R(z))] + i\zeta(z)} \quad (1.5)$$

Where E_o is a complex field amplitude, $r^2/(2(R(z))$ defines transverse coordinates $r = \sqrt{x^2 + y^2}$ on a hemisphere of uniform phase with a radius of curvature $R(z)$, k is the wave number, $W(z)$ is the radius from the beam axis that contains $(1 - 1/e^2) \times 100\%$ of the integrated beam power, and ζ is the Gouy phase [5].

An important consideration to make for any sufficiently long arm length (like that used for LIGO), is avoiding significant power loss due to beam divergence for the designed Michelson arm length, sans sufficiently large core optics. LIGO and most other terrestrial GW detectors manage with curved end mirrors that match and focus the impinging hemispherical wavefronts; symmetrically back-propagate them in each arm to maintain optimal interference at the beamsplitter. Mode overlap η provides a useful metric of this optimization:

$$\eta = \left| \int E_x E_y dA \right|^2 / (P_x P_y) \quad (1.6)$$

Significant length offsets between the arms reduce the integrated phasefront overlap and contribute to sub-optimal interference of the beam mode wavefronts at the beamsplitter. And even without 2D beam profiles at the output, measuring the dark and bright fringe power on single element photodiodes and computing contrast a.k.a interferometer visibility (ν) may do just as well without more involved beam mode analysis:

$$\nu = \frac{P_{\max} - P_{\min}}{P_{\max} + P_{\min}} \quad (1.7)$$

Operating on a scale from 0 to 1, $\nu \leq 1$ can be an indication of : mode mismatch at the beam splitter and/or assymetrical optical loss. From a mode matching perspective $\nu = 1$ represents a mode overlap of $\eta = 100\%$ (assuming no optical loss). Though optical losses and aberrations are often asymmetrically introduced between the orthogonal beam paths that often limit optimal interference and are indiscriminately accounted for in contrast measurements.

1.2.2 Fabry-Pérot Michelson (FPMI)

At the time of the LIGO proposal, constraints (physical and financial) for terrestrial gravitational wave detectors required a compact solution for increasing length (L) of the Michelson arms so to increase the beam phasefront lifetime within the Michelson arms. Two proposed arm folding techniques were considered: the Herriot Delay Line and the Fabry-Pérot cavity, though the Fabry-Pérot cavity is currently the predominant choice.

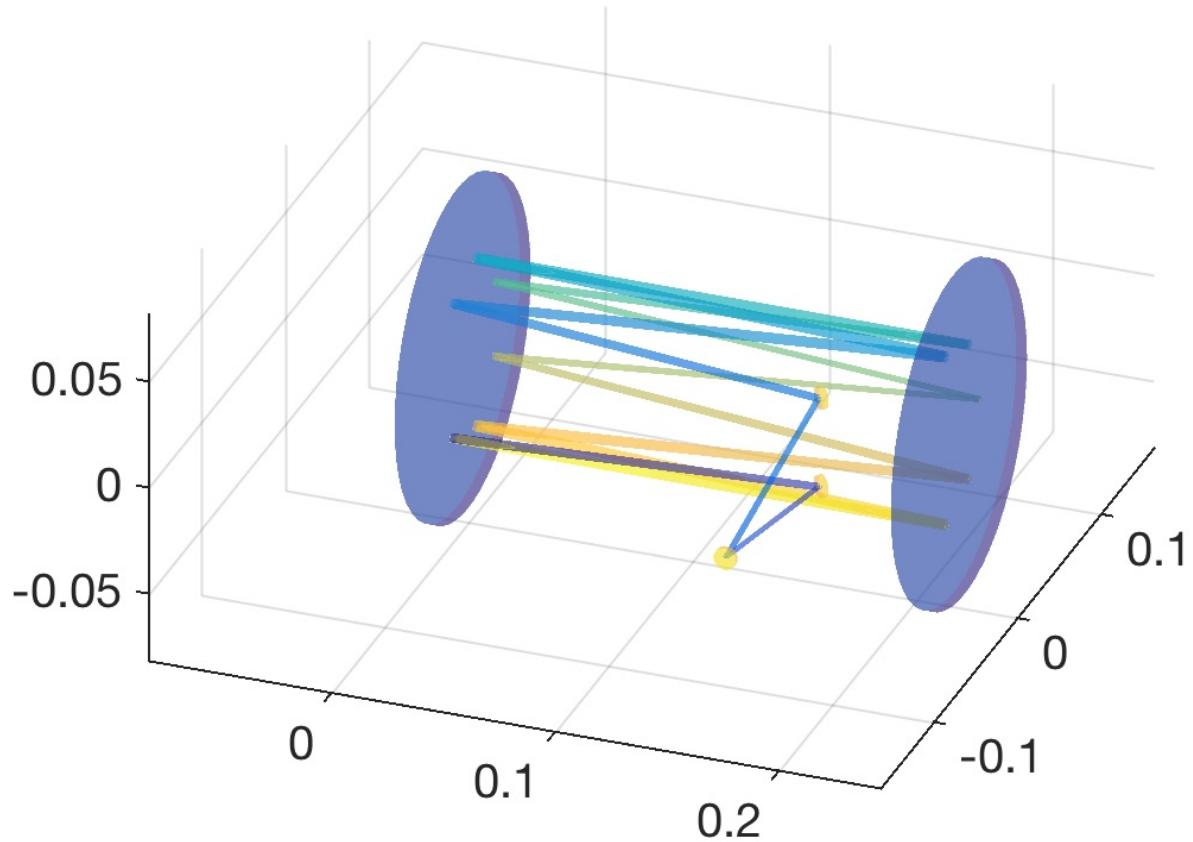


Figure 1.4: A 12 bounce Herriot Delay Line with a small mirror input / output couplers inserted into the beam path.

1.2.2.1 The Fabry-Pérot cavity

To inform of a folding mechanism, consider coherent light encountering an optical cavity with input and output mirror transmission and reflection coefficients of t_1 , r_1 and t_2 , r_2 respectively (assuming lossless mirrors $L_1 + L_2 = 0$).

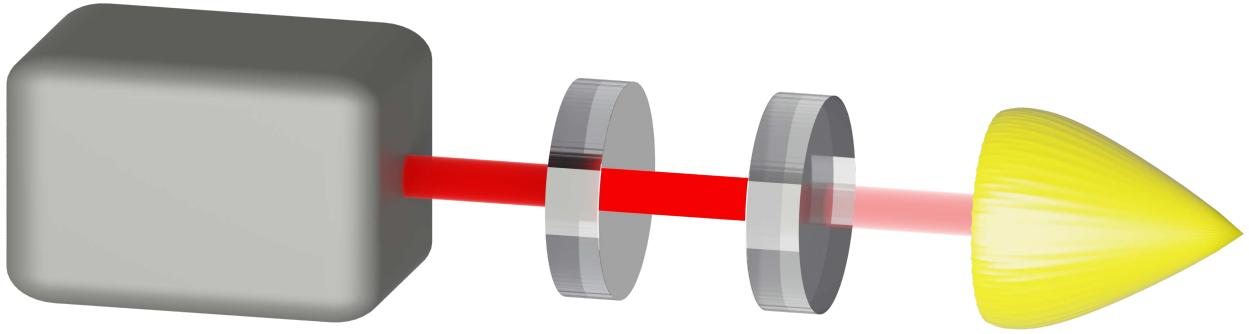


Figure 1.5: Figure of a Fabry Perot Cavity

Light enters the cavity only after passing the input mirror with the trivial solution indicating a field amplitude reduction proportional to the mirror reflection coefficient. Though by tuning the length between the input and end mirrors to an integer multiple of the beam wavelength, circulating light coherently adds with the input, achieving resonance. A cavity of length L , when configured, yields the following cavity reflection and transmission coefficients:

$$r_c = -r_1 + \frac{t_1^2 r_2 e^{-i2kL}}{1 - r_1 r_2 e^{-i2kL}} \quad (1.8)$$

$$t_c = \frac{t_1 t_2 e^{-ikL}}{1 - r_1 r_2 e^{-i2kL}} \quad (1.9)$$

Maintaining resonance for highly reflective mirrors and short wavelength light (i.e. λ 1064nm) requires strict length tuning ($\leq 1\text{e}-7$ [m]).

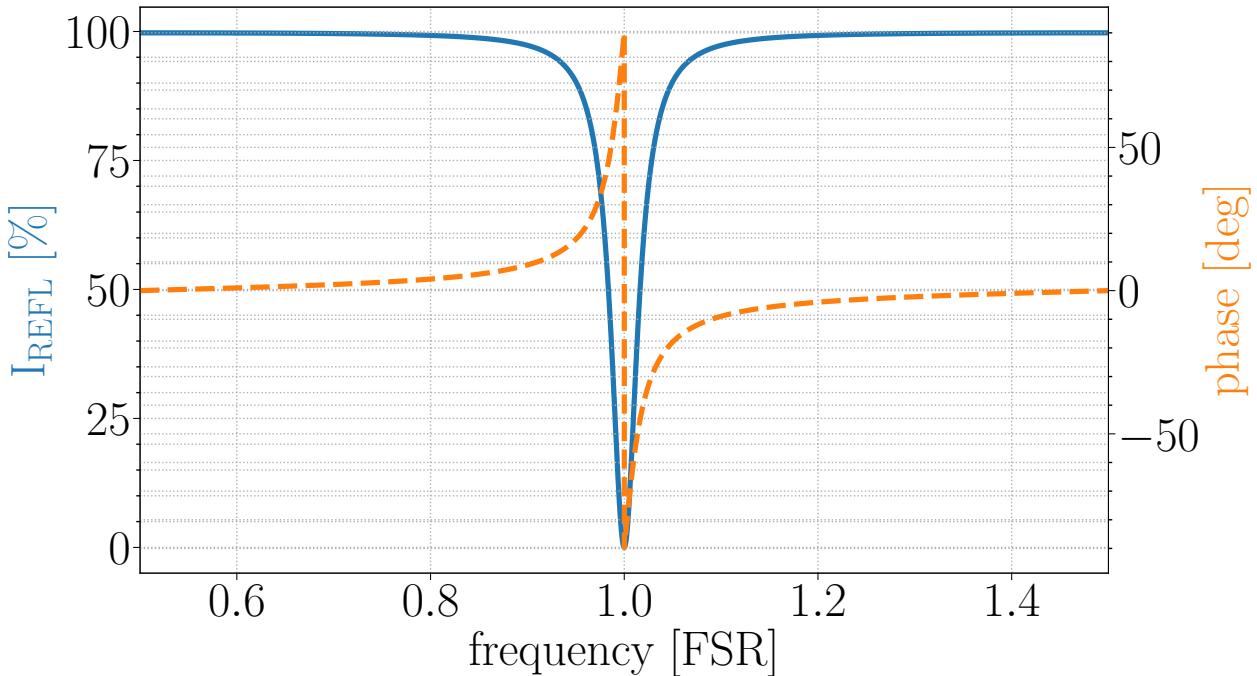


Figure 1.6: Reflected cavity intensity (I_{REFL}) around resonance. The resonance peak full width half maximum is set by mirror reflectivities and is succinctly quantified by the cavity finesse ($\mathcal{F} = \frac{\text{FWHM}_{\text{res}}}{f_{\text{FSR}}} = \frac{\pi\sqrt{r_1 r_2}}{1-r_1 r_2}$).

The ratio between circulating and cavity input power is set by the reflectivity parameters of the cavity mirrors, demonstrating the correlation to how long a given phasefront can remain stored between said mirrors at resonance. This “cavity storage time” ($\tau_s \propto L r_1 r_2$) translates as a length elongation with the phasefront travel history encoded in the arrival time of its photons back at the beam splitter.

1.2.2.1.1 “Arm elongation”

An intuitive analogue of the Fabry-Pérot’s arm elongation capabilities is better illustrated when comparing against a computed Delay Line storage time (with \mathcal{N} number of bounces and length L) [4]:

$$\tau_s^{\text{DL}} = \frac{2\mathcal{N}L}{c} \quad (1.10)$$

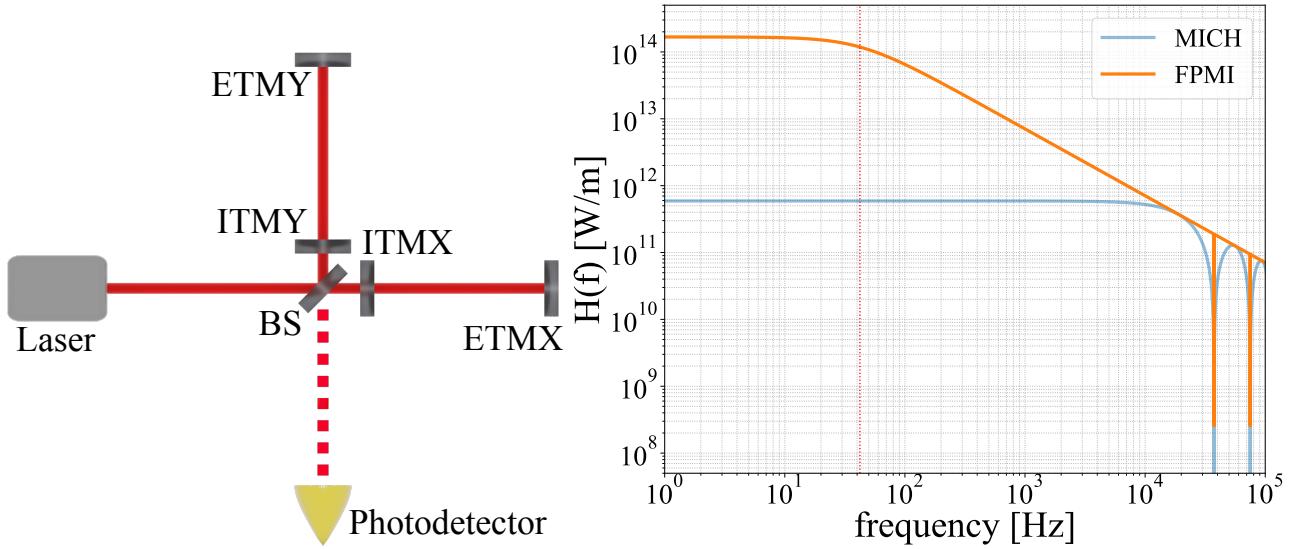


Figure 1.7: [Left] The Fabry-Pérot Michelson optical schema and [Right] an associated optical gain.

$$\tau_s^{\text{FP}} = \frac{L}{c} \frac{r_1 r_2}{1 - r_1 r_2} = \frac{1}{4\pi\mathcal{F}} \quad (1.11)$$

Advanced LIGO, with its 4km length and approximate finesse of 208 correlates to a storage time of $382\mu\text{s}$, whereas the simple Michelson has an arm storage time of $26\mu\text{s}$. The cooresponding optical gain increase is noted in Figure 1.7.

$$H_{\text{FPMI}} = \frac{t_1^2 r_2}{(t_1^2 + r_1^2)r_2 - r_1} \frac{H_{\text{MICH}}}{1 - r_1 r_2 e^{-2i\omega L/c}} \quad (1.12)$$

The noted gain improvements made by adding two mirrors to the optical schema are substantial, though in practice the benefits are contingent upon: 1) maintaining fixed mirror positions within a fraction of the wavelength of the light used and 2) reducing detector bandwidth. But with tools like the Pound-Drever-Hall (PDH) technique and signal recycling to mitigate these respective burdens, it becomes clear that it is a small price to pay [6].

1.2.2.1.2 Gaussian and Higher Order Modes (Mode Matching pt.2)

There are some additional caveats before exploiting any potential enhancement from the Fabry-Pérot when imposing a gaussian beam. First is that of beam divergence, which can quickly limit the stored power between two flat finite sized mirrors (see § 5.3). Though, as addressed for the simple Michelson, curving end mirrors focuses the Gaussian beam power and can also increase resonance robustness curving one or both cavity mirrors.

Additionally mentioned is the importance of the overlap between mirror radii of curvature to the beam phasefront. For resonators this becomes increasingly critical as the placement of mirrors with established curvatures (defining the cavity mode) need to be placed such that they preserve the TEM00 beam mode. Beam optics inform solutions for optimal mirror placement for a given incident beam and vice versa. The exercise sets the importance of matching the **gaussian beam mode** to the spherical mirror **FP resonator mode**, but less obvious are the implications if there is a perturbation from the set solution. Alongside the TEM00 fundamental mode, the paraxial equation § 5.2 also produces families of solutions that exist for the two mirror cavity configuration, characterized by the Hermite-Gauss and Laguerre-Gauss bases:

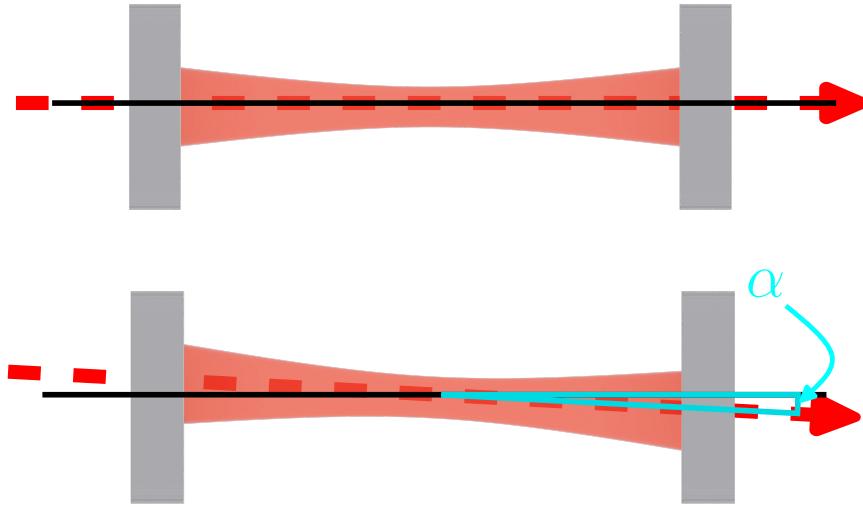
Hermite-Gauss modes

Figure 1.8: Figure depicting cavity-beam alignment (above) vs cavity-beam misalignment (below) with a angle α between the beam and cavity axes.

$$\begin{aligned}
 TEM_{n,m}(x, y, z) = & E_o \frac{\sqrt{[\lambda z_o]/\pi}}{W(z)} \mathbb{H}_n\left(\frac{\sqrt{2}x}{W(z)}\right) \mathbb{H}_m\left(\frac{\sqrt{2}y}{W(z)}\right) \\
 & \times \exp\left(-\frac{(x^2 + y^2)}{W^2(z)}\right) \exp\left(-ikz - ik\frac{x^2 + y^2}{2R(z)} + i(1+n+m)\zeta(z)\right)
 \end{aligned} \tag{1.13}$$

As intensity, power and mode overlap are common computations, the gaussian integrals might be more quickly expressed and computed with the more convenient bra-ket notation:

$$TEM_{n,m}(x, y, z) \rightarrow |U_{n,m}(x, y, z)\rangle$$

Laguerre-Gauss modes

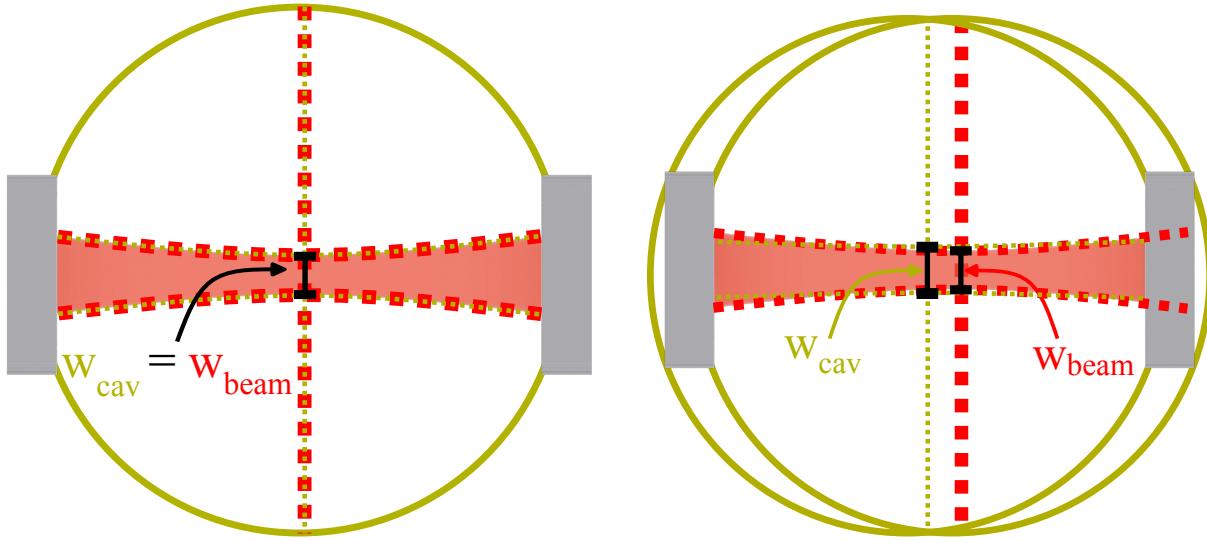


Figure 1.9: Figure depicting cavity-beam mode matching (left) vs. cavity-beam mode mismatch (right). In this particular case, the mode mismatch is a result of changing the macroscopic DC cavity length via a shift of the ETM position towards the ITM (assuming constant mirror ROC and input beam mode). Given that the mirror ROC alongside the beam wavelength remains the same, it is helpful to notice that for the new cavity configuration, arriving at the same mirror ROC within the shorter length requires a larger characteristic beam waist (shortening the Rayleigh length)

$$\begin{aligned}
 TEM_{p,l}(r, \phi, z) = & E_o \frac{\sqrt{[\lambda z_o]/\pi}}{W(z)} \left(\frac{\rho}{W(z)} \right)^p \mathbb{L}_l^p \left(\frac{\sqrt{2}x}{W(z)} \right) \\
 & \times \exp \left(-\frac{\rho^2}{W^2(z)} \right) \exp \left(-ikz - ik \frac{\rho^2}{2R(z)} - ip\phi + i(1+p+2l)\zeta(z) \right)
 \end{aligned} \tag{1.14}$$

$$TEM_{p,l}(r, \phi, z) \rightarrow |U_{p,l}(r, \phi, z)\rangle$$

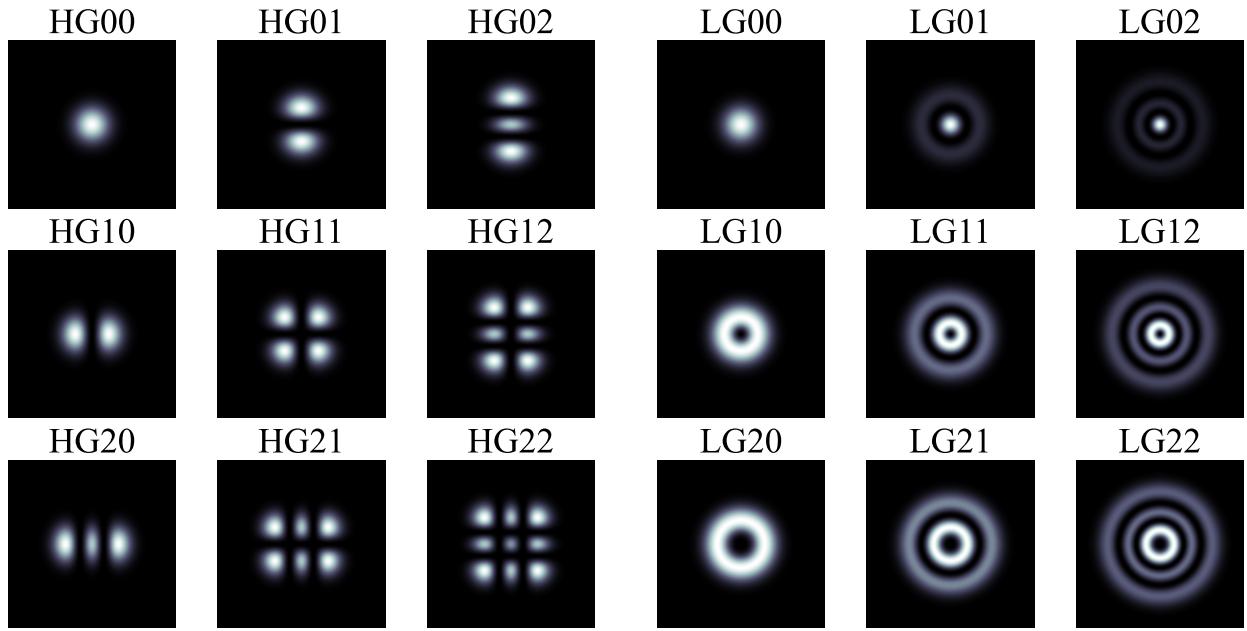


Figure 1.10: Various transverse mode intensity profiles, with Hermite-Gauss modes ($HG_{n,m}$, $[n,m \leq 2]$) to the left and Laguerre-Gauss modes ($LG_{l,m}$, $[l,m \leq 2]$) to the right.

These HG and LG field solutions arise from added perturbations in the beam-cavity alignment and mode matching respectively. The presence of power in these modes when sweeping a mirror along the beam axis (ΔFSR) indicate optical loss, while simultaneously providing useful feedback in reducing it through the construction and maintainence of a TEM00 beam-resonator coupling [7].

1.2.3 Dual-Recycled Fabry-Pérot Michelson (DRFPMI)

Recycling mirrors are an extension of the FPMI that exploits otherwise wasted optical power by providing a means of enhancing the optical gain and bandwidth of the instrument. Strategic tuning of mirror coating parameters and positions at symmetric and anti-symmetric ports can incorporate power recycling and signal recycling respectively.

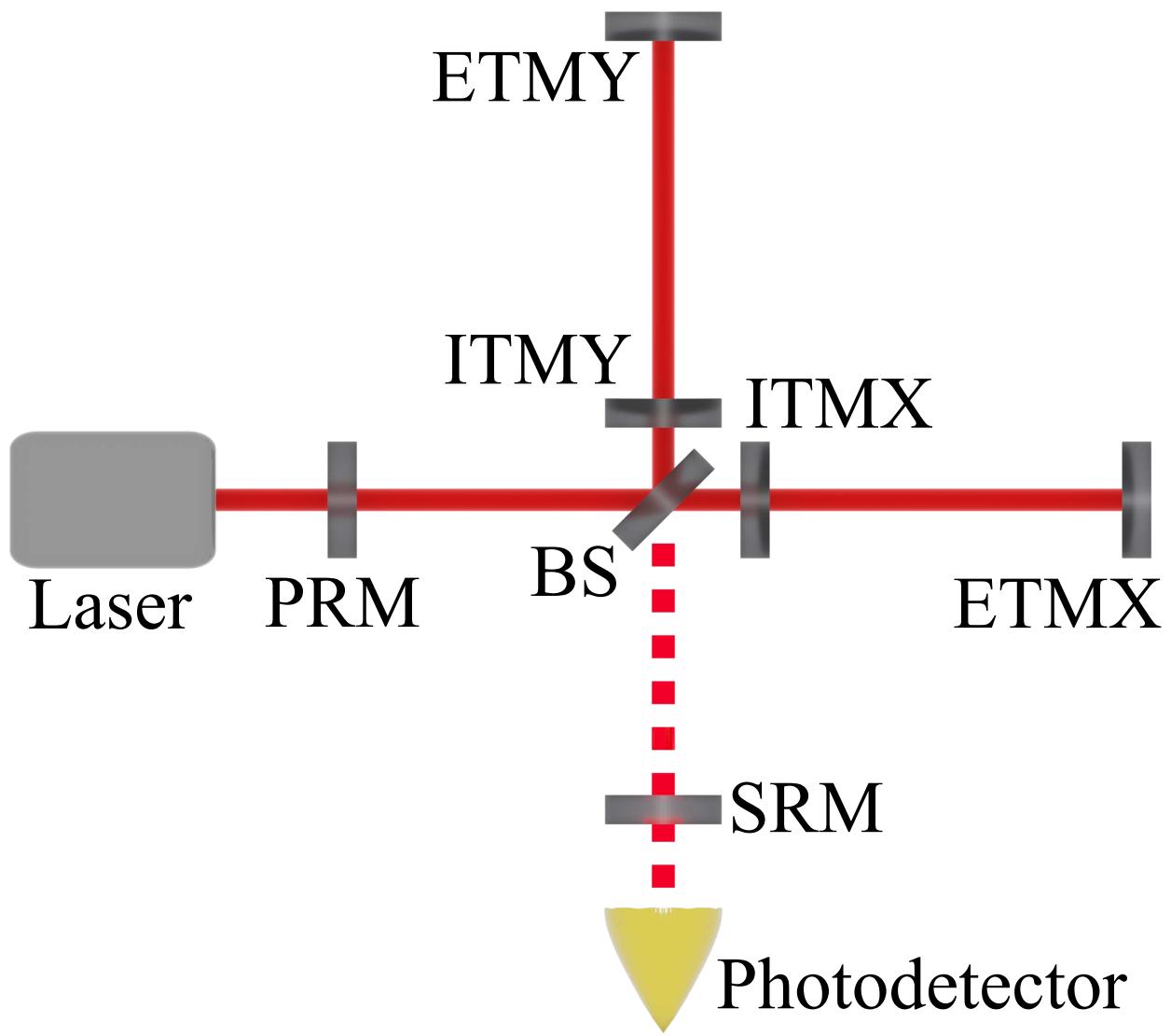


Figure 1.11: A simplified Dual-Recycled Fabry-Pérot Michelson optical schema

1.2.3.1 Power Recycling

When operating a FPMI at a dark fringe, a significant amount of power is reflected back to the symmetric port as mentioned in § 1.2.2.1 leading to wasted optical power if simply dumped. Placing an additional highly reflective mirror at the symmetric port while maintaining resonance of the carrier to the arms, you can reintroduce (or “recycle”) power back to the arm cavities. A PDH loop is utilized for carrier resonance, while macroscopic mirror positioning of the PRM is informed by the choice of optical sideband frequency required

when applying PDH. This recycling gain is also sensitive to cavity arm Finesse \mathcal{F} and round trip loss \mathcal{L}_{RT} :

$$G_{\text{PR}} = \frac{(1 - r_{\text{PRM}}^2)}{1 - r_{\text{PRM}}[1 - (\mathcal{F}\mathcal{L}_{\text{RT}})/\pi]} \quad (1.15)$$

With maximum recycling gain:

$$G_{\text{PR}}^* = \frac{\pi}{2\mathcal{F}\mathcal{L}_{\text{RT}}} \left[\frac{1}{1 - \frac{\mathcal{F}\mathcal{L}_{\text{RT}}}{2\pi}} \right] \quad (1.16)$$

1.2.3.2 Signal Recycling

As may be inferred, this technique requires a mirror installation at the anti-symmetric port, though with a more nuanced approach than that of power recycling. Placing a partially reflective mirror at the output port, it is understood that light leakage coming from the PRFPMI at the anti-symmetric port (caused by differential arm motion) is re-introduced to the arms, but the reflectivity of the new mirror cannot be set too high to prevent attenuating the PRFPMI output. And even then detector enhancement only comes after exacting a cost dependent on signal recycling cavity tuning. This cost resides between a trade off of detector bandwidth for increased detector gain or vice versa, with the operating point between the maxima of these two detector characteristics set as a function of the microscopic signal recycling cavity length (phase) tuning [8]:

$$H_{\text{DRFPMI}} = G_{\text{PR}} P_{\text{in}} L \Omega \left[\frac{t_{\text{ITM}}^2 r_{\text{ETM}}}{(t_{\text{ITM}}^2 + r_{\text{ITM}}^2) r_{\text{ETM}} - r_{\text{ITM}}} t_{\text{SRC}} \frac{e^{-i2\pi Lf/c} \sin(2\pi f/c)}{2\pi Lf} \frac{\sin(\phi_0)}{1 - r_{\text{SRC}} r_{\text{ETM}} e^{-i4\pi Lf/c}} \right] \quad (1.17)$$

$$t_{\text{SRC}} = \frac{t_{\text{SRM}} t_{\text{ITME}} e^{i\phi_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{i2\phi_{\text{SRC}}}}, \quad (1.18)$$

$$r_{\text{SRC}} = \frac{r_{\text{ITM}} - r_{\text{SRM}} e^{i2\phi_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{i2\phi_{\text{SRC}}}} \quad (1.19)$$

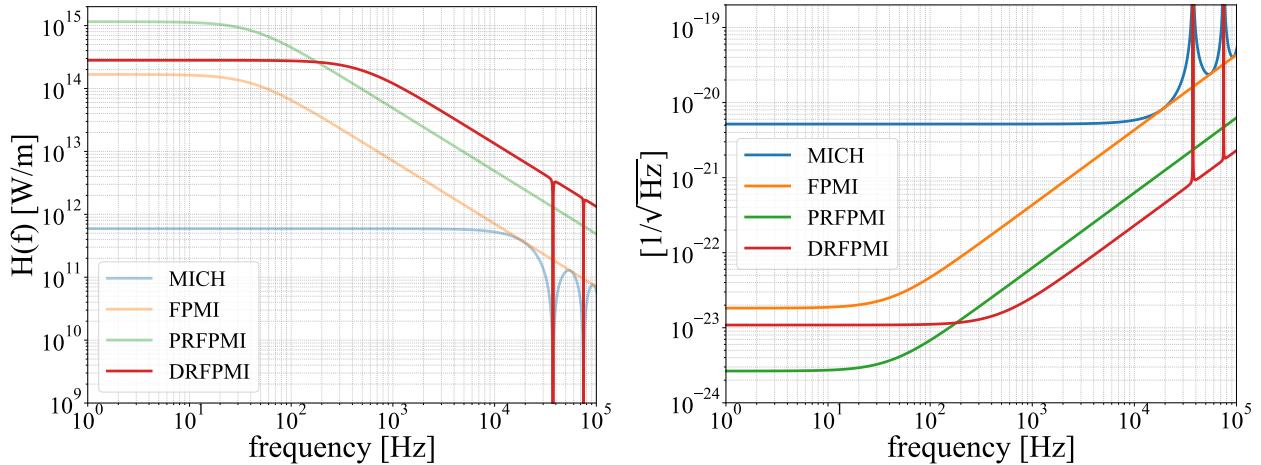


Figure 1.12: [Left] Comparison of all optical gain functions [Right] Coorelated shot noise strain sensitvity. Code and more detailed derivations used to generate optical gain and sensitivity curves can be found in § 5.1

1.3 ALIGO

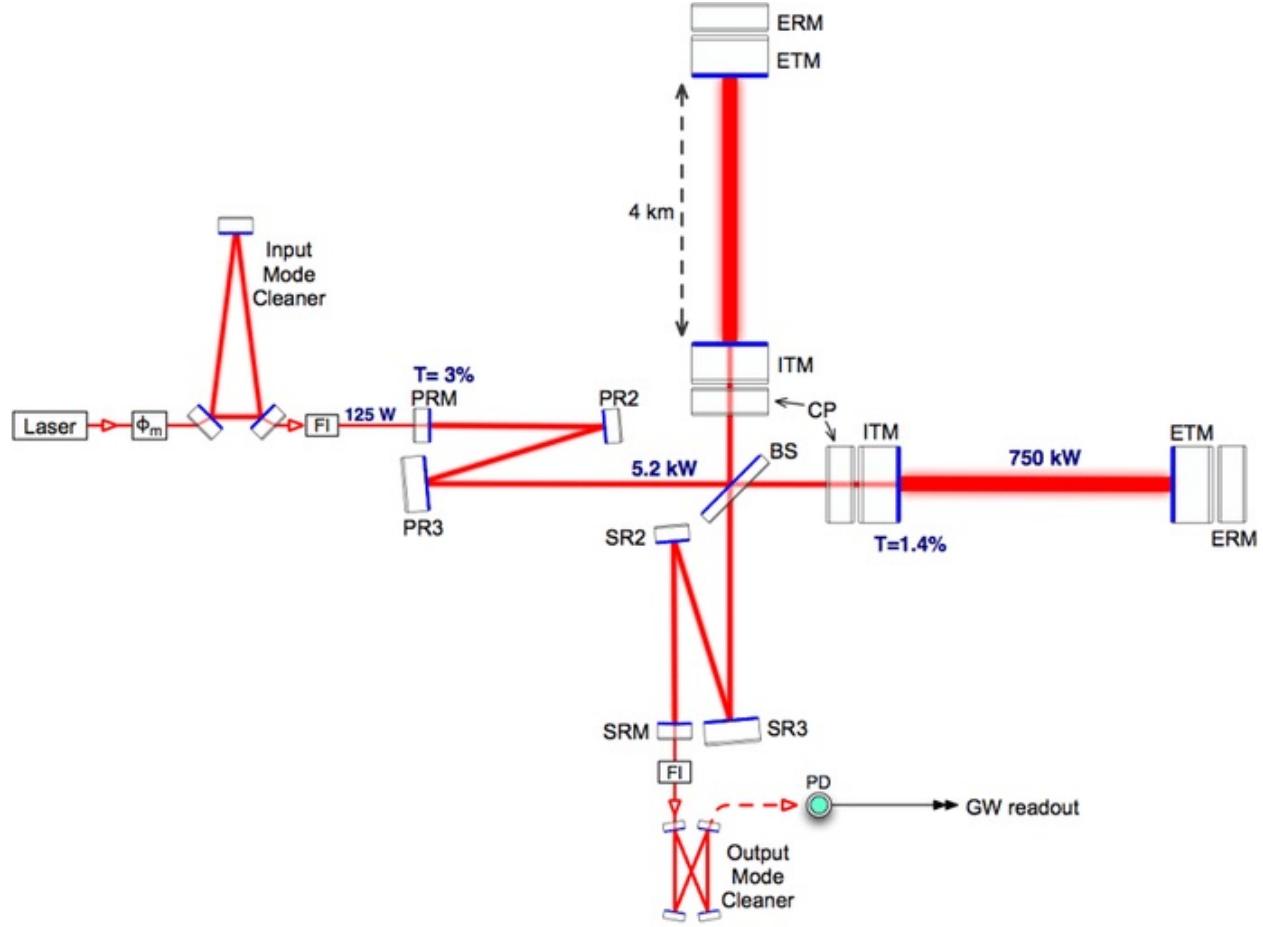


Figure 1.13: DRFPMI configuration used in ALIGO

“Core optics” (Recycling mirrors, Beam splitter, and FP arm cavity mirrors) are suspended with quadruple pendulum suspensions decoupling seismic activity from the mirror positions to as low a frequency as possible.

1.3.1 Thermodynamic considerations

Discussions prior to now still have yet to discuss most practical considerations required to operate a DRFPMI as a gravitational wave observatory. For the sake of transitioning to the niche body of this work, I provide a brief discussion of select detector features: 1) adap-

tive optics for high power operation, and 2) the thermodynamics of highly reflective mirror coatings that impose a fundamental limit to gravitational wave interferometer sensitivity.

1.3.1.1 Adaptive Optics

As mentioned in §1.2.2.1.2 , the microscopic longitudinal control of the mirror positions is only half the story for Gaussian beams and further consideration of macroscopic mirror position and radii of curvature are needed to maximize resonant power in the fundamental (TEM00) mode. Failure to plan and maintain this “mode matching” condition results in beam mode to cavity mode mismatch, scattering power into higher order Laguerre-Gauss modes. Additionally, even with ultra-low absorption HR mirror coatings and fused silica substrates, aLIGO circulating power is estimated to reach ≥ 750 kW, introducing a differential defocus to the arm cavities by [m^{-1}]; which can introduce significant optical loss due to mode mismatch (esp. for a coupled cavity configuration) [9]. As DRFPMIs like aLIGO approach designed sensitivity, instances of mode mismatch are a two-fold threat with optical loss to higher order modes that impact the ability to produce squeezed light states [10]. The further motivation and the implemented solution for aLIGO during O3a is discussed in Chapter 2.

1.3.1.2 Coating Thermal Noise

Generally speaking, sensed differential arm motion is more often than not produced by sources that are not gravitational waves. The sum and categorization of motion from non gravitational wave sources (both known and unknown) at a given point in time form what is known as a DARM noise budget. This tool aids commissioners in understanding the current limits of GWDs and additionally facilitates focused hypotheses for detector improvements / upgrades. One rapidly approaching limit is the coating thermal noise from the HR $\text{SiO}_2\text{TiO}_2 : \text{Ta}_2\text{O}_5$ Fabry-Pérot cavity mirror coatings which arises from the mirror surface position observable influenced by energy dissipated by the coating by way of its acoustic

degrees of freedom and causing uncoorelated phase fluctuations between the two arm cavities. Part of the upgrade discussions for current and future GWDs are HR coating materials with different material composition solely for the purpose of lowering this coating thermal noise. A promising candidate with 5 times lower thermal noise properties is the HR crystalline GaAs/Al_{0.92}Ga_{0.08}As stack. This alone motivates thorough inquiry into more of its properties, some of which are discussed in Chapter 3.

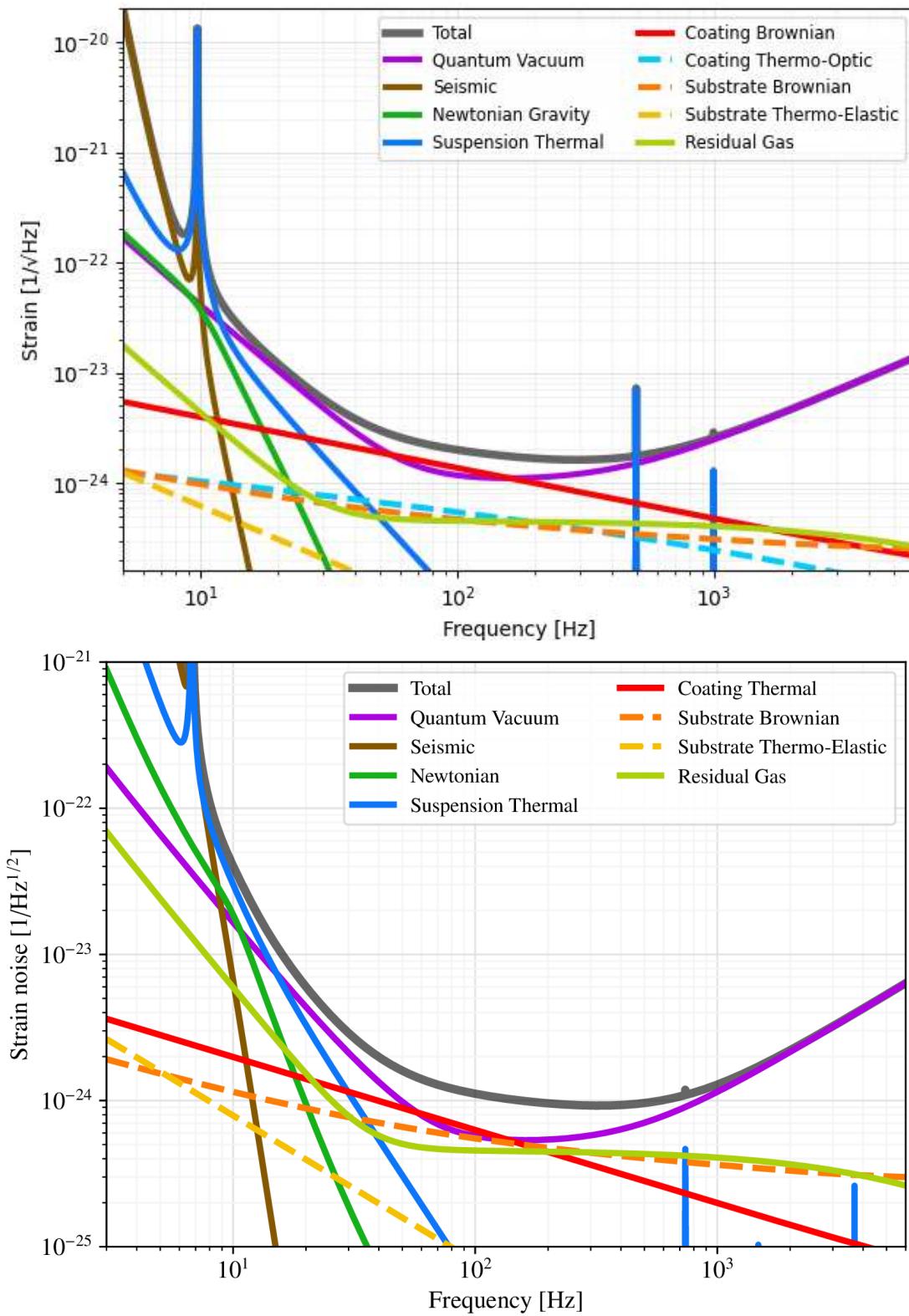


Figure 1.14: [Top] Noise budget for the next generation A+ interferometer using $\text{SiO}_2\text{TiO}_2 : \text{Ta}_2\text{O}_5$ coatings [Bottom] Noise budget for a parallel next generation A \sharp interferometer using $\text{GaAs}/\text{Al}_{0.92}\text{Ga}_{0.08}\text{As}$ coatings [11].

Chapter 2

Comissioning Adaptive Optics for O3a

2.1 Motivation

As seen in § 1.2, increasing detector input power leads to a direct sensitivity increase to gravitational waves. And even using optics with ultra-low absorption ($\approx 328 \text{ ppb} \pm 84 \text{ ppb}$), significant thermo-optic effects persist, especially with a designed circulating arm power of 750 kW in the Fabry-Pérot cavity arms [9, 12]. Thermal aberrations produced from the high circulating carrier power include substrate lensing and relatively smaller lensing from the differential HR surface curvature. The time varying optical path length change integrated over the carrier phasefront produces mode mismatch and contributes to the accumulated optical loss throughout the GWD, reducing sensitivity two-fold: 1) by loss of usable readout power, and 2) reduced efficacy producing squeezed light states in lowering the detector quantum noise limit.

During O3a the LIGO Hanford observatory increased circulating arm power beyond 180 kW; emphasizing importance on properly tuned thermal compensation in O3 to avert arm-cavity/carrier-beam mode mismatch. Detailed in this chapter is a summary of related comisioning efforts at LHO to prepare and preserve interferometer mode matching including but not exclusive to: a primer on the ALIGO adaptive optics schema (TCS), citations on the

initial computed O3a TCS pre-load, the development and implementation of real-time digital filtering for an improved ring heater actuation response by a factor of ≈ 6 , and the impacts of high absorption points aka point absorbers discovered on arm cavity test masses along with efforts to mitigate them.

2.1.1 Thermal Compensation System

High power beams, even propagated by ultra low absorption mirror substrates and coatings, can impart a surface pressure that imposes non-negligible thermo-optic distortions via thermo-refractive and thermo-elastic effects [13]. The ALIGO adaptive optic system is intended to address the problem of dynamic mode mismatch in high power interferometry; as high power operation is a requirement in reaching designed sensitivity. The system is comprised of a feedback control system that uses four Hartmann wavefront sensors (HWS) combined with thermal actuators of two varieties: annular ring heaters and CO₂ lasers heating [14].

2.1.1.1 Actuation

Both ITMs and ETMs (Fabry-Pérot arm cavity mirrors) are strategically monitored for differential lensing, but both are not prescribed equal actuation treatment. All arm cavity mirrors do possess negative lens ring heater actuation in the form of a wound nichrome wire annulus that outlines the outer barrel of the mirror substrate; while CO₂ lasers, though not imaged onto the ITMs directly, are instead imaged onto a fused silica compensation plate (CP)¹ placed promptly before the FP arm input coupling mirror [15].

2.1.1.2 Sensing Optical Path Distortion

Quantifying thermal distortion from both carrier as well as thermal actuators is performed with a set of four Hartmann wavefront sensors; each one measuring differential optical path

¹Decouples CO₂ laser noise from the highly sensitive FP input test mass position

distortion at each FP cavity test mass. The sensor probe beams ² make a double pass through the test mass mirror substrate for all arm cavity mirrors and map the HR mirror surface; while the two input test mass sensors at the interferometer vertex make an additional double pass through the compensation plate (CP). Wavefront distortion maps are then used to compute relevant Zernike polynomial coefficients (i.e. $Z_{n=2}^{l=0}$) in real time to inform of differential defocus in diopters [16].

2.1.2 Thermo-optic transients

Thermo-optic transfer functions from high power carrier beams and implemented thermo-optic actuators suggest that these thermal transients are worthy of consideration to avoid long periods of mode mismatch while sampling points in actuation space.

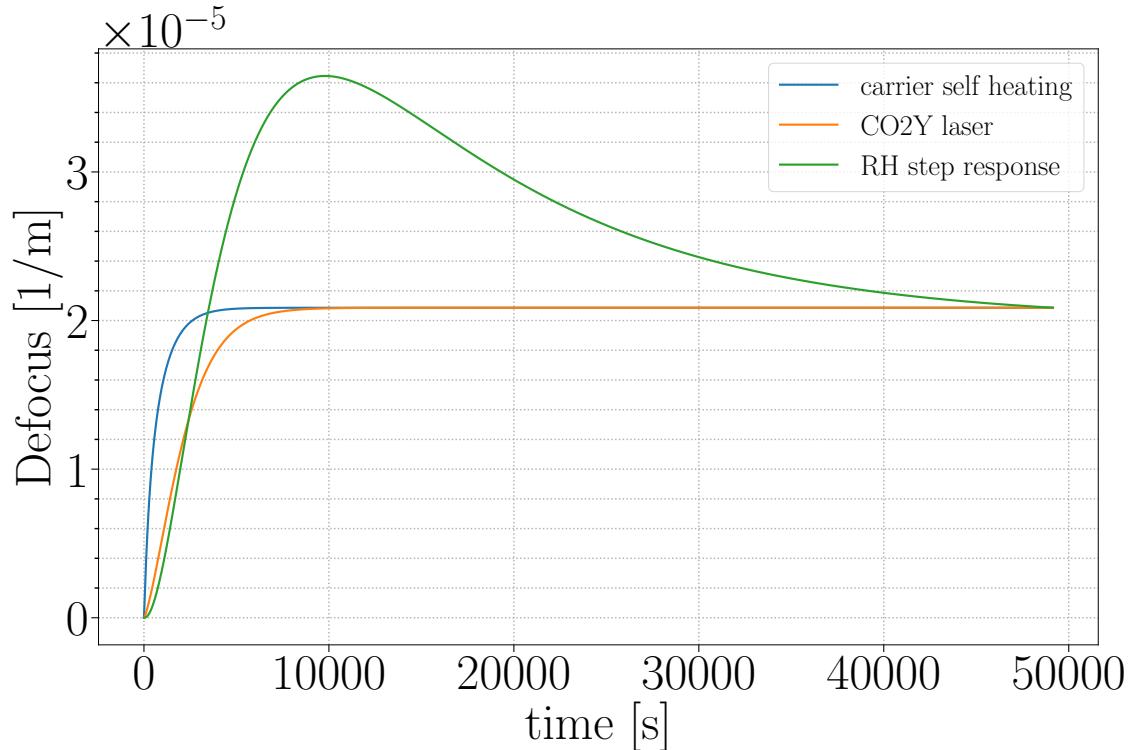
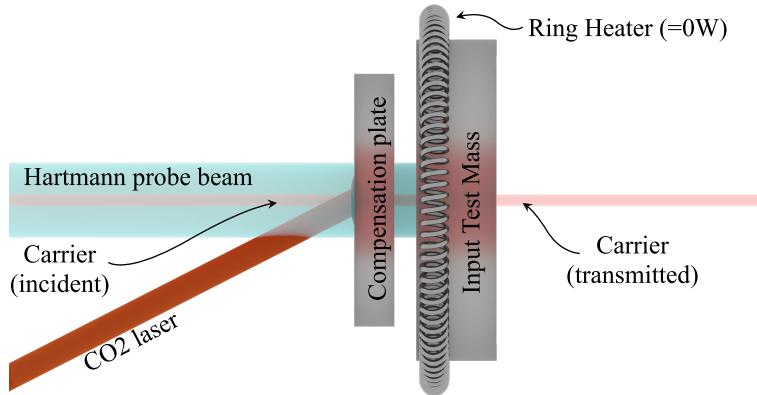
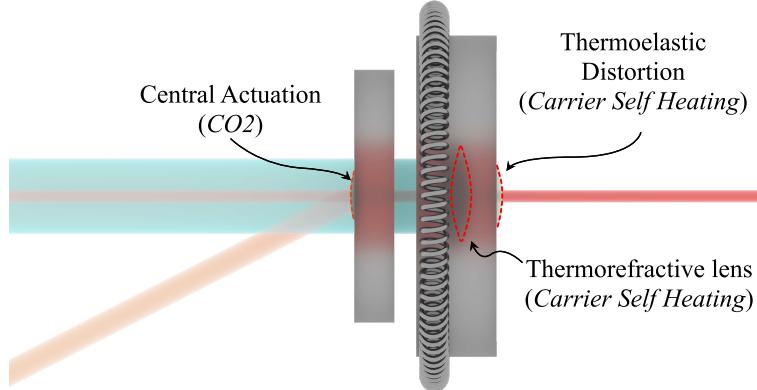


Figure 2.1: Transient defocus responses computed from carrier beam self heating and TCS actuation best fit filters (central CO2 laser heating and annular ring heating) § 5.6.

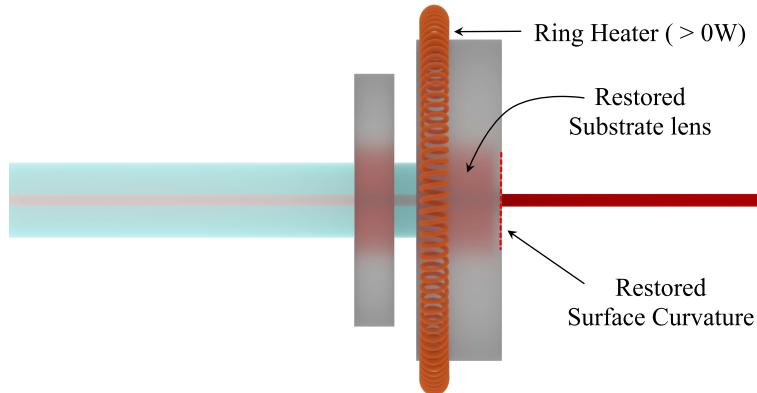
²Differing wavelengths of 800 nm and 833 nm are chosen for the X and Y arms in order to mitigate crosstalk between HWS chains and other auxiliary systems



(a) CO₂ actuator set to replicate projected carrier thermo-optic response, with an off resonance circulating beam.



(b) Arm cavity resonance, with reduced CO₂ central actuation power and increased arm cavity input power. The uniform thermo-optic distortion from the high power circulating carrier imposes a differential thermo-refractive lens and thermo-elastic HR surface change to the ITM, placing an upper limit to the circulating carrier power without annular ring heater actuation.



(c) Maximum circulating arm power, with annular heating and no central CO₂ actuation. The careful timing and calibration of the CO₂ / RH actuators can allow designed power / GW detector sensitivity to be reached.

Figure 2.2: ALIGO thermal compensation design at the input of a single Fabry-Pérot arm cavity. Though not the only location of thermal mode matching actuators, a careful look here demonstrates their capabilities and motivates carefully constructing a thermal pre-load strategy before commissioning and fine tuning during commissioning of the detector at high power.

The thermo-optic time constant of the carrier beam self-heating is similar to that seen from CO₂ laser / CP central actuation, though demonstrably different from annular ring heating. Because of this, LHO applies central CO₂ heating and static annular ring heating to a power level that respectively mimics and actuates for projected thermal deformation from circulating resonant carrier in the Fabry-Perot arm cavities. Once DRFPMI coupled cavities are configured or “locked”, the input carrier power is gradually increased while CO₂ laser power is simultaneously decreased in order to mitigate any possible differential thermo-optic response from the arm cavity test masses when reaching maximum power.

2.2 Dynamic Thermal Compensation

Transient ring heater actuation from a radially symmetric thermal aberration ($\Psi(t, r)$) is realized in [17] and measured here:

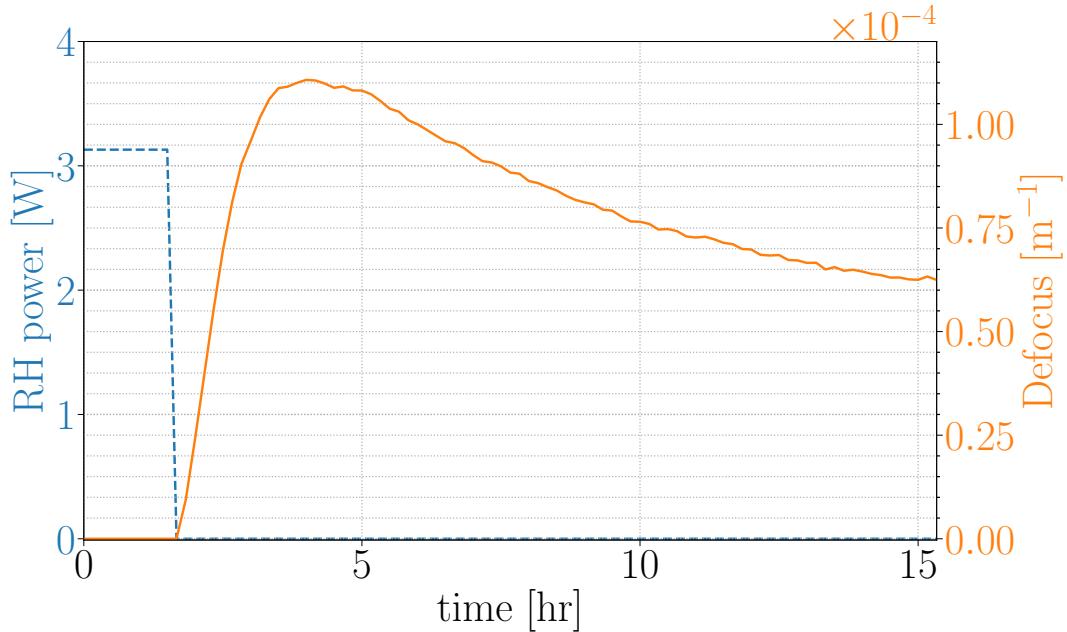


Figure 2.3: ITMY thermo-optic response to a 3.13 [W] combined power reduction to the top and bottom ring heater elements. It's after ≈ 12 hours after the ring heater power control step do you start to see a small enough steady state differential defocus ($\frac{da_{sp}}{dt}$) and can assume a steady state thermal lens.

The measured transient thermo-optic response exhibits differential defocusing for ≈ 12

to 15 hours once the ring heater power has been changed; and with a large enough power steps, these adjustments to ring heater power can significantly stall precious detector observing/comissioning time due to differential mode matching. Thermo-optic time constants are reduced by applying real time digital filtering to ring heater power controls. The desired thermo-optic response is one that resembles a step from one defocus state to another with no overshoot.

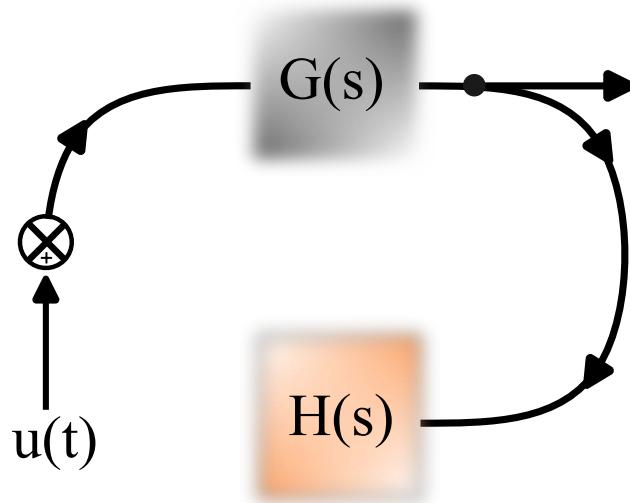


Figure 2.4: A simple controls diagram depicting the concept of a RH control pre-filter ($G(s)$) taking a step input ($u(t)$) and transforming it before reaching the thermo-optic plant ($H(s)$) with output can also be generalized to an adaptive optics control loop.

The RH power control transient ($u(t)$) resembles that of a Heaviside function ($\Theta(t)$), motivating an inversion of the response function which provides a reasonable first order filter correction. Therefore, the prescription for creating an input filter is realized through inverting the known RH step response with additional low passing at high frequency to avoid any high frequency control instability ³.

³see chapter 5 for more detail

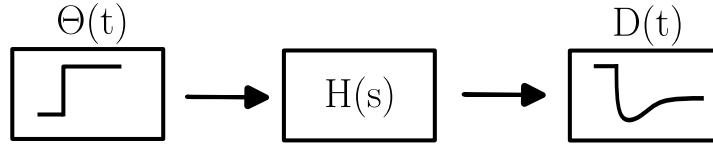


Figure 2.5: A pictograph of the plant system (test mass mirror and annular ring heater) transforming the ring heater power control step to a time-varying thermo-optic response. The example of this can be seen in Fig [2.3]

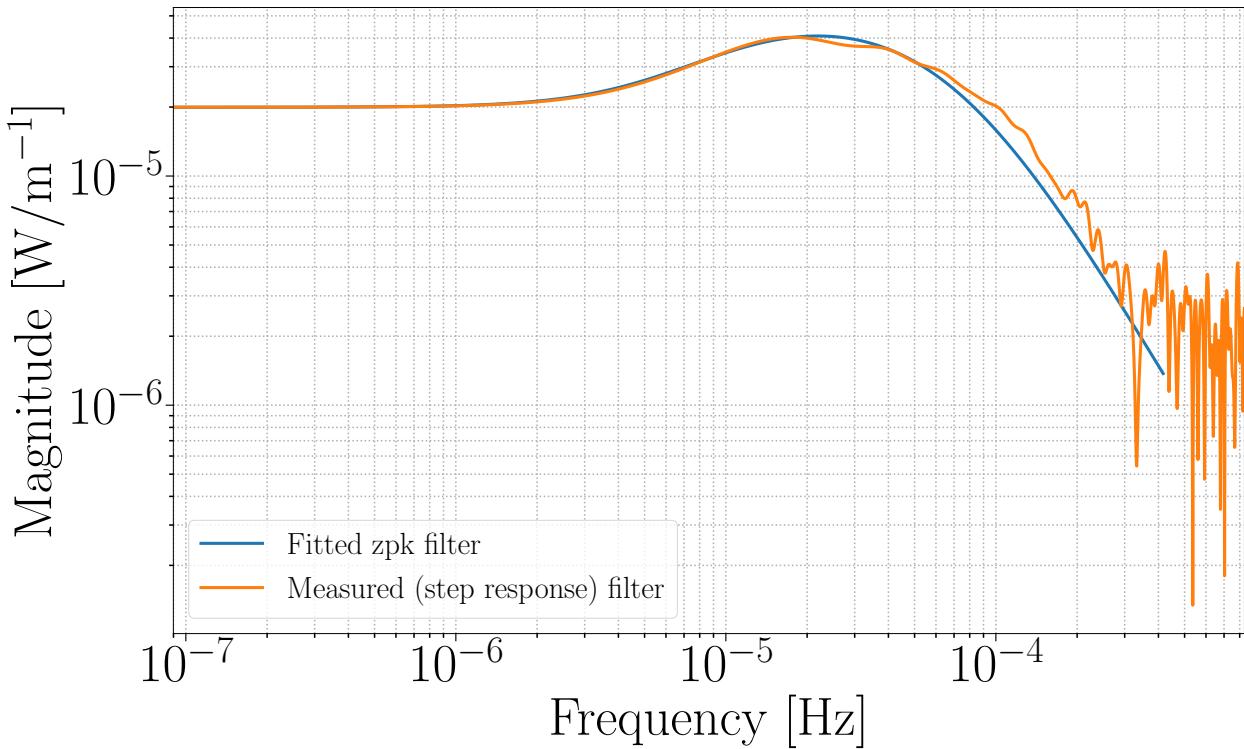


Figure 2.6: Showing the PSD of the RH response (normalized by the input RH power) over a an ≈ 12.5 hour period. The zpk model of the fitted filter ($H(s)$) = $9.2545e-12 \left(\frac{(s+3.14210e-5)}{(s+8.168e-5)(s+0.0003142)(s+0.0005969)} \right)$

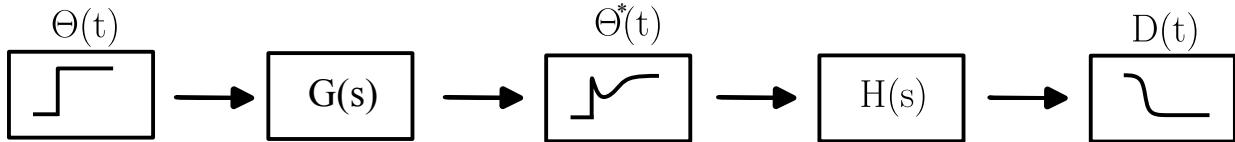


Figure 2.7: A pictograph showing the system with real time digital filtering for an improved thermo-optic response. The RH input filter ($G(s)$) is created by inverting the plant filter combine with a low pass and added poles to the zpk model to ensure stability. The time series $\Theta^*(t)$ illustrates the modified RH control input for an improved defocus responce ($D(t)$).

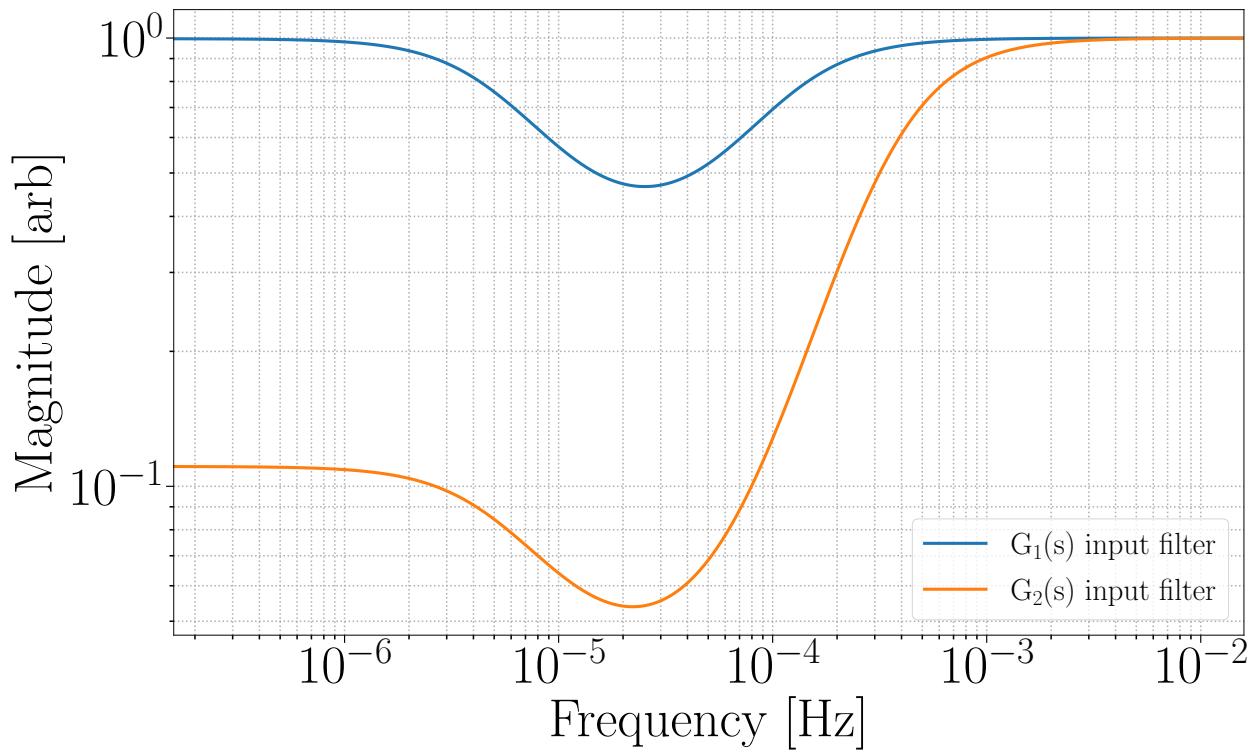


Figure 2.8: Two of the constructed RH pre-filter options:

The zpk models of the two RH pre-filters plotted in Figure 2.8:

$$\begin{aligned}
 G_1(s) &= \frac{(s + 8.16814090e-5)(s + 3.14159265e-4)(s + 5.96902604e-4)}{(s + 3.14159265e-5)(s + 6.99400000e-4)(s + 6.99400000e-4)} \\
 G_2(s) &= \frac{(s + 8.16814090e-5)(s + 3.14159265e-4)(s + 5.96902604e-4)}{(s + 3.14159265e-5)(s + 2.09820000e-3)(s + 2.09820000e-4)}
 \end{aligned} \tag{2.1}$$

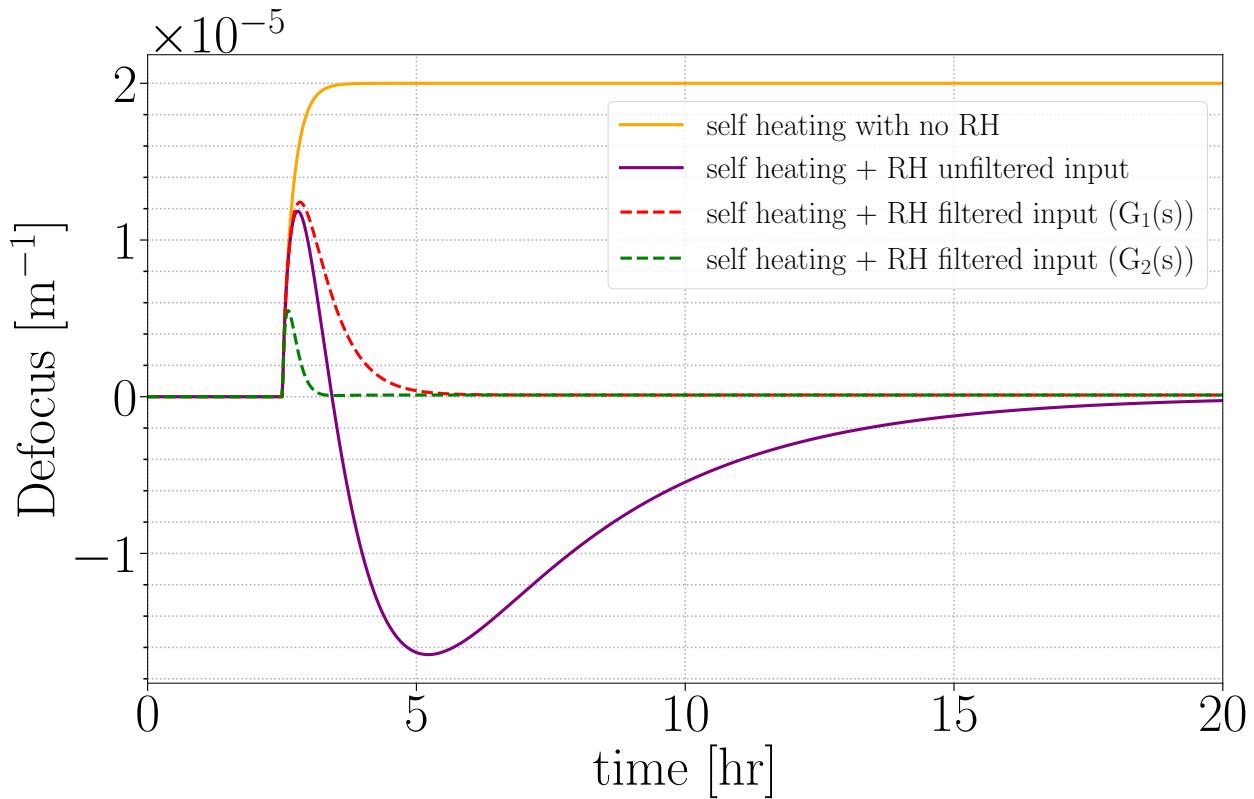


Figure 2.9: Comparison of the natural RH response and the response with the pre-filtered input against the transient carrier absorption response. “self heating” indicates a dataset generated by a COMSOL simulation computing the transient thermo-optic response from 1 [W] of absorbed carrier beam and scaled down. “self heating + RH unfiltered input” demonstrates the simultaneous transient response when changing the ring heater power alongside the carrier thermo-optic transient. The dotted curves indicate the elimination of long defocus period when changing RH power alongside the carrier transient using the two RH pre-filters provided Figure 2.8.

2.2.1 Reducing Parametric Instabilities

Another symptom of resonant high power optical cavities are parametric instabilities (PI); induced by the opto-mechanical interaction between test mass acoustic modes and higher order optical modes. PI's present a threat to achieving designed detector sensitivity, even driving the detector to lockloss. Passive methods of mitigating PI's by way of acoustic mode dampers (AMD) demonstrate significant reductions of problematic mechanical modes though some (i.e. @ 15 kHz) remained problematic during O3a. Lingering PI's required manual intervention by way of adjusting test mass / cavity geometry to disrupt persistent

modes and is now a much more feasible solution with DTC [18].

2.2.2 Limitations

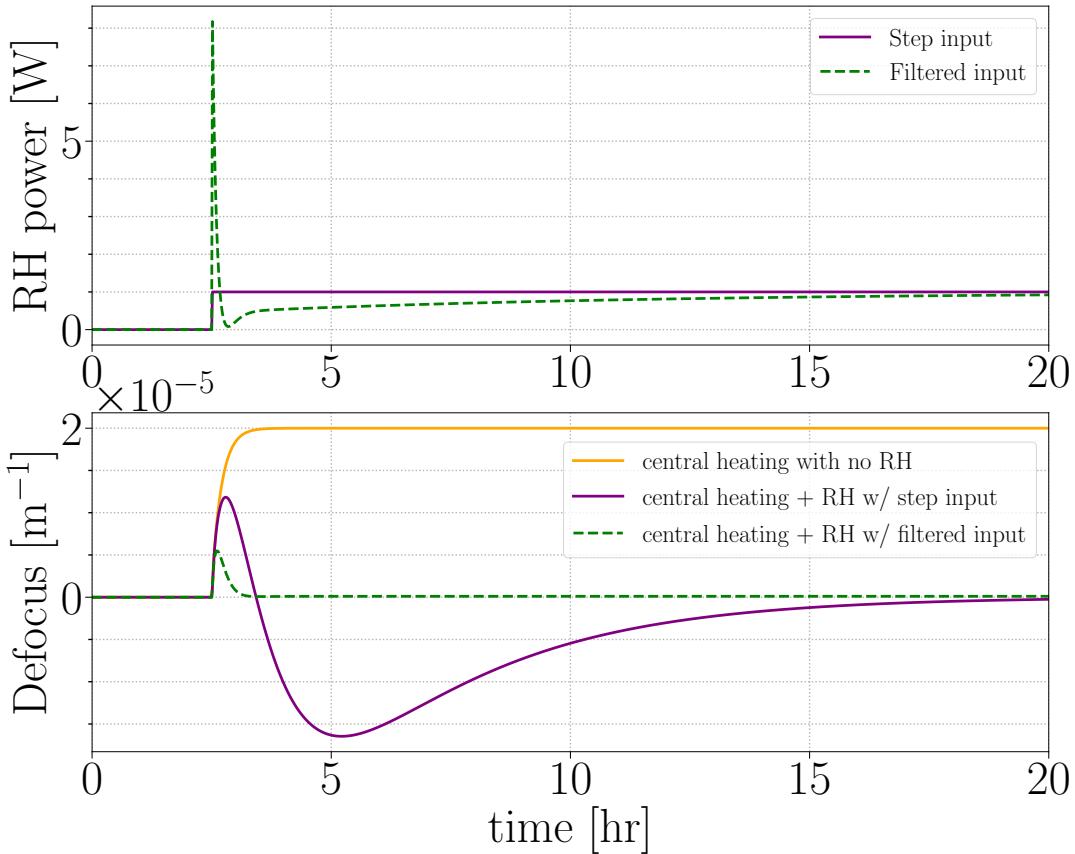


Figure 2.10: Comparison of the natural RH response and the response to the filtered input with RH power

When constructing the control pre-filter important considerations the physical limits of the thermal actuators need to be considered. For the ALIGO RHs, a 10 W limit is set on RH power control and sets a threshold on the high frequency gain of the pre-filter [19]. Failure to consider this limit may cause the actuator output to rail the initial power swing from your actuator response.

There must also be considerations when we cannot assume the scaling of this function is not linear and time-invariant. Concurrent RH changes an example of this as the filter is designed with initial thermal equilibrium in mind. There is room to develop a more robust Dynamic

Thermal Compensation strategy, with enough modeling, independent measurements spanning an actuation space, and integration of Kalman Filtering.

2.3 A priori TCS pre-load methodology for O3a

Preserving arm cavity resonance requires countering the positive thermal lens defocus of the nominal test mass lens induced by high circulating interferometer arm cavity power. Preparing for uniform test mass distortion from the carrier beam requires calibrated and well established thermal actuator settings; informing a ‘pre-load’ of the TCS actuators using test mass absorption measurements. Initial order of magnitude estimates of wavefront distortion from ultra-low absorption fused silica test masses under the influence of a centered high power gaussian beam as well as annular ring heater actuation are available [13, 17]; though variations of the absorption between any two test mass mirrors are accounted for through calibrated defocus measurements using the Hartmann wavefront sensors sensitive to auxiliary beams imaged onto the test mass mirror surfaces.

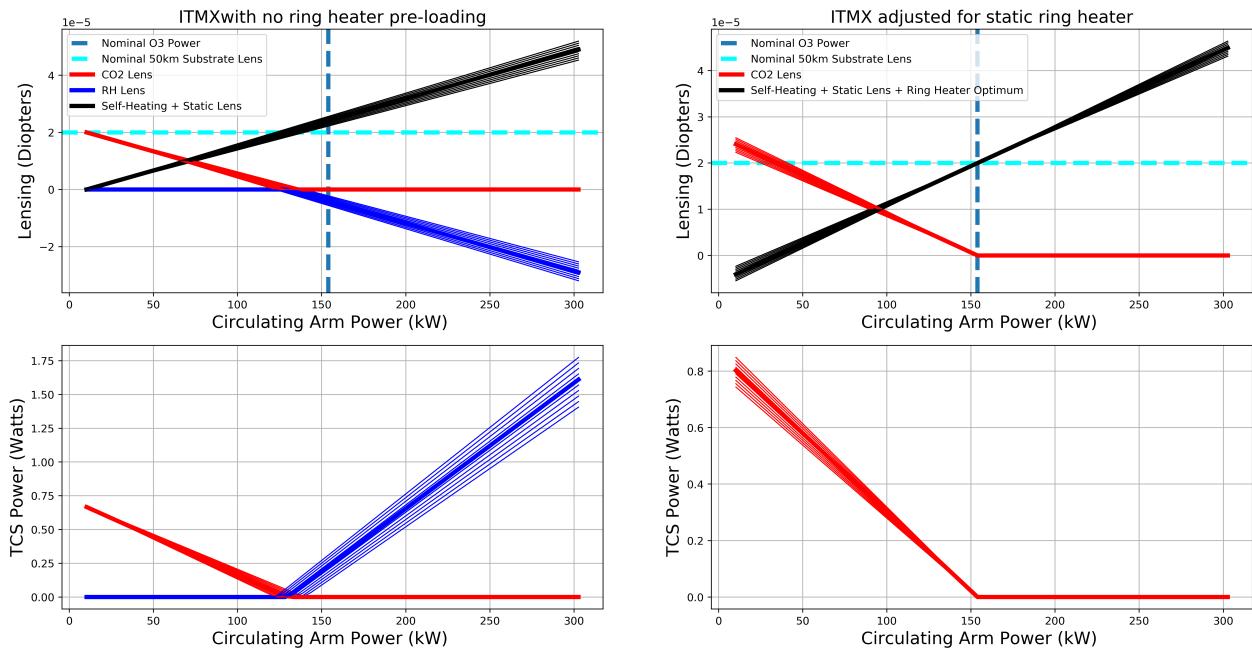


Figure 2.11: ITMX initial pre-load estimate O3a as provided in [9]

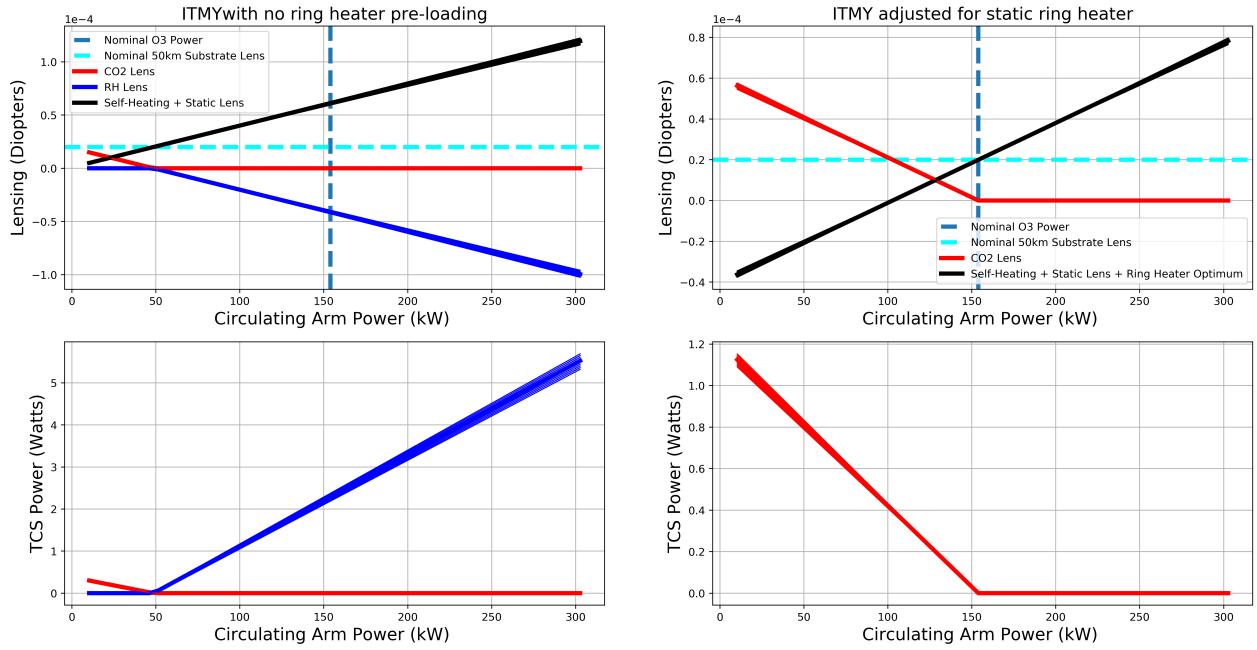


Figure 2.12: ITMY initial pre-load estimate O3a as provided in [9]

2.4 A posteriori thermal compensation for O3a

While approaching designed arm cavity power, the presence of non-uniform high absorption points on the test mass coating surface imposed limits to reaching designed power and hence designed sensitivity; simultaneously, this lead to a significant deviation from the original TCS pre-load algorithm. The current thermal actuation solution is designed to control the TEM00 beam waist size and location, though adjustments and modifications of current actuators were tried. Assessment of these absorbers helps inform of ways to mitigate their effects with the hopes of reaching the designed detector sensitivity. We begin here with a characterization of these high absorbtion points including: characteristic optical path distortion profiles measured on the Hartmann wavefront sensors, as well as some impacts on interferometer operations at high power. The findings mentioned here indicate that these absorbers pose a risk to maintaining and reaching designed circulating power in the arm cavities if no further proactive measures are taken, or are not sufficient to bypass detector

symptoms; whether they are a result of preventable surface particulates or can be masked with an improved adaptive optics schema.

2.4.1 Point absorption in O3a

A significant number of lockloss events during the O3a comissioning period were a direct result of select optical sideband power degradation used to maintain the delicate coupled cavity configuration during interferometer thermalization at input power ≥ 100 kW circulating in the FP arms. This was quickly realized to be caused by high absorption points discovered on ITMY.

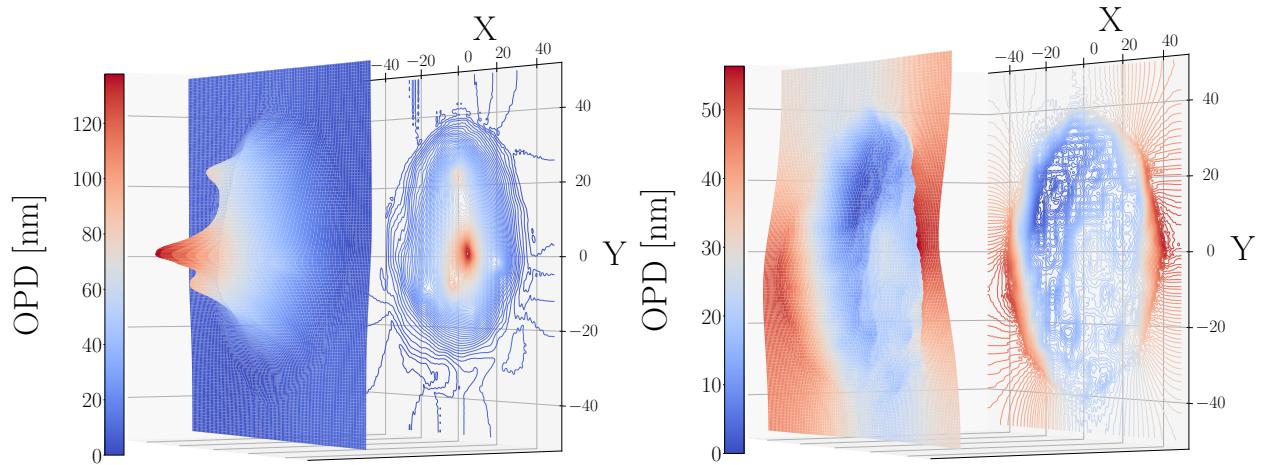


Figure 2.13: An isometric view of point absorption vs uniform cooling of the LHO ITMY.

Also, while sustaining interferometer DC readout the recycling cavities would generate higher order modes, sustained by a Output Mode Cleaner (OMC) co-resonance which contaminates the carrier field at the output photodiode. With continued observation and comissioning at high power (≥ 180 kW) point absorbers were also discovered on ETMX, though difficulties profiling due to unavoidable probe beam noise on that wavefront sensor.

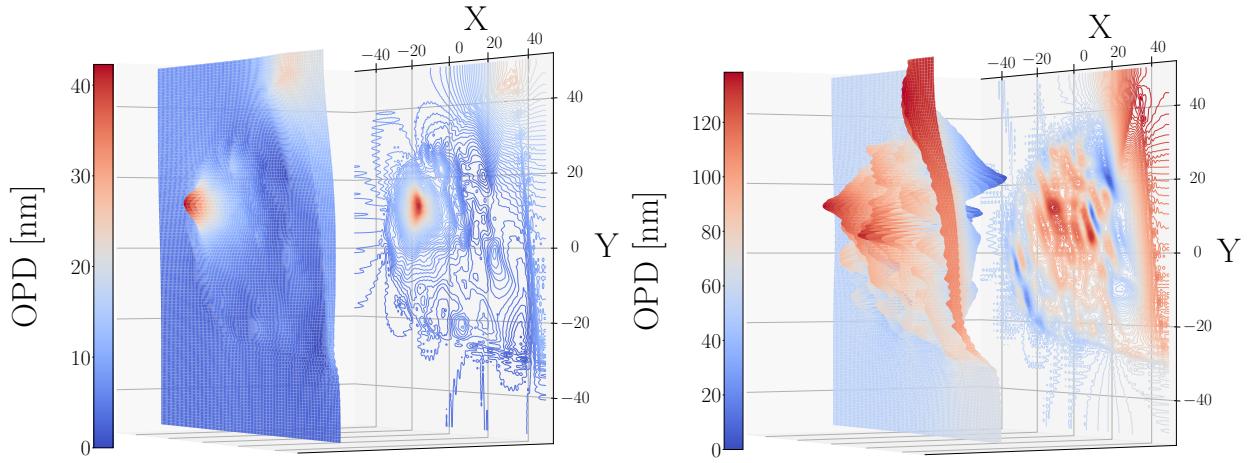


Figure 2.14: An isometric view of uniform absorption vs point absorption of LHO ETMX. The rippling / edge effects are a consequence of the Hartmann probe beam clipping on the baffle due to misalignment of in-vaccum optics.

2.4.1.1 Reduced Power Recycling Gain

As informed by Equation 1.15 the PRG can provide a reasonable metric for the arm losses in the FPMI and in fact suffered notable reductions with interferometer thermalization. At LHO there are reported PRG losses of $\approx 13\%$ increasing IFO input power from 2W to 30W, while LLO notes a $\approx 18\%$ change while increasing input power from 2W to 40 W [20].

2.4.1.2 Frequency noise

A beneficial feature of interferometry is the ability to filter common mode noise (i.e. laser intensity and frequency noise), though these benefits are contingent upon the ability to preserve arm symmetry. Intentional asymmetries, for example the Schnupp asymmetry⁴ are well known and designed such that common mode noise remains low. Meanwhile, common arm mode interference is exacerbated proportionally to the amount of mode mismatch.

⁴A macroscopic differential length offset introduced in order to pass optical sidebands for normal interferometer functionality and stability

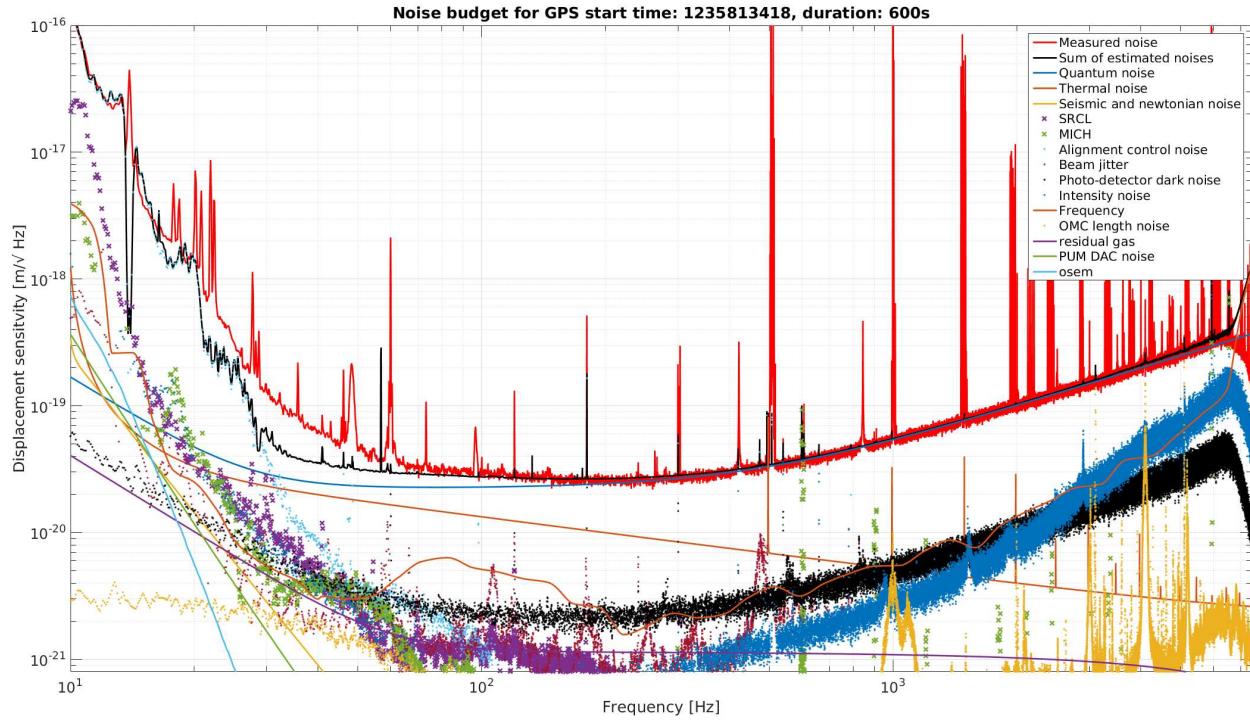


Figure 2.15: A DARM noise budget during O3a [21]. Beneficial features of interferometry, i.e. common mode noise rejection requires adequate mode overlap at the beam splitter. Destruction of such symmetry as mode overlap proportionally destroys such benefits and demonstrate a presence in DARM. Here in this noise budget both Frequency and Intensity noise are about 2 times above the anticipated noise level.

Thorough investigation of the LIGO frequency noise determined the proportional increase of the LIGO frequency and intensity noise with increased power [22].

2.4.1.3 Control signals

Holding delicate resonance conditions throughout the coupled cavity configuration necessitates a resonable optical sideband relative carrier gain (dBc) for the PDH loops to function. And it is with careful planning that the macroscopic recycling cavity geometry is planned so that sideband resonance boosts the signal. The point absorber on an ITM directly impacted these sideband buildups and frequently lead to a lockloss.

2.4.1.4 Higher order modes at DCPD

For the detectors' DC readout configuration, power from higher order modes at the anti-symmetric port obscure the single element photodiode from measuring a pure TEM00 mode with the increased shot noise [23]. Among other higher order modes, the point absorber allowed resonance of a notable higher order mode (9th order Hermite-Gauss) from the 9MHz sideband which was co-resonant with the Output Mode Cleaner and in turn imparted intensity noise to the DCPD [24].

2.5 Results

With these high absorption points dramatically impacting interferometer high power operations, a variety of efforts to mitigate their effects were sampled. Using DTC the amount of sampling possible TCS states increased (for varied RH states specifically) though the current modes of actuation proved to still be too few and/or mixed to rely on TCS alone with a static beam position on ITMY.

2.5.1 Custom CO₂ mask

Attempts at increasing uniform distortion with a machined CO₂ mask § 5.8 was tried. The intention was to actuate by imaging a negative of the optical path distortion from the high absorption points onto the surface with the CO₂ laser combined with a static ring heater offset [25]. The installation location of the mask and size was established using the relevant propagation and imaging techniques applied to the CO₂ actuation field while mitigation of the aforementioned impacts provided commissioners with mixed metrics of success [26, 27]. With varying interferometer conditions and an involved parameter space, alongside the slow nature of thermal actuation, attempts at restoring uniform absorption proved to be not as straightforward to improving the overall interferometer condition [15, 22].

2.5.2 Beam position offset

Commissioners found that the simplest and most accessible solution to reducing point absorber symptoms (i.e. decreasing power recycling gain, control sideband buildup, and reduced 9MHz intensity noise) was to shift the beam spot position on the offending test masses [28] though this was not without having to change the entire interferometer alignment along with the loops that control it [29]. Utilizing this solution also simplifies and reduces the TCS solution back to what was detailed in § 2.3.

Chapter 3

Electro-optic study of a GaAs/Al_{0.92}Ga_{0.08}As coated mirror

3.1 Motivation

Contributions of categorized noises for gravitational wave detectors are organized in a “noise budget” (i.e. Figure 1.14 and Figure 2.15): comprised of a collection of technical (noise imposed by the practical operation of the detector) and fundamental (inherent physical limitations of the DRFPMI design) noise that limit gravitational wave detection. Understanding how much differential phase noise is imparted on the interferometer carrier light passing through and reflecting from core optics is crucial.

3.1.1 Coating Thermal Noise

One source of noise in high precision optical experiments operating at room temperature (and higher due to high power resonant beams), can be realized through Brownian thermal noise and the Fluctuation dissipation theorem.

Brownian Noise

In 1827 the Scottish botanist Robert Brown observed a constant motion of pollen particulates on the surface of water; witnessing randomized collisions of the water molecules holding a kinetic energy proportional to the temperature ($k_B T$) [30]. It is because of these documented observations we name the phenomena Brownian motion. And although the observations were on motion of particulates in liquids and gases, solids also exhibit similar fluctuations through their modes of dissipation.

Fluctuation Dissipation Theorem

Any movement / fluctuations at finite temperatures of the core optic components is especially important for GWDs and becomes more clear when reviewing the fluctuation dissipation theorem (FDT). Derived by H.B. Callen and T.A. Welton, the theorem states that for a randomly fluctuating linear force ($F_x^2(f)$) [31]:

$$F_x^2(f) = 4k_B T \Re[Z] \quad (3.1)$$

Where $\Re[Z]$ is the real part of the impedance of the system. This impedance directly relates to equations of motion:

$$Z = \frac{F}{\dot{x}} \quad (3.2)$$

Another useful form is the power spectrum of the fluctuating motion:

$$x^2(f) = \frac{4k_B T}{(2\pi f)^2} \Re[Y] \quad (3.3)$$

Where Y is the inverse impedance or admittance. The root mean square (RMS) displacement ($x^2(f)$) as informed by the FDT facilitates modelling and budgeting notable Brownian noise sources that fundamentally limit LIGO (i.e. by choice of materials used for highly

reflective mirror coatings). Though adequate modelling of internal force couplings for the aforementioned components provides a more complete picture.

Internal friction in materials and loss angle

Zener provides a model of the internal friction of materials incorporating anelasticity into the equations of motion [32]:

$$F = k(1 + i\phi)x + m\ddot{x} \quad (3.4)$$

Where m is mass attached to a spring with a spring constant $k(1 + i\phi)$ incorporating the degree of anelasticity ϕ . From equations 3.5 and 3.3 we perform a Laplace transform and acquire the following form of admittance:

$$Y(s) = \frac{\dot{x}(s)}{F(s)} = \frac{-s}{k(1 + i\phi) + ms^2} \quad (3.5)$$

Or more transparently the Fourier representation since we assume a linear time invariant system:

$$Y(\omega) = \frac{\dot{x}(\omega)}{F(\omega)} = \frac{-i\omega}{k(1 + i\phi) - m\omega^2} = \frac{k\omega\phi - i\omega(k - m\omega^2)}{(k - m\omega^2)^2 + k^2\phi^2} \quad (3.6)$$

Plugging equation Equation 3.6 back into Equation 3.3:

$$x^2(f) = \frac{2k_B T}{\pi} \frac{k\phi}{(k - 4\pi^2 m f^2)^2 + k^2 \phi^2} \quad (3.7)$$

Computing the admittance from a Gaussian beam impinging upon a HR mirror can require expansion of all individual mechanical degrees of freedom of the test mass system across a relevant frequency range, and with such an approach convergence is not guaranteed. Saulson and Gonzalez provide an alternative method to computing the admittance coined the “direct approach” by Levin when computing the noise from a Gaussian beam on a LIGO HR test mass. The admittance can be acquired through:

$$\Re[Y] = \frac{W_{\text{diss}}}{F_o^2} \quad (3.8)$$

W_{diss} is the dissipated power from the system due to an oscillating force F_o . This form of the admittance reveals an important result of the fluctuation dissipation theorem where an undriven system with a dissipative actor, imparts motion to the degrees of freedom via a driving force by virtue of that same actor at finite temperatures. This direct approach also allows the surface pressure applied by the Gaussian beam to interrogate which mechanical modes of the test mass impose a significant energy when Equation 3.8 is plugged into Equation 3.3. In the case of the gaussian beam / uncoated test mass studied by Levin [33]:

$$S_x(f) = \frac{4k_B T}{f} \frac{1 - \sigma^2}{\pi^3 E_o r_o} I \phi \left[1 - O\left(\frac{r_o}{R}\right) \right] \quad (3.9)$$

Where σ and E_o are the Poisson ratio and Young's modulus respectively, and $O(\frac{r_o}{R})$ is a correction term as a function of the small beam radius (r_o) relative to the mirror radius (R).

Thermal noise of HR mirror coatings

Further investigations into the beam/optic system utilizing this approach and elasticity theory led to a deeper understanding about Brownian thermal noise contributions from LIGO test masses (substrate, suspensions, HR coating). Levin mentions, with further detail from Harry, that the noise contributed by a lossy mirror coating is proven to be the most significant contributor of brownian thermal noise [34]. Hong provides a power spectral density [35]:

$$S_j^X = \frac{4k_B T \lambda \phi_x^j (1 - \sigma_j - 2\sigma_j^2)}{3\pi^2 f Y_j (1 - \sigma_j)^2 \omega_o^2} \quad (3.10)$$

Where X represents bulk and shear with j = odd (material 1) and j = even (material 2) alternating layers representing high and low index materials j = odd (material 1) j = even (material 2) for an HR coating.

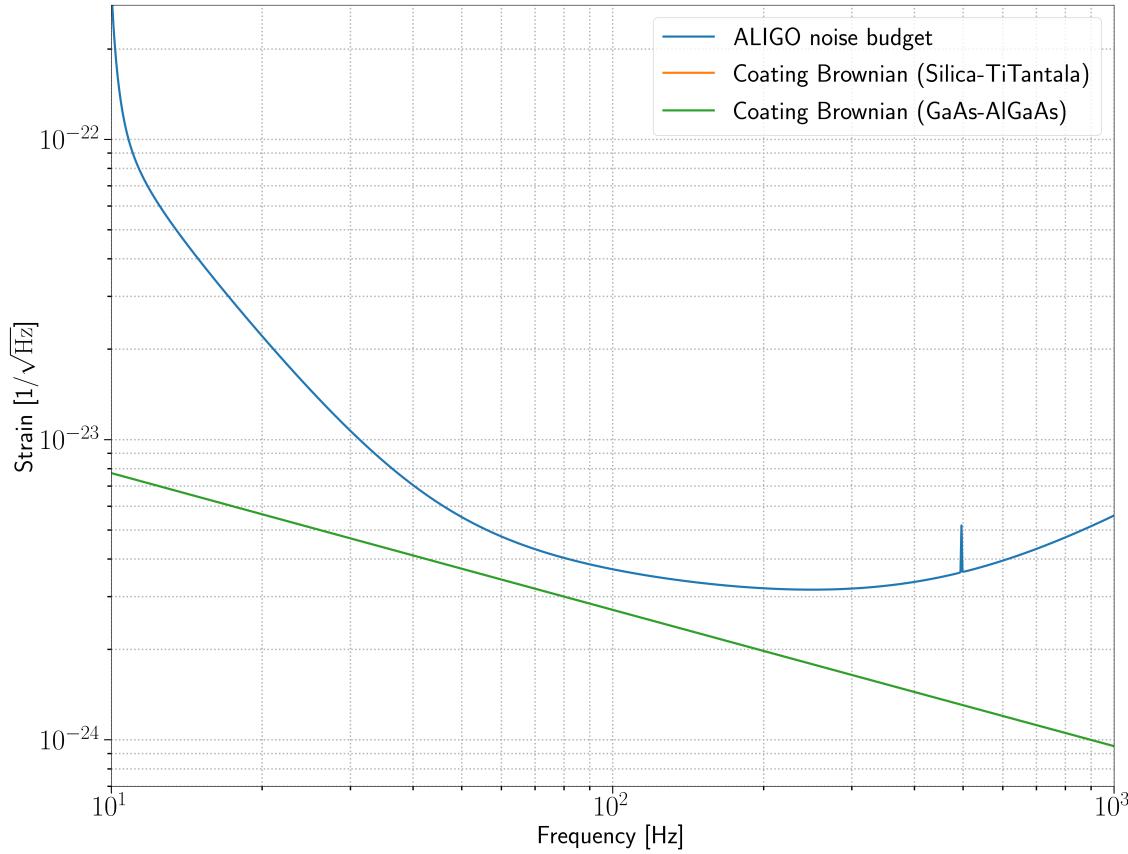


Figure 3.1: **ALIGO noise budget placeholder** for silica-tantala, and gaas-algaas brownian noise comparison

As aLIGO approaches designed sensitivity, the thermal noise limit set by SiO₂TiO₂ : Ta₂O₅ HR coatings becomes an immediate limit to further improvements. Though there are proposals for the usage of alternative mirror coating solutions to push down this thermal noise limit for increased detector sensitivity [36]. GaAs/Al_{0.92}Ga_{0.08}As is a crystalline coating candidate showing promise for next generation detectors with reduced coating Brownian noise by a factor of 10, cooresponding to a potential strain reduction by a factor of 5 [37] when compared to the current aLIGO coating thermal noise limit.

3.1.2 Coating Electro-optic Noise

Applying crystalline HR mirror coatings to the core optics indiscriminately may introduce notable side effects; one being linear electro-optic noise of GaAs/Al_{0.92}Ga_{0.08}As (dn/dE), also known as the Pockels effect [38]. Although estimated to be nearly two orders of magnitude below the A $^\sharp$ strain noise floor ($\approx 10^{-26}$), direct measurement is still merited and adequately motivates a thorough study of electro-optical properties of GaAs/Al_{0.92}Ga_{0.08}As coatings. The rest of this chapter discusses such a study by detailing: 1) birefringence in zincblende materials, 2) a preliminary estimation of differential phase noise of light reflected from a GaAs/Al_{0.92}Ga_{0.08}As coating stack caused by electric field noise are computed while considering potential impacts to the current generation gravitational wave detectors, and 3) a short experimental optical cavity designed to interrogate an estimate of dn/dE from a calibrated differential length PDH locked signal with a normal electric field driven across a HR GaAs/Al_{0.92}Ga_{0.08}As coating “witness” sample.

3.2 Birefringence in zincblende materials

3.2.1 The Indicatrix

The two index solutions for a uniaxial crystal given a general plane wave with unit wave vector \vec{k} can be found via a convenient geometrical construction known as the “index ellipsoid”.

The construction begins when considering a constant electric energy density (U_e) surface in the \vec{D} space; which forms an ellipsoid:

$$\frac{D_x}{\varepsilon_x} + \frac{D_y}{\varepsilon_y} + \frac{D_z}{\varepsilon_z} = 2U_e\varepsilon_o \quad (3.11)$$

With redefined coordinates $(\vec{D}/\sqrt{2U_e\varepsilon_o}) \rightarrow \vec{r}$ and setting $\varepsilon_i = n_i^2$:

$$\frac{x^2}{n_x^2} + \frac{y^2}{n_y^2} + \frac{z^2}{n_z^2} = 1 \quad (3.12)$$

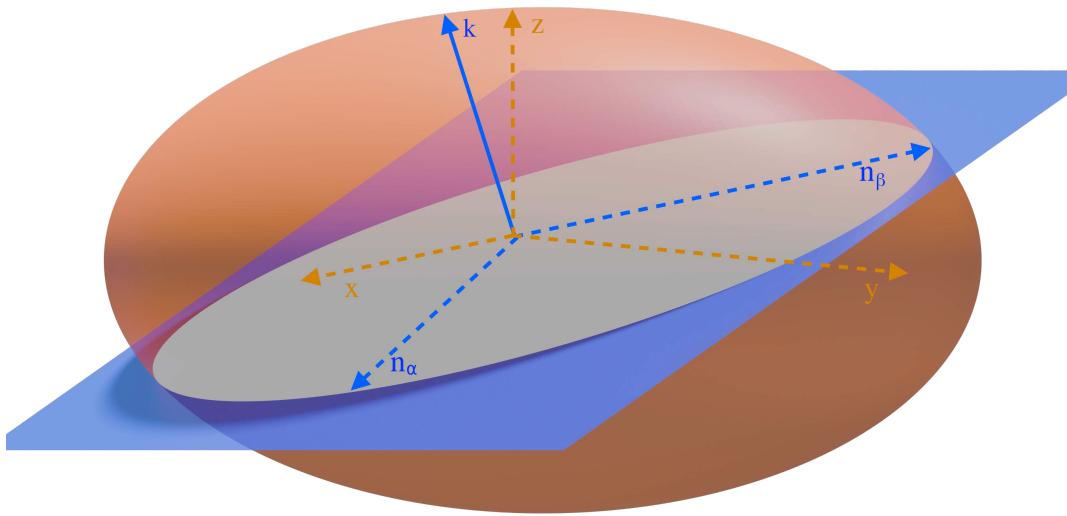


Figure 3.2: A surface of uniform energy density (U_E) forming an ellipsoid in D-space for a generalized uniaxial crystal with general wavefront propagation indicated by a plane normal \hat{k}' where the major and minor axes of the ellipse cross section indicate slow and fast axes n_β and n_α respectively.

This equation for the ellipsoid is known as the indicatrix. Given the co-planar solution demonstrated in § 5.9.2, we can impose the normal of the plane $\vec{r} \cdot \vec{s} = 0$:

$$\vec{r} \cdot \vec{s} = xs_x + ys_y + zs_z = 0 \quad (3.13)$$

Equation 3.11 and Equation 3.13 both contribute constraints to the method of finding extrema using Lagrange multipliers for the function:

$$r^2 = x^2 + y^2 + z^2 \quad (3.14)$$

The Lagrangian (\mathcal{L}) with the introduced multipliers (λ_1, λ_2) then becomes:

$$\mathcal{L}(\vec{r}, \vec{s}, \lambda_1, \lambda_2) = x^2 + y^2 + z^2 + \lambda_1(xs_x + ys_y + zs_z) + \lambda_2\left(\frac{x^2}{\varepsilon_x} + \frac{y^2}{\varepsilon_y} + \frac{z^2}{\varepsilon_z} - 1\right) \quad (3.15)$$

With the generated system of equations from the Lagrange multiplier method ($\partial F_i / \partial x_i = 0$, and $\partial F_j / \partial \lambda_j$) where index $i = x, y, z$ and $j = 1, 2$ we obtain a system of 3 equations:

$$i\left(1 - \frac{r^2}{\varepsilon_i}\right) + s_i\left(\frac{xs_x}{\varepsilon_x} + \frac{ys_y}{\varepsilon_y} + \frac{zs_z}{\varepsilon_z}\right) = 0 \quad (3.16)$$

The result is verified when substituting $r \rightarrow \frac{\vec{D}}{\sqrt{E \cdot \vec{D} \varepsilon_o}}$ back which recovers Equation 5.76.

3.2.2 GaAs and Al_{0.92}Ga_{0.08}As crystal classification

GaAs as well as Al_{1-x}Ga_xAs are both within the $F\bar{4}3m$ space group. Crystals of this space group are commonly known as zincblende crystals; a common crystal configuration named after zinc sulfide (ZnS). Also categorized as a cubic crystal, their crystallographic structure displays linear optical isotropy when stress free and no DC and/or slowly varying electric fields are present [39].

Zincblende structures, like the crystalline materials in question can exhibit birefringent properties when under influence of mechanical stresses and static / low-frequency electric fields (E_{STLF}); characterized by photoelastic and electro-optic effects respectively. For realistic mirror coatings, heteroepitaxial bonding between GaAs/Al_{0.92}Ga_{0.08}As layers (potentially from a noticeable difference in lattice cell constant) may produce an intrinsic strain within the HR stack and can lead to the existence of a static non-negligible birefringence throughout the coating layers [40, 41].

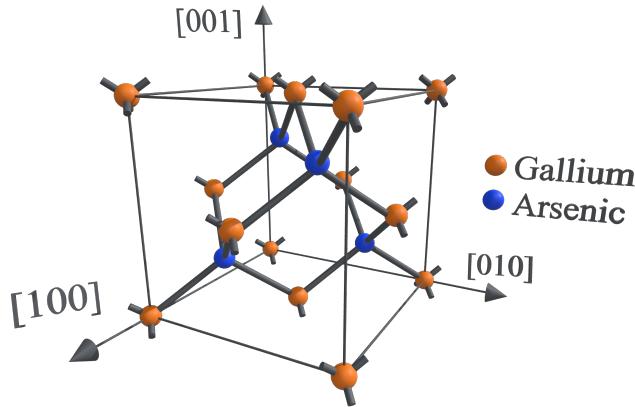


Figure 3.3: The unit cell of gallium arsenide (GaAs) with associated miller indices as coordinate axes

3.2.3 Linear electro-optic effect (Pockel's effect)

For non-centrosymmetric crystalline media there exists a non-zero rank 2, 6×3 tensor (r_{ij}) connecting low-frequency¹ electric fields $\vec{E}_{STLF}(f) = [E_x(f), E_y(f), E_z(f)]$ directly to the indicatrix [42, 43]:

$$\begin{bmatrix} \left(\frac{1}{\Delta n^2}\right)_1 \\ \left(\frac{1}{\Delta n^2}\right)_2 \\ \left(\frac{1}{\Delta n^2}\right)_3 \\ \left(\frac{1}{\Delta n^2}\right)_4 \\ \left(\frac{1}{\Delta n^2}\right)_5 \\ \left(\frac{1}{\Delta n^2}\right)_6 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \\ r_{41} & r_{42} & r_{43} \\ r_{51} & r_{52} & r_{53} \\ r_{61} & r_{62} & r_{63} \end{bmatrix} \begin{bmatrix} E_x(f) \\ E_y(f) \\ E_z(f) \end{bmatrix} \quad (3.17)$$

The i index runs over the terms in the indicatrix equation:

$$\left(\frac{1}{\Delta n_x^2}\right)x^2 + \left(\frac{1}{\Delta n_y^2}\right)y^2 + \left(\frac{1}{\Delta n_z^2}\right)z^2 + 2\left(\frac{1}{\Delta n_{xz}}\right)xz + 2\left(\frac{1}{\Delta n_{yz}}\right)yz + 2\left(\frac{1}{\Delta n_{xy}}\right)xy = 1 \quad (3.18)$$

¹“low frequency” meaning orders of magnitude smaller relative to an optical field

3.2.3.1 r_{ij} for zincblende crystals ($r_{\bar{4}3m,ij}$)

The form of the electro-optic tensor for zincblende crystals (including GaAs and Al_{0.92}Ga_{0.08}As) reduces such that $r_{ij} = r_{41} = r_{52} = r_{62} \neq 0$ with all other terms being zero:

$$r_{\bar{4}3m,ij} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ r_{41} & 0 & 0 \\ 0 & r_{52} & 0 \\ 0 & 0 & r_{63} \end{bmatrix} \quad (3.19)$$

Where $r_{41} = r_{52} = r_{63}$

3.2.4 New principal (electro-optic) dielectric axis for zincblende structures

In general the principle dielectric axes of the new ellipsoid do **not** coincide with the axes of the ellipsoid of the unperturbed crystal. The form of the index ellipsoid for a zincblende crystalline material accounting for the electro-optic tensor and some generalized DC electric field \vec{E} expressed in terms of the crystallographic axes is given by:

$$\left(\frac{1}{n_o^2}\right)x^2 + \left(\frac{1}{n_o^2}\right)y^2 + \left(\frac{1}{n_o^2}\right)z^2 + 2r_{41}E_xyz + 2r_{41}E_yxz + 2r_{41}E_zxy = 1 \quad (3.20)$$

Where we have set $n_x = n_y = n_z = n_o$ for zincblende structures. Visualizing the above as a tensor:

$$\begin{bmatrix} \left(\frac{1}{n_o^2}\right) & r_{41}E_x & r_{41}E_y \\ r_{41}E_x & \left(\frac{1}{n_o^2}\right) & r_{41}E_z \\ r_{41}E_y & r_{41}E_z & \left(\frac{1}{n_o^2}\right) \end{bmatrix} \quad (3.21)$$

3.2.5 The photoelastic effect

General stresses and strains of a material may also cause transformations to the indicatrix connected by the rank 4 elasto-optical tensor p_{ijkl} :

$$\left(\frac{1}{\Delta n^2} \right)_{ij} = p_{ijkl} \epsilon_{kl} \quad (3.22)$$

Where the strain (ϵ) relates to stress (σ) using the generalized Hooke's law:

$$\begin{aligned} \epsilon_{ij} &= K_{ijkl} \sigma_{kl} \\ \sigma_{ij} &= C_{ijkl} \epsilon_{kl} \end{aligned} \quad (3.23)$$

A connection is also formed between the elasto-optical tensor (p) to the piezo-optical tensor (π):

$$\begin{aligned} p_{ijkl} &= \pi_{ijkl} C_{klrs} \\ \pi_{ijrs} &= p_{ijrs} K_{rskl} \end{aligned} \quad (3.24)$$

3.2.6 The generalized indicatrix

Both forms of the induced birefringence (electro-optic and photo-elastic) can be incorporated into a condensed form [43]:

$$\left(\frac{1}{\Delta n^2} \right)_{ij} = r_{ijk} E_k + p_{ijkl} \epsilon_{kl} \quad (3.25)$$

3.2.7 EO Modulation (Application)

Imparting phase modulations onto an optical carrier field is a common application of the electro-optic effect. Electro-optic modulators (EOMs) or Pockel cells are sold as a standard optical components usually composed of a monolithic crystalline material sandwiched between two capacitor plates connected to a single electrical input port (typically coaxial

for RF) designed to take in a voltage input of frequency (Ω) within a specified modulation amplitude and frequency bandwidth. When the field amplitude across the crystal is driven by a voltage controlled oscillation, the amplitudes of the electro-optic tensor vary linearly.

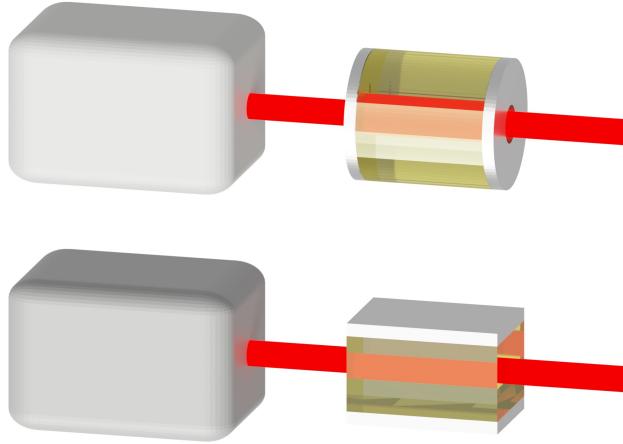


Figure 3.4: Longitudinal and transverse electro-optic modulators

The voltage amplitude of the signal input is proportional to the strength of the modulated phase on the optical carrier frequency (ω); commonly quantified in terms of a modulation index (β):

$$E_{\text{out}} = E_o e^{i\omega t + \beta \sin(\Omega t)} \approx E_o [J_0(\beta) e^{i\omega t} + J_1(\beta) e^{i(\omega + \Omega)t} - J_1(\beta) e^{i(\omega - \Omega)t}] \quad (3.26)$$

Where we have approximated with the Jacobi-Anger expansion utilizing Bessel functions of the first kind ($J_n(x)$) [44]:

$$e^{iz \sin(\theta)} \approx J_0(z) + 2 \sum_{n=1}^{\infty} i^n J_n(z) \sin(n\theta) \quad (3.27)$$

3.26 sufficiently demonstrates that, to the first order, a carrier field that is phase modulated is also, in essence, imparting power to separate optical sideband fields separated in frequency by an integer multiple of the modulation $n \cdot \Omega$. Typically Ω is a chosen frequency

used for optical heterodyne detection; while for a noise-driven modulation, the phase coupling is coorelated to the local directionally relevant E-field spectra alongside the propogation length of the beam propogation within the electro-optic media.

3.3 Electro-optic noise of a GaAs / Al_{0.92}Ga_{0.08}As stack

A comprehensive survey of relevant birefringent properties of a HR GaAs/Al_{0.92}Ga_{0.08}As mirrorstack is due, and for this body of work includes: 1) crystal coordinate considerations when asserting an optical axis on a highly reflective crystalline stack manufactured by the Thorlabs crystalline coatings division, 2) citations of coating parameters and observed intrinsic birefringence from the highly reflective coating stack in question, 3) analysis of the differential electro-optic effect on the phase of a reflected beam, and 4) estimating the the differential phase noise in LIGO based on preliminary electric field measurements measured at LHO.

3.3.1 Static Birefringence / Miller indices from a HR GaAs / Al_{0.92}Ga_{0.08}As coating

Thorlab's crystalline coatings division grows their HR crystalline optical coating such that the coating surface is drawn out in the [100] plane, meaning that beam with a wavevector along the optical (z) axis draws a parallel line to the normal of said plane. Therefore since the beam's polarized E-field oscillates only within that plane, any differential splitting of the beam polarization occurs solely between rotated [010] and [001] axes. This allows us to restrict our interest to a field where $E_z \neq 0$ and $E_x = E_y = 0$ and compute the eigenvalues ($\lambda_{x',y',z'}^*$ / eigenpolarizations ($\vec{x}', \vec{y}', \vec{z}'$) which lead us to the relevant eigenindices ($n_{x',y'}$) ²:

²There may appear to be an inconsistency between the miller indices and optical axes, but because of the isotropy of zincblende crystals prior to the field perturbation coordination of these axes is not quite relevant

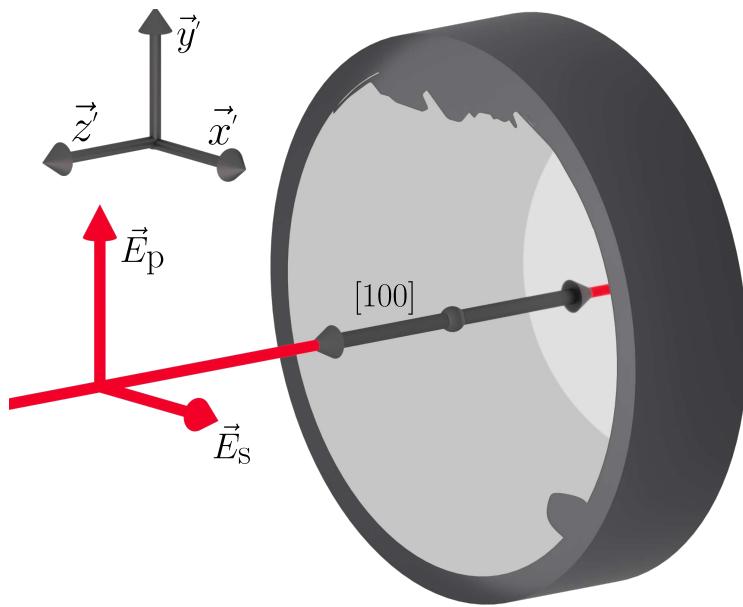


Figure 3.5: The beam propagation axis with respect to the GaAs/Al_{0.92}Ga_{0.08}As crystal axis. The axis formed by the [100] plane normal is drawn parallel with the beam axis (z-axis) and the polarizations of incident and reflected beam oscillate along vectors within the plane formed by the normal of that axis. The topmost coordinate axis triad is drawn to depict world vectors that can help visualize the plane of computed eigenvectors. Depicted here are also defects at the top and bottom right of the coating due to overhandling but do not effect the results of this study.

$$\begin{aligned}\lambda_{x'} &= \left(\frac{1}{n_o^2} - r_{41}E_z \right) \\ \lambda_{y'} &= \left(\frac{1}{n_o^2} + r_{41}E_z \right) \\ \lambda_{z'} &= \frac{1}{n_o^2}\end{aligned}\tag{3.28}$$

And the principal axes / eigenpolarizations are found when solving for the eigenvectors:

$$\begin{aligned}\vec{x}' &= \frac{1}{\sqrt{2}} (0, -1, 1) \\ \vec{y}' &= \frac{1}{\sqrt{2}} (0, 1, 1) \\ \vec{z}' &= (1, 0, 0)\end{aligned}\tag{3.29}$$

$$\lambda_{x'} x'^2 + \lambda_{y'} y'^2 + \lambda_{z'} z'^2 = 1\tag{3.30}$$

The eigenindices ($n_\alpha = n_{x'}$, $n_\beta = n_{y'}$) are therefore:

$$\begin{aligned} n_{x'} &= \sqrt{\lambda_{x'}} = \sqrt{\frac{1}{n_o^2} - r_{41}E_z} \\ n_{y'} &= \sqrt{\lambda_{y'}} = \sqrt{\frac{1}{n_o^2} + r_{41}E_z} \end{aligned} \quad (3.31)$$

And with $n_o r_{41} E_z \ll 1$:

$$\begin{aligned} n_{x'} &\approx n_o + \frac{1}{2} n_o^3 r_{41} E_z \\ n_{y'} &\approx n_o - \frac{1}{2} n_o^3 r_{41} E_z \end{aligned} \quad (3.32)$$

3.3.2 Electro-optic coupling to the reflected phase of a HR mirror coating

Analytic estimate

Fejer and Bonilla takes on an analytical approach to finding the impact of the electric field to the change in phase of the light through a crystalline anisotropic thin film ($\lambda/4$) stack. The construction builds off of a pre-defined derivation of thermo-optic noise calculations for the HR stack and assuming a large enough number of high-low index coating pairs [45, 46]:

$$\left| \frac{\partial \phi}{\partial E} \right| = -\pi \frac{r_{41}}{2} (n_{\text{high}} n_{\text{low}}^2 + n_{\text{low}} n_{\text{high}}^2) \frac{n_{\text{high}}}{n_{\text{low}}} \quad (3.33)$$

With $n_{\text{low}} = n_{\text{Al}_0.92\text{Ga}_0.08\text{As}} = 2.9369$, $n_{\text{high}} = n_{\text{GaAs}} = 3.4786$, and $r_{41} = -1.33 \times 10^{-12}$

The estimated differential phase from the electro-optic effect with a 1064nm E-field propagating along the [110] axis of the HR GaAs/Al_{0.92}Ga_{0.08}As stack:

$$\left| \frac{\partial \phi}{\partial E} \right| = 4.0253 \times 10^{-12} \frac{[\text{rad}]}{[\text{V/m}]}$$

With κ being a scalar parameter.

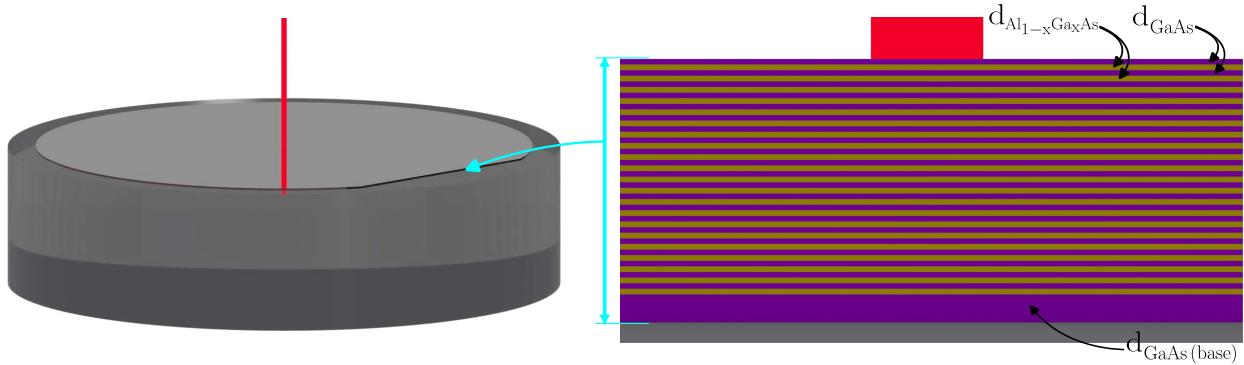


Figure 3.6: The beam propagation axis (\vec{S} , $[-100]$) with respect to the GaAs/Al_{0.92}Ga_{0.08}As crystal axes. The axis formed by the [100] plane normal is drawn parallel with the beam axis (z-axis) and the polarizations of incident and reflected beam oscillate along vectors within the plane formed by the normal of that axis. The coating is grown with a flat indicating a line within the [0-11] plane; where the plane normal points towards the sample center.

Numerical estimate

In the appendix of [47] Ballmer constructs a coating layer transfer function for a given coating layer k with index n_k , and thickness d_k , defining right and left propagating modes $\psi^{R,L}$ respectively:

$$\begin{bmatrix} \psi^R \\ \psi^L \end{bmatrix}_{k+1} = Q_k D_k \begin{bmatrix} \psi^R \\ \psi^L \end{bmatrix}_k$$

D_k applies the phase ($\phi_k = 4\pi n_k d_k / \lambda_0$) from a given coating layer, and Q_k applies the transfer function between high-low/low-high index layers transition:

$$D_k = \begin{bmatrix} e^{-i\phi_k/2} & 0 \\ 0 & e^{i\phi_k/2} \end{bmatrix} \quad (3.34)$$

$$Q_k = \frac{1}{2n_{k+1}} \begin{bmatrix} n_{k+1} + n_k & n_{k+1} - n_k \\ n_{k+1} - n_k & n_{k+1} + n_k \end{bmatrix} \quad (3.35)$$

Defining a HR coating stack, the total transfer matrix from vaccum Q_0 to the N th coating layer is:

$$M = Q_N D_N \dots Q_k D_k \dots Q_1 D_1 Q_0 \quad (3.36)$$

And the partial derivative at the kth coating layer is:

$$\frac{\partial M}{\partial \phi_k} = Q_N D_N \dots Q_k \begin{bmatrix} e^{-i\phi_k/2} & 0 \\ 0 & e^{i\phi_k/2} \end{bmatrix} \begin{bmatrix} -i/2 & 0 \\ 0 & i/2 \end{bmatrix} Q_{k-1} D_{k-1} \dots Q_1 D_1 Q_0 \quad (3.37)$$

The above representing a collective differential phase manifesting as a sum of these phase components. This explicit perturbed phase at the kth layer for the electro-optic effect ($\partial n_k / \partial E$) is found when:

$$\frac{\partial \phi_k}{\partial E} = \frac{4\pi d_k}{\lambda} \frac{\partial n_k}{\partial E} = \pm \frac{2\pi}{\lambda} n_k^3 d_k r_{41,k} \quad (3.38)$$

Where the electro-optic coefficients r_{41} for GaAs and Al_{1-x}Ga_xAs [48, 49, 41]:

$$\begin{aligned} r_{41,\text{GaAs}} &= -1.33 \times 10^{-12} & [\text{m/V}] \\ r_{41,\text{Al}_{1-x}\text{Ga}_x\text{As}} &= -(1.33 - 0.45x) \times 10^{-12} & [\text{m/V}] \end{aligned} \quad (3.39)$$

Rather than tagging on the phases individually, an easier computation is found when relying on the relationship between the transmission (t) and reflectivity (r) to a general transfer matrix (in our case M):

$$\begin{bmatrix} 1 \\ r \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} t \\ 0 \end{bmatrix}$$

And using this relation, differentiating the reflectivity with respect to ϕ_k :

$$\frac{\partial r}{\partial \phi_k} = - \left(\frac{1}{M_{21}} \frac{\partial M_{21}}{\partial \phi_k} - \frac{1}{M_{22}} \frac{\partial M_{22}}{\partial \phi_k} \right) \frac{M_{21}}{M_{22}}$$

The differential reflectivity is normalized by the total reflectivity and taking the imaginary component as noted in Equation 3.37:

$$\frac{\partial \phi_c}{\partial \phi_k} = \text{Im} \left(\frac{1}{r} \frac{\partial r}{\partial \phi_k} \right) = \left(\frac{1}{M_{21}} \frac{\partial M_{21}}{\partial \phi_k} - \frac{1}{M_{22}} \frac{\partial M_{22}}{\partial \phi_k} \right) \quad (3.40)$$

The impact of a differential electric noise field (E_{STLF}) on M due to the electro-optic effect on the k th layer, we use the chain rule:

$$\left| \frac{\partial \phi_c}{\partial E_{\text{STLF}}} \right| = \left| \frac{\partial \phi_c}{\partial \phi_k} \frac{\partial \phi_k}{\partial E} \right| \quad (3.41)$$

The coating to be studied consists of 36 $\lambda/4$ thick layers of GaAs interspersed with 35 layers of $\lambda/4$ thick Al_{0.92}Ga_{0.08}As. GaAs forms the top and bottom layer to prevent oxygen absorption from the AlGaAs layer. The GaAs layers have an index of $n_{\text{GaAs}} = 3.480$ and a thickness of $\Delta d_{\text{GaAs}} = 76.43$ nm while the low index Al_{0.92}Ga_{0.08}As layers are $n_{\text{Al}_{0.92}\text{Ga}_{0.08}\text{As}} = 2.977$ with thickness $\Delta d_{\text{Al}_{0.92}\text{Ga}_{0.08}\text{As}} = 89.35$ nm. With the constructed matrices, we apply these parameters and compute a differential phase of:

$$\left| \frac{\partial \phi_c}{\partial E_{\text{STLF}}} \right| = 3.9 \times 10^{-11} \frac{[\text{rad}]}{[\text{V}/\text{m}]} \quad (3.42)$$

3.3.3 Initial projected DARM coupling

Measured field spectra acquired from installed electric field meters located within LHO and LLO ETMX and ETMY vacuum chambers can help translate how much DARM coupling can occur from electro-optic coating noise. For O3 the EFM's were located next to the test mass mirrors and measured a consistent $3 [\mu\text{V}/\text{m}/\sqrt{\text{Hz}}]$ @ 100 Hz [22]. This along with computed estimate allows us to create an upper limit for what this noise might be assuming incoherent fields between the end stations and a flat frequency response within LIGO's bandwidth. An initial differential phase noise estimate of $\approx 4.5 \times 10^{-10}$ [rad/m/V], alongside measured LHO ambient field noise measured during O3 we compute an initial strain noise estimate [46, 22]:

$$\frac{\partial L}{\partial E} = \frac{\lambda}{4\pi} \left| \frac{\partial \theta_c}{\partial E} \right| = 3.81 \times 10^{-17} \frac{[\text{m}]}{[\text{V}/\text{m}]}$$

$$h_{\text{EO noise}} = 2 \frac{\Delta L_{\text{EO noise}}}{L} = 2 \frac{\partial L}{\partial E} \times \left| \frac{E_{\text{noise}}}{L_{\text{arm}}} \right| = 5.715 \times 10^{-26} \text{ [Hz]} \quad (3.43)$$

3.4 Electro-optic measurement apparatus

In seeking a calibrated estimate of the electro-optic effect for the GaAs/Al_{0.92}Ga_{0.08}As mirror coating stack, we sought to drive an electro-optic response of a mirror sample from Thorlab's crystalline mirror coatings division placed within a custom longitudinal Pockels cell mirror mount. The assembly, with the installed sample, assumed the end mirror position within a two mirror Fabry-Perot cavity, while resonance of a circulating Nd:YAG 1064nm carrier beam was held by a Pound-Drever-Hall servo. As seen in the prior section, the size of the imparted phase noise for currently existing gravitational wave detector configurations is estimated to be small but notable. Investigation through measurement of said effect requires detection methods with sufficient sensitivity for the differential phase noise imparted by the effect. Details and specifications of the detection schema are discussed along with relevant measurements and results.

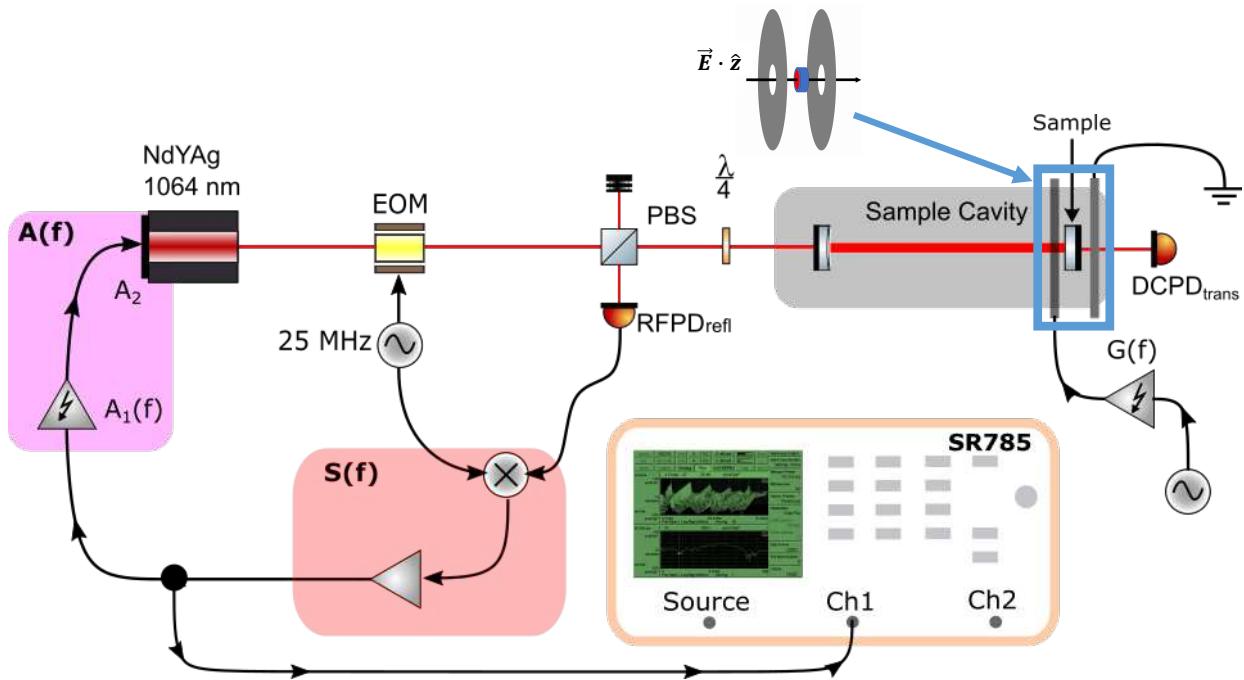


Figure 3.7: A simplified and modular schematic of the PDH servo used along with an electrostatic drive mount design comprised of a disk capacitor sandwiching the HR AlGaAs sample, a high voltage amplifier, and a signal / network analyzer.

Measurability of the electro-optic effect is contingent upon two initial design criteria: the sensitivity of the optical plant to be implemented in the PDH servo, and the maximum achievable electric field strength along the beam axis ($|E_z|_{\max}$).

3.4.1 PDH servo

The Pound-Drever-Hall technique, originally and commonly used for laser frequency stabilization to an ultra-stable length reference, allows the tracking of the linear phase response of an input carrier field through cavity resonance. The servo fully realizes the ability of an optical cavity to act as a length / frequency discriminator:

$$\frac{\Delta f}{f} = \frac{\Delta L}{L} \quad (3.44)$$

The alternative side-of-fringe lock provides a linear response in intensity, which is adequate for some applications but with reduced sensitivity due to the required power reduction by

operating off resonance. Measurements of phase are extracted through an optical heterodyne; the co-propagation of a separate (but phase-locked) optical field with a known frequency separation to the carrier reflected from the cavity input. The PDH servo bypasses the need for a complicated phase-locked two laser configuration by imposing a phase modulation onto the carrier field via an electro-optic modulator (aka Pockels cell) mentioned in section § 3.2.7. Setting a photodiode of area (A_{PD}) in reflection of the cavity with a coefficient of $r_{\text{cav}}(\omega, L)$, we measure the reflected power of the input field given by Equation 3.26 :

$$\begin{aligned} P_{\text{refl}} &\approx \frac{|E_{\text{refl}}|^2}{A_{\text{PD}}} \\ &\approx \frac{E_0^2}{A_{\text{PD}}} \left\{ J_0^2 |r_{\text{cav}}(\omega, L)|^2 + J_1^2(\beta) |r_{\text{cav}}(\omega + \Omega, L)|^2 - J_1^2(\beta) |r_{\text{cav}}(\omega - \Omega, L)|^2 + \right. \\ &\quad \left. J_0 J_1(\beta) [r_{\text{cav}}(\omega, L) r_{\text{cav}}^*(\omega + \Omega, L)] - J_0 J_1(\beta) [r_{\text{cav}}(\omega, L) r_{\text{cav}}^*(\omega - \Omega, L)] \right\} \end{aligned} \quad (3.45)$$

The two trailing terms in the above equation for P_{refl} generate a beat frequency term between the carrier and lower and upper sidebands. The magnitude and sign of these beat terms directly relate to the phase of the reflected carrier field and can be measured and transformed to the error signal seen in Figure 3.8 using resonant electronics (tuned to a chosen sideband frequency) for amplification and a mixer for demodulation.

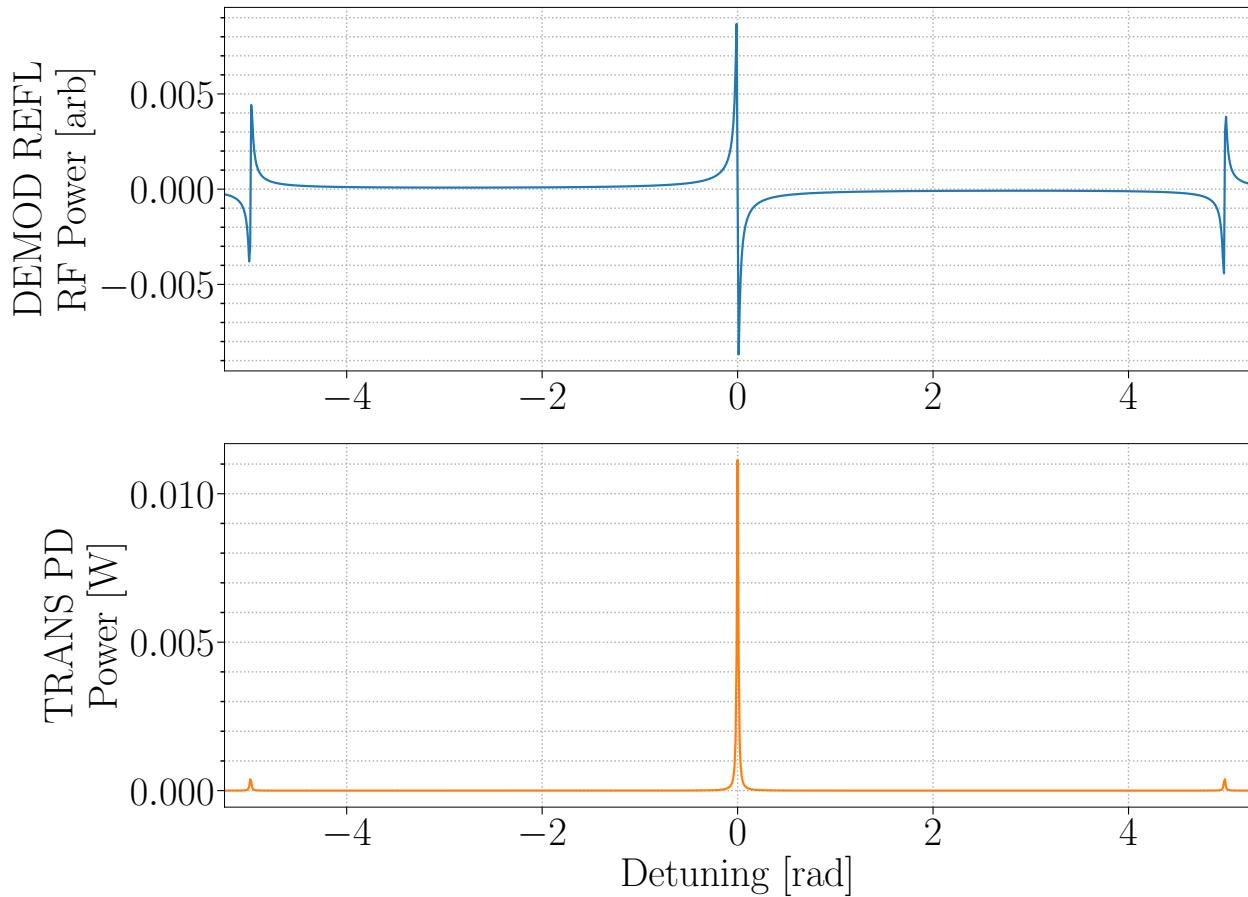


Figure 3.8: By imposing 25 MHz RF sidebands, a pair of reflected reference fields near carrier resonance are off cavity resonance while beating with the carrier and provide a linear response after demodulating the sideband power. With the introduction of high and low frequency sideband fields, their presence is also detected through the DCPDs and PDH error signal. Their separations from carrier resonance are equal in phase (length, and frequency).

With this linearity and sensitivity at cavity resonance, implementation into PID feedback is the next task as any small detuning of the cavity can be registered as a drift from the loop's zero point and fed back to an actuator with an estimated calibration gain factor. When implemented into a low-noise design, this servo can also be used for a high sensitivity lock-in measurement; and with well characterized instrumentation, calibration of the induced differential phase of the light within the stable reference cavity into differential length (or frequency).

3.4.2 Servo Parameters

The quantity we are attempting to measure is a differential length on the order of 3.3×10^{-18} [m/(V/m)], motivating a short cavity design as the relative differential length (phase) change scales with the sensitivity Equation 3.44. Considerations of the lab mirror inventory and mode matching criteria lead us to two candidate plano-concave (ROC = 0.333m) HR IBS coated sample input couplers; one from CVI Melles-Griot and another from ATFilms. When paired with the plano-plano GaAs/Al_{0.92}Ga_{0.08}As mirror from the Crystalline Mirror Solutions (CMS) division of Thorlabs we create a 0.1665 m long cavity.

The implemented servo design uses a light source from a Mephisto 2000 NE Nd:YAG (1064nm) laser with a 25 MHz phase modulation from a New Focus Model 4003 IR resonant phase modulator. As indicated in the figure above, the electronics chain can be decomposed into various filter components: $S(f)$, $A(f)$, and $A_{\text{thermal}}(f)$

3.4.2.1 Sensing S(f)

Sensing electronics are composed of a single element photodiode mounted to a transimpedance amplifier PCB that redirects photocurrent to DC and RF outputs. The RF path is constructed so the RF signal is boosted before being passed to the FSS mixer where it is demodulated by mixing the 25 MHz oscillator phased with variable cable length. Once demodulated, the measured beat signal while sweeping through resonance generates the PDH error signal profile Figure 3.8.

- 25 MHz RFPD
 - Transimpedance measurement (necessary? or should I just use the mixer out PDH to summarize PD/mixer response)
- Frequency Stabilization servo (modified MIT FSS (DCCD980536)) (LTspice model in appendix)

3.4.2.2 Actuation A(f)

- Mephisto 2220 PZT response (capacitance estimated from HVA drive measurement with and without connection to PZT)
- Channel 3 of SVR 350-3 BIP High Voltage Amplifier from Piezomechanik GmbH with Pomona box (elog 412)

3.4.2.3 Low frequency servo (Thermal loop)

- Passed signal from FSS → integrators → Laser thermal actuator input

3.4.2.4 OLG(f)

Isn't quite $A(f) * S(f)$ as stated. Doesn't entirely account for the optical plant. How the measurement is taken (important to take between installations to account for the changes in the optical plant) (elog 831)

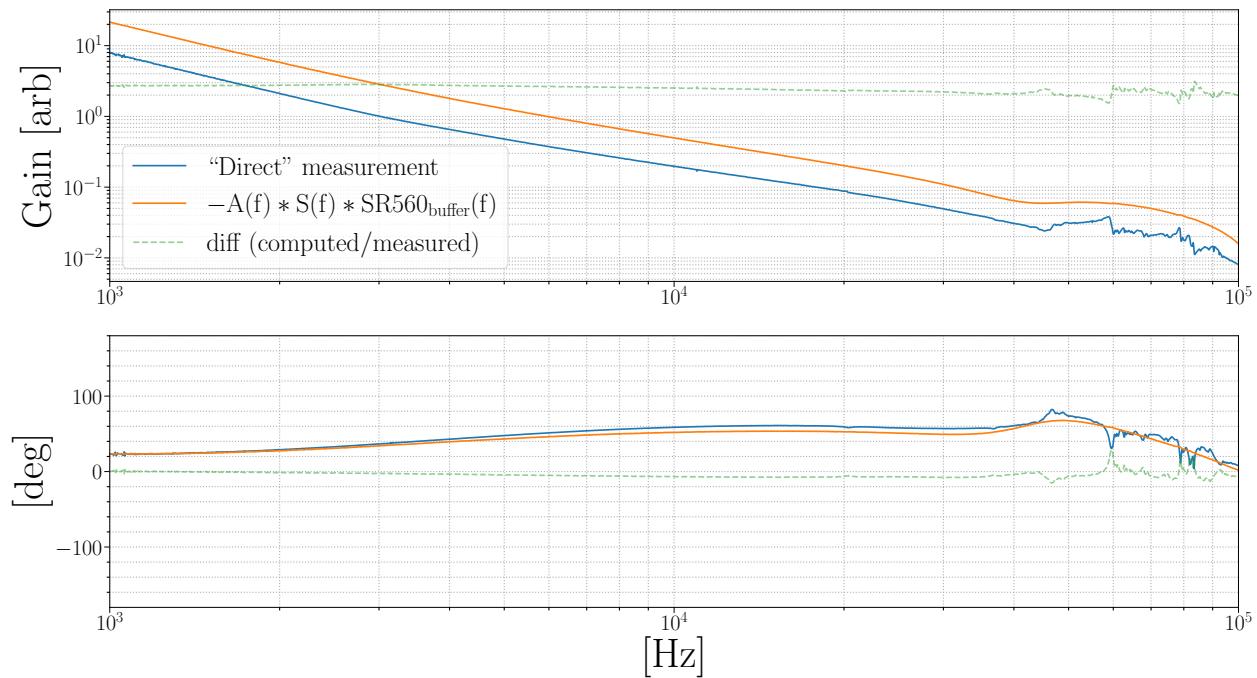


Figure 3.9: Comparison of the open loop gain measurement against the multiplied servo electronics measurements. The maximum gain difference is about a factor of 2.8 which is low passed to a difference of 2.0.

3.4.3 Longitudinal Pockels Cell mirror mount assembly

Maximizing a controlled and well defined electric field ($|E_z|$) within the coating while requiring a through beam to and through the HR coating lead us to a design very similar to that of a longitudinal pockels cell. The most common assembly in for this study is comprised of two electrodes with a 3mm central aperture which is chosen to be at least 5 times larger than the beam size at the plate locations; to avoid significant beam clipping while maximizing field strength at the coating region of interest. There is also a required separation of at least 1/4" accounting for the thickness of the optical sample. Considering these constraints, modelling the system and computing the estimated field strength screened by the coating is the next step to the construction of the assembly.

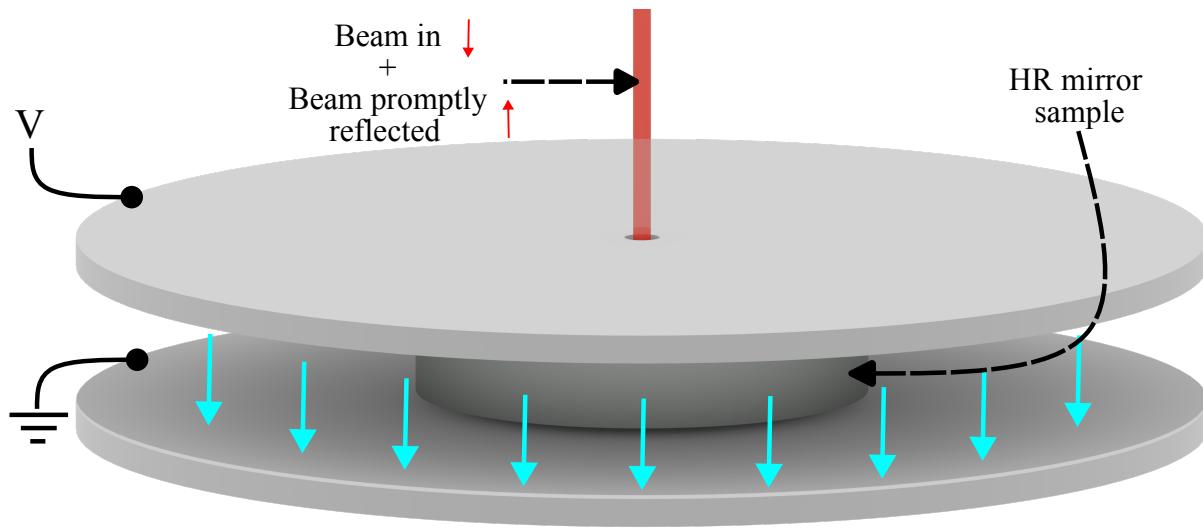


Figure 3.10: Concept image of the longitudinal Pockels cell assembly

3.4.3.1 Modeling

The field screened by the coating can be computed from Gauss' Law:

$$\nabla \cdot D = \rho_{\text{free}} \quad (3.46)$$

There is no free charge within the region of interest ($\rho_{\text{free}} = 0$), though the optic sample fused

silica substrate with the AlGaAs coating imposes dielectric material between the plates. Boundary conditions are expressed in terms of the differential plate potential V , so it is natural to first solve the potential (V) for all relevant system coordinate points.

$$\nabla^2 V = 0 \quad (3.47)$$

Boundary Conditions

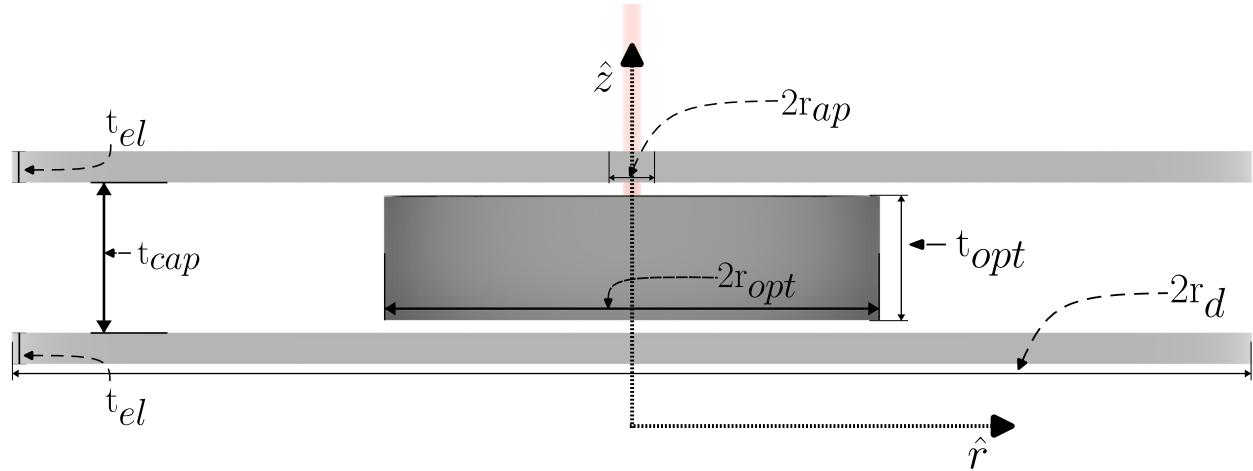


Figure 3.11: Side view of the longitudinal pockels cell mount. The figure is annotated with relevant parameters to build the numerical model: the finite thicknesses of the electrode plates (t_{el}), radius of the aperture at the center of the disk (r_{ap}), radius of the disk (r_d), thickness of the optic (t_{opt}), and radius of the optic substrate (r_{opt})

Substrate: $-t_{opt} < z < t_{opt}$ and $r < r_{opt}$

Coating $t_{opt} < z < t_{opt} + t_{coat}$ and $r < r_{opt}$

Driven Electrode (V): $t_{cap} < z < t_{cap} + 2t_{el}$ and $r_{ap} < r < r_d$

Grounded Electrode (GND): $-t_{cap} - 2t_{el} < z < -t_{cap}$ and $r_{ap} < r < r_d$

Numerical Recipe (Finite Differencing)

We exploit the chosen optic / disk symmetry about the polar angle ($\partial V / \partial \theta = 0$) and compute for the longitudinal (z) and radial (r) coordinates with the use of the appropriate Laplacian:

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{\partial^2}{\partial z^2} \right] (\varepsilon V) = 0 \quad (3.48)$$

Where ε is the dielectric

Observing equation Equation 3.47 we parse the non-zero expression into it's individual parts:

$$\left[\underbrace{\frac{\partial^2}{\partial z^2}}_{(c)} + \underbrace{\frac{\partial^2}{\partial r^2}}_{(b)} + \underbrace{\frac{1}{r} \frac{\partial}{\partial r}}_{(a)} \right] (\varepsilon V) = 0 \quad (3.49)$$

Term (a) Starting with the first derivative, we use the central difference approximation:

$$\frac{\partial}{\partial r} \rightarrow \frac{f(r+h, z) - f(r-h, z)}{2h} \rightarrow \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \quad (3.50)$$

Term (b) Second derivative approximation, we use the standard 2d laplace stencil

$$\frac{\partial}{\partial r^2} \rightarrow \frac{f(r+h, z) - 2f(r, z) + f(r-h, z)}{h^2} \rightarrow [1 \quad -2 \quad 1] \quad (3.51)$$

Term (c) Equivalent to the second derivative approximation used above:

$$\frac{\partial}{\partial z^2} \rightarrow \frac{f(r, z+h) - 2f(r, z) + f(r, z-h)}{h^2} \rightarrow [1 \quad -2 \quad 1] \quad (3.52)$$

To build the stencil terms at the boundaries, we look at the specialized finite difference condition @ $r = 0$, with the symmetry about $r = 0$ allowing the application of a ghost point $V(-h, z)$ as $V(h, z)$:

$$\frac{\partial V}{\partial r} = 0 \rightarrow V(h, z) = V(-h, z) \quad (3.53)$$

$$\frac{\partial^2 V}{\partial r^2} = \frac{2}{h} \left(\frac{V(h, z) - V(0, z)}{h} \right) \quad (3.54)$$

Equation 3.53 alone does not define $V(0, z)$, to establish the form of this point, we proceed to (Taylor) expand the function about it:

$$V \approx V_0 + C_1 r + C_2 r^2 + \mathcal{O}(r^4) \quad (3.55)$$

Symmetry about the origin imposes an even function of V :

$$\frac{\partial V}{\partial r} \approx C_1 + 2C_2 r + \mathcal{O}(r^3)$$

$$\frac{1}{r} \frac{\partial V}{\partial r} \approx 2C_2 + \mathcal{O}(r^2) \quad (3.56)$$

Where $C_1 = 0$ to avoid a singular point.

$$\frac{\partial^2 V}{\partial r^2} \approx 2C_2 + \mathcal{O}(r^2) \quad (3.57)$$

Substituting Equation 3.56 and Equation 3.57 back into equation Equation 3.49 :

$$\nabla^2(\varepsilon V) = \frac{\partial^2}{\partial z^2} + 4C_2 \quad (3.58)$$

The radial portion of the operator $\nabla^2 V$ given Equation 3.55 and Equation 3.54:

$$\left(\frac{r_0}{h} - \frac{2}{h^2} \right) C_1 + \left(\frac{r_h}{h} + \frac{2}{h^2} \right) C_2 h^2 = 4C_2 \quad (3.59)$$

Where again, we found $C_1 = 0$:

$$(r_h * h + 2)C_2 = 4C_2$$

$$\begin{aligned} r_h &= 2/h , \\ r_0 &= -2/h \end{aligned} \tag{3.60}$$

Now meshgrid coordinates are set:

$$z_{\text{indexing}} \rightarrow \left[\begin{array}{cccccccccccc} z_0 & \cdots & \cdots \\ z_1 & \cdots & \cdots \\ z_2 & \cdots & \cdots \\ z_3 & \cdots & \cdots \\ z_4 & \cdots & \cdots \\ z_5 & \cdots & \cdots \\ z_6 & \cdots & \cdots \\ z_7 & \cdots & \cdots \\ z_8 & \cdots & \cdots \\ \vdots & \ddots & \cdots \\ z_n & \cdots & \cdots \end{array} \right]$$

ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	\cdots	ρ_n
\vdots	\ddots	\vdots								
\vdots	\ddots	\vdots								

Parallel computation of the potential over the entire meshgrid is done by vectorizing the potential:

$$V = \begin{bmatrix} V(\rho_0, z_0) \\ V(\rho_1, z_0) \\ \vdots \\ V(\rho_n, z_0) \\ \hline V(\rho_0, z_1) \\ V(\rho_1, z_1) \\ \vdots \\ V(\rho_n, z_1) \\ \hline \vdots \\ \hline V(\rho_0, z_{n-1}) \\ V(\rho_1, z_{n-1}) \\ \vdots \\ V(\rho_n, z_{n-1}) \\ \hline V(\rho_0, z_n) \\ V(\rho_1, z_n) \\ \vdots \\ V(\rho_n, z_n) \end{bmatrix}$$

Inspired by the second-order elliptic equation, operators are modified to incorporate the aforementioned boundary conditions [50]:

$$\mathcal{L}_{cyl} = \begin{bmatrix} O_n & O_n & O_n & O_n & O_n & \cdots & \cdots & \cdots & \cdots & \cdots & O_n \\ \mathcal{I}_n & \mathcal{K}_n^* & \mathcal{I}_n & O_n & \vdots \\ O_n & \mathcal{I}_n & \mathcal{K}_n^* & \mathcal{I}_n & O_n & O_n & O_n & O_n & O_n & O_n & \vdots \\ O_n & O_n & \mathcal{I}_n & \mathcal{K}_n^* & \mathcal{I}_n & O_n & O_n & O_n & O_n & O_n & \vdots \\ \vdots & O_n & O_n & \ddots & \ddots & \ddots & O_n & O_n & O_n & O_n & \vdots \\ \vdots & O_n & O_n & O_n & \ddots & \ddots & \ddots & O_n & O_n & O_n & \vdots \\ \vdots & O_n & O_n & O_n & O_n & \ddots & \ddots & \ddots & O_n & O_n & \vdots \\ \vdots & O_n & O_n & O_n & O_n & O_n & \mathcal{I}_n & \mathcal{K}_n^* & \mathcal{I}_n & O_n & \vdots \\ \vdots & O_n & O_n & O_n & O_n & O_n & O_n & \mathcal{I}_n & \mathcal{K}_n^* & \mathcal{I}_n & O_n \\ \vdots & O_n & \mathcal{I}_n & \mathcal{K}_n^* & \mathcal{I}_n \\ O_n & \cdots & \cdots & \cdots & \cdots & \cdots & O_n & O_n & O_n & O_n & O_n \end{bmatrix}$$

$$\mathcal{K}_n^{(1)} = \frac{1}{h} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & \vdots \\ 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & \frac{1}{2^{(n-4)}} & 0 & \vdots \\ \vdots & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2^{(n-3)}} & 0 & \frac{1}{2^{(n-3)}} & 0 \\ \vdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2^{(n-2)}} & 0 & \frac{1}{2^{(n-2)}} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{K}_n^{(2)} = \frac{1}{h^2} \begin{bmatrix} -6 & 4 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots \\ 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & 0 & 0 & \ddots & -4 & 1 & 0 & \vdots \\ \vdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{K}_n^* = \mathcal{K}_n^{(1)} + \mathcal{K}_n^{(2)}$$

$$\mathcal{I}_n = \frac{1}{h^2} \left(\text{eye}(n) - (\text{zeros}(n)[n, n] + 1) \right)$$

$$O_n = \text{zeros}(n)$$

Where the above matrix is a tri-diagonal block sparse matrix with square embedded diagonal identity (\mathcal{I}_n) matrices with n non-zero diagonal elements, and sparse tridiagonal kernel (\mathcal{K}_n) matrices. O_n presents zero matrices with $n \times n$ sharing the dimensions of \mathcal{K}_n and \mathcal{I}_n . ³

³Not completely clear to the author at first, the motivation behind the structural choice of sparse block diagonals is a symptom of continuously vectorizing the computation. Some standard computing libraries may be proactive about this vectorization and apply it when recognized in for-loops. Even so, the explicit structuring of the boundary conditions listed here is intended to inform of the reasoning behind the modified numerical recipe to establish a compartmentalization between the scientific motivation alongside computing tools and methods.

Computing Potential:

$$V_{n+1} = (\mathcal{I}_{N^2} + \mathcal{L}_{cyl})V_n \quad (3.61)$$

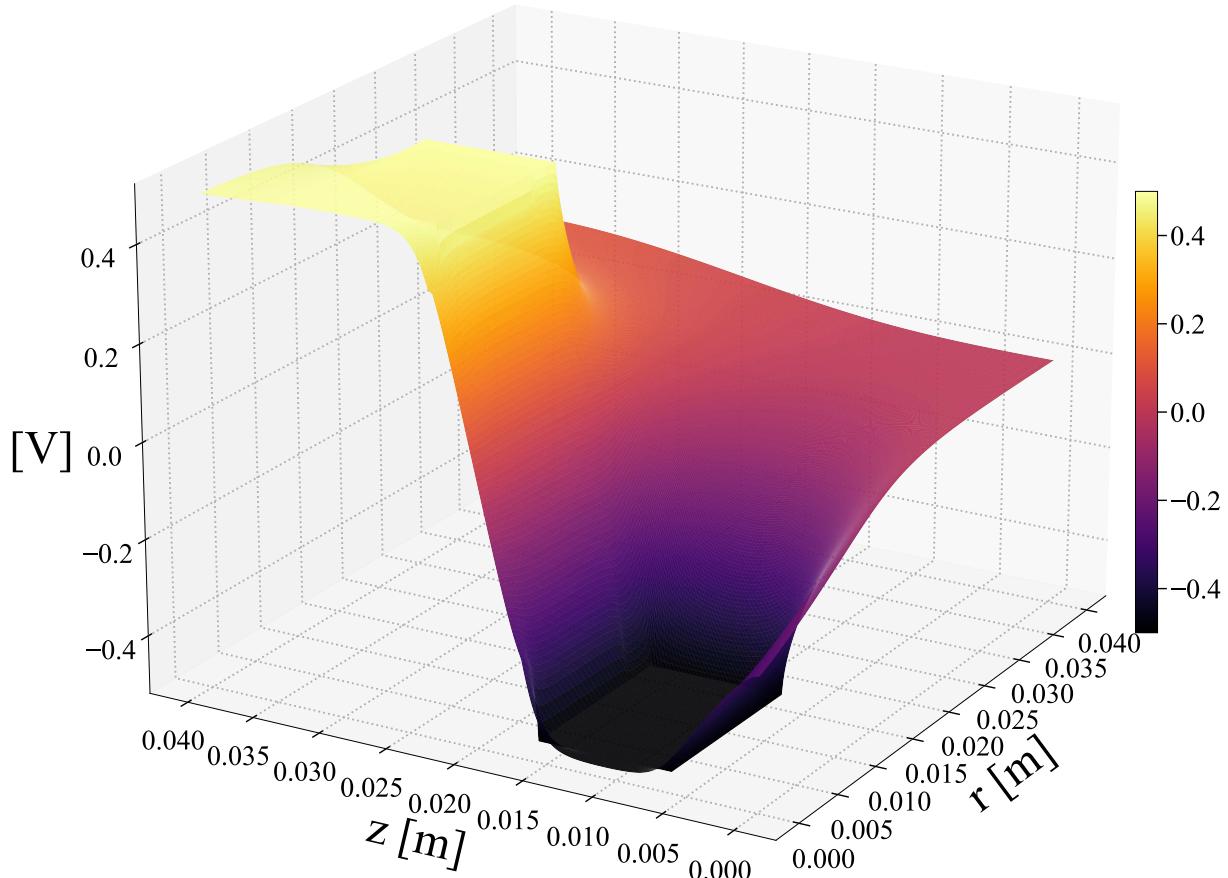


Figure 3.12: Numerically computed potential map estimate ($V(z, r)$ in cylindrical coordinates)

The computed E_z screened by the coating at $r = 0$ for $V = 1$ is estimated to be 13.3 [V/m] and will be included in the calibration as a pockels cell conversion efficiency of 13.3 [(V/m)/V]

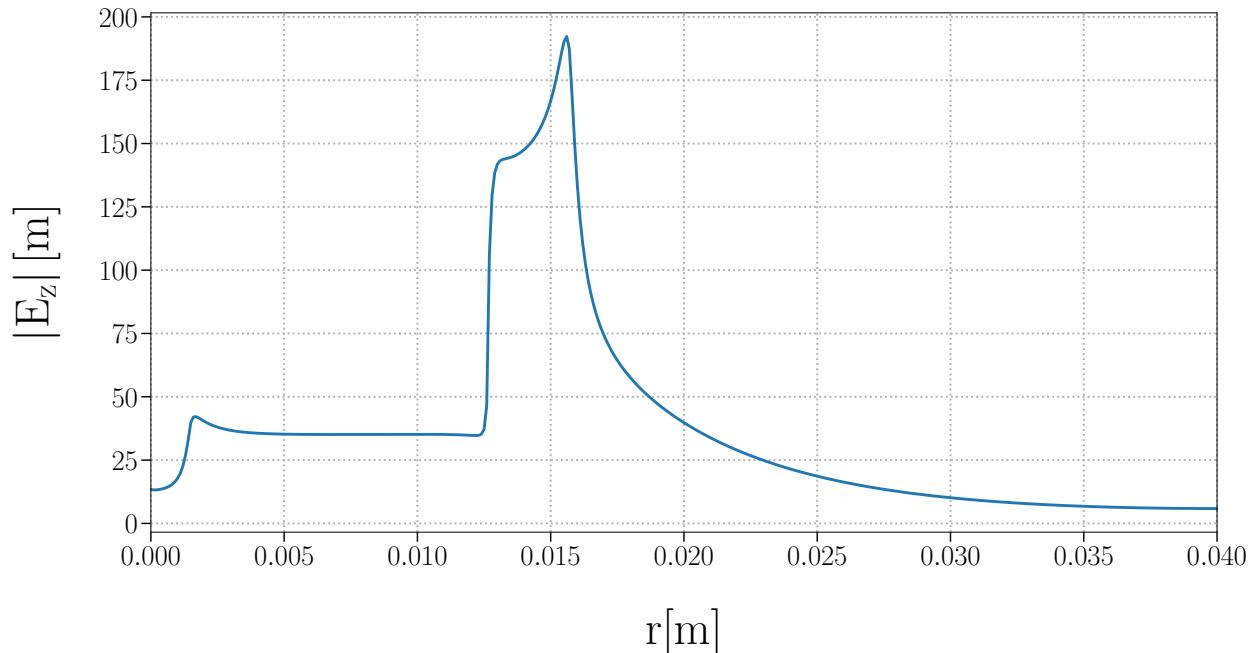


Figure 3.13: Plot of the $|E_z|$ field cross section sampled about the optic HR coating surface.

3.4.4 Servo Overview

3.4.5 Calibration

The Electric field coupling can be expressed as a measured voltage:

$$V = \frac{1}{1+G} \cdot n_{\text{laser}} - \frac{G}{1+G} \cdot C \cdot E \cdot A_2 \cdot V_{\text{in}} - \frac{F \cdot A_1}{1+G} \cdot n_S - \frac{A_1}{1+G} \cdot n_F$$

With G representing the open loop gain ($G = A_1 \cdot F \cdot S \cdot L$). The feedback signal can also be represented as:

$$\begin{aligned} V_{\text{out}} &= F \cdot S \cdot L \cdot (V + C \cdot E \cdot A_2 \cdot V_{\text{in}}) + F \cdot n_S + n_F \\ &= \frac{F \cdot S \cdot L}{1+G} \cdot C \cdot E \cdot A_2 \cdot V_{\text{in}} + \frac{F \cdot S \cdot L}{1+G} \cdot n_{\text{laser}} + \frac{F}{1+G} \cdot n_S + \frac{1}{1+G} \cdot n_F \end{aligned}$$

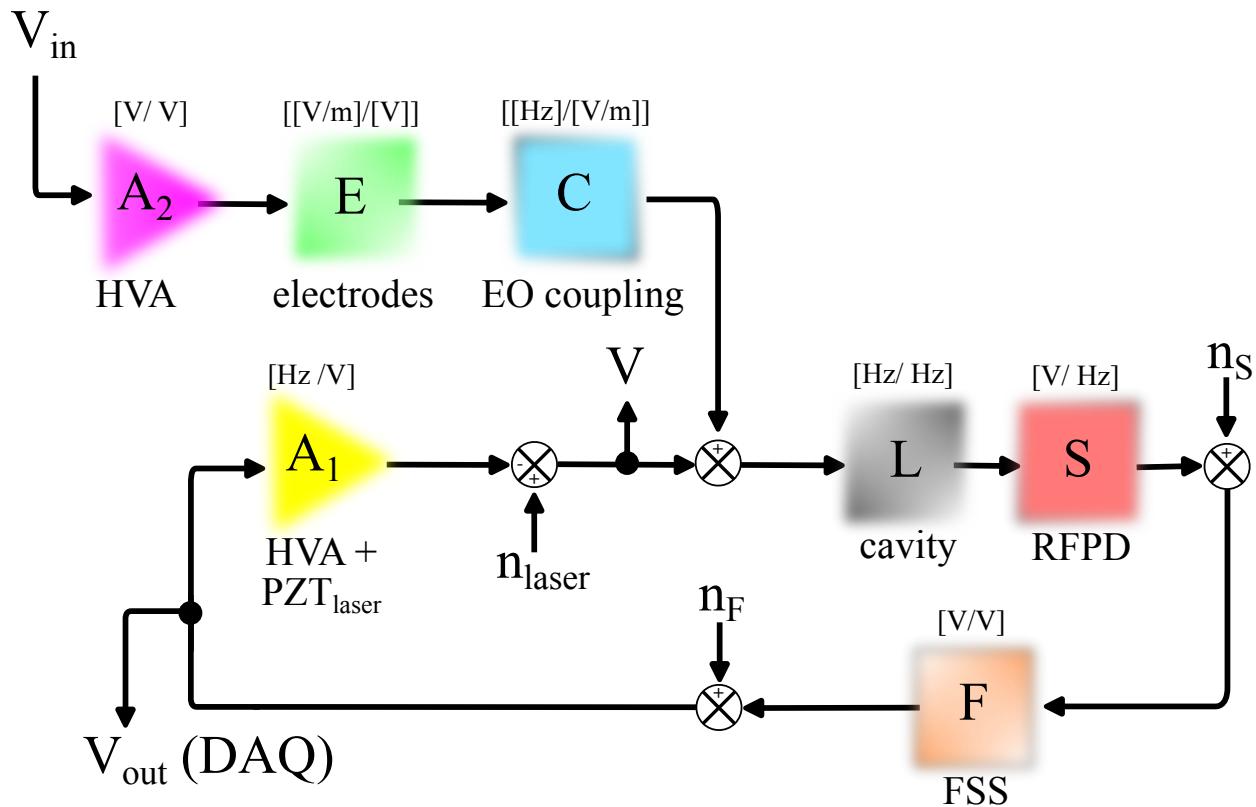


Figure 3.14: A controls diagram of the designed servo.

Therefore, the transfer function ($\frac{V_{out}}{V_{in}}$) :

$$\frac{V_{out}}{V_{in}} = \frac{F \cdot S \cdot L}{1 + G} \cdot C \cdot E \cdot A_2 + \frac{F \cdot S \cdot L}{1 + G} \cdot \frac{n_{laser}}{V_{in}} + \frac{F}{1 + G} \cdot \frac{n_S}{V_{in}} + \frac{1}{1 + G} \cdot \frac{n_F}{V_{in}}$$

If the induced excitation is larger than the noise we approximate the last equation:

$$\frac{V_{out}}{V_{in}} \approx \frac{F \cdot S \cdot L}{1 + G} \cdot C \cdot E \cdot A_2 = \frac{G}{1 + G} \cdot C \cdot E \cdot \frac{A_2}{A_1}$$

3.4.6 Assembly Mount Solution

To maintain the aforementioned boundary conditions in situ, an optical mount made of MACOR, a machinable ceramic, was constructed and installed. With the material's high Young's modulus (66.9 GPa), and a moderate Poisson ratio (.29) [51] making it by far the most durable / non-conductive mounting solution tried.

An optical mount for the sample made with MACOR, along with spherical glass bearings with a $.48 \pm .01$ cm \varnothing , and a McMaster-Carr 8-32, 1/2" ceramic screw were used to clamp the optical sample within a bored $25.74 \pm .5$ mm \varnothing barrel. Two 1.24" \varnothing holes were also bored at a 9 mm depth about the front and back side of the optical mount to accomodate for a flush fit of copper electrodes. The construction suggests a $1 \pm .5$ mm clearance between the front and back surface of the sample to the electrode plates.

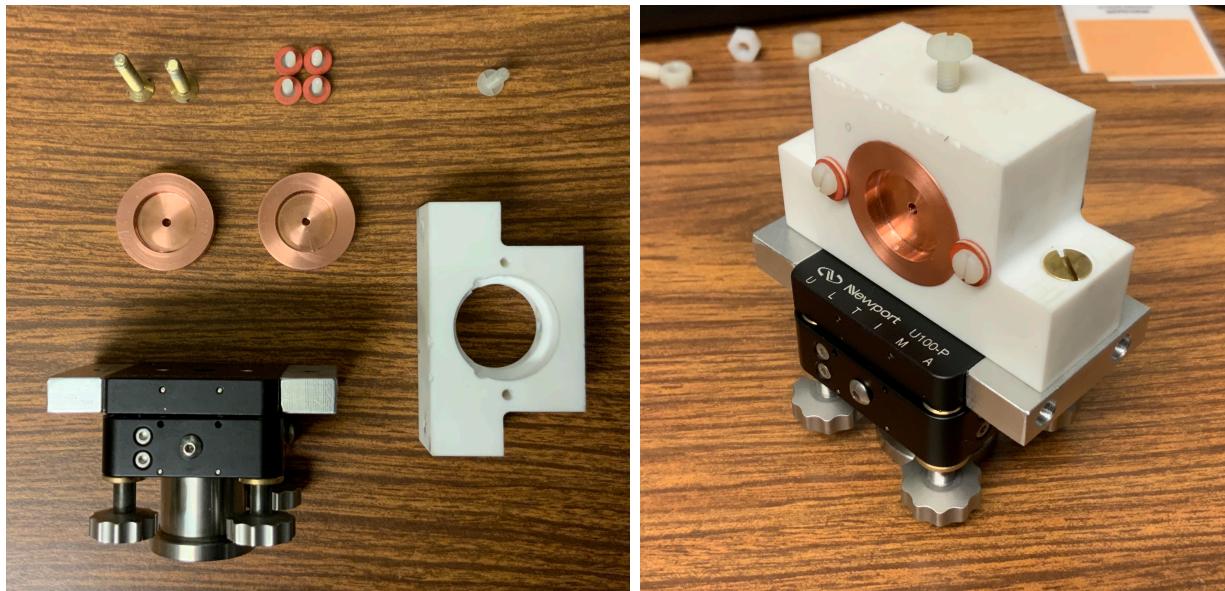


Figure 3.15: Assembly 3: a disassembled configuration and b an isometric view of the assembled configuration.

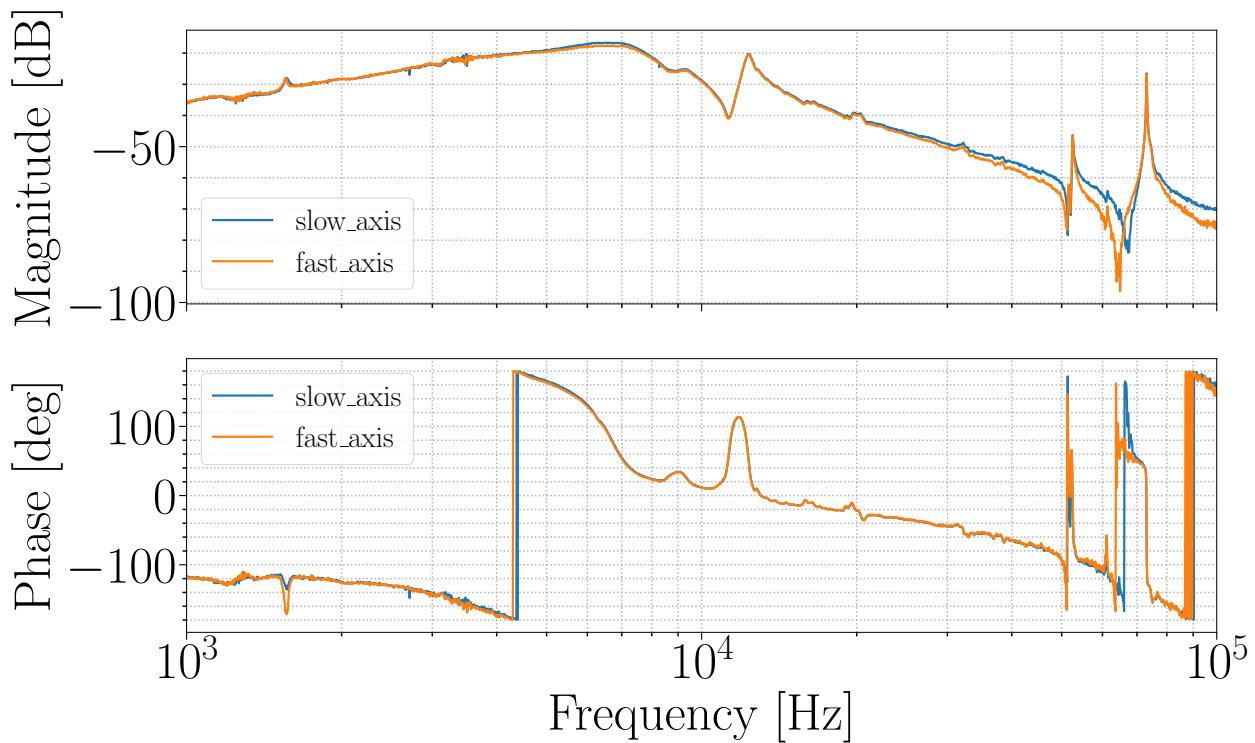


Figure 3.16: Uncalibrated measurement with the polarization aligned along the slow and fast GaAs / Al_{0.92}Ga_{0.08}As coating axes using the MACOR mount

3.4.7 Measured birefringence from HR GaAs/Al_{0.92}Ga_{0.08}As mirror

To identify the fast and slow axes, the crystalline HR mirror is temporarily paired with a highly reflective input mirror to form a high finesse cavity. The DC power in reflection of the cavity was probed while the laser frequency was linearly swept about resonance and various input laser polarizations sampled using a half waveplate ($\lambda / 2$). The resulting measurements exhibit a split cavity resonance when the beam polarization is not co-aligned with one of the two eigenaxes set by the HR sample birefringence.

From coating manufacturers it's estimated that static birefringence can arise from the intrinsic strain between hetero-epitaxial layers of GaAs/Al_{0.92}Ga_{0.08}As. [37, 41]

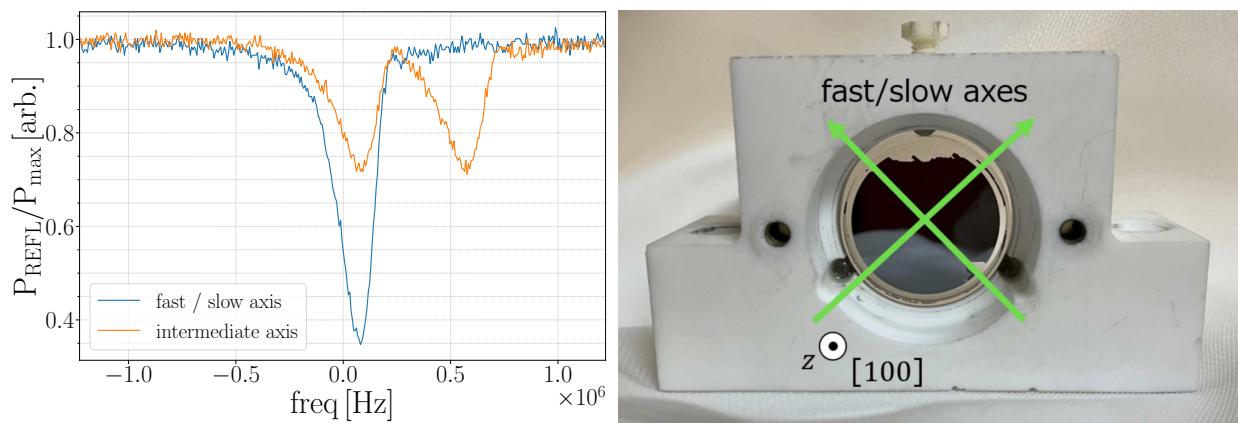


Figure 3.17: [Right] Lines running parallel to fast and slow axes. It was understood prior that a flat would be normal to one of the two eigenaxes. Measurement of the split resonance [Left] suggests no flat to be present and the damage seen is due to excessive handling. This split has the indicated frequency separation of 500 kHz.

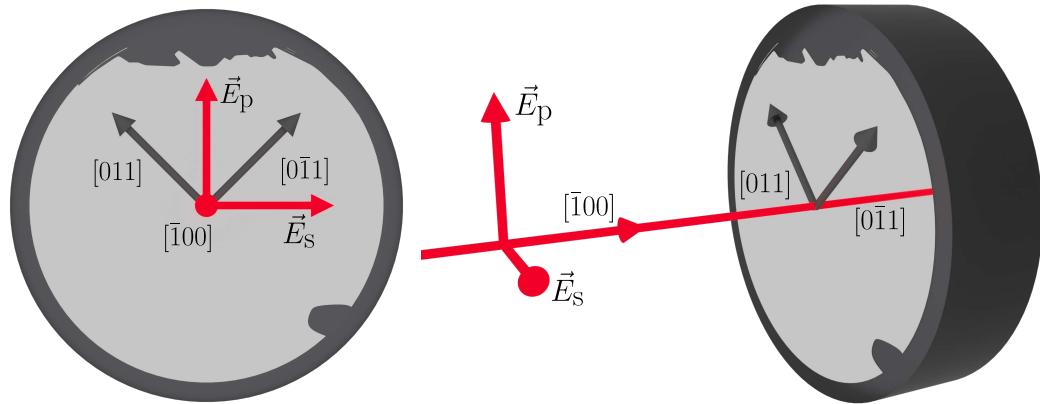


Figure 3.18: Input beam orientation with respect to the coating miller indices / fast and slow axes, with a [Left] Normal view and [Right] isometric view.

3.5 Results

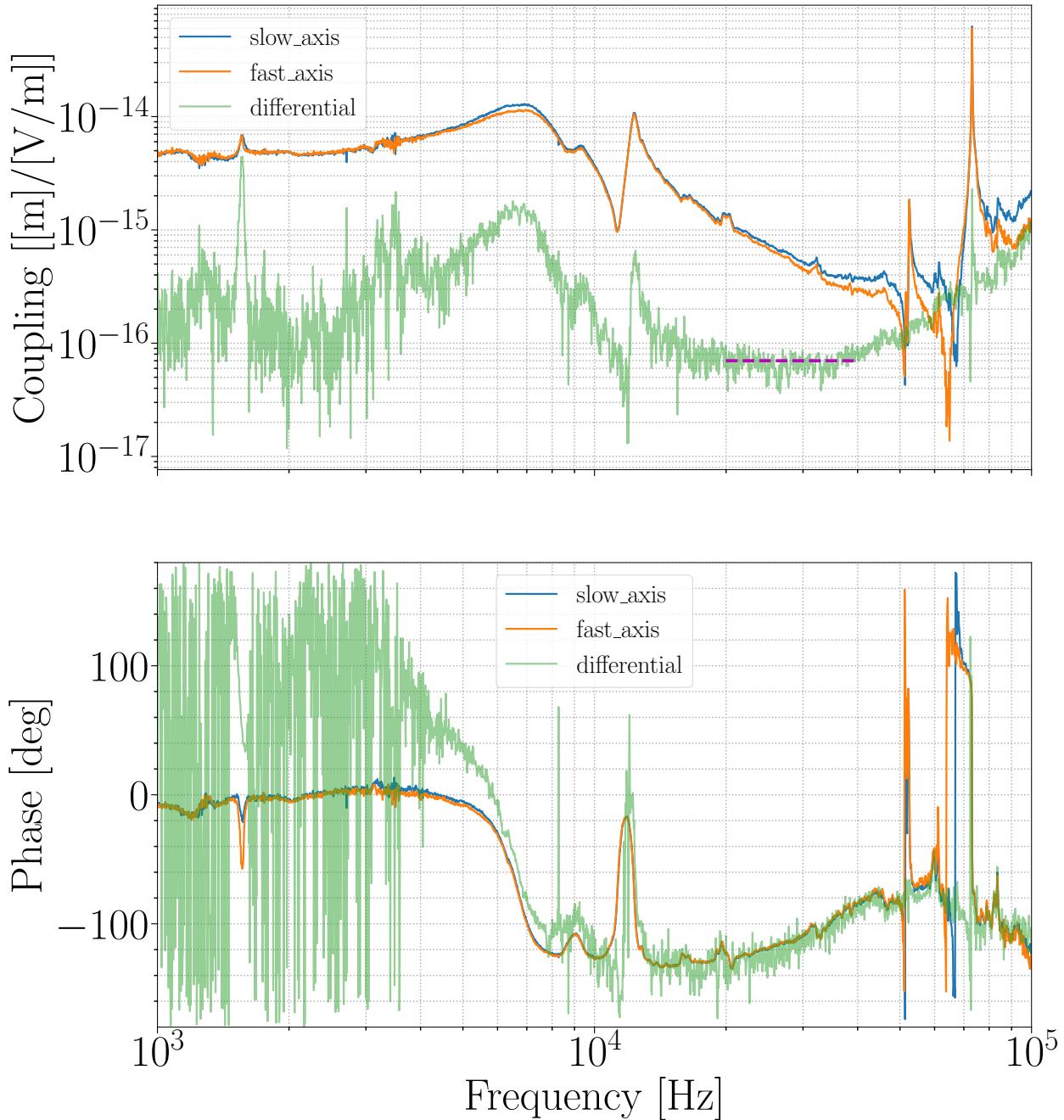


Figure 3.19: Calibrated measurement of the conversion factor of field strength [V/m] to length [m] when the beam input polarization is aligned along the slow and fast GaAs / Al_{0.92}Ga_{0.08}As coating axes.

From this experiment an upper threshold estimate between 20 kHz - 40 kHz is $\approx 1.1 \times 10^{-17}$ [m]/[V/m] though at higher frequency, frequency dispersion of the electro-optic coef-

ficients may explain the increase. The uniform flat response is extended to frequencies at which LIGO operates 10 Hz - 1 kHz as prior work has demonstrated such a trend from 10 Hz to several 10s of kHz.

The significantly improved thermal noise performance of GaAs/Al_{0.92}Ga_{0.08}As HR coatings make these crystalline coatings a prime candidate for the core optics in current and future gravitational wave detectors. The measured upper threshold of the electro-optic noise is demonstrated to be $\sim 10^{-26}$ [1/ $\sqrt{\text{Hz}}$] two orders of magnitude below the designed noise floor $\sim 10^{-24}$ [1/ $\sqrt{\text{Hz}}$] of current and future gravitational wave detectors.

3.5.1 Acousto-optical coupling

A significant barrier to a low cavity differential length noise was encountered when mounting the optics in accessible non-conductive materials. Firstly, most commercial optical mounts are constructed with conductive materials which proves problematic when working to isolate the coating from non-normal field gradients within the coating volume of interest. For this reason, efforts were focused on developing a suitable mounting solution that would provide adequate isolation from any uncontrolled field magnitudes while driving a field normally incident on the surface with enough strength and uniformity across the beam area to extract a measurement of the differential length change from the Pockels effect. The mounts studied span different geometries and different material properties. The varying geometrical assemblies rendered various responses and for this reason some discussions are divided accordingly. A hard lesson learned (while the answer being readily available in collaboration literature) was to select a material that tend towards narrow acoustic resonances while achieving low noise off resonance and at a bandwidth of interest. The most stable solution revealed that ringing of the sample acoustic modes may be responsible. For reassurance, we compare displacement spectra against that of a nearly identical cavity with a flat non-crystalline mirror coating to perform a null measurement reference; ruling out the Pockels effect and providing information for noise investigations.

Drive coupling seen in the transfer function measurements with the MACOR mounts appeared consistent. Varying the mechanical configurations (i.e. differential electrode and / or optic set screw settings) to the slightest degree left us to discover a mechanical action via the excitations from the sample-mount acoustic modes from the construction are rung when driving the voltage on electrodes plates. Tracking consistent mechanical action for assemblies prior to Assembly 3 were difficult due to inconsistencies in mechanical settings between some measurements. When more consistency was realized (i.e. utilizing a MACOR mount and careful tracking of the sample set screw torque), it was more transparent that the sample-mount acoustic modes were responsible with similar features between the GaAs/Al_{0.92}Ga_{0.08}As coating and the non-crystalline HR coating from ATFilms. The cleanest versions of these measurements with coorespondence of FEA suggest the acousto-optical modes with highest contributions of cavity length noise at regions below 10 kHz and 40 kHz - 80 kHz.

Chapter 4

Conclusion

4.1 Adaptive optics comissioning

Since the comissioning that had taken place in O3a, there has been heavy consideration into how point absorbers manifest on the core optic surfaces and ways to mitigate their effects if unavoidable. Some of these considerations include but are not exclusive to: non-invasive pre-installation measurements of ITM surface / coating quality [52], upgrades to TCS to expand upon the current thermal compensation actuation modes, and scheduled vaccum chamber venting specifically for replacing an offending test mass mirror.

Increasing interferometer input power is an inevitability to reaching designed detector sensitivity and developing mode matching contingency plans seem to be a natural progression of the current adaptive optics schema. The technique for improved ring heater transient response represents a small iterative step of pushing the existing and future thermal mode matching infrastructure towards a larger scale evolving adaptive optics schema. Some next immediate steps within the author's limited view:

- Improved measurement and filter fitting of the RH thermo-optic response
- Improving the thermo-optic transient of the CO₂ actuators with unique pre-filters using

the procedure listed in § 5.5.1

- Improving TCS synchronization when raising input power to potentially improve the thermo-optic carrier transient from the high circulating arm power.
- Optimization of simultaneous thermo-optic transients between actuators and the carrier

4.2 GaAs/Al_{0.92}Ga_{0.08}As **Electro-optic noise**

Though with limited sensitivity an upper limit with the experiment is established. The discovery of driven mechanical couplings within the longitudinal pockels cell mount are discussed and were shown to be a major limitation with driven electric field injections indicated and have lead to an improved dual-polarization locked experimental design for improved sensitivity. Some next steps that can be taken with a similar experiments:

- More narrow investigation into improved mechanical isolation to improve SNR at low frequency
- Careful modelling of opto-mechanical resonances to improve a model to inform a required separation (if any) between the photo-elastic and electro-optic effects

Chapter 5

Appendix

5.1 Interferometer Configurations (code)

5.1.1 ifo_configs.py

```
1 import numpy as np
2
3 # Bode tools
4 def bode_amp(H):
5     """
6         Returns amplitude information on transfer function (H)
7     """
8     return np.sqrt(np.real(H)**2 + np.imag(H)**2)
9
10 def bode_ph(H):
11     """
12         Returns phase information on transfer function (H)
13     """
14     return (180/np.pi)*np.arctan(np.imag(H)/np.real(H))
15
16 # some constants:
17 cee = np.float64(299792458) ## speed of light [m/s]
18 h_bar = (6.626e-34)/(2*np.pi) ## planck's constant
19
20
21 # IFO params
22 def finesse(r_i, r_e):
23     """
```

```

24     r_i : ITM reflectivity coefficient
25     r_e : ETM reflectivity coefficient
26     """
27     return np.pi*np.sqrt(r_i*r_e)/(1-(r_i*r_e))
28
29
30 # Michelson frequency response
31 def mich_freq_resp(freq, Length, phi_0, P_in, OMEGA):
32     """
33     MICHELSON FREQUENCY RESPONSE CALCULATOR
34     freq : standard (gravitational wave) frequency [Hz]
35     Length : Michelson ifo arm length [m]
36     phi_0 : static differential arm length tuning phase [rad]
37     P_in : input power [W]
38     """
39     return (P_in*OMEGA*np.sin(phi_0))*Length*
40         np.exp((-1j*Length*2.0*np.pi*freq)/cee)*
41         np.sin((Length*2.0*np.pi*freq)/cee)/(Length*2.0*np.pi*freq)
42
43 def fpmi_freq_resp(freq, r_1, t_1, r_2, L, phi_0, P_in, OMEGA, low_pass=False):
44     """
45     FABRY PEROT MICHELSON FREQUENCY RESPONSE CALCULATOR
46     freq : standard (gravitational wave) frequency [Hz]
47     r_1, t_1, r_2: Assuming arm symmetry where the ITM has r_1, t_1 coefficients
48             and the ETM has a r_2 reflectivity coefficient.
49             Also assumes no loss. [arb]
50     OMEGA: OPTICAL angular frequency [rad Hz]
51     Length: Michelson ifo arm length [m]
52     phi_0 : static differential arm length tuning phase [rad]
53     """
54     if low_pass:
55         f_pole = 1/(((4*np.pi*L)*np.sqrt(r_1*r_2))/(cee*(1-r_1*r_2)))
56         fpmi_resp = 1/(1 + 1j*(freq/f_pole))
57     else:
58         fpmi_resp = ((t_1**2 * r_2)/((t_1**2 + r_1**2)*r_2 - r_1))*(
59             (mich_freq_resp(freq, L, phi_0, P_in, OMEGA)/
60             (1-r_1*r_2*np.exp(-1j*L*4.0*np.pi*freq/cee))))
61     return fpmi_resp
62
63 def PRG(L_rt, Finn, r_PRM, max=0):
64     """
65     POWER RECYCLING GAIN (@ optimal reflectivity)
66     * Assuming a FPMI with symmetric arms *
67     L_rt : Round trip loss
68     Finn : Cavity finesse

```

```

69 """
70 if max == 1:
71     G_PR = np.pi/(2*Finn*L_rt*(1-((Finn*L_rt)/(2*np.pi))))
72 else:
73     G_PR = (1-r_PRM**2)/(1-r_PRM*(1-(Finn/np.pi)*L_rt))**2
74
75
76 return G_PR
77
78 def drfpmi_freq_resp(freq, G_PRC_opt, r_1, t_1, r_2, r_SRM, t_SRM, phi_SRC, L,
79                      phi_0, P_in, OMEGA):
80 """
81 DUAL RECYCLED FABRY PEROT MICHELSON FREQUENCY RESPONSE
82 CALCULATOR
83
84 freq: standard (gravitational wave) frequency [Hz]
85 G_PRC_opt: maximum power recycling gain (optimal) [arb]
86 r_1: ITM reflection coefficient [arb]
87 t_1: ITM transmission coefficient [arb]
88 r_2: ETM reflection coefficient [arb]
89 r_SRM: Signal recycling mirror reflection coefficient [arb]
90 t_SRM: Signal recycling mirror transmission coefficient [arb]
91 L: Length of the Fabry-Perot arms [m]
92 OMEGA: OPTICAL angular frequency [rad Hz]
93 """
94 r_SRC = (r_1 - r_SRM*np.exp(1j*2*phi_SRC))/(
95         (1 - r_1*r_SRM*np.exp(1j*2*phi_SRC)))
96 t_SRC = t_1*t_SRM*np.exp(1j*phi_SRC)/(1 - r_1*r_SRM*np.exp(1j*2*phi_SRC))
97
98 return ((t_1**2 * r_2)/((t_1**2 + r_1**2)*r_2 - r_1))*(
99     G_PRC_opt*t_SRC*(P_in*L*OMEGA*np.exp((-1j*L*2.0*np.pi*freq)/cee)*
100     np.sin((L*2.0*np.pi*freq)/cee)/(L*2.0*np.pi*freq))/(
101     (1-r_SRC*r_2*np.exp(-1j*L*4.0*np.pi*freq/cee)))
102
103
104 # Shot noise
105 def N_shot(OMEGA, P_in):
106 """
107 Interferometer shot noise calculator
108 OMEG: OPTICAL angular frequency [rad Hz]
109 Length : ifo arm length [m]
110 phi_0 : static differential arm length tuning phase [rad]
111 P_in : Input power [W]
112 """
113 return np.sqrt(2*h_bar*OMEGA*P_in)

```

5.1.2 MICH

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import os
4 import sys
5 sys.path.insert(0,'..')
6 plt_style_dir = '../..//stash/'
7 fig_exp_dir = '../..//figs/'
8 from ifo_configs import N_shot
9 from ifo_configs import mich_freq_resp as MICH
10 from ifo_configs import bode_amp, bode_ph
11 %matplotlib inline
12 if os.path.isdir(plt_style_dir) == True:
13     plt.style.use(plt_style_dir + 'ppt2latexsubfig.mplstyle')
14 plt.rcParams["font.family"] = "Times New Roman"

```

```

1 # Some parameters
2 cee = np.float64(299792458)
3 h_bar = (6.626e-34)/(2*np.pi)
4 OMEG = np.float64(2*np.pi*cee/(1064.0*1e-9))
5 L = np.float64(4000.0)
6 nu = np.arange(1, 1000000, 1)
7 PHI_0 = np.pi/2 #[rad]
8 P_IN = 125 #[W]

```

5.1.2.1 Derivation

For the simple Michelson we know that a change in arm length correlates to light at the AS port We also know that a differential arm length corresponds to a difference in phase of the light that impinges upon the BS For a gravitational wave we can quantify the phase difference in this following way:

$$\phi_A - \phi_B = \int_{t-2L/c}^t \Omega \left[1 + \frac{1}{2} h(t) \right] dt - \int_{t-2L/c}^t \Omega \left[1 - \frac{1}{2} h(t) \right] dt \quad (5.1)$$

The phase difference can then be quantified by:

$$\phi_A - \phi_B = \int_{t-2L/c}^t \Omega h(t) dt \quad (5.2)$$

where

$$h(t) = h_0 e^{i\omega t} \quad (5.3)$$

Ω is the **optical angular frequency**

After evaluating this integral we get:

$$\Delta\phi = \phi_A - \phi_B = \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c} \cdot h_0 e^{i\omega t} \quad (5.4)$$

Where the first term in the phase difference carries all the time independent frequency information. This is what we are calculating below.

For the sake of being explicit, we are going to plot:

$$\Delta\phi(\omega) = h_0 \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c} \quad (5.5)$$

This accounts for the differential phase as a function of gravitational wave frequency, though we have not established the amount of optical gain the Michelson offers. This can be understood through a first order taylor approximation about a selected Michelson offset angle ϕ_0 :

$$P(\omega, \phi_0) = \frac{P_{\text{in}}}{4} [r_x^2 + r_y^2 - 2r_x r_y \cos(\phi_0 + \Delta\phi(\omega))] \quad (5.6)$$

$$P(\omega, \phi_0) \approx \frac{P_{\text{in}}}{4} \left[r_x^2 + r_y^2 - 2r_x r_y (\cos(\phi_0) - \Delta\phi(\omega) \cdot \sin(\phi_0)) \right] = \frac{P_{\text{in}}}{2} \left[1 - (\cos(\phi_0) - \Delta\phi(\omega) \cdot \sin(\phi_0)) \right] \quad (5.7)$$

Where we define a response gain function H_{MICH} :

$$H_{\text{MICH}}(\omega, \phi_0) = \frac{P_{\text{in}}}{2} \cdot \Delta\phi(\omega) \cdot \sin(\phi_0) \quad (5.8)$$

```

1 H = MICH(nu, L, PHI_0, P_IN, OMEG)
2
3 fig, ax1 = plt.subplots()
4 ax1.set_xlabel('frequency [Hz]')
5 ax1.set_ylabel('H$\cdot\mathbf{W/m}$', color='C0')
6 #ax1.plot(w/(FSR), F_w_cc_modsq*100)
7 ax1.loglog(bode_amp(H), linewidth=7.5, color='C0')
8 #plt.ylim([10e-6, 10e0])
9 ax2 = ax1.twinx()
10 #ax2.plot(w/(FSR), (180/np.pi)*np.arctan(F_w_cc.imag/F_w_cc.real), '--')
11 ax2.semilogx(nu,(180/np.pi)*np.arctan(np.imag(H)/np.real(H)), '--', linewidth=7.5,
12               color='C1')
13 #plt.xlabel('frequency [FSR]')
14 plt.xlim([1,1e5])
15 plt.ylabel('phase [deg]', color='C1')
16 fig.savefig(fig_exp_dir + 'INTRO/mich_fr.pdf', dpi=300, bbox_inches='tight')
```

Though with the provided frequency dependence and optical gain, we still need to understand a starting noise floor spectra and compare to our anticipated limiting noise

Shot noise

* A fundamental limit imposed by the statistical nature of photon counting

* The photon counting follows Poisson statistics

* Photon counting variance (variance is equal to the mean)

$$\langle (n - \bar{n})^2 \rangle = \frac{P\Delta t}{\hbar\Omega} \quad (5.9)$$

* Power variance:

$$\langle (P - \bar{P})^2 \rangle = \hbar\Omega\bar{P}\Delta t \quad (5.10)$$

* PSD of the measured power between two uncorrelated moments in time:

$$S_P(\omega) = \lim_{T \rightarrow \infty} \frac{2}{T} \left\langle \left| \int_{-T}^T (P(t) - \bar{P}) e^{-i\omega t} dt \right|^2 \right\rangle \quad (5.11)$$

$$= \lim_{T \rightarrow \infty} \frac{2}{T} \int_{-T}^T \hbar\Omega\bar{P} dt \quad (5.12)$$

$$= 2\hbar\Omega\bar{P} \quad (5.13)$$

* Where the ASD is:

$$[S_P(\omega)]^{1/2} = [2\hbar\Omega\bar{P}]^{1/2} \quad (5.14)$$

The signal to noise is established by dividing the frequency dependent optical gain times the gravitational wave ASD ($[S_h(\Omega)]^{1/2}$) by the noise ASD:

$$\text{SNR} = G_{\text{opt}}(\omega)[S_h(\omega)]^{1/2}/S_N(\omega) = H_{\text{MICH}}/[S_P]^{1/2} = \left(\frac{\Delta\phi(\omega)}{h_0} \frac{P_{\text{in}}}{2} \sin(\phi_0) \right) / [2\hbar\Omega\bar{P}]^{1/2} \quad (5.15)$$

This is to say that for the stated gravitational wave ASD, and for an SNR of 1, we establish the following threshold for detector:

$$[S_h(\omega)]^{1/2} \{ \text{SNR} \geq 1 \} \geq \frac{[S_N(\omega)]^{1/2}}{H_{\text{MICH}}(\omega)} \quad (5.16)$$

Where

$$\frac{[S_N(\omega)]^{1/2}}{H_{\text{MICH}}(\omega)} = \frac{[2\hbar\omega\bar{P}]^{1/2}}{\Delta\phi(\omega)[P_{\text{in}}/2]\sin(\phi_0)} = \left(\frac{\hbar\Omega}{\omega P_{\text{in}}} \right)^{1/2} \frac{[r_x^2 + r_y^2 - 2r_x r_y \cos(\phi_0)]^{1/2}}{\sin(L\omega/c)} e^{iL\omega/c} \quad (5.17)$$

1 S_h = N_shot(OMEG, P_IN)

```

2 print(S_h)

1 #ax1.plot(w/(FSR), F_w_cc_modsq*100)
2 plt.loglog(nu, S_h/bode_amp(H), linewidth=7.5, color='C0')
3 plt.ylim([1e-21, .5e-14])
4 plt.xlabel('frequency [Hz]')
5 plt.ylabel('$\mathbf{S}_h$');
6     '\mathbf{[ 1 / \sqrt{\mathbf{Hz}}]}$'
7 #ax2_ = ax1_.twinx()
8 #ax2.plot(w/(FSR), (180/np.pi)*np.arctan(F_w_cc.imag/F_w_cc.real), '--')
9 #ax2_.semilogx(nu,(180/np.pi)*np.arctan(np.imag(S_h)/np.real(S_h)), '--',
10                 linewidth=7.5,color='C1')
11 #plt.xlabel('frequency [FSR]')
12 plt.xlim([1,1e5])
13 plt.grid(visible=True)
14 #plt.subplots_adjust(hspace = 1)
15 #plt.ylabel('phase [deg]',color='C1')
16 #plt.tight_layout(rect=[0,0,1,1])
17 #plt.title('')
18 #plt.subplots_adjust(bottom=.1, top=.85) #, right=.8, left=.1)
19 plt.savefig(fig_exp_dir + 'INTRO/mich_sensi.pdf', dpi=300, bbox_inches='tight')

```

5.1.3 FPMI

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import scipy.signal as sig
4 import os
5 import sys
6 sys.path.insert(0,'..')
7 plt_style_dir = '.../stash/'
8 fig_exp_dir = '.../../figs/'
9 from ifo_configs import mich_freq_resp as MICH
10 from ifo_configs import fpmi_freq_resp as FPMI
11 from ifo_configs import N_shot, bode_amp, bode_ph
12 if os.path.isdir(plt_style_dir) == True:

```

```

13 plt.style.use(plt_style_dir + 'ppt2latexsubfig.mplstyle')
14 plt.rcParams["font.family"] = "Times New Roman"
15 line_width=7.5

```

Let's start with the simple Fabry Perót cavity. The following are equations that characterize the circulating and reflected fields (both critical to measuring the phase response of the FP cavity to GWs):

$$E(t) = t_1 E_{in} + r_1 r_2 E(t - 2T) e^{-i\Delta\phi(t)} \quad (5.18)$$

$$E_r(t) = -r_1 E_{in} + t_1 r_2 E(t - 2T) e^{-i\Delta\phi(t)} \quad (5.19)$$

$T = L/c$ is the time it takes light to reach the end of the cavity and $\Delta\phi(t)$ is the phase rotation.

We can define the static phase rotation (no GW passing through) as :

$$\Delta\phi = 2kL = 4\pi L/\lambda_{opt} \quad (5.20)$$

And if L is tuned just right $2kL = 2\pi n$ so the cavity is just tuned for resonance

If we put a gravitational wave in the mix we redefine this phase rotation as such that:

$$\Delta\phi = \frac{\omega_0}{2} \int_{t-\frac{2L}{c}}^t h(t') dt' \quad (5.21)$$

This assumes that the static phase rotation satisfies $2\omega_0 L/c = 2\pi n$. Which is the same thing that we said above but with different symbols (because we're fancy ;D)

Say that we have something that does throw the cavity slightly off resonance.. doesn't have to be a gravitational wave... but that's what we hope for. ANYWAY...

If the $\Delta\phi$ becomes such that the cavity is thrown off resonance we get a time dependent intra-cavity field:

$$E(t) = \bar{E} + \delta E(t) \quad (5.22)$$

and if the phase rotation ($\Delta\phi$) is super small... which is pretty much guaranteed with gravity waves, we can say:

$$e^{i\Delta\phi} = 1 - i\Delta\phi \quad (5.23)$$

Using equations 5.22 and 5.23 in 5.18 we get:

$$\bar{E} + \delta E(t) = t_1 E_{in} - r_1 r_2 \bar{E} + r_1 r_2 \delta E(t - 2T) - i r_1 r_2 \bar{E} \Delta\phi(t) \quad (5.24)$$

We can parse this into time dependent and time independent terms:

$$\bar{E} = t_1 E_{in} - r_1 r_2 \bar{E} \quad (5.25)$$

$$\delta E(t) = r_1 r_2 \delta E(t - 2T) - i r_1 r_2 \bar{E} \Delta\phi(t) \quad (5.26)$$

Since the time dependent phase information is encoded in 5.26 we will take the laplace transform of this equation to yield:

$$\delta E(s) = -i \frac{r_1 r_2 \bar{E}}{1 - r_1 r_2 e^{-2sT}} \Delta\phi(s) \quad (5.27)$$

YAS! we are now one step closer to getting a useful expression for the phase response. But again.. what does this last equation mean? That last equation is how the change in the electric field directly relates to a small perturbation in phase (which could be either a small change in laser frequency or length modulation)

Now.. we're not done yet because that last expression does not tell us the entire story yet.. we want to see how this effects the phase differential with the **reflected** electric field.

To do this.. we have to combine equations 5.18 and 5.19. (an easy way to do this is to get rid of the $r_2 E(t - 2T) e^{-i\Delta\phi(t)}$) :

$$E_r(t) = \frac{t_1}{r_1} E(t) - \frac{t_1^2 + r_2^2}{r_1} E_{in} \quad (5.28)$$

if the cavity is unperturbed:

$$\bar{E}_r = \left(\frac{r_2(r_1^2 + t_1^2) - r_1}{t_1} \right) \bar{E} \quad (5.29)$$

and if we perturb the cavity we see that the change in the intra-cavity field is directly related to the change in the reflected field:

$$\Delta\phi_r(s) \equiv \frac{\delta E(s)}{\bar{E}} = \frac{t_1^2 r_2}{(t_1^2 + r_1^2)r_2 - r_1} \frac{\Delta\phi(s)}{1 - r_1 r_2 e^{-2sT}} \quad (5.30)$$

This implies that there is an additional frequency dependent factor in your phase shift and this translates into your FPMI transfer function as:

$$H_{FPMI}(\omega_g) = \frac{2\Delta\phi_r(\omega_g)}{h(\omega_g)} = \frac{t_1^2 r_2}{(t_1^2 + r_1^2)r_2 - r_1} \frac{H_{MI}(\omega_g, L)}{1 - r_1 r_2 e^{-2i\omega_g L/c}} \quad (5.31)$$

Whew.... that was a lot.... now let's code it up Since we can seperate the calculation into two.. I'm going to parse out the calculation between the constant Fabry Perót term and the term with the frequency dependence. But first, lets set up our parameters for our FPMI:

```

1 # Some parameters
2 cee = np.float64(299792458)
3 OMEG = np.float64(2*np.pi*cee/(1064.0*1e-9))
4 L = np.float64(4000.0)
5 nu = np.arange(1, 1000000, 1)
6 nat_nu = [np.float64(i*2*np.pi) for i in nu]
7 h_0 = np.float64(1)
8
9 PHI_0 = np.pi/2 #[rad]
```

```

10 P_IN = 25
11
12 T_1 = .014
13 #T_1 = 25e-6
14 T_2 = 50e-6
15 R_1 = 1-T_1
16 R_2 = 1-T_2
17
18 t_1 = T_1**.5
19 r_1 = R_1**.5
20 r_2 = R_2**.5

```

Now we can compute:

$$H_{FPMI}(\omega_g) = \frac{t_1^2 r_2}{(t_1^2 + r_1^2)r_2 - r_1} \cdot \frac{H_{MI}(\omega_g, L)}{1 - r_1 r_2 e^{-2i\omega_g L/c}} \quad (5.32)$$

1 H_FPMI = FPMI(nu, r_1, t_1, r_2, L, PHI_0, P_IN, OMEG)

We estimate the FP's pole frequency

$$1 - r_1 r_2 e^{-2i\omega_g L/c} = 0 \quad (5.33)$$

therefore when:

$$e^{-i\omega_g L/c} = \frac{1}{\sqrt{r_1 r_2}} \quad (5.34)$$

we acquire the pole frequency ω_{pole} as indicated in the low pass

$$f_{pole} = \frac{1}{4\pi\tau_s} = \frac{c}{4\pi L} \frac{1 - r_1 r_2}{\sqrt{r_1 r_2}} = \frac{\nu_{FSR}}{2\pi} \frac{1 - r_1 r_2}{\sqrt{r_1 r_2}} = \frac{\nu_{FSR}}{\mathcal{F}} \quad (5.35)$$

Also, understanding that the cavity Finesse can be defined as

$$\mathcal{F} = \frac{\pi\sqrt{r_i r_e}}{1 - r_i r_e} \quad (5.36)$$

we also can invert for a high value of finesse $\mathcal{F} >> \pi$:

$$r_i r_e \approx 1 - \frac{\pi}{\mathcal{F}} \quad (5.37)$$

```

1 f_pole = 1/(((4*np.pi*L)*np.sqrt(r_1*r_2))/(cee*(1-r_1*r_2)))
2 def fpml_lp(freq, cav_pole):
3     return 1/(1 + 1j*(freq/cav_pole)) #*np.exp(1j*freq/cav_pole))
4 H_FPMI_LP = fpml_lp(nu, f_pole)

```

Might as well compare it to our Michelson response:

$$H_{MI}(\omega_g) = \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c} \quad (5.38)$$

```

1 H_MICH = MICH(nu, L, PHI_0, P_IN, OMEG)
2
3 fig, ax1 = plt.subplots()
4 ax1.set_xlabel('frequency [Hz]')
5 ax1.set_ylabel('H$\backslash$mathdefault{FPMI} $\backslash$; $\backslash$mathdefault{ [W / m] } $', color='C0')
6 #ax1.plot(w/(FSR), F_w_cc_modsq*100)
7 ax1.loglog(bode_amp(H_FPMI), label='FPMI', linewidth=line_width,color='C0')
8 #ax1.loglog(w,H_MI_modsq, label= 'MICH', linewidth= 5)
9 #ax1.loglog(w,H_FPMI_LP_modsq*H_FPMI_modsq[0], label='FPMI LP',
10             linewidth = 20.0, alpha=0.25,color='C2')
11 #ax1.axvline (x=f_pole,ymin=1e-13, color='red', linestyle='dotted', linewidth=3)
12 ax2 = ax1.twinx()
13 ax2.semilogx(nu,bode_ph(H_FPMI),'--', linewidth=line_width, color='C1')
14 #ax2.semilogx(w,(180/np.pi)*np.arctan(np.imag(H_MI)/np.real(H_MI)), '--')
15 #ax2.semilogx(w,(180/np.pi)*np.arctan(np.imag(H_FPMI_LP)/np.real(H_FPMI_LP)),
16                 linestyle='--', linewidth=20.0,dashes=(4,10),alpha=.25, color='C2')
17 plt.xlim([1,1e5])
18 plt.ylabel('phase [deg]', color='C1')
19 #fig.savefig('../figs/INTRO/fpmi_fr.pdf', dpi=300, bbox_inches='tight')

```

```

1 Text(0, 0.5, 'phase [deg]')
2
3 plt.loglog(nu,bode_amp(H_MICH), label= 'MICH', linewidth= line_width, alpha=.5)
4 plt.loglog(nu,bode_amp(H_FPMI), label='FPMI', linewidth=line_width)
5 #plt.loglog(nu,bode_amp(H_FPMI_LP)*bode_amp(H_FPMI)[0], label='FPMI LP',
6             linewidth = 20.0, alpha=0.25)

```

```

5 plt.axvline (x=f_pole,ymin=1e-11, color='red', linestyle='dotted', linewidth=3.0)
6 plt.ylim([5e7, 5e14])
7 plt.xlim([1e0, 1e5])
8 #plt.grid(visible=True, which='minor', axis='y')
9 plt.xlabel('frequency [Hz]')
10 plt.ylabel('H(f) $\mathbf{[W/m]}$')
11 lgd=plt.legend()
12 plt.savefig('../figs/INTRO/fpmi_fr.pdf', dpi=300, bbox_inches='tight')

```

```

1 plt.semilogx(nu,bode_ph(H_MICH), '--', label='MICH', linewidth= line_width, alpha=.5)
2 plt.semilogx(nu,bode_ph(H_FPMI), '--', label='FPMI', linewidth= line_width)
3 #plt.semilogx(nu,bode_ph(H_FPMI_LP),linestyle='--', linewidth=3.0,dashes=(3,10))
4 plt.xlim([1,100000])
5 plt.ylabel('phase [deg]')
6 plt.xlabel('Frequency [Hz] ')
7 lgd=plt.legend()

```

```
1 Sh_noise = N_shot(OMEG, P_IN)
```

```

1 plt.loglog(nu,Sh_noise/bode_amp(H_MICH), label= 'MICH',
            linewidth= line_width, alpha=.5)
2 plt.loglog(nu,Sh_noise/bode_amp(H_FPMI), label= 'FPMI', linewidth=line_width)
3 #plt.loglog(nu,Sh_noise/(bode_amp(H_FPMI_LP)*bode_amp(H_FPMI)[0]),
4             label='FPMI LP', linewidth = 20.0, alpha=0.25)
5 #plt.axvline (x=f_pole,ymin=1e-11, color='red', linestyle='dotted', linewidth=3)
6 plt.ylim([1e-23, 1e-16])
7 plt.xlim([1e0, 1e5])
8 plt.xlabel('frequency [Hz]')
9 plt.ylabel('H(f) $\mathbf{[1/\sqrt{Hz}]}$')
10 lgd=plt.legend()
11 fig.savefig('../figs/INTRO/fpmi_sensi.pdf', dpi=300, bbox_inches='tight')

```

5.1.4 DRFPMI

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import scipy.signal as sig

```

```

4 import os
5 import sys
6 sys.path.insert(0,'..')
7 plt_style_dir = '.../stash/'
8 fig_exp_dir = '.../../figs/'
9 import ifo_configs as ifco
10 if os.path.isdir(plt_style_dir) == True:
11     plt.style.use(plt_style_dir + 'ppt2latexsubfig.mplstyle')
12 plt.rcParams["font.family"] = "serif"
13 plt.rcParams["font.serif"] = ["Times New Roman"] + plt.rcParams["font.serif"]
14 line_width=7.5

```

Up to this point we can understand how the FPMI repsonse function works:

$$H_{FPMI}(\omega_g) = \frac{2\Delta\phi_r(\omega_g)}{h(\omega_g)} = \frac{t_1^2 r_2}{(t_1^2 + r_1^2)r_2 - r_1} \frac{H_{MI}(\omega_g, L)}{1 - r_1 r_2 e^{-2i\omega_g L/c}} \quad (5.39)$$

```

1 # Some parameters
2 cee = np.float64(299792458)
3 OMEG = np.float64(2*np.pi*cee/(1064.0*1e-9))
4 L = np.float64(4000.0)
5 nu = np.arange(1, 1000000, 1)
6 nat_nu = [np.float64(i*2*np.pi) for i in nu]
7 h_0 = np.float64(1)
8
9 T_1 = .014
10 #T_1 = 25e-6
11 T_2 = 50e-6
12 R_1 = 1-T_1
13 R_2 = 1-T_2
14
15 t_1 = T_1**.5
16 r_1 = R_1**.5
17 r_2 = R_2**.5
18
19 PHI_0 = np.pi/2
20 P_IN = 25

```

POWER RECYCLING

With all the power going to the symmetric port, the nominal operating state of the FPMI involves a significant amount of dumped / wasted power. Placing a mirror at the symmetric port can allow that power to be recycled. Though considerations must be made to maximize the amount of recycling gain you can acquire with your GW detector. This is dependent on the placement of the power recycling mirror (PRM) and its reflectivity, transmission, and loss coefficients.

But first, the field at the symmetric port:

$$E_{\text{SYM}} = \frac{E_i}{2} e^{2ikl} (r_{\text{FP,X}} + r_{\text{FP,Y}}) \quad (5.40)$$

This is realized through observing the circulating power between the PRM and the short Michelson:

$$E_{\text{PRC}} = \frac{t_{\text{PRM}}}{1 - r_{\text{PRM}} r_{\text{FPMI}} e^{2ik(L_{\text{PRC2BS}} + L_{\text{SMICH}})}} E_{\text{in}} \quad (5.41)$$

Where:

$$L_{\text{SMICH}} = l_x + l_y \quad (5.42)$$

Now let's observe the cavity reflection parameter:

$$r_{\text{FP}} = -r_1 + \frac{t_1^2 r_2 e^{i2kL}}{1 - r_1 r_2 e^{i2kL}} = -\frac{\mathcal{F}}{\pi} \left[-\left(\frac{r_1}{r_2}\right)^{1/2} + \left(\frac{r_2}{r_1}\right)^{1/2} (r_1^2 + t_1^2) \right] \quad (5.43)$$

But with loss considerations:

$$r_{\text{FP}} = -r_1 + \frac{t_1^2 r_2 e^{-t_{\text{RT}}/\tau_{\text{loss}}} e^{i2kL}}{1 - r_1 r_2 e^{-t_{\text{RT}}/\tau_{\text{loss}}} e^{i2kL}} \approx -\frac{\mathcal{F}}{\pi} \left[\frac{-r_1 + r_2(r_1^2 + t_1^2)(1 - \mathcal{L}_{\text{RT}})}{\sqrt{r_1 r_2}} \right] \quad (5.44)$$

we know that $t_1^2 \ll r_1^2$:

$$r_{\text{FP}} \approx -\frac{\mathcal{F}}{\pi} \left[\frac{r_1(-1 + (1 - \pi/\mathcal{F})(1 - \mathcal{L}_{\text{RT}}))}{\sqrt{r_1 r_2}} \right] \approx -\left(\frac{r_1}{r_2}\right)^{1/2} \frac{\mathcal{F}}{\pi} \left[-\pi/\mathcal{F} - \mathcal{L}_{\text{RT}} + (\mathcal{L}_{\text{RT}}\pi)/\mathcal{F} \right] \quad (5.45)$$

And $\mathcal{L}_{\text{RT}} \ll 1$ with $r_1/r_2 \approx 1$ we get:

$$r_{\text{FP}} \approx -1 + \frac{\mathcal{F}}{\pi} \mathcal{L}_{\text{RT}} \quad (5.46)$$

If we're operating at a dark fringe, at the symmetric port we see superimposed fields:

$$E_{\text{SYM}} = \frac{E_i}{2} \left[r_{\text{FPX}} e^{2ik_x} + r_{\text{FPY}} e^{2ik_y} \right] \quad (5.47)$$

Where we assume that the short Michelson arms and reflection coefficients are roughly equal ($_x = _y$, $r_{\text{FPX}} = r_{\text{FPY}}$)

We also can average the short Michelson arm lengths $(_x + _y)/2$ such that the effective reflection coefficient is: $r_{\text{FPMI}} = e^{2ik}(-1 + \frac{\mathcal{F}}{\pi} \mathcal{L}_{\text{RT}})$ Knowing this we create the following expression for the circulating power within the cavity:

$$P_{\text{PRC}} = \frac{|t_{\text{PRM}}|^2}{|1 - r_{\text{PRM}} r_{\text{FPMI}} e^{2ik(L_{\text{PRC2BS}} + L_{\text{SMICH}})}|^2} P_{\text{in}} \quad (5.48)$$

where $|t_{\text{PRM}}|^2 = 1 - |r_{\text{PRM}}|^2$ and given a carrier resonance condition we want to maximize the power with a variable PRM reflectivity:

$$\frac{\partial P_{\text{PRC}}}{\partial r_{\text{PRM}}} = \frac{2r_{\text{PRM}}^2(r_{\text{FPMI}} - r_{\text{PRM}})}{(1 - r_{\text{PRM}} r_{\text{FPMI}})^3} = 0 \quad (5.49)$$

which sets $r_{\text{PRM}} = r_{\text{FPMI}}$ On resonance, the power recycling gain ($G_{\text{PR}} = \frac{P_{\text{PRC}}}{P_{\text{in}}}$):

$$G_{\text{PR}} = \frac{\pi}{2\mathcal{F}\mathcal{L}_{\text{RT}}} \left[\frac{1}{1 - \frac{\mathcal{F}\mathcal{L}_{\text{RT}}}{2\pi}} \right] \quad (5.50)$$

```

1 r_FPMI = -r_1 + (T_1*r_2)/(1-r_1*r_2)
2 T_PRM = .03
3 R_PRM = 1-T_PRM
4 t_PRM = (T_PRM)**.5
5 r_PRM = (R_PRM)**.5
6 G_PRC = 1/(1-r_PRM*(r_FPMI))

```

```

1 L_rt = 75e-6
2 Finn = (np.pi*np.sqrt(r_1*r_2))/(1-r_1*r_2)
3 print(Finn)

```

1 444.0741558169753

```
1 r_FPMI_approx = (1 - Finn*L_rt/np.pi)
```

```
1 r_range = np.arange(.9,1,1/(2**16))
```

```
1 G_PRC_ = ifco.PRG(L_rt, Finn, r_range, max=0)
```

```
1 G_PRC_opt = ifco.PRG(L_rt, Finn, r_FPMI, max=1)
```

```

1 plt.plot(r_range, G_PRC_, linewidth=line_width)
2 plt.axhline(G_PRC_opt, linestyle='--', linewidth=line_width, color='r')
3 plt.xlim(r_range[0], r_range[-1])
4 plt.xlabel('$\mathbf{r_{\text{PRM}}}$ [arb]')
5 plt.ylabel('$\mathbf{G_{\text{PRC}}}$ [arb]')

```

1 Text(0, 0.5, '\$\mathbf{G_{\text{PRC}}}\$ [arb]')

```
1 G_PRC_actual = ifco.PRG(L_rt, Finn, r_PRM, max=1)
```

```

1 H_FPMI = ifco.fpmi_freq_resp(nu, r_1, t_1, r_2, L, PHI_0, P_IN, OMEG)
2 H_FPMI_LP = ifco.fpmi_freq_resp(nu, r_1, t_1, r_2, L, PHI_0, P_IN, OMEG,
3                                low_pass='True')

```

```
1 H_PRFPMI = ((G_PRC_actual)**.5)*H_FPMI
```

We estimate the FP's pole frequency

$$1 - r_1 r_2 e^{-2i\omega_g L/c} = 0 \quad (5.51)$$

therefore when:

$$e^{-i\omega_g L/c} = \frac{1}{\sqrt{r_1 r_2}} \quad (5.52)$$

we acquire the pole frequency ω_{pole} as indicated in the low pass

$$f_{\text{pole}} = \frac{1}{4\pi\tau_s} = \frac{c}{4\pi L} \frac{1 - r_1 r_2}{\sqrt{r_1 r_2}} = \frac{\nu_{\text{FSR}}}{2\pi} \frac{1 - r_1 r_2}{\sqrt{r_1 r_2}} = \frac{\nu_{\text{FSR}}}{\mathcal{F}} \quad (5.53)$$

Might as well compare it to our Michelson response:

$$H_{\text{MI}}(\omega_g) = \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c} \quad (5.54)$$

```

1 H_MI = ifco.mich.freq_resp(nu, L, PHI_0, P_IN, OMEG)
2
3 plt.loglog(nu, ifco.bode_amp(H_MI), label= 'MICH', linewidth= line_width, alpha=.3)
4 plt.loglog(nu, ifco.bode_amp(H_FPMI), label= 'FPMI', linewidth= line_width, alpha=.3)
5 plt.loglog(nu, ifco.bode_amp(H_PRFPMI), label= 'PRFPMI', linewidth = line_width)
6 #plt.axvline (x=f_pole,ymin=1e-11, color='red', linestyle='dotted', linewidth=3)
7 plt.xlim([1e0, 1e5])
8 plt.ylim([1e9,2e15])
9 plt.xlabel('frequency [Hz]')
10 plt.ylabel('H(f) [$\mathbf{W/m}$]')
11 lgd=plt.legend()
12 plt.savefig('../figs/INTRO/prfpmpo_fr.pdf', dpi=300, bbox_inches='tight')

1 plt.semilogx(nu,(180/np.pi)*np.arctan(np.imag(H_FPMI)/np.real(H_FPMI)), '--',
2               linewidth=line_width)
3 plt.semilogx(nu,(180/np.pi)*np.arctan(np.imag(H_MI)/np.real(H_MI)), '--',
4               linewidth=line_width)
5 plt.semilogx(nu,(180/np.pi)*np.arctan(np.imag(H_PRFPMI)/np.real(H_PRFPMI)),
6               linestyle='--', linewidth=line_width,dashes=(3,10))
7 plt.xlim([1,100000])
8 plt.ylabel('phase [deg]')
9 plt.xlabel('Frequency [Hz]')

1 Text(0.5, 0, 'Frequency [Hz]')
```

SIGNAL RECYCLING

Initially not used in early iterations of LIGO (initial LIGO and enhanced LIGO) signal recycling imagines using a partially reflective mirror at the anti-symmetric port. And at first glance it seems to not very much make sense to have a mirror at detector output as you would potentially attenuate gravitational wave signals by said mirror reflection coefficient.

While true, it is important to analyze the multi-state configurations offered by such a mirror with various microscopic length tuning configurations. What do I mean by this? Well, it helps to start imagining by analogy of couple cavity relationship as established in the power recycling discussion. The relationship of the differential signal output of the PRFPMI with respect to the newly placed mirror at the anti-symmetric port is represented by the following:

$$t_{\text{SRC}} = \frac{t_{\text{ITM}} t_{\text{SRM}} e^{i(k+\Omega/c)_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{2i(k+\Omega/c)_{\text{SRC}}}} \quad (5.55)$$

$$r_{\text{SRC}} = \frac{r_{\text{ITM}} - r_{\text{SRM}} e^{2i(k+\Omega/c)_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{2i(k+\Omega/c)_{\text{SRC}}}} \quad (5.56)$$

as $k > \Omega_{\text{gw}}/c$ for $1 < \Omega_{\text{gw}} < 5 \cdot 10^3$

Therefore with a pre-defined $T_{\text{ITM}} + R_{\text{ITM}} + L_{\text{ITM}} = 1$ the coupled cavity pole AND gain is a function of the SRM reflectivity and microscopic length tuning:

$$t_{\text{SRC}} = \frac{t_{\text{ITM}} t_{\text{SRM}} e^{ik_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{2ik_{\text{SRC}}}} \quad (5.57)$$

$$r_{\text{SRC}} = \frac{r_{\text{ITM}} - r_{\text{SRM}} e^{2ik_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{2ik_{\text{SRC}}}} \quad (5.58)$$

We now observe the tuning extrema: - On resonance $2ik_{\text{SRC}} = 2i\phi_{\text{SRC}} = 0$:

$$r_{\text{SRC}, \phi_{\text{SRC}}=0} = \frac{r_{\text{ITM}} - r_{\text{SRM}}}{1 - r_{\text{ITM}} r_{\text{SRM}}} \quad (5.59)$$

- On resonance $2ik_{\text{SRC}} = 2i\phi_{\text{SRC}} = \frac{\pi}{2}$:

$$r_{\text{SRC}, \phi_{\text{SRC}}=\pi} = \frac{r_{\text{ITM}} + r_{\text{SRM}}}{1 + r_{\text{ITM}}r_{\text{SRM}}} \quad (5.60)$$

$$\begin{aligned} H_{\text{DRFPMI}} = G_{\text{PR}} P_{\text{in}} L \Omega & \left[\frac{t_{\text{ITM}}^2 r_{\text{ETM}}}{(t_{\text{ITM}}^2 + r_{\text{ITM}}^2)r_{\text{ETM}} - r_{\text{ITM}}} \frac{t_{\text{SRM}} t_{\text{ITM}} e^{i\phi_{\text{SRC}}}}{1 - r_{\text{ITM}} r_{\text{SRM}} e^{i2\phi_{\text{SRC}}}} \times \right. \\ & \frac{e^{-i2\pi Lf/c} \sin(2\pi f/c)}{2\pi L f} \times \\ & \left. \frac{\sin(\phi_0)}{1 - [(r_{\text{ITM}} - r_{\text{SRM}} e^{i2\phi_{\text{SRC}}}) / (1 - r_{\text{ITM}} r_{\text{SRM}} e^{i2\phi_{\text{SRC}}})] r_{\text{ETM}} e^{-i4\pi Lf/c}} \right] \end{aligned}$$

```

1 l_SRC = 56 #[m]
2 T_SRM = .30
3 R_SRM = 1-T_SRM
4 t_SRM = T_SRM**.5
5 r_SRM = R_SRM**.5
6 phi_SRC = np.pi

1 H_DRFPMI = ifco.drfpmi_freq_resp(nu, G_PRC_opt, r_1, t_1, r_2, r_SRM, t_SRM,
2                               phi_SRC, L, PHI_0, P_IN, OMEG)

1 bode_test=False
2 if bode_test:
3     fig, ax1 = plt.subplots()
4     ax1.set_xlabel('frequency [Hz]')
5     ax1.set_ylabel('H$\backslash$mathdefault{FPMI}$ [$\backslash$mathdefault{W/m}] ', color='C0')
6     #ax1.plot(w/(FSR), F_w_cc_modsq*100)
7     ax1.loglog(nu, ifco.bode_amp(H_FPMI), label='FPMI', linewidth=line_width,
8                 linestyle=':', color='C0')
9     ax1.loglog(nu, ifco.bode_amp(H_PRFPMI), label='PRFPMI',
10                linewidth=line_width, color='C0')
11    ax1.loglog(nu, ifco.bode_amp(H_DRFPMI), label='DRFPMI',
12                linewidth=line_width, color='C1')
13    ax1.legend()
14    #ax1.loglog(w,H_MI_modsq, label= 'MICH', linewidth= 5)
15    #ax1.loglog(w,H_FPMI_LP_modsq*H_FPMI_modsq[0], label='FPMI LP',
16                linewidth = 20.0, alpha=0.25,color='C2')

```

```

17 #ax1.axvline (x=f_pole,ymin=1e-13, color='red', linestyle='dotted', linewidth=3)
18 ax2 = ax1.twinx()
19 ax2.semilogx(nu, ifco.bode_ph(H_FPMI), '--', linewidth=7.5, color='C0', alpha=.3)
20 ax2.semilogx(nu, ifco.bode_ph(H_DRFPMI), '--', linewidth=7.5, color='C1', alpha=.3)
21 ax2.grid(b=False, which='both', axis='y')
22 #ax2.semilogx(w,(180/np.pi)*np.arctan(np.imag(H_MI)/np.real(H_MI)), '--')
23 #ax2.semilogx(w,(180/np.pi)*np.arctan(np.imag(H_FPMI_LP)/np.real(H_FPMI_LP)),
24                 linestyle='--', linewidth=20.0,dashes=(4,10),alpha=.25, color='C2')
25 plt.xlim([1,1e5])
26 plt.ylabel('phase [deg]', color='C1', alpha=.5)

```

```

1 plt.loglog(nu, ifco.bode_amp(H_MI), label= 'MICH', linewidth= line_width, alpha=.4)
2 plt.loglog(nu, ifco.bode_amp(H_FPMI), label= 'FPMI', linewidth= line_width, alpha=.4)
3 plt.loglog(nu, ifco.bode_amp(H_PRFPMI), label= 'PRFPMI', linewidth = line_width,
4             alpha=.4)
5 plt.loglog(nu, ifco.bode_amp(H_DRFPMI), label= 'DRFPMI', linewidth = line_width)
6 #plt.axvline (x=f_pole,ymin=1e-11, color='red', linestyle='dotted', linewidth=3)
7 plt.xlim([1e0, 1e5])
8 plt.ylim([1e9, 2e15])
9 plt.xlabel('frequency [Hz]')
10 plt.ylabel('H(f) [$\mathbf{W/m}$]')
11 lgd=plt.legend()
12 plt.savefig('../figs/INTRO/drfpmp_f.pdf', dpi=300, bbox_inches='tight')

```

```

1 plt.semilogx(nu,ifco.bode_ph(H_MI), '--', linewidth=line_width,
2               alpha=.4, label='MICH')
3 plt.semilogx(nu,ifco.bode_ph(H_FPMI), '--', linewidth=line_width,
4               alpha=.4, label='FPMI')
5 plt.semilogx(nu,ifco.bode_ph(H_PRFPMI),linestyle='--',
6               linewidth=line_width,dashes=(3,10), alpha=.4, label='PRFPMI')
7 plt.semilogx(nu,ifco.bode_ph(H_DRFPMI), '--', linewidth=line_width, label='DRFPMI')
8 plt.xlim([1,100000])
9 plt.ylim([-91,91])
10 plt.ylabel('phase [deg]')
11 plt.xlabel('Frequency [Hz]')
12 plt.legend()

```

```
1 Sn = ifco.N_shot(OMEG, P_IN)
```

```

1 plt.loglog(nu, Sn/ifco.bode_amp(H_MI), label= 'MICH', linewidth=line_width)
2 plt.loglog(nu, Sn/ifco.bode_amp(H_FPMI), label='FPMI', linewidth=line_width)
3 plt.loglog(nu, Sn/ifco.bode_amp(H_PRFPMI), label='PRFPMI', linewidth=line_width)
4 plt.loglog(nu, Sn/ifco.bode_amp(H_DRFPMI), label='DRFPMI', linewidth=line_width)
5 #plt.axvline (x=f_pole,ymin=1e-11, color='red', linestyle='dotted', linewidth=3)
6 plt.ylim([1e-24,2e-19])
7 plt.xlim([1e0, 1e5])
8 plt.xlabel('frequency [Hz]')
9 plt.ylabel('$(\mathbf{1} / \sqrt{\mathbf{Hz}})$')
10 lgd=plt.legend()
11 plt.savefig('../figs/INTRO/strain_compare.pdf', dpi=300, bbox_inches='tight')

```

5.2 Paraxial equation

The general three dimensional wave equation for an E-field $E(x, y, z, t)$ is provided by Maxwell:

$$\left(\nabla^2 + \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) E(x, y, z, t) = 0$$

For a coherent beam ($k = \frac{2\pi}{\lambda}$), we analyze the purely spatial component of the solution and select a longitudinal propagation (\vec{z}) direction such that our solution will look like the following (utilizing Helmholtz's equation):

$$E(x, y, z) = E_0(x, y, z)e^{-ikz}$$

The above wavefunction combined with the Helmholtz equation requires the complex form of E_0 and obeys the paraxial equation [53]:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - 2ik \frac{\partial}{\partial z} \right) E_0(x, y, z) = 0 \quad (5.61)$$

5.3 Cavity stability criteria ($G(g_1, g_2)$)

Using spherical mirror resonators to match the phasefront of the beam mode is standard practice that has some additional geometric considerations to maximize resonance for a given beam. Choosing two mirrors with ROCs (R_1, R_2) and a set distance between them d , an implicit containment condition is set on the resonator [5]:

$$0 \leq \left(1 + \frac{d}{R_1}\right) \left(1 + \frac{d}{R_2}\right) \leq 1$$

Where we define $g_i = 1 + \frac{d}{R_i}$ so that we define a single parameter for the two mirror resonator G :

$$0 \leq G(g_1, g_2) \leq 1 \quad (5.62)$$

5.4 The Equipartition theorem and the Fluctuation dissipation theorem

The Fluctuation Dissipation Theorem connects fluctuations on a microscopic level to fluctuations of macroscopic observables, and allows one to bypass having to venture into overly involved microscopic processes; a profound finding for experiments that are or will become thermal noise limited. Though after revisiting the equipartition theorem, there might still be some confusion how the two statements might contradict each other. We quickly revisit the 1D harmonic oscillator to provide some clarity:

$$1/2(k\bar{x}^2) = 1/2(k_B T) \quad (5.63)$$

\bar{x}^2 indicates an average position which the theorem indicates when root square mean motion is assumed. This is to say that FDT by no means is a modification of our understanding

of the equipartition theorem but rather enriches providing insight on the microscopic fluctuating phenomenon when measuring the power spectral density of the fluctuations [4]:

$$x^2 = \frac{k_B T}{\pi^2 f^2} \Re(Y) \quad (5.64)$$

5.5 RH control pre-filter

5.5.1 recipe

The following is a brief recipe to build a filter that can better optimize the RH thermo-optic response:

1. Fit step response to a zpk filter $H(s)$ (see Figure 2.6)
2. Invert fitted filter ($H(s) \rightarrow H^{-1}(s)$)

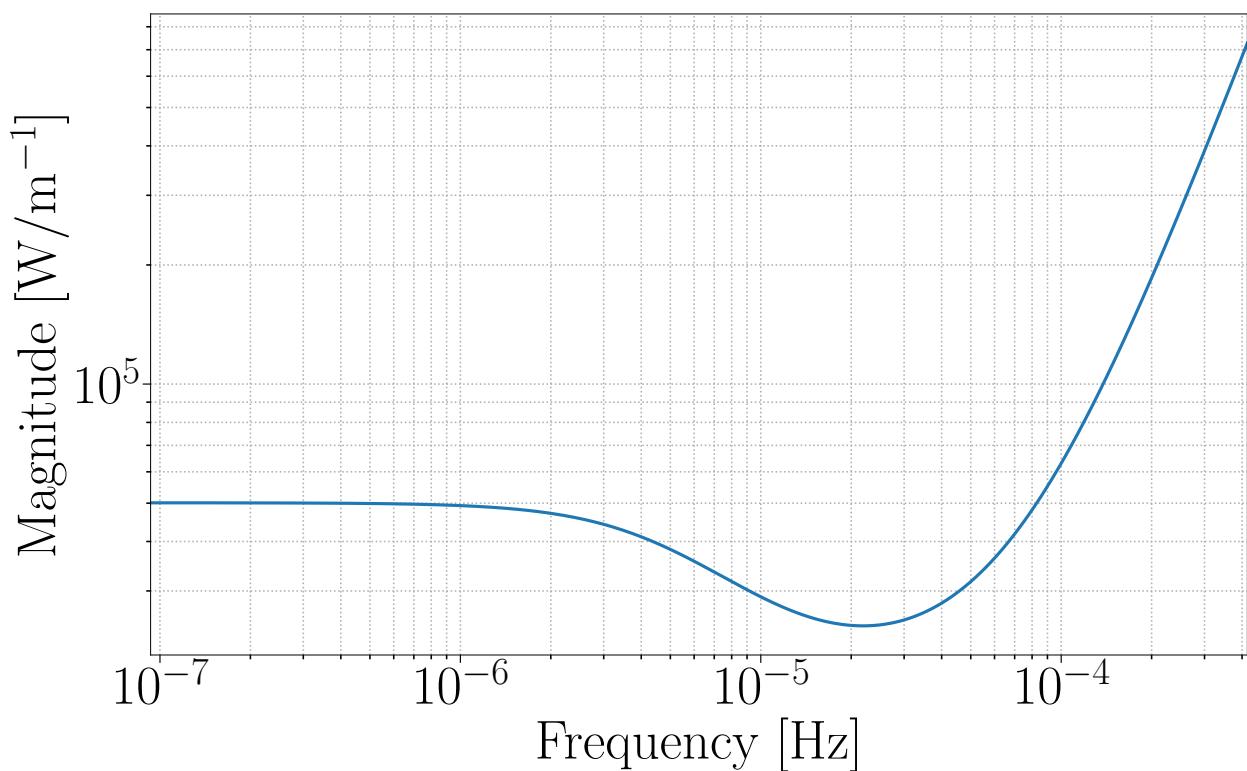


Figure 5.1: Fitted zpk filter, inverted.

3. Apply correction filter $G(s)$ for stability and speed tuning ($H^{-1}(s) * G(s)$)

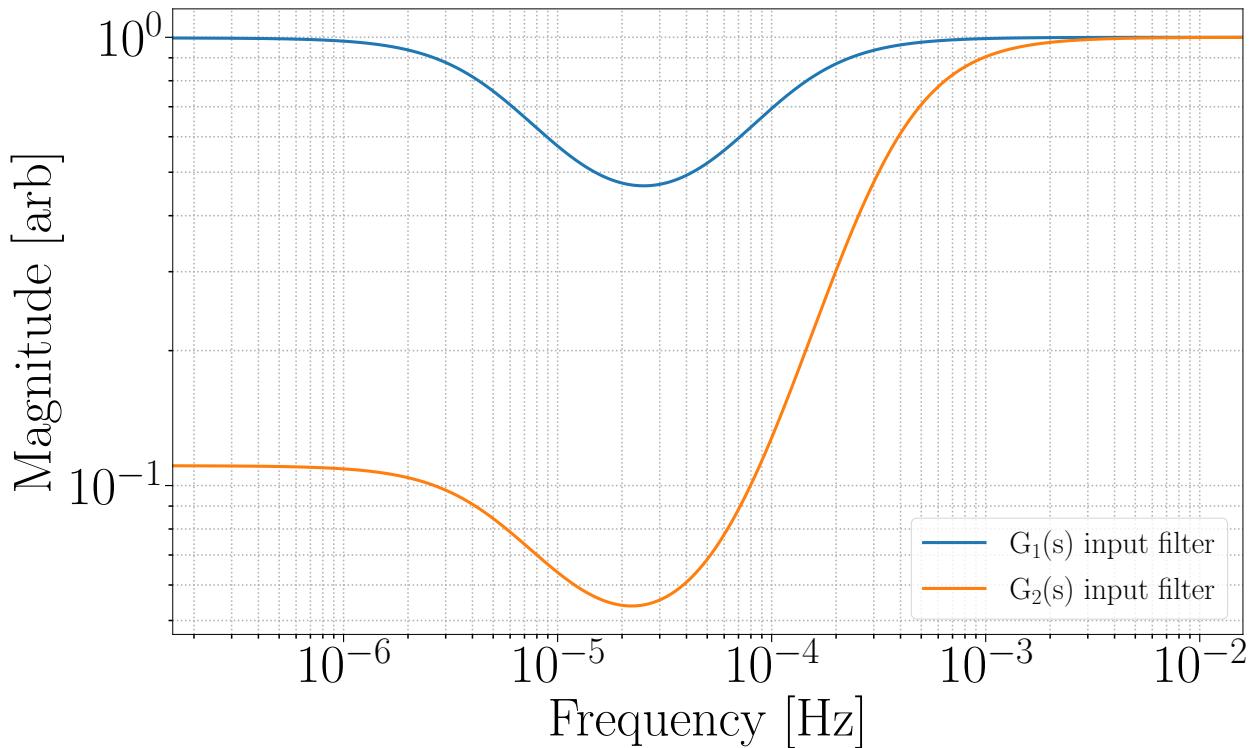


Figure 5.2: Fitted zpk filter to transient response of self heating COMSOL model.

5.5.2 code

```

1 import matplotlib
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from scipy import signal
5 import h5py
6 import os
7
8 plt_style_dir = '../my-python/matplotlib/stylelib/'
9 if os.path.isdir(plt_style_dir) == True:
10     plt.style.use(plt_style_dir + 'ppt2latex')
11 plt.rcParams["font.family"] = "Times New Roman"

```

```

1 # Establish default color array
2 prop_cycle = plt.rcParams['axes.prop_cycle']
3 colors = prop_cycle.by_key()['color']

```

```

4 lin_thickness=4
1 ## Set figure saving directory
2 thesis_dir = '../doc/figures/python/'
3 thesis_dir='../dissertation/figs/TCS/IRHF/'
```

Generating / plotting plant filter

```

1 ITMYRH_data = np.loadtxt('../data/ITMY_trend_10min_int_longer.dat')
2 t = np.arange(0,len(ITMYRH_data[:,0][2:]))*60.0*10.0
3 normalize = 3.13
4 print(len(t))
5 data_in = ITMYRH_data[:,1][2:]
6 b, a = signal.butter(2, .2)
7 #data_new = signal.filtfilt(b,a,data_in)
8 data_new = data_in
9 plt.figure()
10 ir = (data_new[1:] - data_new[:-1])/normalize
11 ir_new = ir
12 fig1 = plt.figure(figsize=(13,10))
13 plt.plot(t, data_new, label='Step response', linewidth=lin_thickness)
14 #plt.plot(t[:len(t)-1], ir, label= 'Impulse response')
15 plt.xlabel('time [s]')
16 plt.ylabel('Defocus [m$^{-1}$]')
17 #plt.legend(fontsize='medium')
18 plt.show()
19
20 Fs = 1/(t[2]-t[1])
21 #print(Fs)
22
23 [F,H]=signal.freqz(ir_new,1, worN=3000, whole=False)
24 fig2 = plt.figure(figsize=(13,10))
25 plt.loglog(F*Fs/(2*np.pi), abs(H), label='Plant filter', linewidth=lin_thickness)
26 plt.ylabel('Magnitude [m$^{-1}$/W]')
27 plt.xlabel('Frequency [Hz]')
28 plt.legend()
29 plt.show()
30
31 print(max(ir_new))
```

```

1 adj_data = data_new + abs(min(data_new))
2 mod_data = np.concatenate([np.zeros((10,)), adj_data])
3 mod_t = np.arange(0,len(mod_data))*60.0*10.0/(3600)
4 mod_rh_inp = np.concatenate([np.ones((10,))*3.13, np.zeros(adj_data.shape)])

```

```

1 fig, ax1 = plt.subplots()
2
3 ax1.set_xlabel('time [hr]')
4 ax1.set_ylabel('Primary-axis')
5 ax1.plot(mod_t, mod_rh_inp,'--', linewidth=lin_thickness, color = colors[0])
6 ax1.tick_params(axis='y', labelcolor=colors[0])
7 ax1.set_ylabel('RH power [W]', color=colors[0])
8 #ax1.grid(b=False, which='minor', linestyle='--')
9 #ax1.grid(b=False, which='major', linestyle='--')
10 ax1.minorticks_off()
11 ax1.set_xlim([0,mod_t[-1]])
12 ax1.set_ylim([-0.01,4])
13
14 ax2 = ax1.twinx()
15 ax2.plot(mod_t, mod_data, linewidth=lin_thickness, color = colors[1])
16 ax2.set_ylabel('Defocus [m$^{-1}$]',color= colors[1])
17 ##plt.grid(b=True, which='minor', linestyle='--')
18 ##plt.grid(b=True, which='major', linestyle='--')
19 ##plt.minorticks_on()
20 ax2.set_xlim([0,mod_t[-1]])
21 ax2.tick_params(axis='y', labelcolor=colors[1])
22 ax2.ticklabel_format(style='sci', axis='y', scilimits=(0,-5))
23
24 ax2.set_ylimits([-0.003e-4,1.2e-4])
25
26 fig.savefig(thesis_dir + 'Meas_response.pdf', dpi=300, format='pdf', bbox_inches='tight')

```

```
1 print('Only plots up to the nyquist frequency: {} Hz'.format(F[-1]*Fs/(2*np.pi)))
```

1 Only plots up to the nyquist frequency: 0.0008330555555555556 Hz

```

1 zeros = 5.0e-6
2 fit_zeros = -2.0*np.pi*zeros
3 poles = np.array([1.3e-5, 5.0e-5 ,9.5e-5])

```

```

4 fit_poles = -2.0*np.pi*poles
5
6 k = 1 #This gain is not initially correct
7
8 s1 = signal.ZerosPolesGain(fit_zeros, fit_poles, k)
9 F_2, H_2 = signal.freqresp(s1, F*(Fs/2.0))
10
11 #[F_2,H_2] = signal.freqs(b_2, a_2)
12 k_new = abs(H[0])/abs(H_2[0])
13
14 plt.loglog(F_2/(2*np.pi), abs(H_2)*k_new, label='Fitted zpk filter', linewidth=lin_thickness)
15 plt.loglog(F/(2*np.pi)*Fs, abs(H), label='Measured (step response) filter',
16             linewidth=lin_thickness)
17 plt.ylabel('Magnitude [W/m$^{-1}$]')
18 plt.xlabel('Frequency [Hz]')
19 plt.legend()
20 #plt.title('RH plant filter (H(s))')
21 plt.xlim([0,(F[-1]/(2*np.pi)*Fs)])
22 print(k_new) #Split out the new gain
23 ##plt.grid(b=True,which='minor')
24 ##plt.grid(b=True,which='major')
25 ##plt.minorticks_on()
26
27 model_zpk = signal.ZerosPolesGain(fit_zeros, fit_poles,k_new)
28
29 plt.savefig(thesis_dir+'RH_plant_filter_fit.pdf',bbox_inches = 'tight')

```

1 model_zpk

```

1 ZerosPolesGainContinuous(
2 array([-3.14159265e-05]),
3 array([-8.16814090e-05, -3.14159265e-04, -5.96902604e-04]),
4 9.729529652779821e-12,
5 dt: None
6 )

```

Now to invert the plant filter (just swapping the poles and the zeros and inverting gain)
 $(H^{-1}(s))$

```

1 inv_model = signal.ZerosPolesGain(fit_poles, fit_zeros, 1/k_new)
2 F_3, H_3 = signal.freqresp(inv_model, F*(Fs/2.0))
3 fig4 = plt.figure()
4 plt.loglog(F_3/(2*np.pi), abs(H_3), label='Fitted zpk Filter', linewidth=lin_thickness)
5 plt.ylabel('Magnitude [W/m$^{-1}$]')
6 plt.xlabel('Frequency [Hz]')
7 #plt.title('RH inverse filter ([H(s)]$^{-1}$)')
8 plt.xlim([0, F_3[-1]/(2*np.pi)])
9 ##plt.grid(b=True, which='minor', linestyle='--')
10 ##plt.grid(b=True, which='major', linestyle='--')
11 #plt.minorticks_on()
12 plt.savefig(thesis_dir+'RH_inv_filt.pdf', bbox_inches = 'tight')

```

Stabilize the high frequencies to DC (Generating $H^{-1}(s) * G_n(s)$)

Will also attempt to reduce the time constant

```

1 #pole_test = .0001113 + 1e-4
2 Hinv_G_1_filt = signal.ZerosPolesGain(fit_poles, [fit_zeros, -2.0*np.pi*.0001113129672, -2.0*np.pi*.
3     0001113129672], 1)
4 pole_shift = 3
5 Hinv_G_2_filt = signal.ZerosPolesGain(fit_poles, [fit_zeros, -2.0*np.pi*.0001113129672*pole_shift,
6     -2.0*np.pi*.0001113129672*pole_shift], 1)
7
8 ## Plotting
9 freq = np.arange(10e-7, 10e-2, 1e-7)
10 F_4, H_4 = signal.freqresp(Hinv_G_1_filt, freq)
11 F_5, H_5 = signal.freqresp(Hinv_G_2_filt, freq)
12
13 fig5= plt.figure()
14 plt.loglog(F_4/(2*np.pi), abs(H_4), label='RH input filter', linewidth=lin_thickness)
15 #plt.loglog(F_5/(2*np.pi), abs(H_5), label='Livingston filter', linewidth=lin_thickness)
16 #plt.legend(fontsize='xx-large')
17 plt.xlim([F_4[0]/(2*np.pi), F_4[-1]/(2*np.pi)])
18 #plt.grid(b=True, which='minor', linestyle='--')
19 #plt.grid(b=True, which='major', linestyle='--')
20 ##plt.minorticks_on()
21 plt.ylabel('Magnitude [arb]')

```

```

22 plt.xlabel('Frequency [Hz]')
23 #plt.title('Real-time RH filter [H(s)]$^{-1}$', **title_font)
24
25 plt.savefig(thesis_dir+'RH_input_filt.pdf',bbox_inches='tight')

```

1 Hinv_G_1filt

```

1 ZerosPolesGainContinuous(
2 array([-8.16814090e-05, -3.14159265e-04, -5.96902604e-04]),
3 array([-3.14159265e-05, -6.99400000e-04, -6.99400000e-04]),
4 1,
5 dt: None
6 )

```

1 Hinv_G_2_filt

```

1 ZerosPolesGainContinuous(
2 array([-8.16814090e-05, -3.14159265e-04, -5.96902604e-04]),
3 array([-3.14159265e-05, -2.09820000e-03, -2.09820000e-03]),
4 1,
5 dt: None
6 )

```

```

1 fig79= plt.figure()
2 plt.loglog(F_4/(2*np.pi), abs(H_4), label='G$_1$(s) input filter', linewidth=lin_thickness)
3 plt.loglog(F_5/(2*np.pi), abs(H_5), label='G$_2$(s) input filter', linewidth=lin_thickness)
4 plt.legend()
5 plt.xlim([F_4[0]/(2*np.pi),F_4[-1]/(2*np.pi)])
6 #plt.grid(b=True,which='minor',linestyle='--')
7 #plt.grid(b=True,which='major',linestyle='--')
8 #plt.minorticks_on()
9 plt.ylabel('Magnitude [arb]')
10 plt.xlabel('Frequency [Hz]')
11 #plt.title('Real-time RH filter [H(s)]$^{-1}$', **title_font)
12
13 plt.savefig(thesis_dir+'RH_input_filt_G1_G2.pdf',bbox_inches='tight')

```

COMSOL self heating filter

Import COMSOL self heating data

```

1 COM_data = np.loadtxt('../data/1W_self_heating_defocus_doublepass.txt')
2 t_com = COM_data[:,0]*3600
3 defocus = COM_data[:,1]/max(COM_data[:,1])

```

```

1 fig6 = plt.figure()
2 plt.plot(t_com/3600,defocus,linewidth=lin_thickness)
3 plt.title('COMSOL self heating time series')
4 plt.xlabel('time [hrs]')
5 plt.ylabel('defocus [arb]')
6 max(defocus)

```

```

1 ir_com = (defocus[1:] - defocus[:-1])
2 t_ir = t_com[:((len(t_com)-1))]

```

```

1 [F_ir,H_ir]=signal.freqz(ir_com, 1, worN=3000,whole=False)
2 Fs_com = 1/(t_com[1]-t_com[0])

```

```

1 zeros_com = np.array([.9e-3,.3e-3])
2 fit_zeros_com = -2.0*np.pi*zeros_com
3 poles_com = np.array([.25e-3,.25e-3,1.6e-3])
4 fit_poles_com = -2.0*np.pi*poles_com
5
6 k_com = 1 #This gain is not initally correct
7
8 zpk_com = signal.ZerosPolesGain(fit_zeros_com, fit_poles_com, k_com)
9 F_com, H_com = signal.freqresp(zpk_com, F_ir*(Fs_com/2.0))
10 k_new_com = abs(H_ir[0])/abs(H_ir[0]*H_com[0])
11
12 fig6 = plt.figure()
13 plt.loglog(F_com/(2*np.pi), abs(H_com)*k_new_com, label='Fitted zpk Filter',
14 linewidth=lin_thickness)
15 plt.loglog(F_ir*Fs_com/(2*np.pi), abs(H_ir)/abs(H_ir[0]), label='Plant filter',
16 linewidth=lin_thickness)
17 plt.ylabel('Magnitude [arb]')
18 plt.xlabel('Frequency [Hz]')
19 plt.title('Self Heating filter')

```

```

1 G_2 = signal.ZerosPolesGain(fit_zeros_com, fit_poles_com, k_new_com)
2 unit_step_testing = np.zeros(np.shape(t_com))

```

```

3 unit_step_testing[t_com>0] = 1
4 [ ,y_self_test, ] = signal.lsim(G_2, unit_step_testing, t_com)

```

```

1 fig7= plt.figure()
2 plt.plot(t_com/3600,defocus,label='measured',linewidth=lin_thickness)
3 plt.plot(t_com/3600,y_self_test,label='fit',linewidth=lin_thickness)
4 plt.title('Self heating time series (fit vs measured)')
5 plt.legend()

```

Generating time series

Step input time series

```

1 unit_step = np.zeros((t.shape[0]*30))
2 t_new = np.arange(0,len(unit_step))*60.0*1.0
3 ## Generating simulated response
4 unit_step[t_new>9000] = 1
5 [t_mod_new,y_mod_sim,xout] = signal.lsim(model_zpk, unit_step, t_new)

```

Conditioned input time series

```

1 unit_step2 = np.zeros((t.shape[0]*30))
2 unit_step2[t_new>(9000)] = pole_shift**2
3
4 [ ,y_inp_inv_L, ] = signal.lsim(Hinv_G_2_filt, unit_step2, t_new)
5 [ ,y_inp_inv_H, ] = signal.lsim(Hinv_G_1_filt, unit_step, t_new)
6 [ ,y_mod_sim_inv_L, ] = signal.lsim(model_zpk, y_inp_inv_L, t_new)
7 [ ,y_mod_sim_inv_H, ] = signal.lsim(model_zpk, y_inp_inv_H, t_new)

```

Self heating time series

```

1 unit_step3 = np.zeros((t.shape[0]*30))
2 t_offset = 0
3 unit_step3[t_new>(9000+t_offset)] = 1
4 [ ,y_sh_resp, ] = signal.lsim(G_2, unit_step3, t_new)

```

Basic Performance

```

1 fig = plt.figure()

```

```

2 plt.subplot(211)
3 plt.plot(t_new/3600, unit_step, linewidth = lin_thickness,label='RH step input')
4 plt.plot(t_new/3600, y_inp_inv_H,'--', linewidth = lin_thickness,
5          color = 'red', label='RH filtered input')
6 plt.ylabel('RH power [W]')
7 #plt.title('RH step input vs. Filtered input')
8 plt.legend()
9 plt.xlim([0, t_new[-1]/3600])
10 #plt.grid(b=True,which='minor',linestyle='--')
11 #plt.grid(b=True,which='major',linestyle='--')
12 #plt.minorticks_on()
13 plt.subplot(212)
14 plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,label = 'RH step input')
15 plt.plot(t_new/3600,-y_mod_sim_inv_H,'--', linewidth = lin_thickness,
16          color='red',label ='RH filtered input')
17 plt.ylabel('Defocus [m$^{-1}]')
18 plt.xlabel('time [hr]')
19 #plt.legend()
20 plt.xlim([0, t_new[-1]/3600])
21 #plt.grid(b=True,which='minor',linestyle='--')
22 #plt.grid(b=True,which='major',linestyle='--')
23 #plt.minorticks_on()
24 plt.ticklabel_format(style='sci', axis='y',scilimits=(0,-5))
25 fig.savefig(thesis_dir+'IRHF_step_vs_filt_step.pdf',bbox_inches='tight')

```

All curves together

```

1 fig = plt.figure()
2 plt.subplot(211)
3 plt.plot(t_new/3600, unit_step,linewidth = lin_thickness,label='RH unfiltered step input')
4 plt.plot(t_new/3600, y_inp_inv_L,'--', linewidth = lin_thickness, color = 'green',
5          label='RH conditioned input (G$\{1\}$(s))')
6 plt.plot(t_new/3600, y_inp_inv_H,'--', linewidth = lin_thickness,color = 'red',
7          label='RH conditioned input (G$\{2\}$(s))')
8 plt.ylabel('RH power [W]')
9 plt.title('RH filtered response w/ self-heating')
10 plt.legend(fontsize='medium')
11 plt.xlim([0,20])

```

```

12 plt.subplot(212)
13 plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,label = 'RH unfiltered step input')
14 plt.plot(t_new/3600,y_sh_resp*20e-6, linewidth = lin_thickness, color='orange',
15         label ='self heating')
16 plt.plot(t_new/3600,-y_mod_sim_inv_L,'--', linewidth = lin_thickness,color='green',
17         label ='RH conditioned input (G$\_{1}\$(s))')
18 plt.plot(t_new/3600,-y_mod_sim_inv_H,'--', linewidth = lin_thickness,color='red',
19         label ='RH conditioned input (G$\_{2}\$(s))')
20 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_L,linewidth = lin_thickness,
21         label='self heating + RH conditioned input (G$\_{1}\$(s)',color='purple')
22 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_H,linewidth = lin_thickness,
23         label='self heating + RH conditioned input (G$\_{2}\$(s))',color='magenta')
24 plt.ylabel('Defocus [m$^{-1}$]')
25 plt.xlabel('time [hr]')
26 plt.legend(fontsize='medium')
27 plt.xlim([0,20])
28 fig.savefig(thesis_dir+'IRHF_compare_self_w_filter_compare.pdf',bbox_inches='tight')

```

```

1 fig8 = plt.figure()
2 plt.rc('font', size=25)
3 plt.plot(t_new/3600,y_sh_resp*20e-6, linewidth = lin_thickness,color='orange',
4         label ='self heating with no RH')
5 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim,linewidth = lin_thickness,
6         label='self heating + RH unfiltered input',color='purple')
7 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_H,'--', linewidth = lin_thickness,
8         label='self heating + RH filtered input (H$\^{-1}\$(s)*G$\_{1}\$(s))',color='red')
9 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_L,'--', linewidth = lin_thickness,
10         label='self heating + RH filtered input (H$\^{-1}\$(s)*G$\_{2}\$(s))',color='green')
11 plt.ylabel('Defocus [m$^{-1}$]')
12 plt.xlabel('time [hr]')
13 plt.ticklabel_format(style='sci', axis='y',scilimits=(0,-5))
14 plt.xlim([0,20])
15 plt.legend(loc='upper right',bbox_to_anchor=(1.0,.95))
16 fig8.savefig(thesis_dir+'IRHF_compare_w_self.pdf',bbox_inches='tight')

```

Set RH upper limit

```

1 upper_lim = np.ones(np.shape(t_new))*40

```

```

1 fig9= plt.figure(figsize=(25,20))
2 plt.rc('font', size=30)
3 plt.subplot(211)
4 plt.plot(t_new/3600, unit_step,linewidth = lin_thickness,
5           label='Step input', color= 'purple')
6 plt.plot(t_new/3600, y_inp_inv_L,'--', linewidth = lin_thickness, color = 'green',
7           label='Filtered input')
8 #plt.plot(t_new/3600, upper_lim,':',linewidth = lin_thickness, color='magenta',
9           label='RH upper limit')
10 #plt.plot(t_new/3600, y_inp_inv_H,'--', linewidth = lin_thickness,color = 'red',
11           label='Filtered input ( $H^{-1}(s)G(s)$ )')
12 #plt.minorticks_on()
13 #plt.grid(b=True,which='minor',linestyle='--')
14 #plt.grid(b=True,which='major',linestyle='--')
15 plt.ylabel('RH power [W]')
16 plt.xlim([0,20])
17 #plt.title('RH responses')
18 plt.legend(fontsize='large')
19 plt.subplot(212)
20 plt.ylabel('Defocus [m $^{-1}$ '])
21 plt.plot(t_new/3600,y_sh_resp*20e-6, linewidth = lin_thickness,color='orange',
22           label ='central heating with no RH')
23 plt.plot(t_new/3600,(y_sh_resp*20e-6 -y_mod_sim),linewidth = lin_thickness,
24           label ='central heating + RH w/ step input',color='purple')
25 plt.plot(t_new/3600,(y_sh_resp*20e-6 -y_mod_sim_inv_L),'--', linewidth = lin_thickness,
26           label ='central heating + RH w/ filtered input',color='green')
27 #plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,
28           # label = 'Unfiltered step input',color='purple')
29 #plt.plot(t_new/3600,y_sh_resp*20e-6-y_mod_sim_inv_L,'--', linewidth = lin_thickness,
30           # color='green',label ='Filtered input ( $H^{-1}(s)G(s)$ )')
31 #plt.plot(t_new/3600,-y_mod_sim_inv_H,'--', linewidth = lin_thickness,color='red',
32           # label ='Filtered input ( $H^{-1}(s)G(s)$ )')
33 #plt.minorticks_on()
34 #plt.grid(b=True,which='minor',linestyle='--')
35 #plt.grid(b=True,which='major',linestyle='--')
36 plt.xlabel('time [hr]')
37 plt.ticklabel_format(style='sci', axis='y',scilimits=(0,-5))

```

```

38 plt.legend(loc='upper right', bbox_to_anchor=(1.0,.95), fontsize='large')
39 plt.xlim([0,20])
40
41 fig9.savefig(thesis_dir+'IRHF_compare_filts_PI_paper.pdf',bbox_inches='tight')

```

```

1 fig9= plt.figure(figsize=(25,20))
2 plt.rc('font', size=30)
3 plt.subplot(211)
4 plt.plot(t_new/3600, unit_step,linewidth = lin_thickness,
5           label='Step input', color= 'purple')
6 #plt.plot(t_new/3600, upper_lim,':', linewidth = lin_thickness, color='magenta',
7 # label='RH upper limit')
8 plt.plot(t_new/3600, y_inp_inv_L,'--', linewidth = lin_thickness, color = 'green',
9 # label='Filtered input( $H^{-1}(s)G^2(s)$ )')
10 plt.plot(t_new/3600, y_inp_inv_H,'--', linewidth = lin_thickness,color = 'red',
11 # label='Filtered input ( $H^{-1}(s)G^1(s)$ )')
12 #plt.minorticks_on()
13 #plt.grid(b=True,which='minor',linestyle='--')
14 #plt.grid(b=True,which='major',linestyle='--')
15 plt.ylabel('RH power [W]')
16 plt.xlim([0,20])
17 #plt.title('RH responses')
18 plt.legend(fontsize='large')
19 plt.subplot(212)
20 plt.ylabel('Defocus [m $^{-1}$ ])
21 plt.plot(t_new/3600,y_sh_resp*20e-6, linewidth = lin_thickness,color='orange',
22           label ='self heating with no RH')
23 plt.plot(t_new/3600,(y_sh_resp*20e-6 -y_mod_sim),linewidth = lin_thickness,
24           label='self heating + RH w/ step input',color='purple')
25 #plt.plot(t_new/3600,(y_sh_resp*20e-6 -y_mod_sim_inv_L),'--', linewidth = lin_thickness,
26 # label='self heating + RH w/ filtered input',color='green')
27 #plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,
28 # label = 'Unfiltered step input',color='purple')
29 plt.plot(t_new/3600,(y_sh_resp*20e-6-y_mod_sim_inv_L),'--', linewidth = lin_thickness,
30           color='green',label ='Filtered input ( $H^{-1}(s)G^2(s)$ )')
31 plt.plot(t_new/3600,(y_sh_resp*20e-6-y_mod_sim_inv_H),'--', linewidth = lin_thickness,
32           color='red',label ='Filtered input ( $H^{-1}(s)G^1(s)$ )')
33 #plt.minorticks_on()

```

```

34 #plt.grid(b=True,which='minor',linestyle='--')
35 #plt.grid(b=True,which='major',linestyle='--')
36 plt.xlabel('time [hr]')
37 plt.ticklabel_format(style='sci', axis='y',scilimits=(0,-5))
38 plt.legend(loc='upper right', bbox_to_anchor=(1.0,.97),fontsize='large')
39 plt.xlim([0,20])
40
41 fig9.savefig(thesis_dir+'IRHF_compare_filt.pdf',bbox_inches='tight')
42 fig9.savefig(thesis_dir+'IRHF_compare_filt.pdf',bbox_inches='tight')

```

```

1 fig9= plt.figure(figsize=(17,15))
2 plt.rc('font', size=25)
3 plt.subplot(211)
4 plt.plot(t_new/3600, unit_step,linewidth = lin_thickness, color= 'purple',
5         label='Step input')
6 plt.plot(t_new/3600, y_inp_inv_L,'--', linewidth = lin_thickness, color = 'green',
7         label='Filtered input ( $H^{-1}(s)G(s)$ )')
8 #plt.plot(t_new/3600, upper_lim,':',linewidth = lin_thickness, color='magenta',
9 #         label='RH upper limit')
10 plt.plot(t_new/3600, y_inp_inv_H,'--', linewidth = lin_thickness,color = 'red',
11 #         label='Filtered input ( $H^{-1}(s)G(s)$ )')
12 #plt.minorticks_on()
13 #plt.grid(b=True,which='minor',linestyle='--')
14 #plt.grid(b=True,which='major',linestyle='--')
15 plt.ylabel('RH power [W]')
16 plt.xlim([0,t_new[-1]/3600])
17 #plt.title('RH responses')
18 plt.legend(fontsize='medium')
19 plt.subplot(212)
20 plt.ylabel('Defocus [ $m^{-1}$ ])'
21 plt.plot(t_new/3600,y_sh_resp*20e-6, linewidth = lin_thickness,color='orange',
22         label ='self heating w/ no RH')
23 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim,linewidth = lin_thickness,
24         label='self heating + step input',color='purple')
25 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_L,'--', linewidth = lin_thickness,
26         label='self heating + filtered input ( $H^{-1}(s)G(s)$ )',color='green')
27 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_H,'--', linewidth = lin_thickness,
28         label='self heating + filtered input ( $H^{-1}(s)G(s)$ )',color='red')

```

```

29 #plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,
30 # label = 'Unfiltered step input',color='purple')
31 #plt.plot(t_new/3600,-y_mod_sim_inv_L,'--', linewidth = lin_thickness,color='green',
32 # label ='Filtered input ( $H^{-1}(s)G(s)$ )')
33 #plt.plot(t_new/3600,-y_mod_sim_inv_H,'--', linewidth = lin_thickness,color='red',
34 # label ='Filtered input ( $H^{-1}(s)G(s)$ )')
35 #plt.minorticks_on()
36 #plt.grid(b=True,which='minor',linestyle='--')
37 #plt.grid(b=True,which='major',linestyle='--')
38 plt.xlabel('time [hr]')
39 plt.xlim([0,t_new[-1]/3600])
40 plt.ticklabel_format(style='sci', axis='y',scilimits=(0,-5))
41 plt.legend(loc='upper right', bbox_to_anchor=(1.0,.97),fontsize='medium')
42
43 fig9.savefig(thesis_dir+'IRHF_compare_filts.pdf',bbox_inches='tight')

```

```

1 fig = plt.figure(figsize=(17,15))
2 plt.subplot(311)
3 plt.plot(t_new/3600, unit_step,linewidth = lin_thickness,
4         label='Unfiltered step input')
5 plt.plot(t_new/3600, y_inp_inv_L,'--', linewidth = lin_thickness, color = 'green',
6         label='Conditioned input ( $G(s)$ )')
7 plt.plot(t_new/3600, y_inp_inv_H,'--', linewidth = lin_thickness,color = 'red',
8         label='Conditioned input ( $G(s)$ )')
9 plt.ylabel('RH power [W]')
10 #plt.title('RH responses')
11 plt.legend(fontsize='small')
12 plt.subplot(312)
13 plt.ylabel('RH Defocus [ $m^{-1}$ '])
14 plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,
15         label = 'Unfiltered step input')
16 plt.plot(t_new/3600,-y_mod_sim_inv_L,'--', linewidth = lin_thickness, color='green',
17         label ='Conditioned input ( $G(s)$ )')
18 plt.plot(t_new/3600,-y_mod_sim_inv_H,'--', linewidth = lin_thickness, color='red',
19         label ='Conditioned input ( $G(s)$ )')
20 plt.legend(fontsize='x-small',loc='upper right')
21 plt.subplot(313)
22 plt.plot(t_new/3600,y_sh_resp*20e-6, linewidth = lin_thickness,color='orange',

```

```

23     label = 'Self heating')
24 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim,linewidth = lin_thickness,
25     label='Self heating + RH unfiltered input',color='C0')
26 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_H,'--',linewidth = lin_thickness,
27     label='Self heating + RH conditioned input (G$\{1\}$(s))',color='red')
28 plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv_L,'--',linewidth = lin_thickness,
29     label='Self heating + RH conditioned input (G$\{2\}$(s))',color='green')
30 plt.ylabel('Total Defocus [m$^{-1}$]')
31 plt.xlabel('time [hr]')
32 plt.legend(fontsize='xx-small')
33 fig.savefig(thesis_dir+'IRHF_compare_all.pdf')

```

$G_1(s) \rightarrow$ The “response function”

For the above scenario we have the following G_s (a double pole low pass at 1.113e-4)

```
G_1 = signal.ZerosPolesGain([], [-2.0np.pi.0001113129672, -2.0np.pi.0001113129672],1)
```

```

1 fig2 = plt.figure(figsize=(15,8))
2 plt.loglog(F_5/(2*np.pi), abs(H_5)*k_upd, label='G$\{1\}$(s)')
3 plt.loglog(F_com/(2*np.pi), abs(H_com)*k_new_com, label='G$\{2\}$(s)')
4 plt.ylabel('Magnitude [arb]')
5 plt.xlabel('Frequency [Hz]')
6 plt.title('G$\{1\}$ vs. G$\{2\}$')
7 plt.legend()

```

The Livingston filter is what we will construct here. To do that, we will first attempt multiplying $G_2(s)$ (the self heating response) to $H^{-1}(s)$

```

1 FILT_LIV_zeros= np.append(fit_zeros_com,fit_poles)
2 FILT_LIV_poles= np.append(fit_poles_com,fit_zeros)
3 FILT_LIV = signal.ZerosPolesGain(FILT_LIV_zeros, FILT_LIV_poles, 1)
4 _, H_G2 = signal.freqresp(FILT_LIV,np.arange(10e-7,10e-3,1e-7))
5 plt.loglog(np.arange(10e-7,10e-3,1e-7)/(2*np.pi), abs(H_G2)/abs(H_G2[0]))

```

Not enough zeros to set high frequency to unity gain (would be an unphysical without one more pole)

```

1 FILT_LIV_poles_2= np.append(FILT_LIV_poles,-0.00020951281288038756)
2
3 FILT_LIV = signal.ZerosPolesGain(FILT_LIV_zeros, FILT_LIV_poles_2, 1)
4 - , H_G2 = signal.freqresp(FILT_LIV,freq)
5 plt.loglog(freq/(2*np.pi), abs(H_G2)/abs(H_G2[0]))
6
7 [ -,y_G2, -] = signal.lsim(FILT_LIV, unit_step, t_new)
8
9 [ -,y_G2_time, -] = signal.lsim(model_zpk, y_G2, t_new)
10
11 fig = plt.figure(figsize=(17,10))
12 plt.subplot(211)
13 plt.plot(t_new/3600, unit_step,linewidth = lin_thickness,
14           label='RH step input')
15 plt.plot(t_new/3600, y_inp_inv,'--', linewidth = lin_thickness,
16           label='G$\{1\}$')
17 plt.plot(t_new/3600,y_G2,'--', linewidth = lin_thickness,color='purple',
18           label ='G$\{2\}$')
19 plt.ylabel('RH power [W]')
20 plt.title('Comparison between RH inverted response with self heating')
21 plt.legend(fontsize='xx-large')
22 plt.subplot(212)
23 plt.plot(t_new/3600,-y_mod_sim, linewidth = lin_thickness,
24           label = 'RH step input')
25 plt.plot(t_new/3600,-y_sh_resp*20e-6, linewidth = lin_thickness,color='magenta',
26           label = 'self heating (negative)')
27 plt.plot(t_new/3600,-y_mod_sim_inv,'--', linewidth = lin_thickness,color='orange',
28           label ='G$\{1\}$')
29 #plt.plot(t_new/3600,y_sh_resp*20e-6 -y_mod_sim_inv,linewidth = lin_thickness,
30 #           label='diff (orange - green)',color='red')
31 plt.plot(t_new/3600,-y_G2_time,'--', linewidth = lin_thickness,color='purple',
32           label ='G$\{2\}$')
33 plt.ylabel('Defocus [m$^{-1}$]')
34 plt.xlabel('time [hr]')
35 plt.legend(fontsize='xx-large')
36 fig.savefig('G1_and_G2.pdf',bbox_inches='tight')

```

5.6 Misc. thermo-optic filters

5.6.1 COMSOL self heating filter

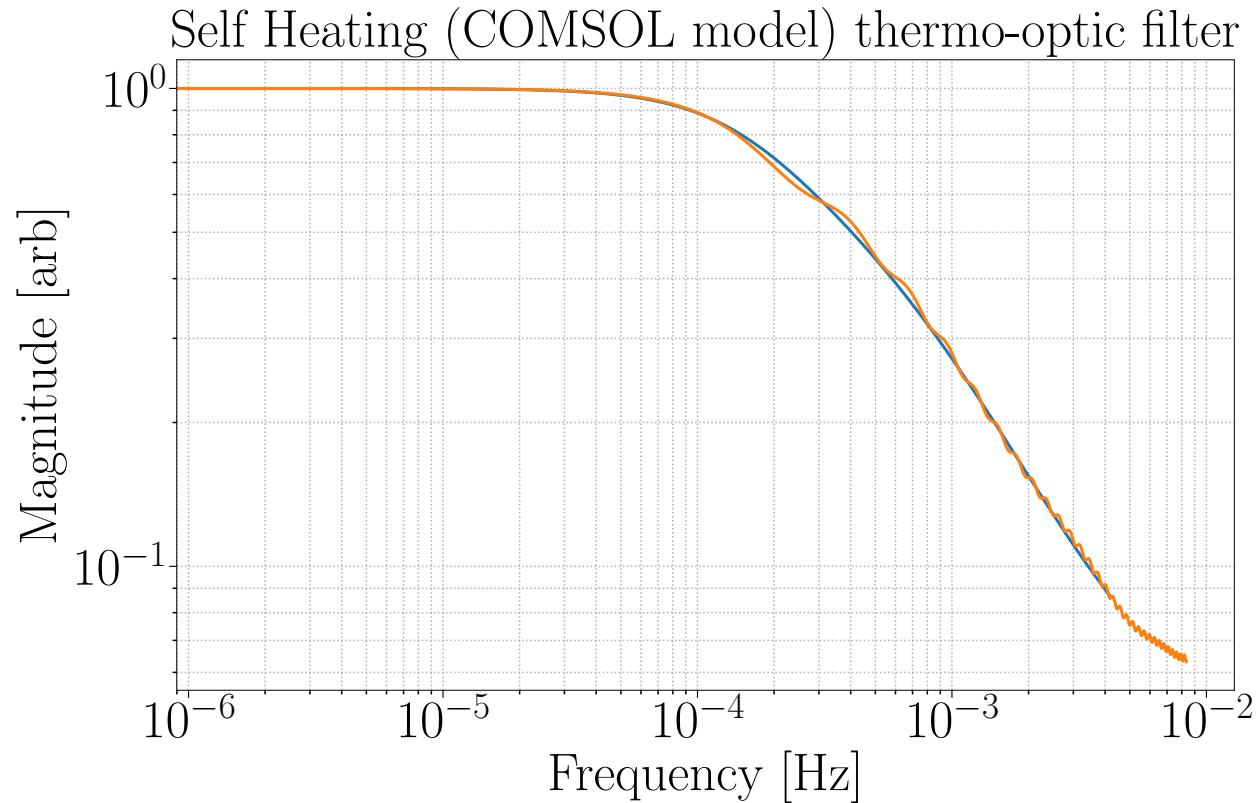


Figure 5.3: Fitted zpk filter to transient response of self heating COMSOL model.

5.6.2 CO₂ filter

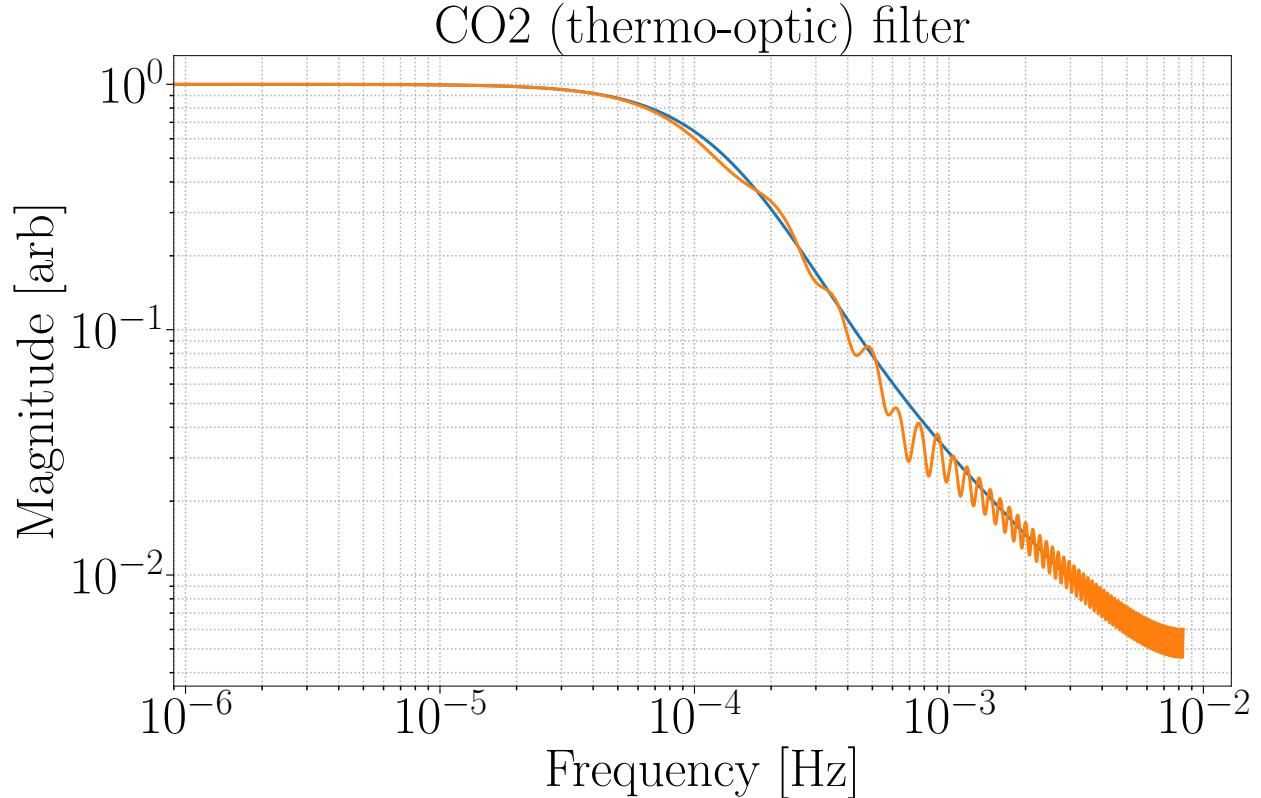


Figure 5.4: Fitted zpk filter to transient CO₂ actuation response.

5.7 Thermo-optic Path Distortion (analytical)

5.7.1 Thermorefractive aberration

Consider an aberration of a substrate with an uninfluenced index of n_0 and a thermo-refractive term ($\frac{dn}{dT}$):

$$n(x, y, z) = n_0 + \frac{dn}{dT}[T(x, y, z) - T_0] \quad (5.65)$$

The above correlates the material index (n) to a path distortion (Ψ) (to first order) from thermal aberrations on a cylindrical substrate volume [13]:

$$\Psi(r) = \frac{dn}{dt} \int_{-h/2}^{h/2} [T(r, z) - T_0] dz \quad (5.66)$$

5.7.2 Thermoelastic aberration

A much more involved derivation with a significantly larger result than above is computed in [54], though best computed for oneself especially for coatings and substrates alternative to $\text{SiO}_2\text{TiO}_2 : \text{Ta}_2\text{O}_5$ and fused silica respectively. It is worth mentioning that the effect for an approximate 1W absorbed power yields a 10 times smaller optical path distortion than that mentioned for the thermal lens [13].

5.7.3 Ring Heater actuation

$$\Psi(t, r) = 2 \frac{dn}{dT} \sum_{m,p=1}^{\infty} A_{m,p} c_p^u \sin(u_m h / 2a) (a/u_m) [1 - e^{-\alpha t}] J_0(\zeta_p r/a) \quad (5.67)$$

[17]

5.8 CO₂ mask

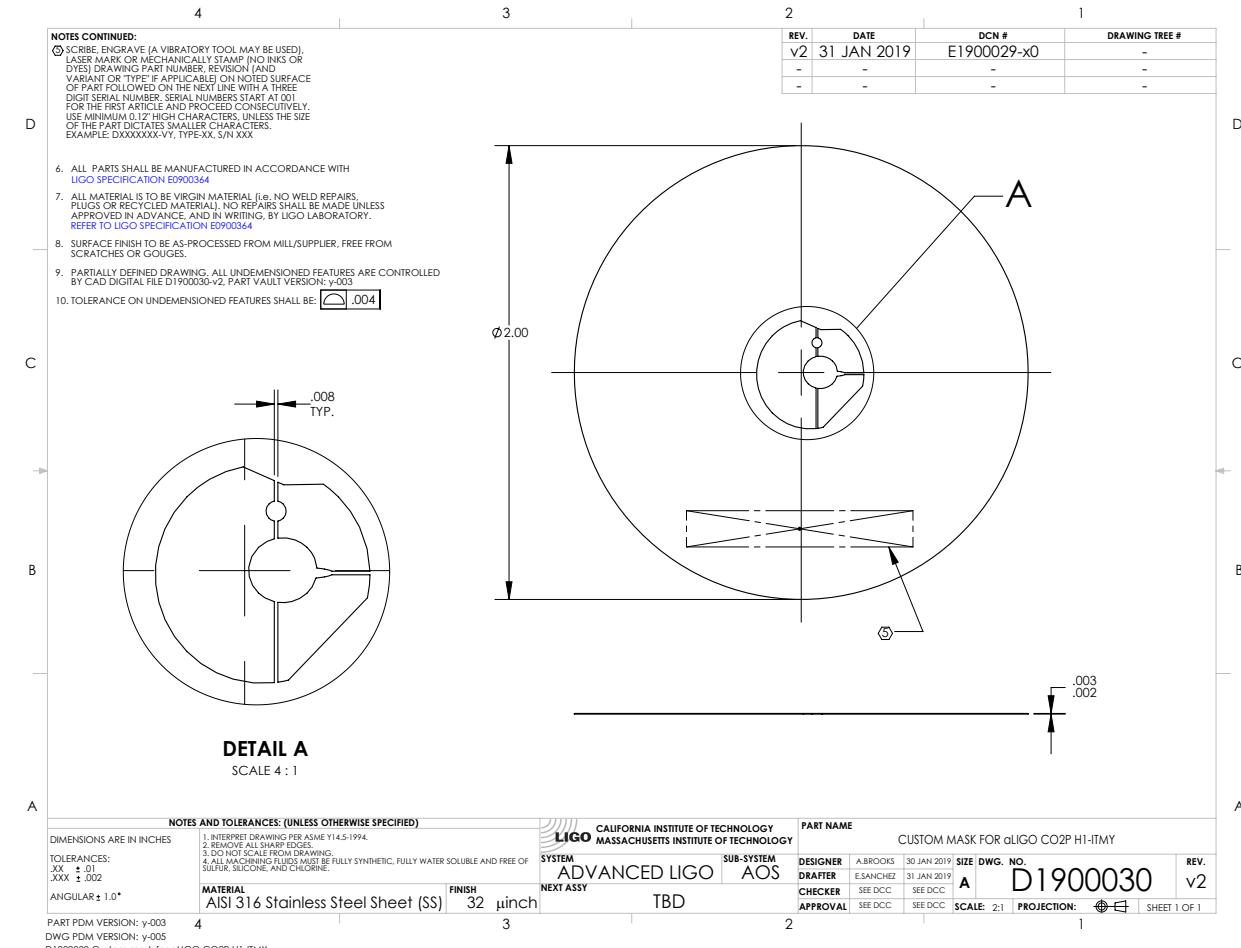


Figure 5.5: A CAD drawing of the first CO₂ mask installed in the CO₂ beam path.

5.9 Anisotropic media

Unlike isotropic media, we do not assume that the index of refraction of anisotropic media is the same for all chosen wave vectors. This is a direct consequence of the birefringence of anisotropic media; characterized by the dielectric, permittivity, and polarization tensors.

5.9.1 Monochromatic plane wave propagation

Revisiting Maxwell's equations for a simple monochromatic plane wave solution provides further direction on how crystalline media may effect incident light. Further elaborating, the following assumptions are made:

$$\vec{E} = E_o e^{(i\omega(\frac{n}{c}\vec{r}\cdot\vec{s}-t))} \quad (5.68)$$

Where n is the index of refraction, c is the speed of light, \vec{r} is the position vector and \vec{s} is the unit wave normal.

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} \quad (5.69)$$

Where \vec{H} is the magnetic field assuming permeability μ , and the generalized displacement vector \vec{D} and electric field vector \vec{E} .

$$\nabla \times \vec{E} = -\mu \vec{H} \quad (5.70)$$

Reducing to only the displacement and electric fields:

$$\vec{D} = \frac{n^2}{\mu} [\vec{E} - \vec{s}(\vec{s} \cdot \vec{E})] \quad (5.71)$$

Maxwell's equations show that the electric field is not necessarily parallel to the displacement field and in most materials with non-zero polarizability tensors and dielectric tensors, it is not. But as specified above, the displacement vector, Electric field and unit wave normal are co-planar while remaining orthogonal to \vec{H} . Assuming we are operating within a coordinate system aligned with the principal dielectric axes, we substitute Equation 5.73 into Equation 5.71:

$$E_i = \frac{n^2 s_i (\vec{E} \cdot \vec{s})}{n^2 - \mu \epsilon_i} \quad (5.72)$$

From here it can be shown that for a general plane wave there exist two unique refractive

index solutions within the constructed dielectric [43].

5.9.2 The Dielectric tensor

Further elaborating on the nature of a generalized dielectric tensor (ε) for any wavevector is required to proceed:

$$D_i = \varepsilon_{ij} E_j \quad (5.73)$$

Where D is the displacement vector, E is the electric field vector, and ε is the dielectric tensor. The displacement vector for isotropic media is retrieved when $i = j$ and $\varepsilon_i = \varepsilon$. To further understand the nature of the dielectric tensor we assert Poynting's theorem providing an energy conservation requirement:

$$\nabla \cdot \vec{S} = \frac{dU}{dt} \quad (5.74)$$

Where $\vec{S} = \vec{E} \times \vec{H}$ is the poynting vector and $U = \frac{1}{8\pi} (\vec{E} \cdot \vec{D} + \vec{B} \cdot \vec{H})$ is the electromagnetic field density. The reader is left to perform the exercise and show that in order for Equation 5.74 to hold true given Equation 5.73

$$\varepsilon_{ij} = \varepsilon_{ji} \quad (5.75)$$

Demonstrating that the dielectric tensor is symmetric - exhibiting only six unique terms. Diagonalizing the tensor, the presence of two unique eigenvectors and eigenvalues indicates the existence of two eigenpolarizations with paired eigenindices.

$$E_i = \frac{n^2 s_i (\vec{E} \cdot \vec{s})}{n^2 - \mu \varepsilon_i} \quad (5.76)$$

Though this result requires revisiting geometrical conditions that are best visualized using a method introduced in the next section [43].

5.9.3 Miller indices for highly reflective GaAs/Al_{0.92}Ga_{0.08}As coatings

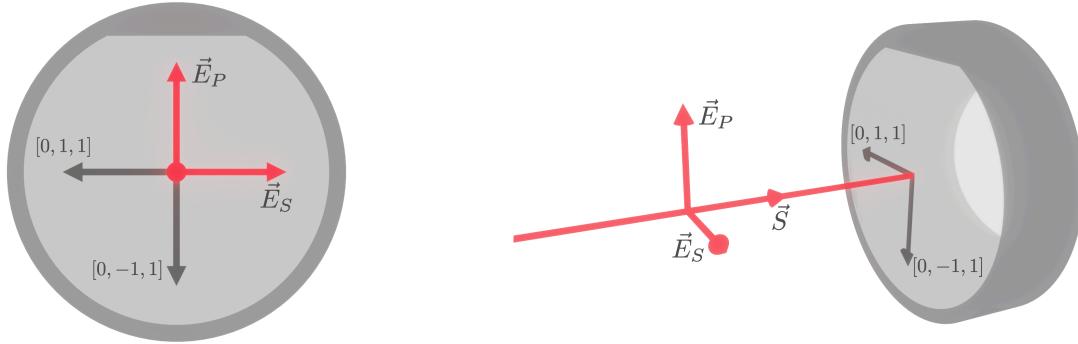


Figure 5.6: The beam propagation axis (\vec{S} , $[-100]$) with respect to the GaAs/Al_{0.92}Ga_{0.08}As crystal axes. The axis formed by the $[100]$ plane normal is drawn parallel with the beam axis (z-axis) and the polarizations of incident and reflected beam oscillate along vectors within the plane formed by the normal of that axis. The coating is grown with a flat tracing a line within the $[0-11]$ plane; where the plane normal points towards the sample center.

Up to this point three varieties of orthonormal coordinates are addressed: the crystal axis (as indicated by Miller index plane normals), the principal dielectric axis (based on diagonalization of the indicatrix), and an optical beam axis (when considering a desired (laser) light propagation). The asserted beam axis can be cited Figure 5.6.

5.10 Mode matching data for Electro-optic sample cavity

5.10.1 Pre MMT beam scan

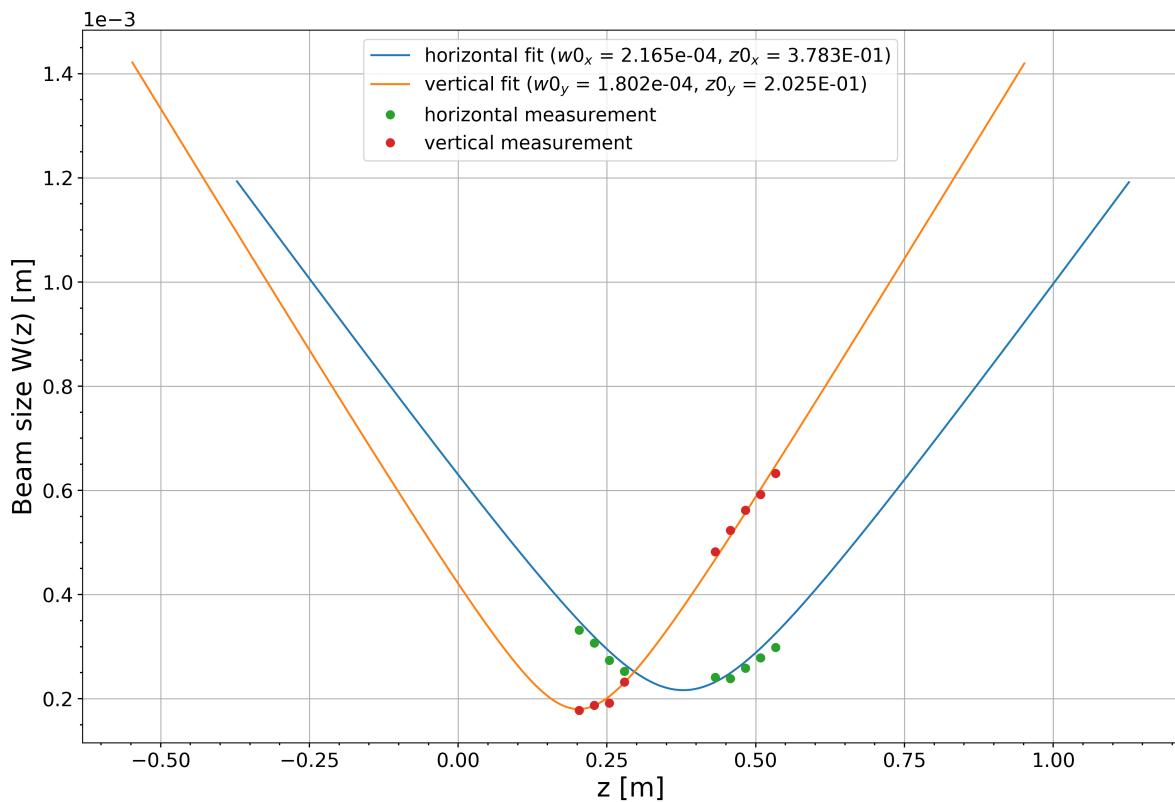


Figure 5.7: Beam scan taken from SM5 (Steering mirror 5)

5.10.2 “Just another mode matching tool” (JAMMT) solution

5.10.3 Post MMT beam scan

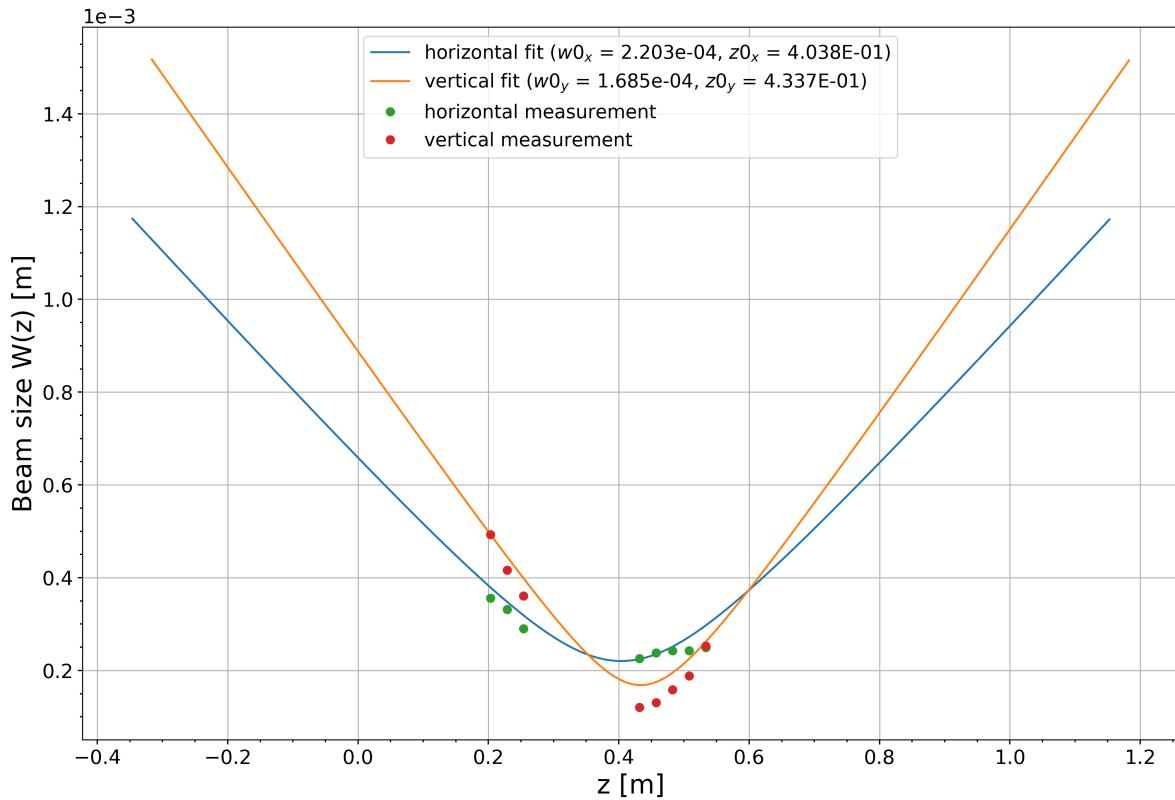


Figure 5.8: Beam scan taken from SM6. Sampling points before SM7 and after the first cavity iris.

5.11 Laser PZT sweep

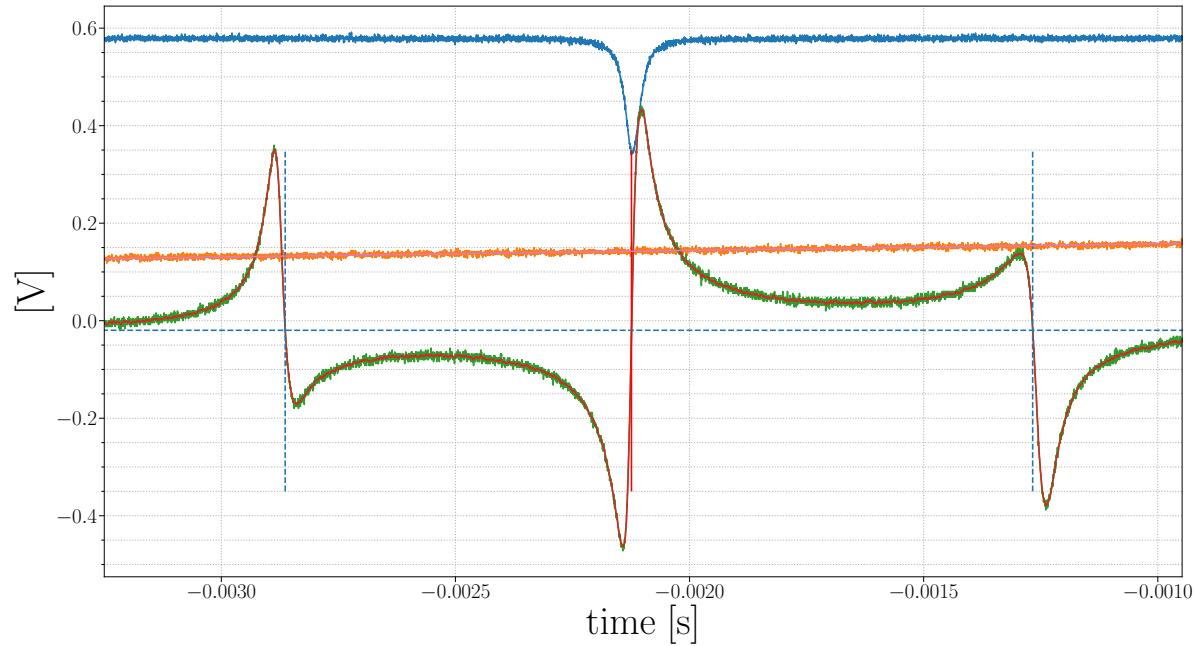


Figure 5.9: Ramping voltage sent to the laser PZT while probing the mixer output. The sweep was performed for sample cavity of length notes

5.12 Voltage drive electronics

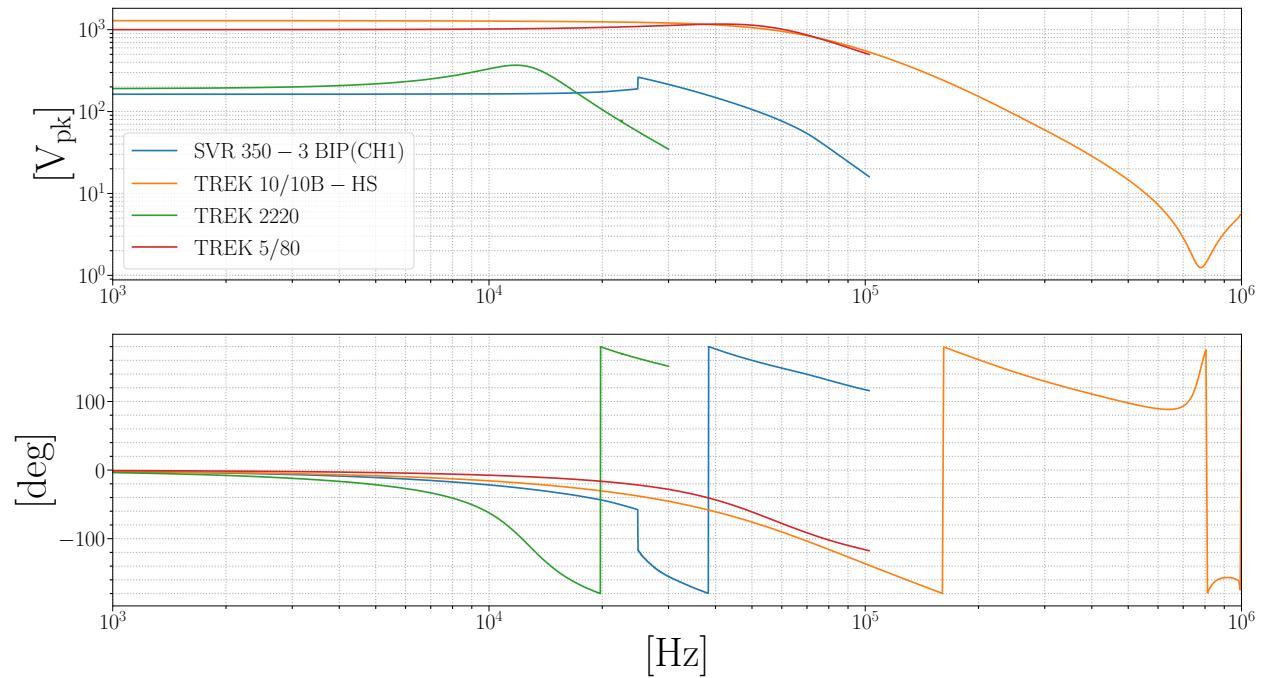


Figure 5.10: Different high voltage amplifier transfer functions used for the study

5.13 Alternate Mounting Solutions (Assemblies 1 → 3)

5.13.1 Assembly 1

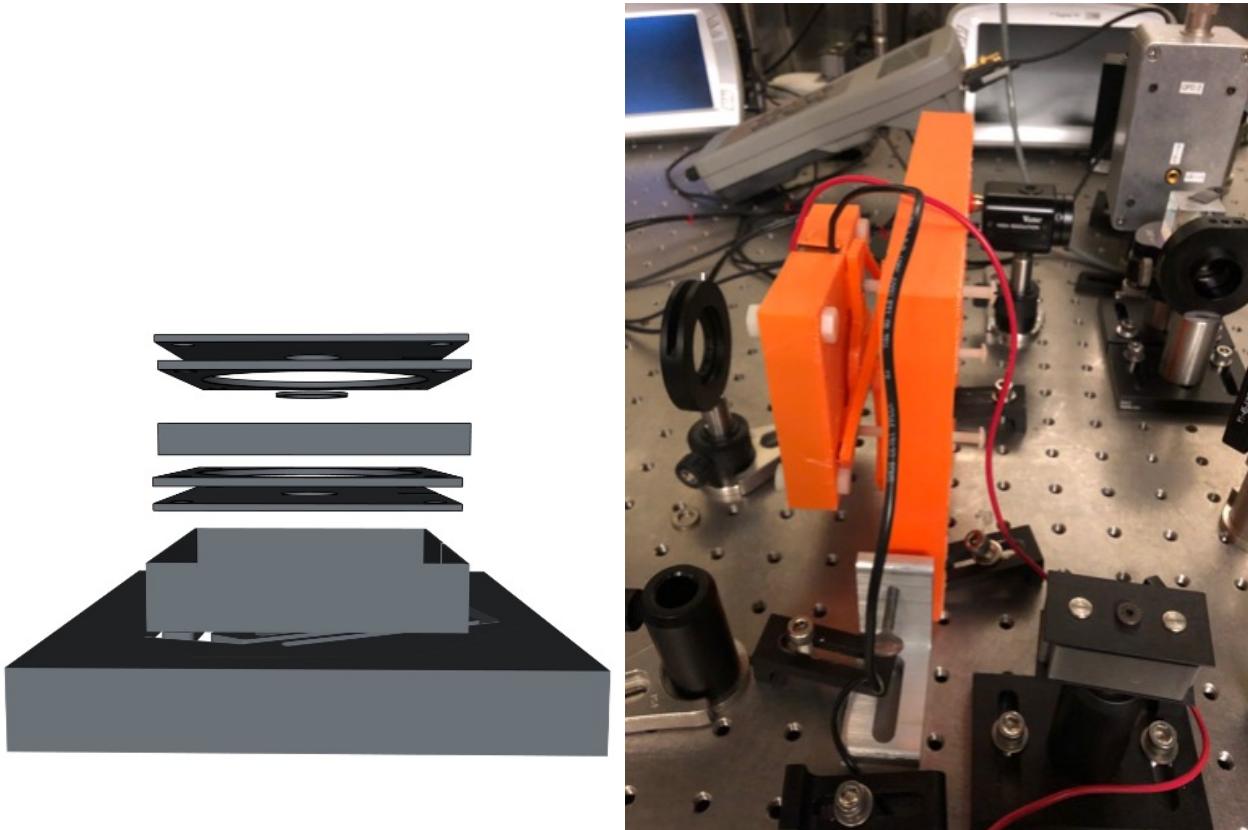


Figure 5.11: Assembly 1 was constructed to meet two criteria: the same solution of housing the sample and electrodes as Assembly 0, but also offer pitch / yaw control via an ortho-planar spring design (Brigham Young University).

The construction revealed flaws; made most obvious when comparing to displacement noise of traditional optical mounts. Pitch and yaw control via the ortho-planar spring were prioritized to avoid metal springs and further mount pieces. The solution became unjustifiable when observing the the displacement noise coupling from the mount.

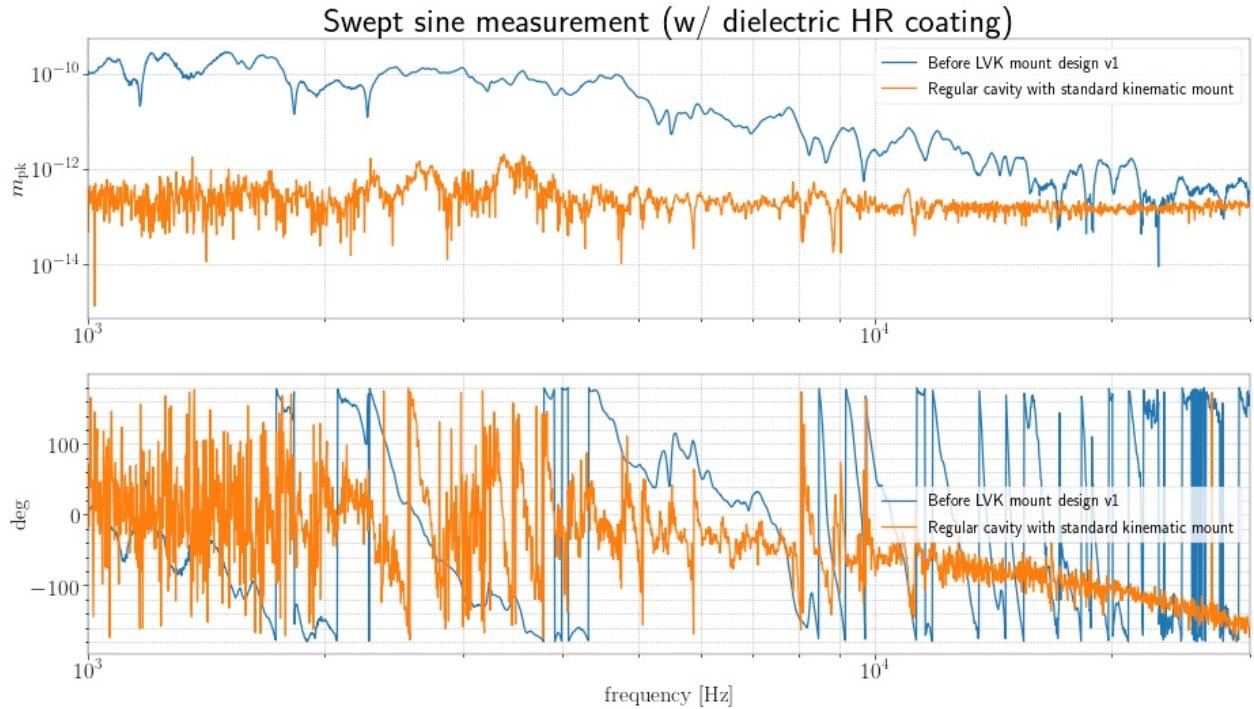


Figure 5.12: Measured displacement spectra for Assembly 1.2 of the longitudinal pockels cell mount compared to the standard kinematic mount. Both measurements were recorded with the the CVI Melles-Griot (amorphous) mirror coating sample installed in the assembly.

Attempts at improving the pitch dithering were tried with side set screws but mostly caused significant misalignment (less power in fundamental mode, locking issues, etc.). This lead to the final alternative for this assembly type; with the same PLA stack indicated in Assembly 1.

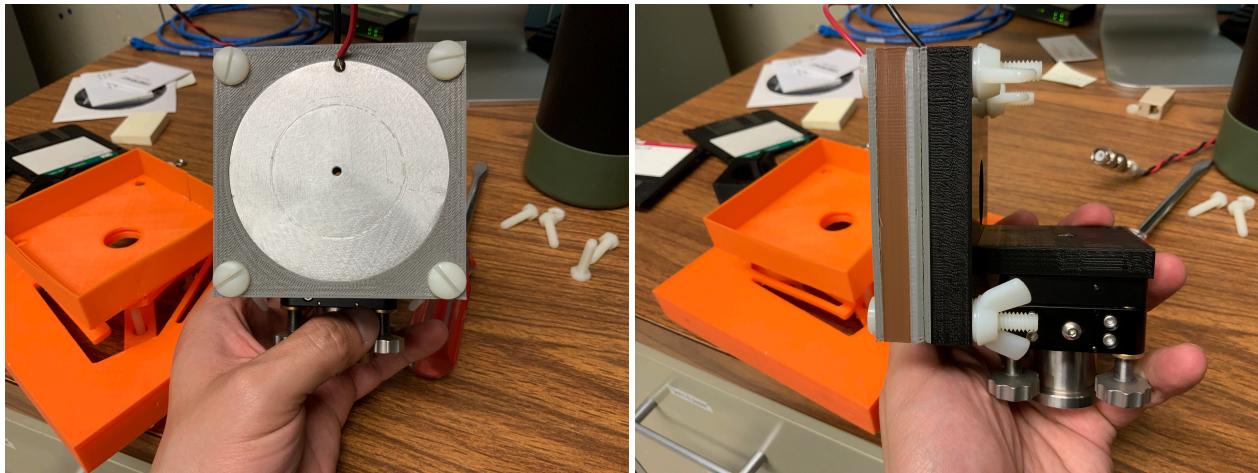


Figure 5.13: A modification implemented with the intention of reducing pitch dithering while still having control of DC YAW

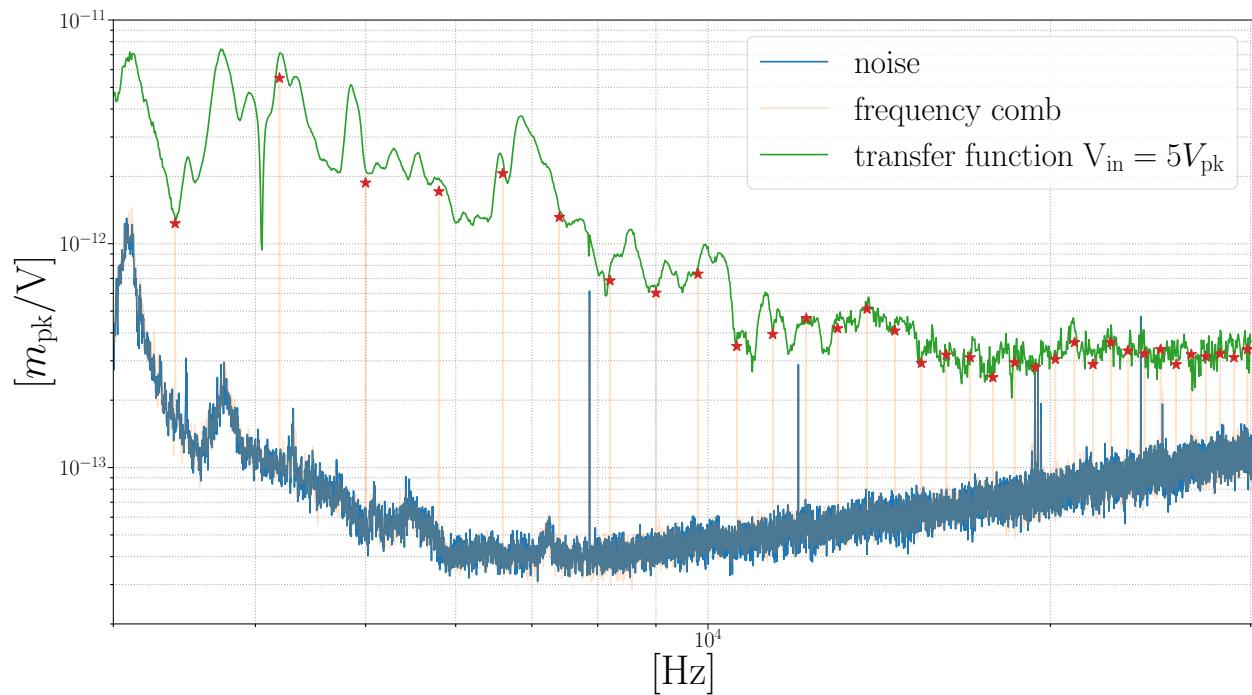


Figure 5.14: Assembly 1.2 and 1.3 transfer function measurement and separate noise displacement spectra measurement with the GaAs/Al_{0.92}Ga_{0.08}As sample installed. Measurements were taken from 3kHz up to 21kHz on using a Stanford Research 785 spectrum analyzer.

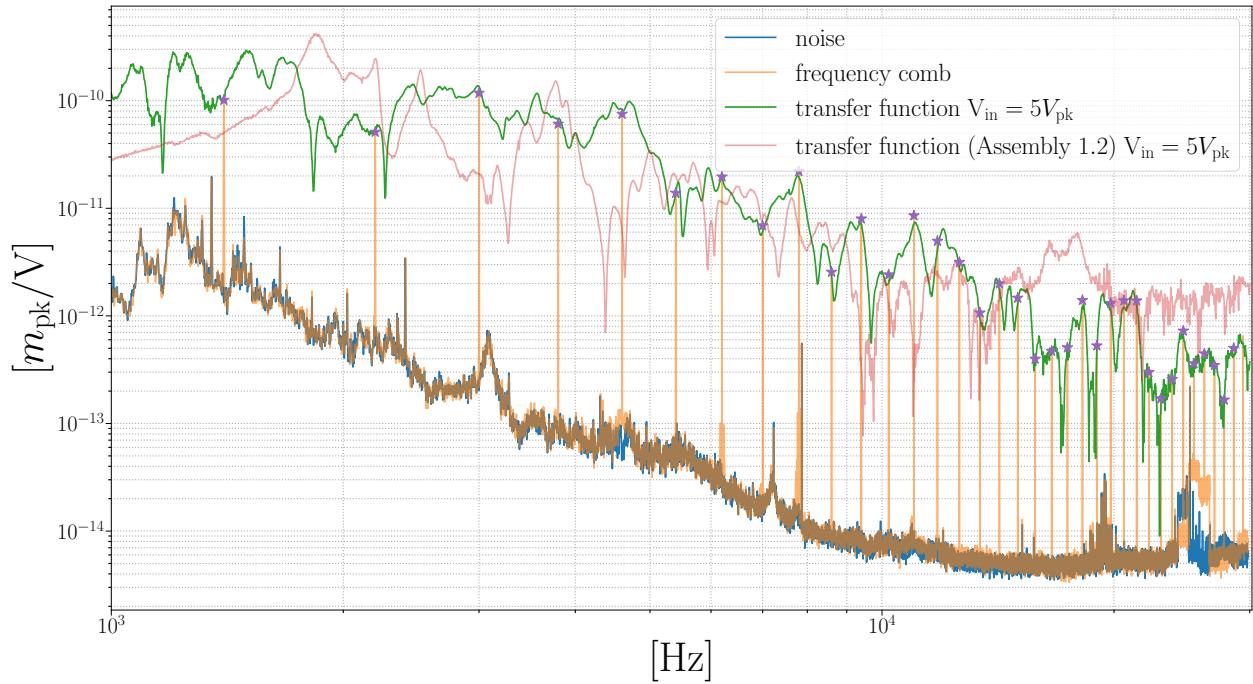


Figure 5.15: Assembly 1.2 and 1.3 transfer function measurement and separate noise displacement spectra measurement with a CVI Melles Griot flat mirror sample ($R \approx$ sample installed). Measurements were taken from 1kHz up to 30kHz on using a Stanford Research 785 spectrum analyzer.

5.13.2 Assembly 2

With considerations after Assembly 1, a more monolithic optical mount design with a simple geometry was imagined. PLA material compliance factoring to the seen drive noise. To see if it could be due to the compliance of the assembly material or printing, we tested this assembly design against different infills of PLA, PETG, and a version machined from solid PVC. With this modification, came also a different plate geometry. The assumption is the region of interest of the injection would not be heavily influenced by the non-circular plate geometry where it matters.

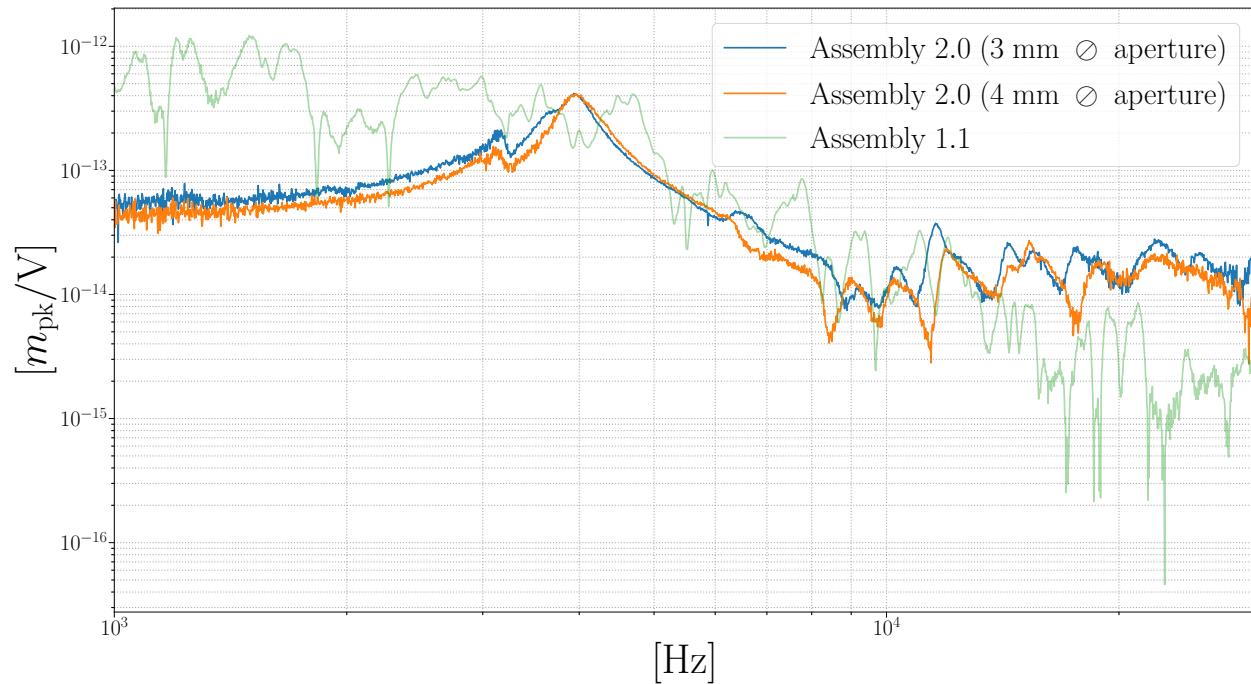


Figure 5.16: Assembly 2.0 transfer function measurements compared to Assembly 1.1, Measurements were taken from 1kHz up to 30kHz on using a Stanford Research 785 spectrum analyzer.

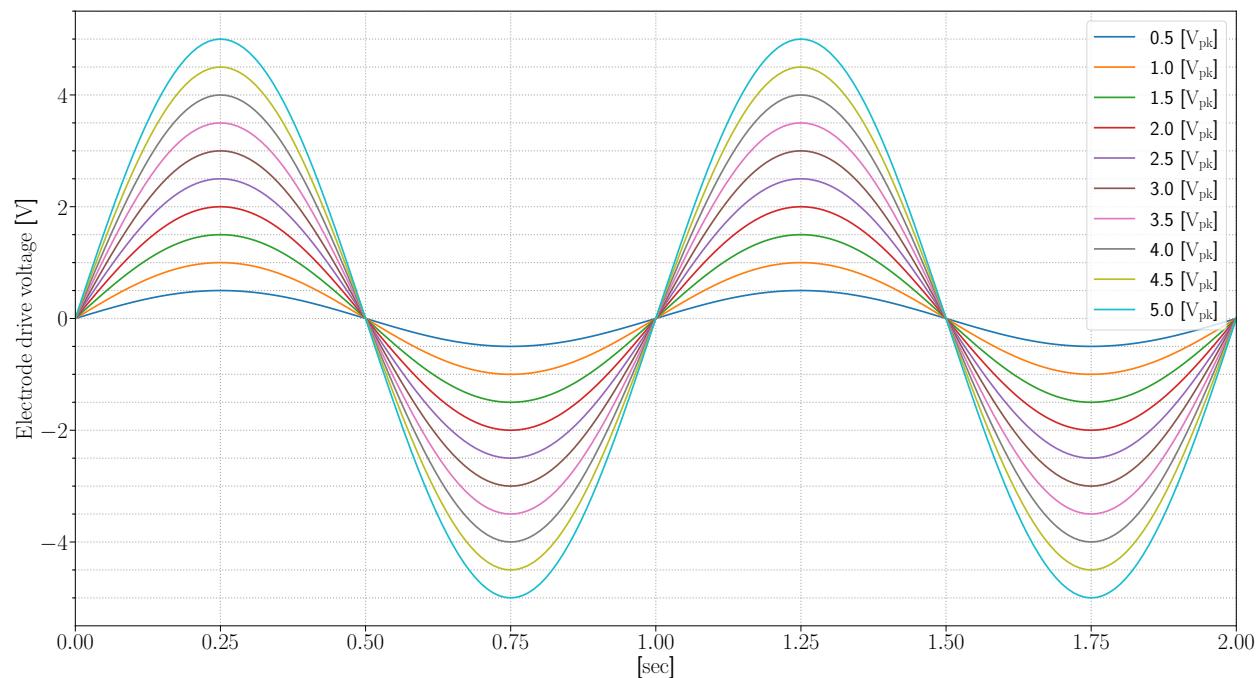


Figure 5.17: The varied drive amplitudes input into the HVA to perform the following test

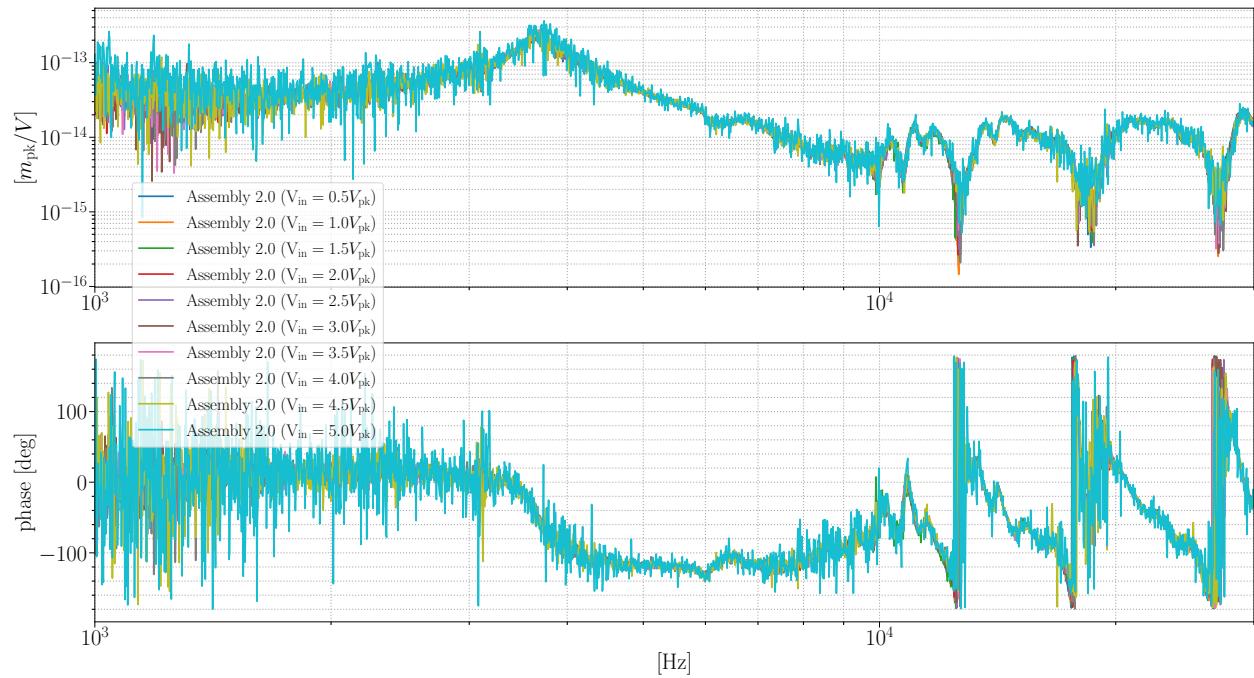


Figure 5.18: Assembly 2.0 transfer function measurements with varied drive amplitudes indicated in Figure 5.17.

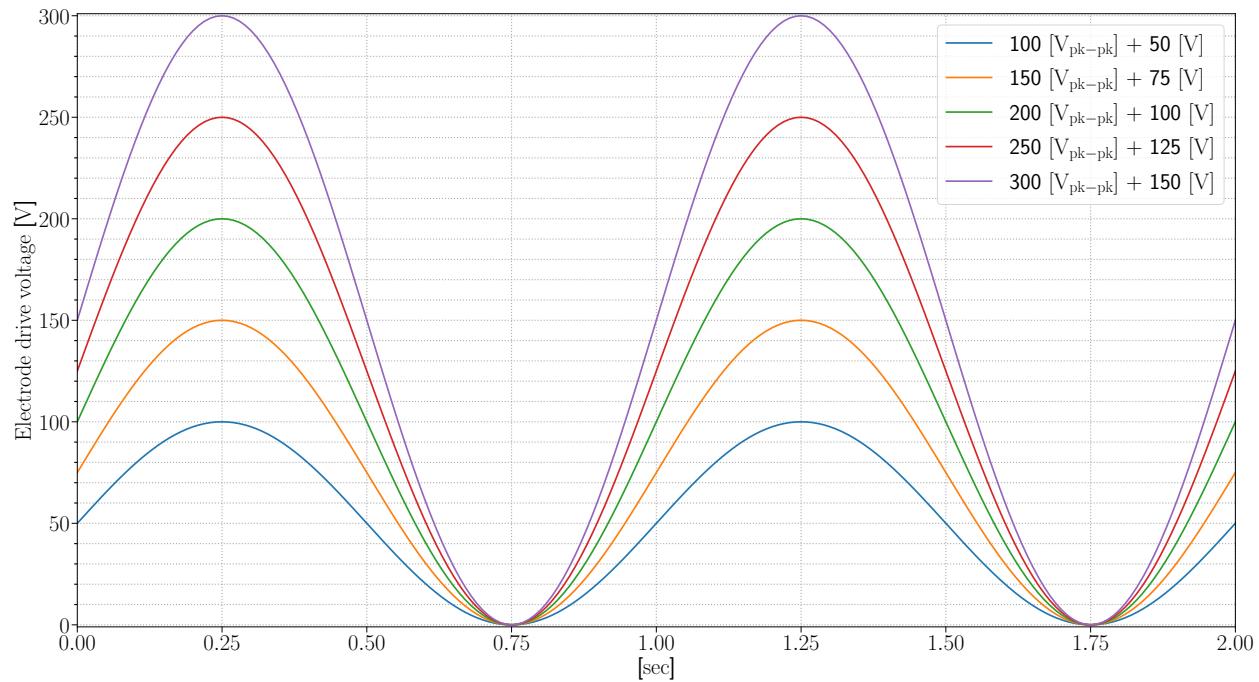


Figure 5.19: The varied drive amplitudes and offsets input into the HVA to perform the following test

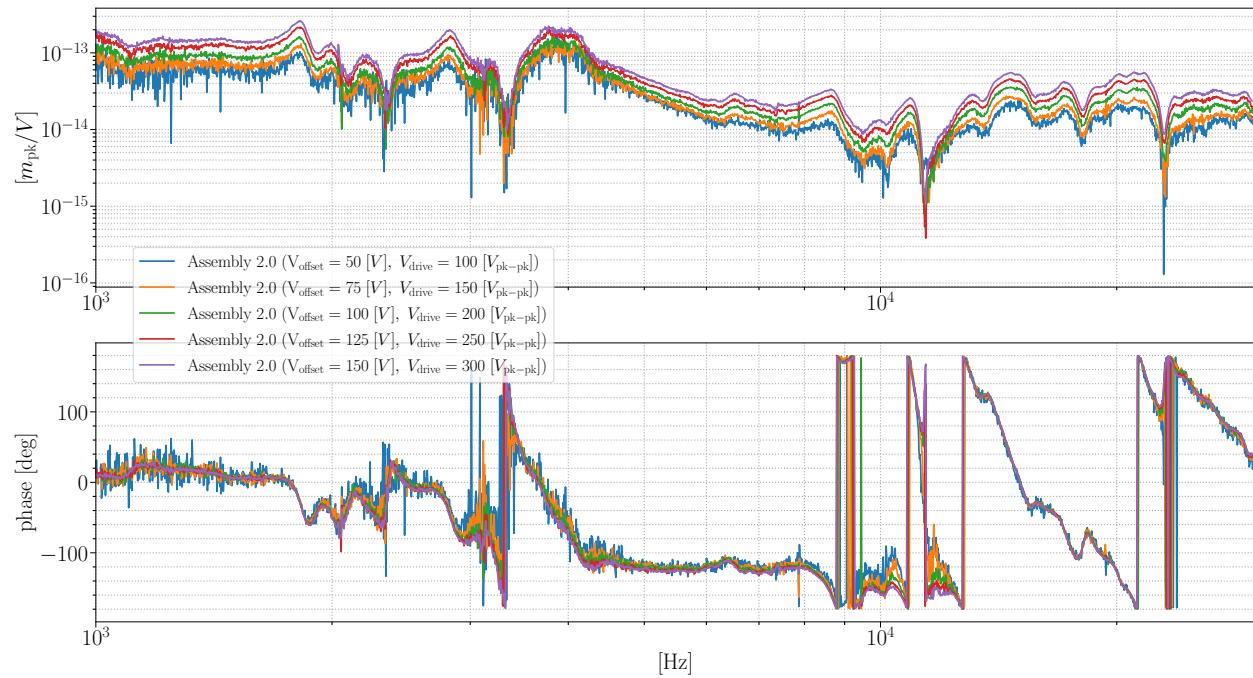


Figure 5.20: Assembly 2.0 transfer function measurements with varied drive amplitudes and offsets indicated in Figure 5.19.

5.14 Assembly blueprints and alternative views

5.14.1 Assembly 1

5.14.1.1 Cross section

5.14.1.2 Electrodes

5.14.1.3 Iteration 1.1

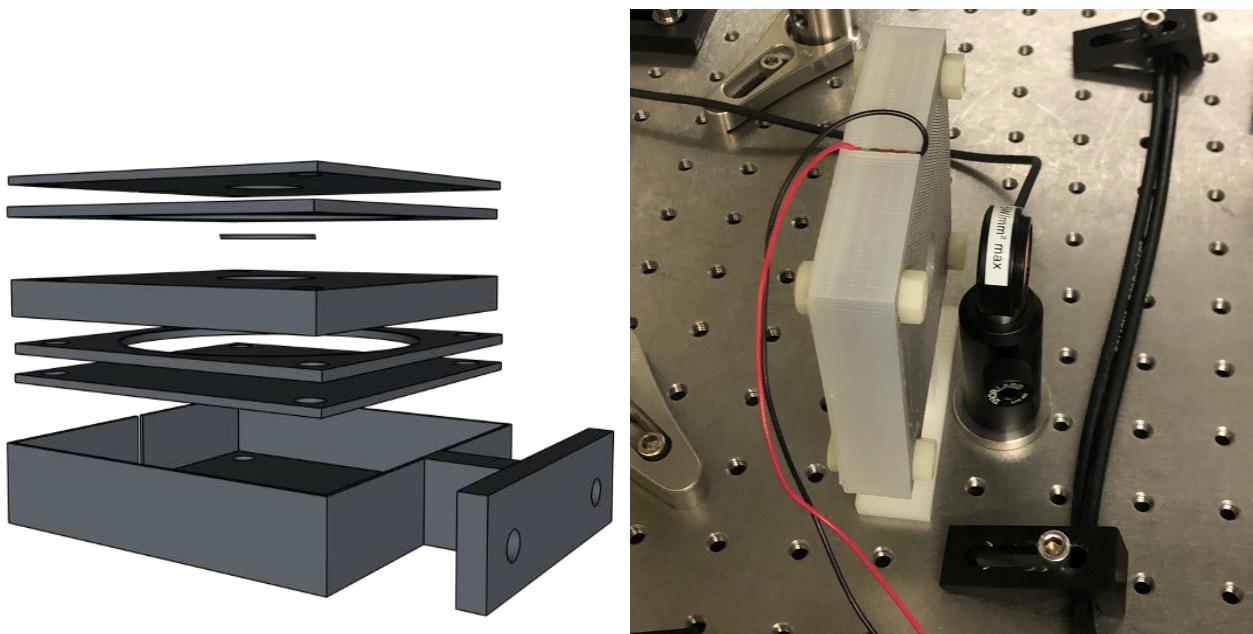


Figure 5.21: Assembly 0 was constructed to meet the criteria of providing a non-conductive housing for the electrode / sample assembly while maintaining a fixed length spacing using parts 3d printed with polylactic acid (PLA).

5.14.1.4 Iteration 1.2

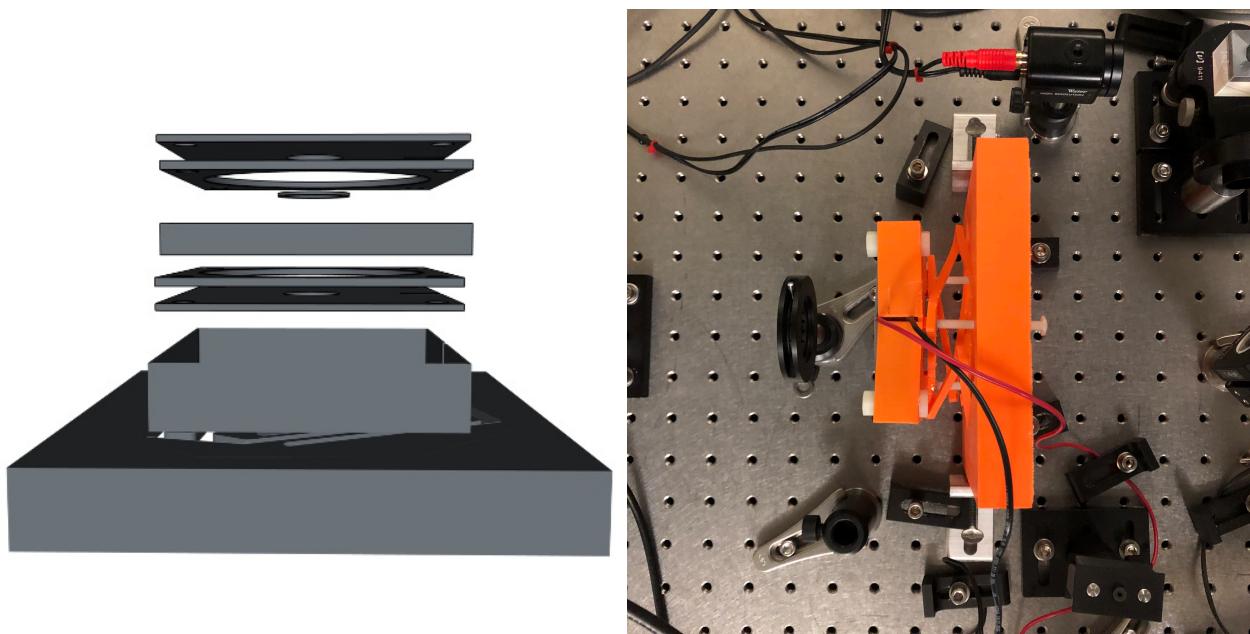


Figure 5.22: Assembly 1 was constructed to meet the criteria of providing a non-conductive housing for the electrode / sample assembly while maintaining a fixed length spacing using parts 3d printed with polylactic acid (PLA).

5.14.1.5 Iteration 1.3

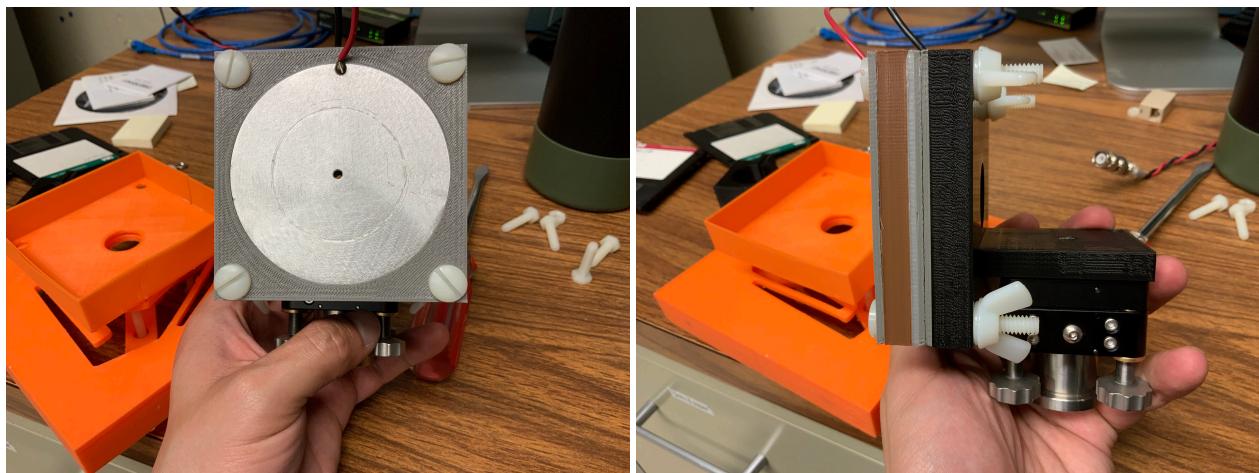


Figure 5.23: A modification implemented with the intention of reducing pitch dithering while still having control of DC YAW

5.14.2 Assembly 2

5.14.2.1 Cross section

5.14.2.2 Electrodes

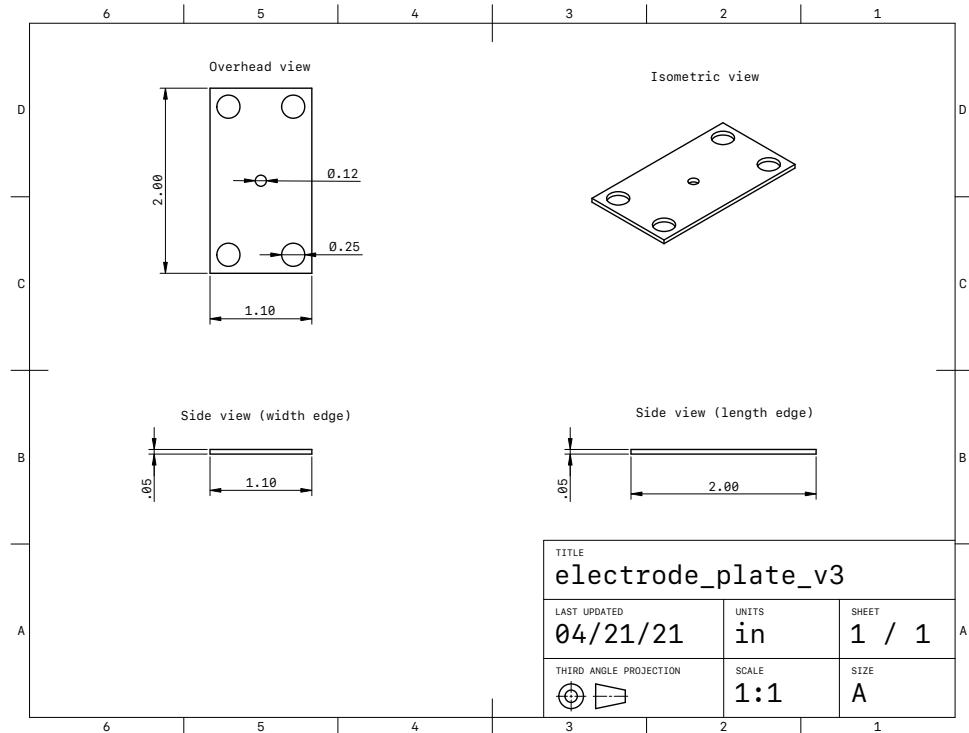


Figure 5.24: Rectangular (.05"X1.1"X2") plates made of aluminum.

5.14.2.3 Iteration 2.1

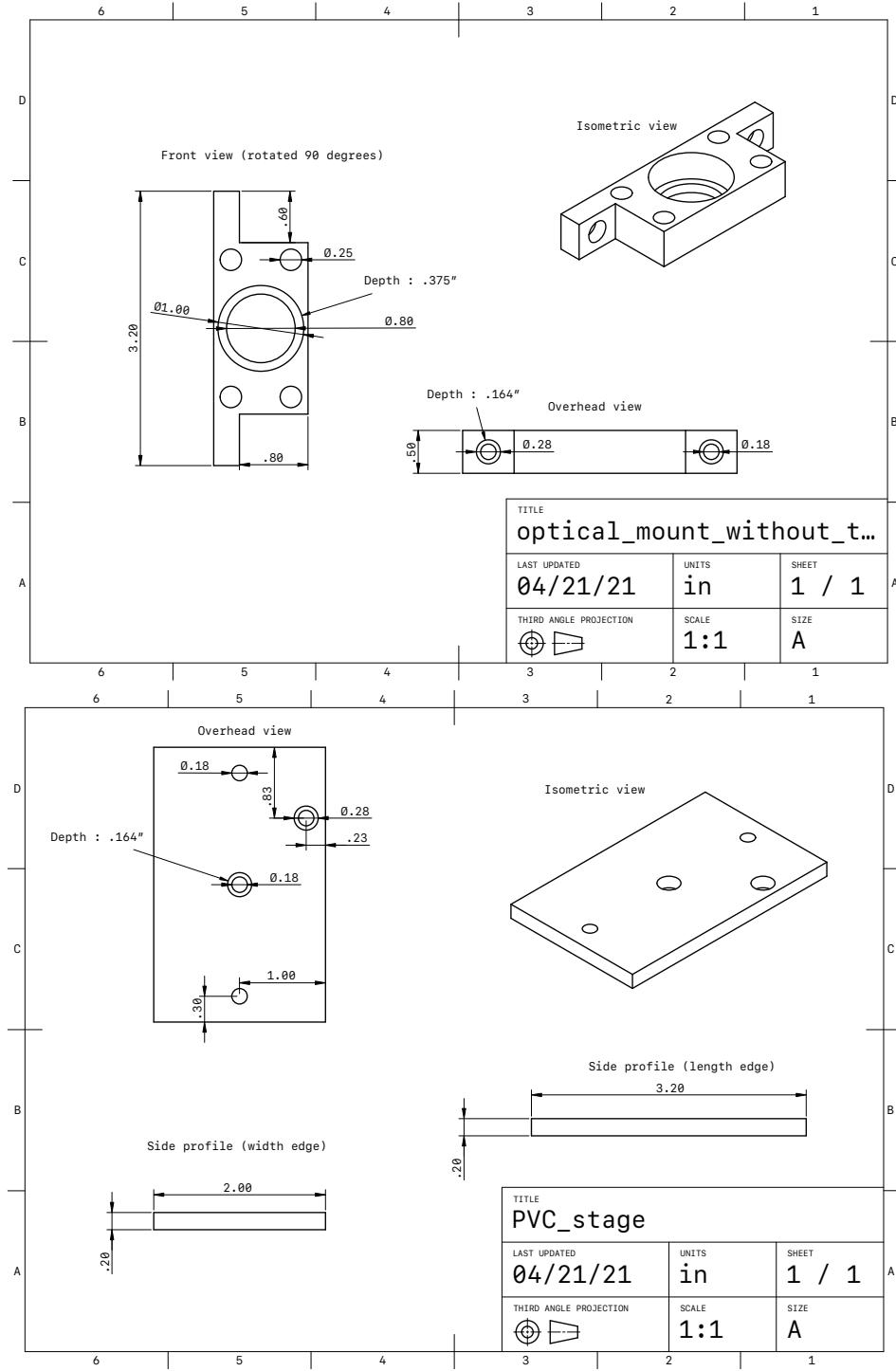


Figure 5.25: A design iteration of the assembly 2 mounts. Materials tried varied from PVC, PLA, and PETG. Quarter inch holes are bored in order to pass nylon screws holding electrode plates fixed to the mount.

5.14.2.4 Iteration 2.2**5.14.2.5 3D printed w/ MACOR spacers**

Insert blueprints

Insert printed results

5.14.3 Assembly 3 [MACOR] (blueprint)

5.14.3.1 Cross section

5.14.3.2 Electrodes

5.14.3.3 Iteration 3.0

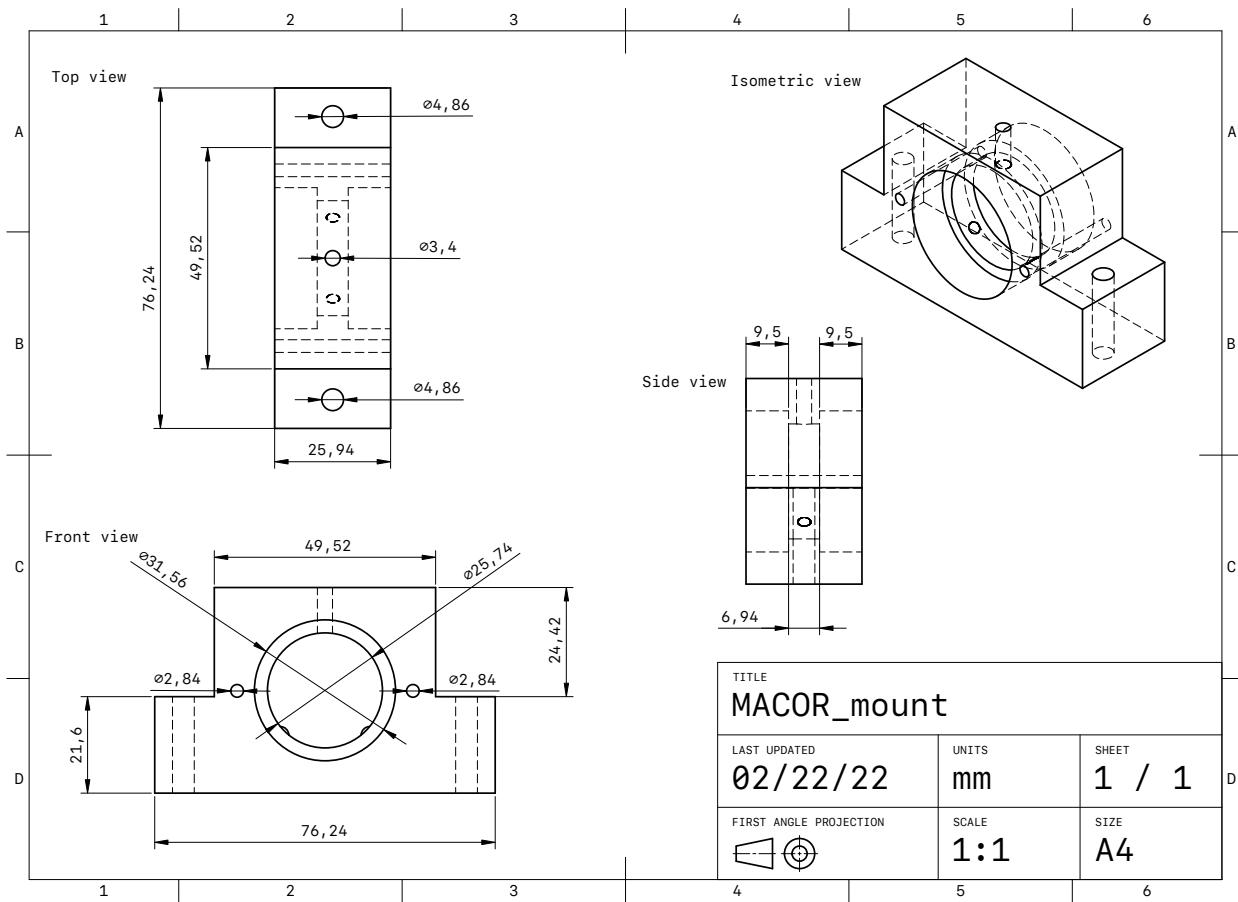


Figure 5.26: MACOR mount design constructed in Shapr3D

5.15 LaplacE code

5.15.1 laplace.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import h5py
4 from scipy.sparse import lil_matrix
5 from scipy.sparse import spdiags
6 import torch
7
8 # Computes Laplace's equation in cartesian and cylindrical coordinates
9 # For some related detailed documentation: Numerical recipies (3rd edition)
10 (Chapter 20 [Partial Differential Equations])
11
12 ## Initialize fields
13
14 def init_coords(pdct):
15     """
16     Looks at params file to start implementing coordinate choices for simulation
17     """
18     if pdct['coords'] == 'cylindrical':
19         if pdct['torch']:
20             i_rho = torch.arange(pdct['origin'][0],pdct['N'][0])
21             i_z = torch.arange(pdct['origin'][1],pdct['N'][1])
22             rho_ = i_rho*torch.tensor(pdct['res'][0])
23             z_ = i_z*torch.tensor(pdct['res'][1])
24             rho, z = torch.meshgrid(rho_, z_, indexing='ij')
25             invrho_ = 1/rho_
26             invrho_[0] = 0
27             invrho, z0 = torch.meshgrid(invrho_, z_, indexing='ij')
28
29     else:
30         i_rho = np.arange(pdct['origin'][0],pdct['N'][0])
31         i_z = np.arange(pdct['origin'][1],pdct['N'][1])
32         rho_ = i_rho*pdct['res'][0]
33         z_ = i_z*pdct['res'][1]
```

```

34     rho = (rho_ * np.ones((pdic['N'][0],1)))
35     rho = rho.reshape(pdic['N'][0]*pdic['N'][1],1)
36     z = (np.ones((pdic['N'][1],1)) * z_-).T
37     z = z.reshape(pdic['N'][0]*pdic['N'][1],1)
38     invrho = 1/rho
39     invrho[rho==0] = 0 # addresses inf elements
40
41     coord_dict = {
42         'coords' : {
43             'rho' : rho, #np.round(rho,abs(int(np.log10(pdic['res'][0])))),
44             'z' : z, #np.round(z,abs(int(np.log10(pdic['res'][0])))),
45             'invrho' : invrho}, #np.round(invrho,abs(int(np.log10(pdic['res'][0])))),
46         'indices' : {
47             'rho' : i_rho,
48             'z' : i_z}
49     }
50
51 #elif pdic['coords'] == 'cartesian':
52
53     return coord_dict
54
55 def indx(icoord1, icoord2, N):
56     """
57     formalized lambda function reshaping potential (vectorizing V):
58     indx = lambda i_rho, i_z : np.int32(i_rho + i_z*(N))
59     """
60     return np.int32(icoord1 + icoord2*N)
61
62 def idx_match(vec,N,step):
63     """
64     Acquire nearest matching ind(ex/ices) for queried location(s) in potential map
65     """
66     idx = np.int32(np.round(vec/step, decimals=0))
67     idx = 1 if idx<1 else N if idx>N else idx
68     return idx
69
70 def init_V(N):

```

```

71     """
72     Initialize (square) potential map
73     """
74     return np.zeros((N**2,1))
75
76
77 def build_lambd(i1, i2, N):
78     """
79     Constructs a matrix for a lambda function,
80     which operates on all available indices in the simulation.
81     Preallocates memory so that the indx function
82     doesn't need to be used twice (reducing computations).
83     """
84     LAMBD = np.array([indx(i, i2, N) for i in i1])
85
86     return LAMBD
87
88 def bc_set(pdct, BC, N, V):
89     """
90     Establishes simulation boundary conditions
91     """
92     #global R, d, step, idx, V, rho, z, bc0set
93
94     #Plate bcs
95     if pdct['coords'] == 'cylindrical':
96
97         #Setting up the edge boundaries (for faster convergence)
98         if not bc0set:
99             rho0 = False
100            rhoend = np.interp(np.arange(0,N), np.array([0,N-1]),
101                            np.array([pdct['back_plate']['voltage'],
102                                     pdct['front_plate']['voltage']])).reshape(N,1)
103            z0 = pdct['back_plate']['voltage']
104            zend = pdct['front_plate']['voltage']
105            edge_vals =np.array(rho0, rhoend, z0, zend)
106            V = bc_edge(pdct, edge_vals, V)
107            bc0set = True

```

```

108
109      #Set potentials
110      for i in range(BC['cont']):
111          V = set_pot(V,BC[i]['coords'],BC[i]['values'],LAMBD)
112
113      # exponential boundary conditions
114      V0 = 0
115      R0 = 1
116      V[idx(np.arange(0,N),N-1)] = V0 +
117          np.exp(-step/R0)*(V[idx(np.arange(0,N), N-2)]-V0)
118      V[idx(np.arange(0,N),0)] = V0 +
119          np.exp(-step/R0)*(V[idx(np.arange(0,N),1)]-V0)
120      V[idx(N-1,np.arange(0,N))] = V0 +
121          np.exp(-step/R0)*(V[idx(N-2, np.arange(0,N))]-V0)
122
123      return V
124
125      # Constructing the operator(s)
126
127  def build_lap(pdct, LAMBD, i_rho):
128      """
129      constructs first order structure of the laplace operator
130      """
131      if pdct['coords'] == 'cylindrical':
132
133          op_shape = (pdct['N'][0]**2, pdct['N'][1]**2)
134
135          if pdct['torch'] == True:
136              idx_1 = LAMBD[0,1:-1]
137              idx_2 = LAMBD[1,1:-1]
138              idx_3 = LAMBD[0,:-2]
139              idx_4 = LAMBD[0,2:]
140              size_ = (pdct['N'][0]-2)**2
141              idx_5 = LAMBD[1:-1,1:-1].reshape(size_)
142              idx_6 = LAMBD[1:-1,:-2].reshape(size_)
143              idx_7 = LAMBD[1:-1,2:].reshape(size_)
144              idx_8 = LAMBD[:-2,1:-1].reshape(size_)
```

```

145     idx_9 = LAMBD[2:,1:-1].reshape(size_)
146     ones_1 = np.ones(idx_1.shape)
147     ones_2 = np.ones(idx_5.shape)
148     const_ = (np.ones((1,i_rho[1:-1].shape[0])).T
149                 *(((1/2)/(i_rho[1:-1])))).reshape(size_)
150     lap1 = torch.sparse_coo_tensor(np.array([idx_1, idx_1]),
151                                     -6*ones_1, op_shape, dtype=torch.float32)
152     lap2 = torch.sparse_coo_tensor(np.array([idx_1, idx_2]),
153                                     4*ones_1, op_shape, dtype=torch.float32)
154     lap3 = torch.sparse_coo_tensor(np.array([idx_1, idx_3]),
155                                     ones_1, op_shape, dtype=torch.float32)
156     lap4 = torch.sparse_coo_tensor(np.array([idx_1, idx_4]),
157                                     ones_1, op_shape, dtype=torch.float32)
158     lap5 = torch.sparse_coo_tensor(np.array([idx_5, idx_5]),
159                                     -4*ones_2, op_shape, dtype=torch.float32)
160     lap6 = torch.sparse_coo_tensor(np.array([idx_5, idx_6]),
161                                     ones_2, op_shape, dtype=torch.float32)
162     lap7 = torch.sparse_coo_tensor(np.array([idx_5, idx_7]),
163                                     ones_2, op_shape, dtype=torch.float32)
164     lap8 = torch.sparse_coo_tensor(np.array([idx_5, idx_8]),
165                                     1 - const_, op_shape, dtype=torch.float32)
166     lap9 = torch.sparse_coo_tensor(np.array([idx_5, idx_9]),
167                                     1 + const_, op_shape, dtype=torch.float32)
168     lap_ = lap1 + lap2 + lap3 + lap4 + lap5 + lap6 + lap7 + lap8 + lap9
169     lap = lap_/(pdict['res'][0]**2)
170 else:
171     lap = lil_matrix(op_shape,dtype=pdict['bitres'])
172     lap[LAMBD[0,1:-1], LAMBD[0,1:-1]] = -6
173     lap[LAMBD[0,1:-1], LAMBD[1,1:-1]] = 4
174     lap[LAMBD[0,1:-1], LAMBD[0,:-2]] = 1
175     lap[LAMBD[0,1:-1], LAMBD[0,2:]] = 1
176     lap[LAMBD[1:-1,1:-1], LAMBD[1:-1,1:-1]] = -4
177     lap[LAMBD[1:-1,1:-1], LAMBD[1:-1,:-2]] = 1
178     lap[LAMBD[1:-1,1:-1], LAMBD[1:-1,2:]] = 1
179     lap[LAMBD[1:-1,1:-1], LAMBD[:,-2,1:-1]]= 1 - ((1/2)/(i_rho[1:-1]))
180     lap[LAMBD[1:-1,1:-1], LAMBD[2:,1:-1]]= 1 + ((1/2)/(i_rho[1:-1]))
181     lap = lap/(pdict['res'][0]**2)

```

```
182
183     #elif pdict['coords'] == 'cartesian':
184
185     return lap
186
187 def build_grad(pdict, LAMBD):
188     """
189     Gradient operators
190     """
191
192     if pdict['coords'] == 'cylindrical':
193         if pdict['torch'] == True:
194
195             idx1 = LAMBD[1:-1,:]
196             idx2 = LAMBD[:-2,:]
197             idx3 = LAMBD[2:,:]
198             idx4 = LAMBD[:,1:-1]
199             idx5 = LAMBD[:,:-2]
200             idx6 = LAMBD[:,2:]
201
202             gradrho1 = torch.sparse_coo_tensor(np.array([idx1, idx2]),
203                                              -1/2, op_shape, dtype=pdict['bitres'])
204             gradrho2 = torch.sparse_coo_tensor(np.array([idx1, idx3]),
205                                              -1/2, op_shape, dtype=pdict['bitres'])
206             GRADrho = (gradrho1+gradrho2)/pdict['res'][0]
207
208             gradrhopos1 = torch.sparse_coo_tensor(np.array([idx1, idx1]),
209                                              -1, op_shape, dtype=pdict['bitres'])
210             gradrhopos2 = torch.sparse_coo_tensor(np.array([idx1, idx3]),
211                                              -1, op_shape, dtype=pdict['bitres'])
212             GRADrhopos = (gradrhopos1+gradrhopos2)/pdict['res'][0]
213
214             gradrhoneg1 = torch.sparse_coo_tensor(np.array([idx1, idx2]),
215                                              -1, op_shape, dtype=pdict['bitres'])
216             gradrhoneg2 = torch.sparse_coo_tensor(np.array([idx1, idx1]),
217                                              -1, op_shape, dtype=pdict['bitres'])
218             GRADrhoneg = (gradrhoneg1+gradrhoneg2)/pdict['res'][0]
```

```

219     gradz1 = torch.sparse_coo_tensor(np.array([idx4, idx5]),
220                                     -1/2, op_shape, dtype=pdict['bitres'])
221     gradz2 = torch.sparse_coo_tensor(np.array([idx4, idx6]),
222                                     -1/2, op_shape, dtype=pdict['bitres'])
223     GRADz = (gradz1+gradz2)/pdict['res'][1]
224
225     gradzpos1 = torch.sparse_coo_tensor(np.array([idx4, idx4]),
226                                         -1, op_shape, dtype=pdict['bitres'])
227     gradzpos2 = torch.sparse_coo_tensor(np.array([idx4, idx6]),
228                                         1, op_shape, dtype=pdict['bitres'])
229     GRADzpos = (gradzpos1 + gradzpos2)/pdict['res'][1]
230
231     gradzpos1 = torch.sparse_coo_tensor(np.array([idx4, idx5]),
232                                         -1, op_shape, dtype=pdict['bitres'])
233     gradzpos2 = torch.sparse_coo_tensor(np.array([idx4, idx4]),
234                                         1, op_shape, dtype=pdict['bitres'])
235     GRADzneg = (gradzpos1 + gradzpos2)/pdict['res'][1]
236
237 else:
238
239     init_spmat = lambda shape, res : lil_matrix(shape, dtype = res)
240
241     op_shape = (pdict['N'][0]**2, pdict['N'][1]**2)
242
243     GRADrho = init_spmat(op_shape, pdict['bitres'])
244     GRADrho[LAMBD[1:-1,:], LAMBD[:2,:]] = -1/2
245     GRADrho[LAMBD[1:-1,:], LAMBD[2:,:]]= 1/2
246     GRADrho = GRADrho/pdict['res'][0]
247
248     GRADrhopos = init_spmat(op_shape, pdict['bitres'])
249     GRADrhopos[LAMBD[1:-1,:], LAMBD[1:-1,:]] = -1
250     GRADrhopos[LAMBD[1:-1,:], LAMBD[2:,:]]= 1
251     GRADrhopos = GRADrhopos/pdict['res'][0]
252
253     GRADrhoneg = init_spmat(op_shape, pdict['bitres'])
254     GRADrhoneg[LAMBD[1:-1,:], LAMBD[:2,:]] = -1
255     GRADrhoneg[LAMBD[1:-1,:], LAMBD[1:-1,:]]= 1

```

```

256     GRADrhoneg = GRADrhoneg/pdict['res'][0]
257
258     GRADz= init_sparmat(op_shape, pdict['bitres'])
259     GRADz[LAMBD[:,1:-1], LAMBD[:,:-2]] = -1/2
260     GRADz[LAMBD[:,1:-1], LAMBD[:,2:]] = 1/2
261     GRADz = GRADz/pdict['res'][1]
262
263     GRADzpos= init_sparmat(op_shape, pdict['bitres'])
264     GRADzpos[LAMBD[:,1:-1], LAMBD[:,1:-1]] = -1
265     GRADzpos[LAMBD[:,1:-1], LAMBD[:,2:]] = 1
266     GRADzpos = GRADzpos/pdict['res'][1]
267
268     GRADzneg= init_sparmat(op_shape, pdict['bitres'])
269     GRADzneg[LAMBD[:,1:-1], LAMBD[:,:-2]] = -1
270     GRADzneg[LAMBD[:,1:-1], LAMBD[:,1:-1]] = 1
271     GRADzneg = GRADzneg/pdict['res'][1]
272
273
274     return GRADrho, GRADhopos, GRADrhoneg, GRADz, GRADzpos, GRADzneg
275
276 def build_disp(pdict, LAMBD):
277     """
278     Displacement operators
279     """
280
281     if pdict['coords'] == 'cylindrical':
282         DISPrhopos = lil_matrix((pdict['N'][0]**2, pdict['N'][1]**2), dtype=pdict['bitres'])
283         DISPrhopos[LAMBD[1,:,:], LAMBD[:-1,:,:]] = 1
284
285         DISPrhoneg = lil_matrix((pdict['N'][0]**2, pdict['N'][1]**2), dtype=pdict['bitres'])
286         DISPrhoneg[LAMBD[:-1,:,:], LAMBD[1,:,:]] = 1
287
288         DISPzpos = lil_matrix((pdict['N'][0]**2, pdict['N'][1]**2), dtype=pdict['bitres'])
289         DISPzpos[LAMBD[:,1:], LAMBD[:,:-1]] = 1
290
291         DISPzneg = lil_matrix((pdict['N'][0]**2, pdict['N'][1]**2), dtype=pdict['bitres'])
292         DISPzneg[LAMBD[:, :-1], LAMBD[:, 1:]] = 1

```

```

293
294     return DISPrhopos, DISPrhoneg, DISPzpos, DISPzneg
295
296 def build_LAP(pdct, coord_dict, lap, grad, disp, chi_e):
297     """
298     full laplace operator (dielectric considerations)
299     """
300
301     if pdct['coords'] == 'cylindrical':
302         GRADrho = grad[0]
303         GRADrhopos = grad[1]
304         GRADrhoneg = grad[2]
305         GRADz = grad[3]
306         GRADzpos = grad[4]
307         GRADzneg = grad[5]
308
309         DISPrhopos = disp[0]
310         DISPrhoneg = disp[1]
311         DISPzpos = disp[2]
312         DISPzneg = disp[3]
313
314         chi_e_half = chi_e/2
315
316         CHI1 = spdiags((1/(1+chi_e_half)).T,0, pdct['N'][0]*pdct['N'][1],
317                         pdct['N'][0]*pdct['N'][1], format='lil')
318         CHI2 = spdiags((chi_e_half*coord_dict['coords']['invrho']).T,0,
319                         pdct['N'][0]*pdct['N'][1], pdct['N'][0]*pdct['N'][1], format='lil')
320         DNEG = spdiags(DISPrhoneg.dot(chi_e_half).T,0, pdct['N'][0]*pdct['N'][1],
321                         pdct['N'][0]*pdct['N'][1], format='lil')
322         DPOS = spdiags(DISPrhopos.dot(chi_e_half).T,0, pdct['N'][0]*pdct['N'][1],
323                         pdct['N'][0]*pdct['N'][1], format='lil')
324         ZNEG = spdiags(DISPzneg.dot(chi_e_half).T,0, pdct['N'][0]*pdct['N'][1],
325                         pdct['N'][0]*pdct['N'][1], format='lil')
326         ZPOS = spdiags(DISPzpos.dot(chi_e_half).T,0, pdct['N'][0]*pdct['N'][1],
327                         pdct['N'][0]*pdct['N'][1], format='lil')
328         LAP = lap + CHI1.dot(CHI2.dot(GRADrho)) + (DNEG.dot(GRADrhopos) -
329             DPOS.dot(GRADrhoneg))/pdct['res'][0] + (ZNEG.dot(GRADzpos) -
330             ZPOS.dot(GRADzneg))/pdct['res'][1])

```

```

330
331     #elif pdict['coords'] == 'cartesian':
332
333     return LAP
334
335 def anal_sol(pdict):
336     z_p1 = pdict['front_plate']['zpos']
337     z_p2 = pdict['back_plate']['zpos']
338     d_plates = z_p1 - z_p2
339     V_p1 = pdict['front_plate']['voltage']
340     V_p2 = pdict['back_plate']['voltage']
341     V_diff = V_p1 - V_p2
342     d_opt = pdict['optic']['thickness']
343     d_sub = pdict['optic']['sub_thickness']
344     d_coat = pdict['optic']['coat_thickness']
345     d_air = pdict['cap_params']['d_air']
346     z_opt = pdict['optic']['z_com']
347     p1_2_opt = z_p1 - (d_opt/2.0) - z_opt
348     opt_2_p2 = z_opt - (d_opt/2.0) - z_p2
349     eps_air = pdict['cap_params']['air_eps']
350     eps_sub = pdict['optic']['sub_eps']
351     eps_coat = pdict['optic']['coat_eps']
352     CoA = pdict['cap_params']['cap_div_area']
353     cap_ratio = CoA
354     air_ratio = d_air/eps_air
355     sub_ratio = d_sub/eps_sub
356     coat_ratio = d_coat/eps_coat
357     V_air = cap_ratio*air_ratio*V_diff
358     V_coat = cap_ratio*coat_ratio*V_diff
359     V_sub = cap_ratio*sub_ratio*V_diff
360
361     #E_front = V_diff/(p1_2_opt + opt_2_p2 + (d_opt-d_coat)/eps_sub + d_coat/eps_coat)
362     #E_sub = V_diff/((p1_2_opt + opt_2_p2 + d_coat/eps_coat)*eps_sub + (d_opt-d_coat))
363     #E_coat = V_diff/((p1_2_opt + opt_2_p2 + (d_opt-(d_coat))/eps_sub)*eps_coat +
364                         d_coat)
365     #E_back = E_front
366     z_anal = z_p2+np.array([0, opt_2_p2, (opt_2_p2 + d_sub), (opt_2_p2 + d_sub + d_coat),

```

```

367     (opt_2_p2 + d_sub + d_coat + opt_2_p2)])
368 V_anal = np.array([V_p2, V_p2 + V_air, V_p2 + V_air + V_sub, V_p2 + V_air +
369     V_sub + V_coat, V_p1])
370 anal_dict = {
371     'V_anal' : lambda z: np.interp(z, z_anal, V_anal)
372     }
373 return anal_dict
374
375 def pltxsect(loc_params, coord_dict, V):
376     if loc_params['cross_section_coord'] == 'z':
377         rho_ = coord_dict['coords']['rho'] == np.around(loc_params['rho'],
378             int(np.abs(np.log10(coord_dict['coords']['rho'][1]))))
379         z_ = np.logical_and(coord_dict['coords']['z']<=loc_params['z1_bound'],
380             coord_dict['coords']['z']>=loc_params['z2_bound'])
381         plt.plot(coord_dict['coords']['z'][np.logical_and(rho_,z_)],
382             V[np.logical_and(rho_, z_)])
383     if loc_params['cross_section_coord'] == 'rho':
384         z_ = coord_dict['coords']['z'] == np.around(loc_params['z'],
385             int(np.abs(np.log10(coord_dict['coords']['z'][1]))))
386         rho_ = np.logical_and(coord_dict['coords']['rho']<=loc_params['rho2_bound'],
387             coord_dict['coords']['rho']>=loc_params['rho1_bound'])
388         plt.plot(coord_dict['coords']['rho'][np.logical_and(rho_,z_)],
389             V[np.logical_and(rho_, z_)])

```

5.15.2 set_params.py

```

1 import numpy as np
2 ## Setting parameters
3 pdict ={
4     'coords' : 'cylindrical' , # coordinate system chosen for simulation box
5     'assembly' : 1 , # Establish plate geometry / location and voltage based on assembly
6     'origin' : np.array([0,0]) , # Origin of the simulation space / map
7     'size' : np.array([.04, .04]) , # Size of simulation box [m]
8     'res' : np.array([1,1])*1e-6 , # relative resolution [coord1, coord2]
9     'iters' : 100000 , # total number of time iterations

```

```

10    'iter_step' : 0.1 , # time step
11    'expbc' : False , # Exponential boundary conditions?
12    'bitres' : 'float32' , # matrix element data type ('float32' vs 'float64')
13    'in2m' : .0254 , # frequently used conversion
14    'torch': True
15 }
16
17 pdict['res_exp'] = np.abs(np.log10(pdict['res'])).astype('int')
18 pdict['aspect'] = pdict['size'][0] == pdict['size'][1]
19 pdict['N'] = (pdict['size']*(1/pdict['res']) + 1).astype('int')
20     # number of points sampled 1 dimension of simulation box
21
22 # Sample parameters
23 pdict['optic'] = {
24     "diam" : 1.0*pdict['in2m'],
25     "thickness" : .25*pdict['in2m'],
26     "z_com" : pdict['size'][1]/2,
27     "sub_eps" : 3.82, # dielectric constant for substrate (fused silica)
28     "coat_eps" : 13.436, # dielectric constant for coating material (AlGaAs / GaAs)
29     "coat_thickness" : 9.5e-6
30 }
31
32 pdict['optic']['sub_thickness'] = pdict['optic']['thickness'] - pdict['optic']['coat_thickness']
33
34 # CHOOSING ASSEMBLY CONFIGURATION
35     # Parameters established to characterize the assembly configurations:
36     # Front and back plate dimensions (usually disk diameters)
37     # Central aperture diameter
38     # Plate positioning along the beam axis with respect to simulation size center
39     # Maximum AC voltage sent on respective plates
40
41 maxhva_settings = {
42     "SVR350" : 210, # [Vpk]
43     "TREK2220" : 220, # [Vpk]
44     "TREK5/80" : 1000, # [Vpk]
45     "TREK10/10B-HS" : 1040 # [Vpk]
46 }

```

```

47
48 if pdict['assembly'] == 0 or pdict['assembly'] == 1:
49     # Setting front and back plate params (including spacing between
50     # them and voltage on respective plates)
51     # This assembly has an assortment of 3d printed spacer components
52     pdict['HVA'] = "SVR350"
53     pdict['mount_zdims'] = {
54         "back_ring" : 1e-3, # +/- 2e-4 [m]
55         "sample_holder" : 9e-3, # +/- 2e-4 [m]
56         "electrode_brace": 3e-3 , # +/- 2e-4 [m]
57         "electrode_backing": 2e-3 # +/- 2e-4 [m]
58     }
59     pdict['front_plate'] = {
60         "diam" : 3* pdict['in2m'], # diameter of plate [m]
61         "hole_diam" : 3e-3, # aperture diameter [m]
62         "thickness" : 1.5e-3,
63         "zpos" : pdict['size'][1]/2 + pdict['mount_zdims'][ "sample_holder"]/2,
64             # location of plate surface (com) [m]
65         "voltage" : maxhva_settings[pdict['HVA']] # Voltage on front plate [V]
66     }
67     pdict['back_plate'] = {
68         "diam" : 3*pdict['in2m'],
69         "hole_diam" : 3e-3,
70         "thickness" : 1.5e-3,
71         "zpos" : pdict['size'][1]/2 - pdict['mount_zdims'][ "sample_holder"]/2,
72         "voltage" : - maxhva_settings[pdict['HVA']]
73     }
74 elif pdict['assembly'] == 2 :
75     # Overall thickness was approximately .5 inches with a
76     # .125 inch lip on one end and .125 gap on the other end of sample.
77     # Once the sample was dropped into the mount with the
78     # surface hugging the PVC lip, it was held down with a nylon set screw
79         (with a rubberized tip.)
80     # This plate used is an aluminum rectangular plate
81     # (will incorporate cartesian coordinates into program soon.)
82     pdict['HVA'] = "SVR350"
83     pdict['mount_zdims'] = {

```

```

84     "sample_holder" : .5*pdict['in2m']
85 }
86 if pdict['coords'] == 'cartesian':
87     pdict['front_plate'] = {
88         "diam" : 0.02794, # diameter of plate [m]
89         "hole_diam" : 3e-3, # aperture diameter [m]
90         "thickness" : 1.27e-3,
91         "zpos" : pdict['size'][1]/2 + pdict['mount_zdims']["sample_holder"]/2,
92             # location of plate surface (com) [m]
93         "voltage" : maxhva_settings[pdict['HVA']]//2
94             # Voltage on front plate [V] (MAX value for associated HVA)
95     }
96     pdict['back_plate'] = {
97         "diam" : 0.02794,
98         "hole_diam" : 3e-3,
99         "thickness" : 1.27e-3,
100        "zpos" : pdict['size'][1]/2 - pdict['mount_zdims']["sample_holder"]/2,
101        "voltage" : - maxhva_settings[pdict['HVA']]//2
102    }
103 elif pdict['coords'] == 'cylindrical':
104     pdict['front_plate'] = {
105         "diam" : 0.02794, # diameter of plate [m]
106         "hole_diam" : 3e-3, # aperture diameter [m]
107         "thickness" : 1.27e-3,
108         "zpos" : pdict['size'][1]/2 + pdict['mount_zdims']["sample_holder"]/2,
109             # location of plate surface (com) [m]
110         "voltage" : maxhva_settings[pdict['HVA']]//2 # Voltage on front plate [V]
111     }
112     pdict['back_plate'] = {
113         "diam" : 0.02794,
114         "hole_diam" : 3e-3,
115         "thickness" : 1.27e-3,
116         "zpos" : pdict['size'][1]/2 - pdict['mount_zdims']["sample_holder"]/2,
117         "voltage" : maxhva_settings[pdict['HVA']]//2
118     }
119 elif pdict['assembly'] == 3 :

```

```

121 #Set front and back plate params
122 pdict['HVA'] = "TREK10/10B-HS"
123 pdict['mount_zdims'] = {
124     "total_zthickness" : 25.94e-3 , # holds both sample and both electrodes [m]
125     "sample_holder" : 6.94e-3 # width of lip that separates sample from electrodes [m]
126 }
127 pdict['front_plate'] = {
128     "diam" : 31.5e-3, # diameter of plate [m]
129     "hole_diam" : 3e-3, # aperture diameter [m]
130     "thickness" : 9.66e-3,
131     "zpos" : pdict['size'][1]/2 + (pdict['mount_zdims']["sample_holder"])/2),
132         # location of plate surface (com) [m]
133     "voltage" : maxhva_settings[pdict['HVA']]//2
134         # Voltage on front plate [V]
135 }
136 pdict['back_plate'] = {
137     "diam" : 31.5e-3,
138     "hole_diam" : 3e-3,
139     "thickness" : 9.66e-3,
140     "zpos" : pdict['size'][1]/2 - (pdict['mount_zdims']["sample_holder"])/2),
141     "voltage" : -maxhva_settings[pdict['HVA']]//2
142 }
143
144 elif pdict['assembly'] == 4 :
145     #Set front and back plate params
146     pdict['mount_zdims'] = {
147         "total_zthickness" : 25.94e-3 , # holds both sample and both electrodes [m]
148         "sample_holder" : 6.94e-3 # width of lip that separates sample from electrodes [m]
149     }
150     pdict['front_plate'] = {
151         "diam" : 31.5e-3, # diameter of plate [m]
152         "hole_diam" : 3e-3, # aperture diameter [m]
153         "thickness" : 9.66e-3,
154         "zpos" : pdict['size'][1]/2 + (pdict['mount_zdims']["sample_holder"])/2),
155             # location of plate surface (com) [m]
156         "voltage" : maxhva_settings[pdict['HVA']]//2 # Voltage on front plate [V]
157     }

```

```

158 pdict['back_plate'] = {
159     "diam" : 31.5e-3,
160     "hole_diam" : 3e-3,
161     "thickness" : 9.66e-3,
162     "zpos" : pdict['size'][1]/2 - (pdict['mount_zdims']['sample_holder'])/2),
163     "voltage" : - maxhva_settings[pdict['HVA']]//2
164 }
165
166 pdict['cap_params'] = {
167     "area" : np.pi*((pdict['front_plate']['diam']/2.0)**2),
168     "d_air": (pdict['mount_zdims']['sample_holder']-pdict['optic']['thickness'])/2.0,
169     "air_eps" : 1.0006
170 }
171
172 pdict['cap_params']['cap_div_area'] = (pdict['optic']['sub_eps']*pdict['optic']['coat_eps']*
173 pdict['cap_params']['air_eps])/((2.0*pdict['optic']['sub_eps']*pdict['optic']['coat_eps']*
174 pdict['cap_params']['d_air']) + (pdict['optic']['sub_eps']*pdict['cap_params']['air_eps']*
175 pdict['optic']['coat_thickness']) + (pdict['optic']['coat_eps']*
176 pdict['cap_params']['air_eps']*(pdict['optic']['sub_thickness'])))
177 pdict['cap_params']['capacitance'] = pdict['cap_params']['cap_div_area']*
178 pdict['cap_params']['area']
179
180 # system location params / metadata
181 pdict['loc_params'] = {
182     'center of optic' : {
183         'cross_section_coord' : 'z',
184         'rho' : 0,
185         'z1_bound' : pdict['front_plate']['zpos'],
186         'z2_bound' : pdict['back_plate']['zpos']
187     },
188     'edge of hole' : {
189         'cross_section_coord' : 'z',
190         'rho' : pdict['front_plate']['hole_diam'],
191         'z1_bound' : pdict['front_plate']['zpos'],
192         'z2_bound' : pdict['back_plate']['zpos']
193     },
194     'edge of optic' : {

```

```
195     'cross_section_coord' : 'z',
196     'rho' : pdict['optic']['diam'] /2,
197     'z1_bound' : pdict['front_plate']['zpos'],
198     'z2_bound' : pdict['back_plate']['zpos']
199 },
200 'edge of plate' : {
201     'cross_section_coord' : 'z',
202     'rho' : pdict['front_plate']['diam']/2,
203     'z1_bound' : pdict['front_plate']['zpos'],
204     'z2_bound' : pdict['back_plate']['zpos']
205 },
206 'halfway out on optic' : {
207     'cross_section_coord' : 'z',
208     'rho' : pdict['optic']['diam']/4,
209     'z1_bound' : pdict['front_plate']['zpos'],
210     'z2_bound' : pdict['back_plate']['zpos']
211 },
212 'front of plate' : {
213     'cross_section_coord' : 'rho',
214     'rho1_bound' : 0,
215     'rho2_bound' : pdict['size'],
216     'z' : pdict['back_plate']['zpos']
217 },
218 'front of optic' : {
219     'cross_section_coord' : 'rho',
220     'rho1_bound' : 0,
221     'rho2_bound' : pdict['size'],
222     'z' : pdict['optic']['z_com'] + pdict['optic']['thickness']/2
223 },
224 'middle of optic' : {
225     'cross_section_coord' : 'rho',
226     'rho1_bound' : 0,
227     'rho2_bound' : pdict['size'],
228     'z' : pdict['optic']['z_com']
229 },
230 'back of optic' : {
231     'cross_section_coord' : 'rho',
```

```

232     'rho1_bound' : 0,
233     'rho2_bound' : pdict['size'],
234     'z' : pdict['optic']['z_com'] - pdict['optic']['thickness']/2
235 },
236 'back plate' : {
237     'cross_section_coord' : 'rho',
238     'rho1_bound' : 0,
239     'rho2_bound' : pdict['size'],
240     'z' : pdict['back_plate']['zpos']
241 }
242 }
```

5.15.3 run.py

The numerical recipe is written up fabulously in Chapter 20 of [50] for any inquiring minds.

```

1 import set_params
2 import laplace
3 import matplotlib.pyplot as plt
4 import numpy as np
5 plt.style.use('stylelib/surftex')
6 #fig_exp_dirama = ".../.../.../dissertation/figs/ALGAAS/"
7 from matplotlib import cm
8 from matplotlib import rcParams
9 import time
10 import torch
```

Params Import

```

1 pdict = set_params.pdict
2 pdict
```

```

1 {'coords': 'cylindrical',
2  'assembly': 1,
3  'origin': array([0, 0]),
4  'size': array([0.04, 0.04]),
5  'res': array([0.0001, 0.0001]),
```

```
6   'iters': 100000,
7   'iter_step': 0.1,
8   'expbc': False,
9   'bitres': 'float32',
10  'in2m': 0.0254,
11  'res_exp': array([4, 4]),
12  'aspect': True,
13  'N': array([401, 401]),
14  'optic': {'diam': 0.0254,
15    'thickness': 0.00635,
16    'z_com': 0.02,
17    'sub_eps': 3.82,
18    'coat_eps': 13.436,
19    'coat_thickness': 9.5e-06,
20    'sub_thickness': 0.0063405},
21  'HVA': 'SVR350',
22  'mount_zdims': {'back_ring': 0.001,
23    'sample_holder': 0.009,
24    'electrode_brace': 0.003,
25    'electrode_backing': 0.002},
26  'front_plate': {'diam': 0.0761999999999999,
27    'hole_diam': 0.003,
28    'thickness': 0.0015,
29    'zpos': 0.0245,
30    'voltage': 210},
31  'back_plate': {'diam': 0.0761999999999999,
32    'hole_diam': 0.003,
33    'thickness': 0.0015,
34    'zpos': 0.0155,
35    'voltage': -210},
36  'cap_params': {'area': 0.004560367311877479,
37    'd_air': 0.001324999999999998,
38    'air_eps': 1.0006,
39    'cap_div_area': 232.07592015102685,
40    'capacitance': 1.0583514401306306},
41  'loc_params': {'center of optic': {'cross_section_coord': 'z',
42    'rho': 0,
```

```
43     'z1_bound': 0.0245,
44     'z2_bound': 0.0155},
45     'edge of hole': {'cross_section_coord': 'z',
46     'rho': 0.003,
47     'z1_bound': 0.0245,
48     'z2_bound': 0.0155},
49     'edge of optic': {'cross_section_coord': 'z',
50     'rho': 0.0127,
51     'z1_bound': 0.0245,
52     'z2_bound': 0.0155},
53     'edge of plate': {'cross_section_coord': 'z',
54     'rho': 0.03809999999999995,
55     'z1_bound': 0.0245,
56     'z2_bound': 0.0155},
57     'halfway out on optic': {'cross_section_coord': 'z',
58     'rho': 0.00635,
59     'z1_bound': 0.0245,
60     'z2_bound': 0.0155},
61     'front of plate': {'cross_section_coord': 'rho',
62     'rho1_bound': 0,
63     'rho2_bound': array([0.04, 0.04]),
64     'z': 0.0155},
65     'front of optic': {'cross_section_coord': 'rho',
66     'rho1_bound': 0,
67     'rho2_bound': array([0.04, 0.04]),
68     'z': 0.023175},
69     'middle of optic': {'cross_section_coord': 'rho',
70     'rho1_bound': 0,
71     'rho2_bound': array([0.04, 0.04]),
72     'z': 0.02},
73     'back of optic': {'cross_section_coord': 'rho',
74     'rho1_bound': 0,
75     'rho2_bound': array([0.04, 0.04]),
76     'z': 0.016825},
77     'back plate': {'cross_section_coord': 'rho',
78     'rho1_bound': 0,
79     'rho2_bound': array([0.04, 0.04]),
```

80 'z': 0.0155}}}}

Initializing coordinates / simulation space

```

1 # initialize coordinates
2 coord_dict = laplace.init_coords(pdict)
3
4 # Imposing a square simulation space
5 N = pdict['N'][0]
6
7 # coord vecs
8 rho = coord_dict['coords']['rho']
9 z = coord_dict['coords']['z']
10 inv_rho = coord_dict['coords']['inv_rho']
11
12 #indices
13 irho = coord_dict['indices']['rho']
14 iz = coord_dict['indices']['z']

```

Potential map initialization (V) with Dielectric tensor initialization (chi_e)

```

1 fV = laplace.anal_sol(pdict)
1 plt.plot(coord_dict['indices']['z']*1e-4,fV['V_anal'](coord_dict['indices']['z']*1e-4))

```

```

1 # intialize potential map, electric susceptibility, and LAMBD operator
2 V = laplace.init_V(N)
3 chi_e = laplace.init_V(N)
4 chi_e_sub = pdict['optic']['sub_eps']-1
5 chi_e_coat = pdict['optic']['coat_eps']-1
6 LAMBD = laplace.build_lambd(irho, iz, N)

```

```

1 # Translating (Dirichlet) boundary conditions to sim
2
3 # Initial value
4
5 ## (For faster convergence) setting edge values
6
7 ### Edge locations
8 r_max = (rho == max(rho))
9 r_min = (rho == min(rho))

```

```

10 z_max = (z == max(z))
11 z_min = (z == min(z))
12
13 # Boundary values
14
15 ## Plate potentials
16 fp = 'front_plate'
17 bp = 'back_plate'
18 loc_fp = np.logical_and(np.logical_and(rho>=pdict[fp]['hole_diam']/2,
19 rho<=pdict[fp]['diam']/2),z == pdict[fp]['zpos'])
20 loc_bp = np.logical_and(np.logical_and(rho>=pdict[bp]['hole_diam']/2,
21 rho<=pdict[bp]['diam']/2),z == pdict[bp]['zpos'])
22 #bc_fp = laplace.BC_dict([[pdict[fp]['hole_diam']/2, pdict[fp]['diam']/2],
23 pdict[fp]['zpos']],pdict[fp]['voltage'],fp, LAMBD)
24 #bc_bp = laplace.BC_dict([[pdict[bp]['hole_diam']/2, pdict[bp]['diam']/2],
25 pdict[bp]['zpos']],pdict[fp]['voltage'],bp, LAMBD)
26
27 # Exponential boundary conditions
28 exp_rend = rho==(max(rho)-pdict['res'][0])
29
30 exp_z0 = z==(min(z)+pdict['res'][1])
31
32 exp_zend = z==(max(z)-pdict['res'][1])
33
34 # Setting sample dielectric
35 loc_sub = np.logical_and(np.abs(z - pdict['optic']['z_com']) <
36 np.round((pdict['optic']['thickness']/2),pdict['res_exp'][1]),
37 (rho<np.round((pdict['optic']['diam']/2),pdict['res_exp'][0])))
38 loc_coat1 = np.logical_and((z==np.round(pdict['loc_params']['front of optic']['z'],
39 pdict['res_exp'][1])), (rho < np.round((pdict['optic']['diam']/2),
40 pdict['res_exp'][0])))
41 loc_coat2 = np.logical_and((z==np.round((pdict['loc_params']['front of optic']['z']-
42 pdict['res'][1]),pdict['res_exp'][1])), (rho < (np.round(pdict['optic']['diam']/2,
43 pdict['res_exp'][0]))))

```

```

1 ## Initialize BCs
2
3 ##### Set susceptibility

```

```

4 chi_e[loc_sub] = chi_e_sub
5 chi_e[loc_coat1] = chi_e_coat
6 chi_e[loc_coat2] = chi_e_coat
7
8 ## Electro-static conditions
9
10 #### Electrode plates
11 V[loc_fp] = pdict[fp]['voltage']
12 V[loc_bp] = pdict[bp]['voltage']
13
14 #### Boundary values
15
16 #### Edge for faster convergance
17 V[z_min] = pdict['back_plate']['voltage']
18 V[z_max] = pdict['front_plate']['voltage']
19 V[r_max] = np.interp(np.arange(0,pdict['N'][0]),
20     np.array([0,pdict['N'][0]-1]), np.array([pdict['back_plate']['voltage'],
21     pdict['front_plate']['voltage']])))
22
23 #### Exponential (Dirichlet) boundary conditions
24 V_exp = lambda V_0, R_0, V , R : V_0 + np.exp(-R/R_0)*(V - V_0)
25 V_char = 0
26 R_char = 1.0
27
28 #rho=rho_max
29 V[r_max] = V_exp(V_char, R_char, V[exp_rend], pdict['res'][0])
30 #z=z_min
31 V[z_min] = V_exp(V_char, R_char, V[exp_z0], pdict['res'][1])
32 #z=z_max
33 V[z_max] = V_exp(V_char, R_char, V[exp_zend], pdict['res'][1])

```

```

1 #### Build operators
2 lap = laplace.build_lap(pdict, LAMBD, irho)
3 grad = laplace.build_grad(pdict, LAMBD)
4 disp = laplace.build_disp(pdict, LAMBD)
5 LAP = laplace.build_LAP(pdict, coord_dict, lap, grad, disp, chi_e)

```

```
1 t = time.time()
```

```

2 ##### run sim
3 for itr in range(0, pdict['iters']):
4
5     V = V + (pdict['res'][0]*pdict['res'][0]*pdict['iter_step']*LAP.dot(V))
6
7     ## Re-applying exponential condition
8     #rho=rho_max
9     V[r_max] = V_exp(V_char, R_char, V[exp_rend], pdict['res'][0])
10    #z=z_min
11    V[z_min] = V_exp(V_char, R_char, V[exp_z0], pdict['res'][1])
12    #z=z_max
13    V[z_max] = V_exp(V_char, R_char, V[exp_zend], pdict['res'][1])
14
15    #### Re-apply Electro-static condition
16    V[loc_fp] = pdict[fp]['voltage']
17    V[loc_bp] = pdict[bp]['voltage']
18 elapsed = time.time() - t
19 print(elapsed)

```

1 84.81229615211487

```

1 fig = plt.figure(figsize = (18.5,21))
2 ax = plt.axes(projection='3d')
3 surf = ax.plot_surface(rho.reshape(N,N), z.reshape(N,N),
4                         V.reshape(N,N),rstride=1,cstride=1,cmap=cm.inferno,alpha=1,
5                         linewidth=10,rasterized=True)
6 fig.tight_layout()
7 ax.view_init(20,210)
8 ax.set_xlabel('r [m]')
9 ax.set_ylabel('z [m]')
10 ax.set_zlabel('[V]')
11 fig.colorbar(surf, shrink=0.4, aspect=20, pad=-0.025)
12 axes_width = fig.get_size_inches()[1]*(fig.subplotpars.right-float(fig.subplotpars.left))
13 right = 1.095
14 left = -.15
15 fig.subplots_adjust(left=left,right=right)
16 fig.set_size_inches((fig.get_size_inches()[0],axes_width/(right-left)))
17 ax.tick_params(axis='both', pad=15)
18 axes_height = fig.get_size_inches()[1]*(fig.subplotpars.top-float(fig.subplotpars.bottom))

```

```

19 top = 1.15
20 bottom=-.09
21 fig.subplots_adjust(top=top,bottom=bottom)
22 fig.set_size_inches((fig.get_size_inches()[0],axes.height/(top-bottom)))
23 #fig.savefig(fig_exp_dir + 'assembly1_sim.pdf', dpi=300, format='pdf')

```

```

1 # Plotting potential and field profiles
2 laplace.pltxsect(pdict['loc_params']['halfway out on optic'], coord_dict, V)
3 plt.plot(coord_dict['indices']['z']*1e-4,fV['V_anal'](coord_dict['indices']['z']*1e-4))
4
5 ## Comparison to analytical solutions

```

5.16 Calibration code

5.16.0.1 Import packages

If it fails the first time try installing the following to a separate conda enviornment: /pydependencies/eo_calibrate.yml

```

1 import glob
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from matplotlib import rc
5 import os
6 import h5py
7
8 plt_style_dir = 'pydependencies/stylelib/'
9
10 if os.path.isdir(plt_style_dir) == True:
11     plt.style.use(plt_style_dir + 'pptsize') # This just sets a python figure style (you can adjust it to your

```

5.16.0.2 Frequently used functions:

```

1 def concat_vecs(directory):
2     """
3         Takes a directory filled with spectra measurements (from SR785) to be

```

```

4 concatenated to a single vector. The output of the function is (frequency
5 vector, amplitude vector)
6 """
7 txtcounter = len(glob.glob1(directory,"*.TXT"))
8 freq = np.zeros((801,txtcounter))
9 freq1 = np.zeros((801,txtcounter))
10 vpk = np.zeros((801,txtcounter))
11
12 columns = range(0,txtcounter)
13 fff = 0
14 vpkn = 0
15 ## import and measurements
16 for i in columns:
17     data = np.loadtxt(directory + str(i).zfill(2) + '.TXT')
18     freq[:,i] = data[:,0]
19     vpk[:,i] = data[:,1]
20     if i == columns[0]:
21         fff = freq[:,i]
22         vpkn = vpk[:,i]
23     elif i == columns[-1]:
24         fff = np.append(fff,freq[:,i])
25         vpkn=np.append(vpkn,vpk[:,i])
26     else:
27         fff = np.append(fff, freq[:,i][:-1])
28         vpkn = np.append(vpkn, vpk[:,i][:-1])
29 return fff, vpkn
30
31 def gen_concat_vecs(directory):
32 """
33 Takes a directory filled with spectra measurements (from SR785) to be
34 concatenated to a single vector. The output of the function is (frequency
35 vector, amplitude vector)
36 """
37 txtcounter = len(glob.glob1(directory,"*.TXT"))
38 columns = range(0,txtcounter)
39
40 data = np.loadtxt(directory + str(0).zfill(2) + '.TXT')
41 master_freq = data[:,0]
42 master_data = data[:,1]
43
44 for i in columns:
45     if i < columns[-1]:
46         data = np.loadtxt(directory + str(i).zfill(2) + '.TXT')
47         data1 = np.loadtxt(directory + str(i+1).zfill(2) + '.TXT')
48         xy, xind, yind = np.intersect1d(data[:,0], data1[:,0], \

```

```

49         return_indices=True)
50
51     if sum(yind>400) != 0:
52         xy2, xind2, yind2 = np.intersect1d(master_freq, data1[390:,0], \
53         return_indices=True)
54         master_freq = np.append(master_freq[:xind2[0]-11], \
55         data1[390:,0])
56         master_data = np.append(master_data[:xind2[0]-11], \
57         data1[390:,1])
58
59     else:
60         if len(xy) != 0:
61             master_freq = np.append(master_freq, data1[yind[-1]+1:,0])
62             master_data = np.append(master_data, data1[yind[-1]+1:,1])
63         else:
64             master_freq = np.append(master_freq, data1[:,0])
65             master_data = np.append(master_data, data1[:,1])
66
67     return master_freq, master_data
68
69 def transfer_function(amplitude, phase):
70     """
71     Takes frequency response data (amplitude and phase) combines it into a
72     transfer function :  $Ae^{(i\phi)}$ 
73     """
74     return 10**((amplitude/20)* np.exp(1j*(phase/180)*np.pi))
75
76 def phase_wrap(phase_array, type='deg'):
77     """
78     Wraps phase from  $-180 \rightarrow +180$  degrees if type == 'rad' then it wraps from
79      $-\pi \rightarrow \pi$ 
80     """
81     if type == 'deg':
82         fin_phase_array = (phase_array + 180) % (2 * 180) - 180
83     if type == 'rad':
84         fin_phase_array = (phase_array + np.pi) % (2 * np.pi) - np.pi
85     return fin_phase_array
86
87 def function_transfer(freq, tf_in):
88     """
89     Converts transfer function back to amplitude and phase data (frequency, amplitude [dB], phase [deg])
90     """
91     db = 20*np.log10(np.abs(tf_in))
92     deg = np.angle(tf_in, deg=True)
93     return freq, db, deg

```

```
94
95 def tf_import(tf_path):
96     """
97     Takes a directory containing amplitude and phase data (dB and deg) and
98     imports the data and outputs (frequency, amplitude, phase)
99     """
100    db = np.loadtxt(tf_path + 'db.TXT')
101    deg = np.loadtxt(tf_path + 'deg.TXT')
102    ff = db[:,0]
103    return ff, db[:,1], deg[:,1]
104
105 def tf_interpolate(new_freq, tf_tuple):
106     """
107     """
108     """
109     new_db = np.interp(new_freq, tf_tuple[0], tf_tuple[1])
110     new_deg = np.interp(new_freq, tf_tuple[0], tf_tuple[2])
111     return new_freq, new_db, new_deg
112
113 def bode_plt(tf_tuple, save_path, lbl, title, ylbl='dB'):
114     ff = tf_tuple[0]
115     db = tf_tuple[1]
116     deg = tf_tuple[2]
117     bode_fig = plt.figure()
118     plt.subplot(211)
119     if not ylbl=='dB':
120         plt.loglog(ff, db, label=lbl)
121     else:
122         plt.semilogx(ff, db, label=lbl)
123     plt.xlim(ff[0], ff[-1])
124     plt.ylabel(ylbl)
125     plt.legend()
126     plt.title(title.replace('_', '\_'))
127     plt.subplot(212)
128     plt.semilogx(ff, deg, label=lbl)
129     plt.xlim(ff[0], ff[-1])
130     plt.legend()
131     plt.xlabel('Frequency [Hz]')
132     plt.ylabel('phase [deg]')
133     plt.savefig(save_path + '/' + title + '.png', dpi=300, bbox_inches='tight')
134     plt.close()
135     return bode_fig
136
137 def h5_import(dir):
138     return h5py.File(dir + '/data.hdf5', 'r')
```

```
139
140 def printname(name):
141     print(name)
142
143 def h5_peek(h5_file):
144     if type(h5_file) == str:
145         h5_data = h5_import(h5_file)
146     else:
147         h5_data = h5_file
148
149     return h5_data.visit(printname)
150
151 def qkh5plt(h5_file,meas,LBL,AXIS,YAX='log',LGND_SIZE=30,PEEK=False):
152     """
153     Plotting tool that allows you to quickly plot any one of the traces from an
154     h5 file.
155
156     h5_file : Can be an already open h5 file or a directory to an h5 file
157
158     meas : The measurement you wish to select from the options in the h5 file
159
160     axis : needs to inherit axis from already established figure
161
162     LBL : Label we want to tag onto the requested dataset
163
164     YAX : can swap between a logarithmic and linear yaxis
165
166     LGND_SIZE : size of legend font
167     """
168
169     if type(h5_file) == str:
170         h5_data = h5_import(h5_file)
171     else:
172         h5_data = h5_file
173
174     if PEEK == True:
175         h5_peek(h5_file)
176
177     if YAX == 'log':
178         AXIS.loglog(h5_data['freq'][:],h5_data[meas][:],label=LBL)
179     elif YAX == 'lin':
180         AXIS.semilogx(h5_data['freq'][:],h5_data[meas][:],label=LBL)
181
182     AXIS.legend(prop={'size':LGND_SIZE})
183     PLT.XLIM(h5_data['freq'][0],h5_data['freq'][-1])
```

5.16.0.3 Input variables

```

1 meas_data_dir = 'measurements/swept/algaas/08_13_2021/meas1/' # directory where the uncalibrated data
2 date = '08_13_2021' # date when measurement was taken ("mm_dd_yyyy")
3 final_dir = 'results/' # directory where the final data will live
4 meas_type = 'swept' # type of measurement taken tag (i.e. noise, swept)
5 spectra_type = 'pk' # spectra type (i.e. pk, rms)
6 sample = 'algaas' # sample tag (i.e. algaas, atfilms, sio2tao5, etc.)
7 xtradir = 'meas1' # this label helps distinguish between measurements taken in a given day
8 cav_length = .165 # recorded length of cavity
9 inp_voltage_swept = 4.65 # voltage sent from SR785 to HVA connected to electrodes
10 plot_saving = False # generate and save .png files for intermediate calibration functions
11 model = False # boolean that decides whether or not the model estimate should be plotted with calibrated

1 labl = date + '_' + meas_type + '_' + spectra_type + '_' + sample # label of the directory containing all
2 if xtradir != 'none': # adjusted label if an extra directory was used
3     labl = date + '_' + meas_type + '_' + spectra_type + '_' + xtradir + \
4         '_' + sample
5 new_final_dir = final_dir + '/' + labl
6
7
8 if os.path.isdir(new_final_dir) == False: # generates the directory containing the results if it doesn't already
9     os.mkdir(new_final_dir)

```

5.16.0.4 Data import

```

1 #HVA and OLG directory search
2 HVA_common_dir = 'measurements/HVASVR_tf/' # HVA and OLG directories
3 OLG_common_dir = 'measurements/OLG/'
4 HVA_dir = HVA_common_dir + 'HVACH3_plus_pomona/' + date + '/'
5 #OLG_dir = OLG_common_dir + date + '/'
6 OLG_dir = OLG_common_dir + sample + '/' + date + '/'
7
8
9 if xtradir != 'none' and meas_type != 'noise':
10     HVA_dir = HVA_dir + xtradir + '/'
11     OLG_dir = OLG_dir + xtradir + '/'
12
13 HVA = tf_import(HVA_dir) # import the HVA and OLG data

```

```

14 OLG = tf_import(OLG_dir)
15 #If the data is a swept frequency measurement
16 if meas.type == 'swept': # import HVA CH1 transfer function data for transfer function measurement
17     HVA_CH1_dir = HVA_common_dir + 'HVACH1/' + date + '/'
18     #HVA_CH1_dir = HVA_common_dir + 'HVACH1_w_LPF/' + date + '/'
19     if xtradir != 'none':
20         HVA_CH1_dir = HVA_CH1_dir + xtradir + '/'
21
22 electrode_type = 'disk' # import low pass measurement from resistor / electrode capacitance
23 if sample == 'sio2ta2o5' or sample == 'atfilms':
24     Electcap_dir = 'measurements/electrode_capacitence/' + \
25         electrode_type + '/' + sample + '/03_29_2021/'
26 if sample == 'atfilms':
27     Electcap_dir = 'measurements/electrode_capacitence/' + \
28         electrode_type + '/' + sample + '/06_04_2021/'
29 if sample == 'algaas':
30     Electcap_dir = 'measurements/electrode_capacitence/' + \
31         electrode_type + '/' + sample + '/03_10_2021/'
32
33 meas_swept = tf_import(meas_data_dir) # import transfer function measurement
34 if plot_saving == True: # plot uncalibrated transfer function measurement if requested
35     bode_plt(meas_swept, new_final_dir, date.replace("_", "\_"), \
36             'Pockels_effect_frequency_response_uncalibrated_dB')
37
38 swept_tf = transfer_function(meas_swept[1], meas_swept[2]) # combine amplitude and phase
39
40 HVA_CH1 = tf_import(HVA_CH1_dir) # HVA CH1 import and interpolation (interpolate to transfer
41 HVA_CH1_inter = tf_interpolate(meas_swept[0], HVA_CH1)
42 HVA_CH1_tf = transfer_function(HVA_CH1_inter[1], HVA_CH1_inter[2])
43
44 #Electrode capacitence transfer function import and interpolation
45 ECAP = tf_import(Electcap_dir) # Import and interpolate LPF measurement (part of frequency depe
46 ECAP_inter = tf_interpolate(meas_swept[0], ECAP)
47 ECAP_tf = transfer_function(ECAP_inter[1], ECAP_inter[2])
48
49 #interpolate related tfs
50 HVA_inter = tf_interpolate(meas_swept[0], HVA) # HVA CH3 and OLG interpolation
51 OLG_inter = tf_interpolate(meas_swept[0], OLG)
52
53 else: # if the measurement is not a transfer function (spectra measurement)
54     spectra = gen_concat_vecs(meas_data_dir) # changed from concat_vecs to gen_concat_vecs (07-25-20
55     #interpolate related tfs
56     HVA_inter = tf_interpolate(spectra[0], HVA) # HVA CH3 and OLG interpolation
57     OLG_inter = tf_interpolate(spectra[0], OLG)
58     if plot_saving == True:

```

```

59      #Spectra plotting
60      plt.loglog(spectra[0], spectra[1], label=lbl.replace("_","\\_"))
61      plt.legend()
62      plt.xlabel('frequency [Hz]')
63      #plt.xlim([spectra[0][0],spectra[0][-1]])
64      if spectra_type == 'pk':
65          plt.ylabel('$$V_{\mathrm{pk}}$$')
66      elif spectra_type == 'rms':
67          plt.ylabel('$$V_{\mathrm{rms}}$$')
68      plt.savefig(new_final_dir + '/v_spectra_' + lbl + '.png',dpi=300, \
69      bbox_inches='tight')
70      plt.close()
71
72
73 if plot_saving == True: # plot and HVACH3 and OLG if requested
74     #HVA plotting
75     bode=plt(HVA, new_final_dir, 'HVA.75\_total\_gain', 'HVACH3+pomona')
76     #OLG plotting
77     bode=plt(OLG, new_final_dir , date.replace('_', '\\_'), 'OLG' )

```

5.16.0.5 Build calibration function

```

1 HVA_tf = transfer_function(HVA_inter[1], HVA_inter[2]) # combine amplitude and phase of HVACH3 and
2 OLG_tf = transfer_function(OLG_inter[1], OLG_inter[2])
3
4 if meas_type == 'swept':
5     volt_divider = False # Voltage divider if you want a smaller normalization factor
6     if volt_divider == True: # Voltage divider with r_1 as the first resistor and r_2 as the resistor connected
7         r_1 = 100000
8         r_2 = 50
9         pom_vdivider = (r_2)/(r_1+r_2)
10
11     swept_tf = swept_tf*pom_vdivider
12 else:
13     pom_vdivider = 1
14
15 stf_unnorm = swept_tf*inp_voltage_swept # Unnormalized transfer function measurement
16
17 s_unnorm = [meas_swep[0], abs(stf_unnorm), np.angle(stf_unnorm, \
18 deg=True)] # Unnormalized transfer function in triad format
19
20 if plot_saving == True: # Plot voltage spectra for transfer function measurement if requested

```

```

21      bode=plt(s_unnorm, new_final_dir, date.replace(' ','\_'), \
22      'Pockels_effect_frequency_response_vspectra', \
23      ylbl='$V\_mathrm{pk}$')
24
25  v_direct = inp_voltage_swept*HVA_CH1_tf*ECAP_tf # This is the voltage directly across the coating
26  vdirec = [meas_swep[0], abs(v_direct), np.angle(v_direct,deg=True)] # Frequency dependent injection
27  if plot_saving==True: # Plot frequency dependent injection voltage if requested
28      bode=plt(vdirec, new_final_dir, date.replace(' ','\_'), \
29      'Potential difference across electrodes', ylbl='$V\_mathrm{pk}$')
30
31
32  #laserV2Hz = 2.0e6
33  #laserV2Hz = 1.4706e6 # measurement from elog 831 (05-24-2021)
34  laserV2Hz = 1.748e6 # Laser PZT response acquired from PDH measurement
35  HzpV = HVA_tf*laserV2Hz # Actuation function  $A(f) = A_1(f)*A_2(f)$ 
36
37  CLG = 1/(1-OLG_tf) # Closed loop gain
38  CAL = OLG_tf*CLG # Loop gain calibration factor
39  #CALVpHz=CAL/HzpV # Calibrated voltage to frequency
40  CALHzpV=HzpV/CAL # Calibration factor using  $A(f)$  and  $OLG(f)$ 
41
42  if meas_type == 'swept': # Calibrate data
43      freq_noise = CALHzpV*stf_unnorm
44  else:
45      freq_noise = abs(CALHzpV)*spectra[1]
46
47  if plot_saving == True and meas_type != 'swept': # if plotting spectra measurement this is plotting and
48      plt.loglog(spectra[0],freq_noise, label= date.replace(' ','\_'), \
49      linewidth=3)
50      #plt.xlim([spectra[0][0], spectra[0][-1]])
51      plt.legend()
52      plt.xlabel('Frequency [Hz]')
53      plt.ylabel('$$\mathit{Hz}\mathit{pk}$$')
54      plt.title("Laser frequency noise from measured voltage noise")
55      plt.savefig(new_final_dir + '/Hz' + '_spectra' + labl + '.png', \
56      dpi=300, bbox_inches='tight')
57      plt.close()

```

5.16.0.6 Cavity params

```

1 c = 299792458 # Cavity parameters
2 lamb = 1.064e-6

```

```

3 nu = c/lamb
4 Lcav = cav_length

```

5.16.0.7 Calibrate voltage to displacement

```

1 #Displacement spect
2 displac_spect = freq_noise*Lcav/nu # Calibrate to displacement spectra
3 if meas_type == 'swept':
4     disp_spect_norm = displac_spect/v_direct # Displacement spectra normalized by the frequency dependent
5
6 model_freq = 10000 # Model estimate
7 marty_estimate = 3.8e-16 # mpk/[V*m]
8 Efield_strength_estimate = 4648 # [V/m] (changed from 6350 to 4648 on 07-13-2021)
9
10 #Displacement spectra
11 if meas_type == 'swept': # Organizing and plotting displacement spectra
12     displac_spect_unnorm = [meas_swept[0], abs(displac_spect), \
13     np.angle(displac_spect, deg=True)]
14     displac_spect_norm = [meas_swept[0], abs(disp_spect_norm), \
15     np.angle(disp_spect_norm, deg=True)]
16     final_fig = bode_plt(displac_spect_unnorm, new_final_dir, \
17     date.replace('_', '\_'), \
18     'Displacement spectra for AlGaAs Pockels effect measurement', \
19     ylbl='Displacement ['$\mathrm{m}$-$\mathrm{pk}$'])
20 else:
21     final_fig = plt.loglog(spectra[0],abs(displac_spect),color='m', \
22     label=lbl, linewidth=3)
23     plt.xlabel('frequency [Hz]')
24     #plt.xlim([spectra[0][0], spectra[0][-1]])
25     #plt.ylabel('$V_{\mathrm{m}}$'.format(spectra_type))
26
27 if model == True and meas_type != 'swept': # If model estimate is requested, will plot model estimate with
28     plt.axhline(y=marty_estimate*Efield_strength_estimate,linestyle='--', \
29     color='k', label='Marty estimate')
30     #plt.xlim([spectra[0][0], spectra[0][-1]])
31     plt.legend()
32     plt.xlabel('Frequency [Hz]')
33     if spectra_type == 'pk':
34         plt.ylabel('Displacement ['$\mathrm{m}$-$\mathrm{pk}$'])
35     if spectra_type == 'rms':
36         plt.ylabel('Displacement ['$\mathrm{m}$-$\mathrm{rms}$'])
37     plt.title("Displacement spectra for AlGaAs Pockels effect measurement")

```

```
38     plt.savefig(new_final_dir + '/' + 'pockels_displacement_spectra' + \
39             'labl + '.png', dpi=300, bbox_inches='tight')
```

5.16.0.8 Save raw data, calibration functions, calibrated displacement function, and other metadata

```
1 with h5py.File(new_final_dir + "/data.hdf5", "a") as f: # Store raw / calibrated data along with metadata
2
3     #Raw data
4     raw = f.create_group("raw")
5     hva_save = f.create_group("raw/hva") # where hva data will be saved
6     hva_save_ch3 = f.create_group("raw/hva/ch3+ponoma")
7     if meas_type == 'swept':
8         freq = f.create_dataset("freq", data=meas_swept[0]) # common frequency vector
9         hva_save_ch1 = f.create_group("raw/hva/ch1")
10        hva_save_ch1.attrs['dir'] = HVA_CH1_dir
11        pomona_vdiv=f.create_dataset("pomona_vdivider",data=pom_vdivider) # Voltage divider factor
12        trans_func = f.create_group("raw/meas_freq_resp")
13        meas_db = f.create_dataset("raw/meas_freq_resp/db", \
14            data=meas_swept[1])
15        meas_deg = f.create_dataset("raw/meas_freq_resp/deg", \
16            data=meas_swept[2])
17        trans_func.attrs['dir'] = meas_data_dir
18        direc_volt = f.create_group("raw/vdirect") # the Vpk voltage and phase information of the signal
19    else:
20        freq = f.create_dataset("freq", data=spectra[0]) # common frequency vector
21        vdata_save = f.create_dataset("raw/v_spect", data=spectra[1])
22        vdata_save.attrs['units'] = spectra_type
23        vdata_save.attrs['dir'] = meas_data_dir # where error signal spectra will be saved
24        cav_length = f.create_dataset("cav_length", data=Lcav)
25        laser_freq = f.create_dataset("laser_freq", data=nu)
26        laserPZTresp = f.create_dataset("laserV2Hz", data=laserV2Hz )
27        hva_save_ch3.attrs['dir'] = HVA_dir
28        olg_save = f.create_group("raw/olg") # where olg data will be saved
29        cal_save = f.create_group("raw/cal") # easily accessible loop calibration factor data
30        olg_save.attrs['dir'] = OLG_dir
31        if meas_type == 'swept':
32            hvadb_save_ch1 = f.create_dataset("raw/hva/ch1/db",data=HVA_CH1[1])
33            hvadeg_save_ch1 = f.create_dataset("raw/hva/ch1/deg", \
34                data=HVA_CH1[2])
35            vdirec_db = f.create_dataset("raw/vdirect/db", data=vdirec[1])
36            vdirec_deg = f.create_dataset("raw/vdirect/deg", data=vdirec[2])
```

```
37 hvadb_save_ch3 = f.create_dataset("raw/hva/ch3+pomona/db", \
38 data=HVA_inter[1])
39 hvadeg_save_ch3 = f.create_dataset("raw/hva/ch3+pomona/deg", \
40 data=HVA_inter[2])
41 olgdb_save = f.create_dataset("raw/olg/db", data=OLG_inter[1])
42 olgdeg_save = f.create_dataset("raw/olg/deg", data=OLG_inter[2])
43 calgain_save = f.create_dataset("raw/cal/gain", data=abs(CAL))
44 caldeg_save = f.create_dataset("raw/cal/deg", data=np.angle(CAL, \
45 deg=True))
46
47
48 #Calibrated data
49 calibra = f.create_group("calibrated")
50 hvatf_save = f.create_group("calibrated/hva")
51 hvach3tf_save = f.create_dataset("calibrated/hva/ch3+pomona", \
52 data=HVA_tf)
53 olgtf_save = f.create_dataset("calibrated/olg", data=OLG_tf)
54 freqnoise_save = f.create_dataset("calibrated/HzpV", data=CALHzpV)
55 if meas_type == 'swept':
56     hvach1tf_save = f.create_dataset("calibrated/hva/ch1", \
57 data=HVA_CH1_tf)
58     displacement_spect = f.create_dataset("calibrated/disp_spect_unnorm"\ \
59 , data=displac_spect_unnorm[1])
60     phase_resp1 = f.create_dataset("calibrated/phase_resp_unnorm", \
61 data=displac_spect_unnorm[2])
62     displacement_spect_norm = \
63         f.create_dataset("calibrated/disp_spect_norm", \
64 data=displac_spect_norm[1])
65     displacement_spect_norm.attrs['units'] = 'm' + spectra_type + '/Vpk'
66     phase_resp2 = f.create_dataset("calibrated/phase_resp_norm", \
67 data=displac_spect_norm[2])
68
69 else:
70     displacement_spect = f.create_dataset("calibrated/disp_spect", \
71 data=displac_spect)
72     displacement_spect.attrs['units'] = spectra_type
73     displacement_spect.attrs['meas_type'] = meas_type
74 f.close()
```

5.17 HVA

5.18 FSS

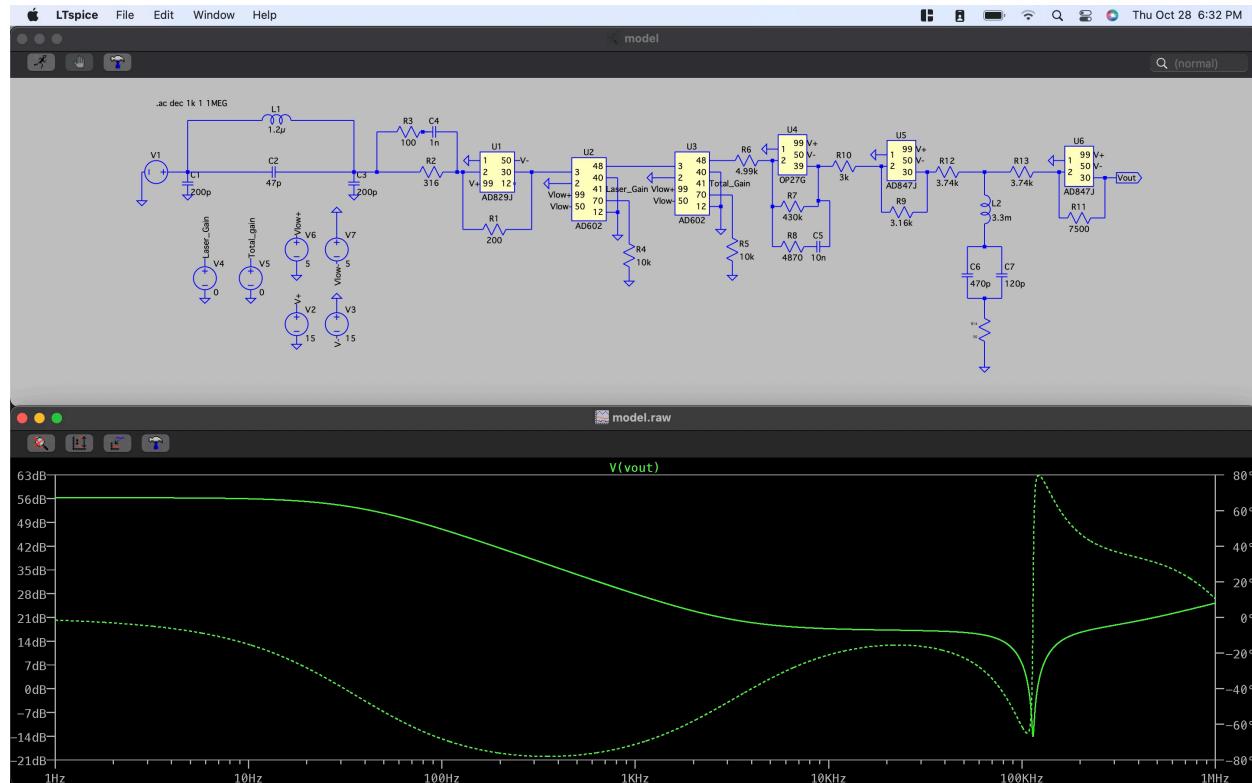


Figure 5.27: The FSS frequency response simulated in LTspice

5.19 Measuring OLG [H]

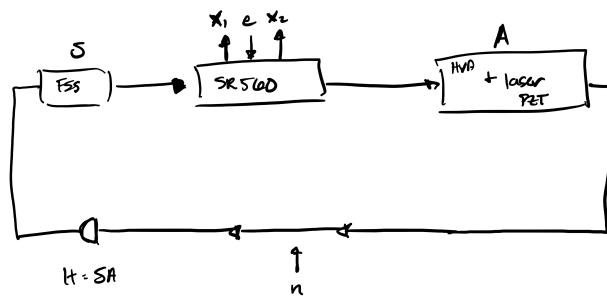


Figure 5.28: Open loop gain measurement drawn diagram

x2 is the PSD taken immediately after the excitation point

$$x_2 = e + He + H^2e + \text{H.O.T.s} = \frac{e + n}{1 - H} \quad (5.77)$$

x1 is the PSD taken immediately prior to the excitation point

$$x_1 = He + H^2e + H^3e + \text{H.O.T.s} = \frac{He}{1 - H} \quad (5.78)$$

We take the transfer function measurement ζ :

$$\zeta = \frac{x_1}{x_2} = \frac{He/(1 - H)}{(e + n)/(1 - H)} \quad (5.79)$$

Assuming the excitation is significantly larger than the noise ($e \gg n$):

$$\zeta \approx H \quad (5.80)$$

References

- [1] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation*. W. H. Freeman and Company, 1973.
- [2] B. P. Abbott *et al.*, “Gw170817: Observation of gravitational waves from a binary neutron star inspiral,” *Phys. Rev. Lett.*, vol. 119, p. 161101, Oct 2017.
- [3] A. H. Nitz, S. Kumar, Y.-F. Wang, S. Kastha, S. Wu, M. Schäfer, R. Dhurkunde, and C. D. Capano, “4-ogc: Catalog of gravitational waves from compact binary mergers,” *The Astrophysical Journal*, vol. 946, p. 59, mar 2023.
- [4] P. Saulson, *Interferometric Gravitational Wave Detectors (2nd edition)*. World Scientific Publishing Co. Pte. Ltd., 2017.
- [5] B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics*. Wiley Series in Pure and Applied Optics, New Jersey: Wiley, 2 ed., 2007.
- [6] E. D. Black, “An introduction to pound–drever–hall laser frequency stabilization,” *American Journal of Physics*, vol. 69, no. 1, pp. 79–87, 2001.
- [7] D. Z. Anderson, “Alignment of resonant optical cavities,” *Appl. Opt.*, vol. 23, pp. 2944–2949, Sep 1984.
- [8] G. Vajente, “Experimental techniques in interferometric gravitational wave detectors.” unpublished (Chapter : Optical Configurations, Accessed : 2018).

- [9] T. Vo, *Adaptive Mode Matching in Advanced LIGO and beyond*. PhD thesis, Syracuse University, Syracuse, NY, 2019.
- [10] E. Oelker, L. Barsotti, S. Dwyer, D. Sigg, and N. Mavalvala, “Squeezed light for advanced gravitational wave detectors and beyond,” *Opt. Express*, vol. 22, pp. 21106–21121, Aug 2014.
- [11] L. S. Collaboration, “Instrument science white paper 2022-2023,” tech. rep., LIGO Scientific Collaboration, 2022. LIGO-T2200384-v2.
- [12] J. Aasi *et al.*, “Advanced ligo,” *Classical and Quantum Gravity*, vol. 32, p. 074001, Mar. 2015.
- [13] P. Hello and J.-Y. Vinet, “Analytical models of thermal aberrations in massive mirrors heated by high power laser beams,” *Journal de Physique*, vol. 51, pp. 1267–1282, 1990.
- [14] A. F. Brooks, *Hartmann Wavefront Sensors for Advanced Gravitational Wave Interferometers*. PhD thesis, University of Adelaide, 2007.
- [15] A. F. Brooks and V. Fafone, “Thermal adaptive optics,” in *Advanced Interferometric Gravitational-Wave Detectors* (D. Reitze, P. Saulson, and H. Grote, eds.), pp. 609–637, World Scientific, 2019.
- [16] T. L. S. Collaboration, J. Aasi, *et al.*, “Advanced ligo,” *Classical and Quantum Gravity*, vol. 32, p. 074001, mar 2015.
- [17] J. Ramette, M. Kasprzack, A. Brooks, C. Blair, H. Wang, and M. Heintze, “Analytical model for ring heater thermal compensation in the advanced laser interferometer gravitational-wave observatory,” *Applied Optics*, vol. 55, p. 2619, Mar 2016.
- [18] T. Hardwick, V. J. Hamedan, C. Blair, A. C. Green, and D. Vander-Hyde, “Demonstration of dynamic thermal compensation for parametric instability suppression in advanced ligo,” *Classical and Quantum Gravity*, vol. 37, p. 205021, sep 2020.

- [19] M. Jacobson, S. O'Connor, and P. Willems, "Final design of aligo tcs ring heaters and associated electronics," tech. rep., LIGO Scientific Collaboration, 2011. LIGO-T1100034-v5.
- [20] C. Cahillane, "Power up thermalization." <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=46553>, 2019. [aligo elog 46553].
- [21] S. Dwyer, "noise budget plots without squeezing." <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47351>, 2019. [aligo elog 47351].
- [22] A. Buikema, C. Cahillane, G. L. Mansell, *et al.*, "Sensitivity and performance of the advanced ligo detectors in the third observing run," *Phys. Rev. D*, vol. 102, p. 062003, Sep 2020.
- [23] T. T. Fricke, N. D. Smith-Lefebvre, R. Abbott, R. Adhikari, K. L. Dooley, M. Evans, P. Fritschel, V. V. Frolov, K. Kawabe, J. S. Kissel, B. J. J. Slagmolen, and S. J. Waldman, "Dc readout experiment in enhanced ligo," *Classical and Quantum Gravity*, vol. 29, p. 065005, Feb. 2012.
- [24] D. Brown, "Model of 9mhz rin higher order mode coupling." <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47253>, 2019. [aligo elog 47253].
- [25] A. Brooks, "Commissioning guide for the custom h1itm co2 mask: Step 1." <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=46976>, 2019. [aligo elog 46976].
- [26] G. Mansell, "Demodulating injected lines during ring heater steps so far." <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47968>, 2019. [aligo elog 47968].
- [27] D. Brown, "Initial itmy mask test and 9mhz rin line." <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47081>, 2019. [aligo elog 47081].

- [28] J. Driggers, “Moving itmy spot position changes 9 mhz rin coupling.” <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47101>, 2019. [aligo elog 27101].
- [29] J. Driggers, “Moving itmy spot position with rest of ifo aligned to match.” <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47117>, 2019. [aligo elog 47117].
- [30] R. B. F. H. M. . R. A. V.P.L.S., “Xvii. a brief account of microscopical observations made in the months of june, july and august 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies,” *The Philosophical Magazine*, vol. 4, no. 21, pp. 161–173, 1828.
- [31] H. B. Callen and T. A. Welton, “Irreversibility and generalized noise,” *Phys. Rev.*, vol. 83, pp. 34–40, Jul 1951.
- [32] C. Zener, *Elasticity and Anelasticity of Metals*. University of Chicago Press, 1948.
- [33] Y. Levin, “Internal thermal noise in the ligo test masses: A direct approach,” *Physical Review D*, vol. 57, p. 659–663, Jan 1998.
- [34] G. M. Harry, H. Armandula, E. Black, D. R. M. Crooks, G. Cagnoli, J. Hough, P. Murray, S. Reid, S. Rowan, P. Sneddon, M. M. Fejer, R. Route, and S. D. Penn, “Thermal noise from optical coatings in gravitational wave detectors,” *Appl. Opt.*, vol. 45, pp. 1569–1574, Mar 2006.
- [35] T. Hong, H. Yang, E. K. Gustafson, R. X. Adhikari, and Y. Chen, “Brownian thermal noise in multilayer coated mirrors,” *Physical Review D*, vol. 87, Apr 2013.
- [36] G. Harry and G. Billingsley, “Fused silica, optics and coatings,” in *Advanced Interferometric Gravitational-Wave Detectors* (D. Reitze, P. Saulson, and H. Grote, eds.), pp. 521–554, World Scientific, 2019.

- [37] G. Cole, W. Zhang, M. Martin, J. Ye, and M. Aspelmeyer, “Tenfold reduction of brownian noise in high-reflectivity optical coatings,” *Nature Photonics*, vol. 7, pp. 644–650, 07 2013.
- [38] M. Abernathy, “Noise in crystalline coatings,” Tech. Rep. G1401060-v1, LIGO Scientific Collaboration, August 2014.
- [39] R. W. Boyd, *Nonlinear Optics, Third Edition*. USA: Academic Press, Inc., 3rd ed., 2008.
- [40] G. D. Cole, W. Zhang, B. J. Bjork, D. Follman, P. Heu, C. Deutsch, L. Sonderhouse, J. Robinson, C. Franz, A. Alexandrovski, M. Notcutt, O. H. Heckl, J. Ye, and M. Aspelmeyer, “High-performance near- and mid-infrared crystalline coatings,” *Optica*, vol. 3, pp. 647–656, Jun 2016.
- [41] S. Adachi, “GaAs, AlAs, and $\text{Al}_x\text{Ga}_{1-x}\text{As}$: Material parameters for use in research and device applications,” *Journal of Applied Physics*, vol. 58, pp. R1–R29, aug 1985.
- [42] A. Yariv, *Quantum Electronics (3rd. ed)*. John Wiley and Sons, 1989.
- [43] J. F. Nye, *Physical properties of crystals (Their representation by tensors and matrices)*. Oxford University Press, 1985.
- [44] F. W. J. Oliver, “Bessel functions of integer order,” in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables (Partially Mathcad-enabled)* (M. Abramowitz and I. Stegun, eds.), pp. 9.1.41–45, U.S. Department of Commerce, NIST, 1972.
- [45] M. F. Edgard Bonilla, “Optical phase perturbations on the reflected wave of a dielectric mirror. application to the electro-optic effect in algaas coatings,” Tech. Rep. T1800528-v1, LIGO Scientific Collaboration, December 2018.

- [46] M. Fejer, “Electro-optic effects in algaas mirrors.” Private LIGO scientific collaboration document, August 2018.
- [47] S. W. Ballmer, “Photothermal transfer function of dielectric mirrors for precision measurements,” *Physical Review D*, vol. 91, Jan 2015.
- [48] N. Suzuki and K. Tada, “Elastooptic and electrooptic properties of gaas,” *Japanese Journal of Applied Physics*, vol. 23, no. 8, pp. 1011–1016, 1984.
- [49] S. Adachi, *The Handbook on Optical Constants of Semiconductors*. WORLD SCIENTIFIC, 2012.
- [50] W. Press, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, 2007.
- [51] Corning, *MACOR (Machinable glass ceramic for industrial applications)*, 2012. Available at <https://www.corning.com/media/worldwide/csm/documents/71759a443535431395eb34ebad091cb.pdf>.
- [52] A. Ananyeva, S. Appert, B. G., A. Brooks, P. Fritschel, S. Gras, M. Kasprzack, K. Kuns, and L. Zhang, “Core optics and point absorber mitigation,” tech. rep., LIGO Scientific Collaboration, 2022. LIGO-G2200853-v2.
- [53] H. Kogelnik and T. Li, “Laser beams and resonators,” *Appl. Opt.*, vol. 5, pp. 1550–1567, Oct 1966.
- [54] P. Hello and J.-Y. Vinet, “Analytical models of transient thermoelastic deformations of mirrors heated by high power cw laser beams,” *Journal De Physique*, vol. 51, pp. 2243–2261, 1990.