

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

Technical Note	LIGO-T1800224-v0	2018/08/03
<b>Thermal State of Advanced LIGO Test Masses: Mirror Degradation Monitor</b> Second Interim Report		
SURF Student: Guadalupe Quirarte, Harvey Mudd College Mentors: Carl Blair, Joseph Betzweizer		

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

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

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

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

# 1 Introduction

The Advanced LIGO (aLIGO) detector is composed of a Michelson interferometer with 4 km Fabry-Perot optical cavities that function as the arms of the interferometer. These cavities utilize high-reflectivity fused silica mirrors that form an optical cavity and interact with high power laser beams. These mirrors are referred as the input test mass (ITM) and the end test mass (ETM) of each arm as shown below in Figure 1. Each cavity has an optical power that ultimately will reach 800 kW. A small portion of this power is absorbed by test masses and then converted into heat. A thermal transient is created in the mirrors when the arm cavities control systems are locked attaining optical resonance in the arm cavities.

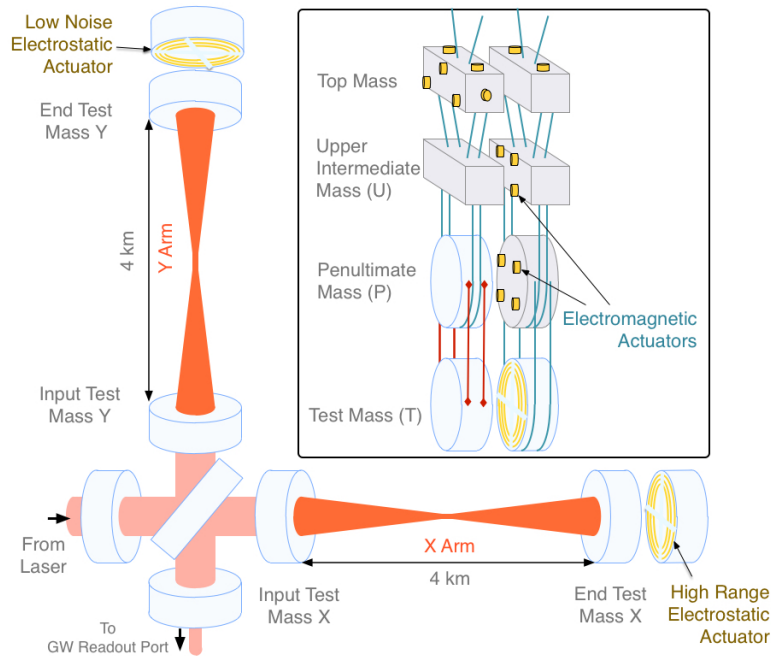


Figure 1: Diagram of an Advanced LIGO interferometer with Fabry-Perot cavities. [11].

This thermal transient's effect on the interferometer is dominated by two mechanisms. The first is known as thermal lensing which is caused by a change of refractive index through the thermo-optic effect. This thermal lensing can cause aberrations in the beam entering and leaving the detector's arm cavities that contribute to a deterioration of the mode matching between the many optical cavities of the interferometer which in turn reduces the sensitivity of the interferometer.[2]

The second effect is the high-reflectivity coatings of the test masses being deformed from thermal expansion. A change of the radius of curvature (ROC) of the mirror is the first order deformation and this shifts the frequency of the transverse optical modes (TEM) resonant in the cavity. This changes the tuning conditions for parametric instabilities. As the mirror warms there is also a relatively small increase in the mechanical mode frequencies due to the small positive thermal dependence of the Young's modulus of the mirror substrates. Parametric instabilities arise when the frequency spacing between the fundamental mode and a TEM mode equals the mechanical mode frequency. The interferometer can become

inoperable and unstable if this condition is met.

These are problems that need to be addressed and investigated. Work has been completed by LIGO researchers to design and develop strategies aimed to compensate these thermal effects such that they do not impact the detector sensitivity. The paper, “Thermal modelling of Advanced LIGO test masses,” discusses how LIGO researchers have built a thermal model of the test masses that applies shifts in mechanical mode frequencies as a probe for the overall temperature of the mirror. This paper provides information on estimates of the coating absorption of the ETMY of the Livingston detector.[2]

LIGO researcher, Simon Tait, has worked with the thermal model outlined above in an effort to create an instantaneous absorption estimate of aLIGO test masses. He has modified the COMSOL thermal model and has written Matlab scripts that extract eigenfrequency and temperature data from the models. He has conducted parameterizations on the data to create an exponential model which he compared with experimental frequency tracking. The experimental frequency tracking integrates a transfer function and the extraction of the coating absorption. His final results focus on the month of July 2017. His code compares the simulated eigenfrequency of the test mass with the measured eigenfrequencies of the test mass. The eigenfrequencies are measured in the light in the transmission of the OUTPUT mode cleaner. As there are several other contributions to the change in eigenfrequency, his code calculates the coating absorption by normalizing the gradient of this shift. Shifts in the mechanical frequencies are tracked through this method during lock periods marked as stable for meeting a specific time threshold.

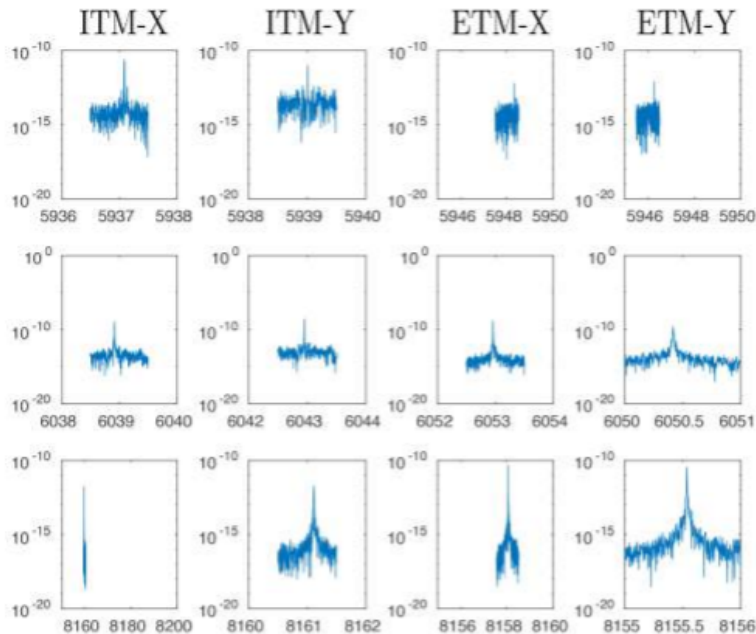


Figure 2: This is the windowed Power Spectral Density of the OMC output showing resonant frequencies of ITMs and ETMs in Hz. [1].

Tests have only been conducted between January and September of 2017. [1] However, further work that can be integrated is implementing a Kalman Filter into the LLO Ther-

mal Compensation System monitoring system to help produce an instantaneous absorption estimate of the LLO test masses. This is what this project is focused on creating.[3]

## 1.1 Motivation

The purpose of this project is to use the eigenmodes of the test mass to measure its thermal state. The ultimate goal is to build a Kalman filter to optimally estimate the eigenfrequencies of the test mass. I will be working with COMSOL models and learning how to analyze and extract useful information from them. I will also be using the LIGO CDS system that allows me to get data from the interferometer archives. Integrating the Kalman filter state space model parameters such as the test mass coating absorption can be inferred. This information will then be used to optimize the thermal compensation, to minimize detrimental thermal effects on the interferometer.

Currently, I have focused on the theory and implementation of the Kalman Filter after having finished the COMSOL model analysis. I have built on the existing COMSOL Multiphysics models from S.Tait [3] and H. Wang [2]. I have also parameterized the simulated eigenfrequency data in preparation for building a Kalman filter state space model. Furthermore, I have built a simplified Kalman filter that estimates eigenfrequencies of the test mass that can be tested on actual data.

Before explaining the work I have done on building the Kalman Filter mirror degradation monitor, it is useful to have a background on mechanisms and work that has been completed beforehand. In the following sections I describe the essential components of the thermal compensation system, introduce the finite element model, the completed parameterization, and the Kalman Filter implementation.

## 2 Progress

### 2.1 Background

The following section describe the interferometer's Thermal Compensation System and the Pound Drever Hall technique utilized for locking the cavities. This is aimed to provide the sufficient background information of the mechanisms involved in the extraction of eigenfrequency data for this project. By understanding these systems, one can have a clearer understanding of the functionality of mechanisms that play a role in providing the control system inputs and outputs for the implementation of a mirror degradation monitor.

#### 2.1.1 The Thermal Compensation System (TCS)

aLIGO's Thermal Compensation System (TCS) is supposed to help mitigate thermal lensing optical distortion effects.

The Ring Heater (RH) heats the test masses on the outside near the rear surface. This produces a negative thermal lens in the optic and decreases the radius of curvature of the concave mirrored surface of the test mass. The  $CO_2$  laser heats an area in the center of

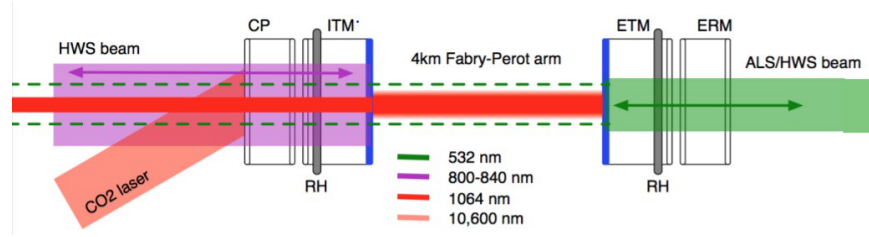


Figure 3: TCS system including ITM, ETM, circular ring heater (RH) to heat the test masses on the outside, a  $CO_2$  laser projects various annular shapes onto the compensation plates, and the Hartmann Wavefront Sensor (HWS) [3].

the compensation plate, this produces a positive thermal lens. The HWS measures the wavefront distortion of the the HWS probe beam. This wavefront distortion is dominated by the thermal lens in the ITM and CP. The purpose of this system is to compensate for the laser power absorbed in the test mass mirror coatings. [1]

The power absorbed into the optic is proportional to the input laser power  $P_{power}$  and  $\alpha_\lambda$  the absorption at the wavelength of the laser. The formation of a thermal lens and thermal expansion are the two effects created by the thermal transient that forms when the cavities control system is locked producing a large laser power in the optical cavity, this locking process will be described below. Figure 4 provides a visual of a the thermoelastic surface deformation and wavefront distortion due to the thermal lens. The thermally distorted mirror, illustrated as a dashed line, shows the change of radius of curvature of the mirrors. This shifts the frequency of the transverse optical modes (TEM) resonant on the cavity by the order of  $10^3 Hz$ . This can dramatically, change the tuning conditions for parametric instabilities, inducing many different instabilities. The change in temperature of the optic results in a small shift in the mechanical eigenmode frequency of the test masses of approximately 0.2 Hz. If the shifted frequency of the TEM modes and the spacing to the fundamental central mode of the cavity become equal to the frequency of the mechanical modes, then the radiation pressure of the optical modes can result in an exponentially increasing excitation of the mechanical modes known as parametric instability (PI) leading to the unlocking of the cavity. [1]

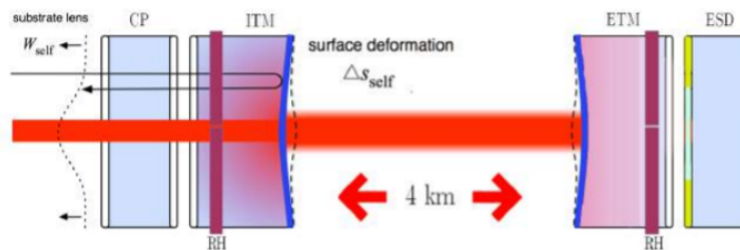


Figure 4: Thermoelastic surface deformation [3].

The cavities are locked through the Pound Drever Hall technique . A basic layout for locking a cavity to a laser is depicted below. This is the setup that permits to measure

length noise in the cavity. The main mechanism behind the setup can be divided into three steps. A beam is sent into the cavity, a photodetector takes in the reflected beam, and then its output goes to a length controlling actuator. With a correct feedback setup, the system will hold at the length of the cavity where light is resonant. [10]

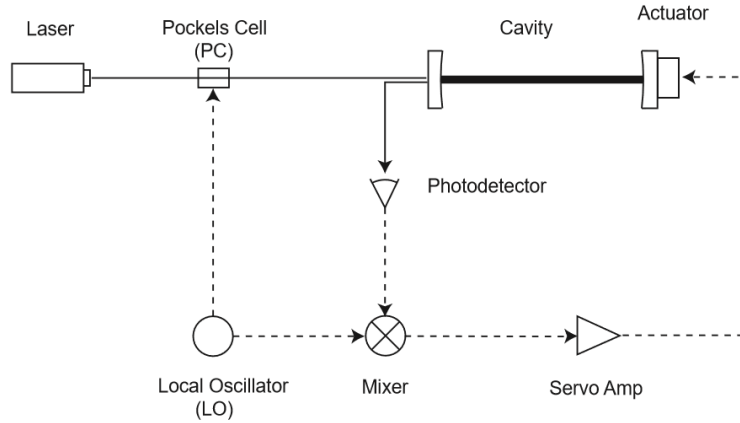


Figure 5: Locking a cavity to a laser. (Dashed lines are signal paths and solid lines are optical paths) [10].

When the cavity is resonant, it also experiences a large laser power build up. A portion of the light is absorbed in the high reflectivity optical coating. Currently, the detectors have design specifications for ETM coatings to have an optical absorption of 0.5 ppm and 99.999 reflectivity at 1064 nm. [1] The input mirror is more complicated as the reflectivity is formed by the ITM and another mirror PRM forming a coupled cavity. The resulting resonance results in 400x more light in the arm cavity compared to the input light. The small portion of this light that is absorbed in the coating dominates the thermal load on the test mass resulting in the deformation as pictured on Figure 4.

As described in this section, understanding the TCS, the locking of a cavity, and the behavior of a cavity when resonant are all necessary to comprehend. This helps to further understand the next steps of the project which involve the thermal analysis of the test masses and the development of the degradation monitor.

## 2.2 Final Element Model

This section focuses on the development of a model of the entire thermal system utilizing COMSOL Multiphysics. The objective of the following procedures is to provide an analytical model of the shift in eigenfrequencies. The results from this section provide an analytic to arrive at a model for use as an input for the mirror degradation monitor Kalman Filter.

### 2.2.1 Method

Mechanical mode frequencies can be described as test mass thermometers. They depend on the dimension of the test mass and on two material properties with a temperature depen-

dence. These properties are the Young's modulus and the Poisson ratio. [2] An analytic way to express the eigenfrequencies of a cylinder is by following the equation below.

$$\omega_m = \beta_m \sqrt{\frac{Y(T)_{bulk}}{\rho(1+\nu)}}$$

Equation 1. The Mechanical Mode Frequency [3]

As shown above, the mechanical mode frequency  $\omega_m$  is directly proportional to the square root of the Young's modulus  $Y(T)_b$  and inversely proportional to the material density  $\rho$  and the Poisson ratio  $\nu$ . The variable  $\beta_m$  is corresponding to a constant dependent on the geometry of the cylinder.

### 2.2.2 Thermal Model

A thermal model of the test mass allows the temperature distribution in the test mass to be estimated for a particular thermal load on the test mass. This thermal model includes the test mass and its surrounding bodies. The accuracy of the aLIGO test mass thermal model can be improved by including thermal coupling. This is done by including the radiative heat transfer between the test mass and the other bodies surrounding it, which are discussed in section 2.1.1 under TCS. [2]

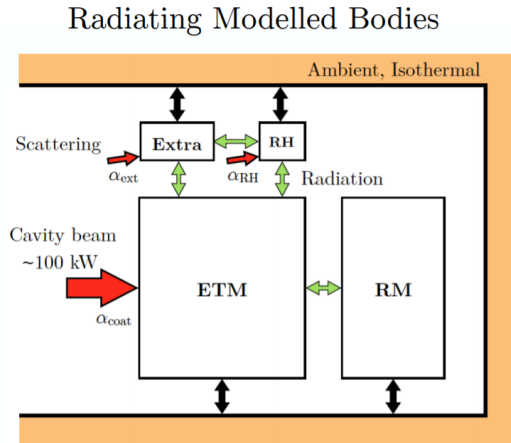


Figure 6: This is a heat transfer model between an end test mass and its surrounding elements, RH, RM, and the extra term represents other complex structures surrounding it. [3].

The model above illustrates the mechanisms involved during the transfer of heat when an arm cavity is locked. A portion of the light is converted into heat and scattered light from the beam covers bodies surrounding the test mass. Longer time constants are introduced to the test mass thermal transient because some of the heat from absorbed light radiatively couples to other structures. This object's increased temperature then radiatively couples heat back to the test mass.[2]

### 2.2.3 COMSOL

A state space model is required for the development of a Kalman Filter. It is a mathematical model that takes a set of input, output, and state variables of a physical system. To create a state space model of the test mass eigenfrequencies an accurate thermal model is required. I used COMSOL Multiphysics heat transfer toolbox to perform this thermal modelling. I modelled an end test mass, ETMY (test masses are nominally identical), and ran studies that pertained to its acoustic modes. I attached the heat transfer module to calculate the temperature of the ETM over time. A solid mechanics study is also attached to the module calculating eigenfrequencies of the ETM over time.

An aLIGO test mass is a cylinder with a 170 mm radius and 200 mm thickness made of Heraeus Suprasil 3001 fused silica.[4] When a 100 kW laser beam is shot at the system, the model utilizes a heat source boundary condition with a Gaussian profile to simulate the area in which the beam hits the mirror surface as shown in figure 10. With a laser beam of this power, the coating absorption of 1 ppm would correspond to a total absorbed energy of 0.1 W. The only input of the model that is given to it is the laser power step function. When the cavity is unlocked it is 0 and when the cavity is locked the measure power is 0.1 W. Figure 7 depicts the inputs and outputs of this thermal model and figure 8 provides a visual of the 2 dimensional thermal model of the whole system. [2]

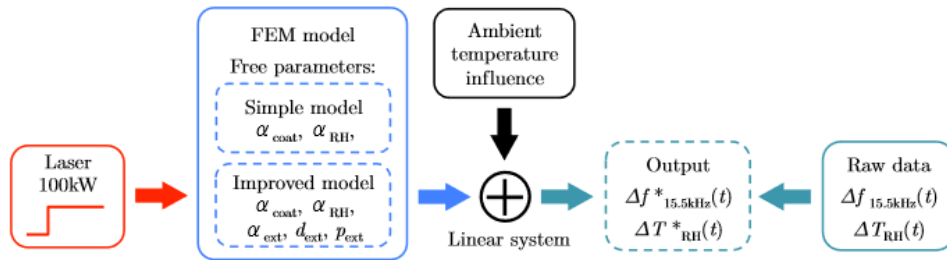


Figure 7: A finite-element model showing the inputs, outputs, and the free parameters to take into account when modelling a test mass. [2].

By running the physics, setting the material properties, and running a time dependence and eigenfrequency study, H. Wangs [2] model was used as a guide to learn how to navigate through the program. I focused on testing the behavior of fused silica. It is a material with a unique behavior as its Youngs modulus has a temperature dependence. This causes the rise in frequency of the mechanical modes of the material. Below is a table of the material properties modelled for this material.



## Rotationally Symmetric Model

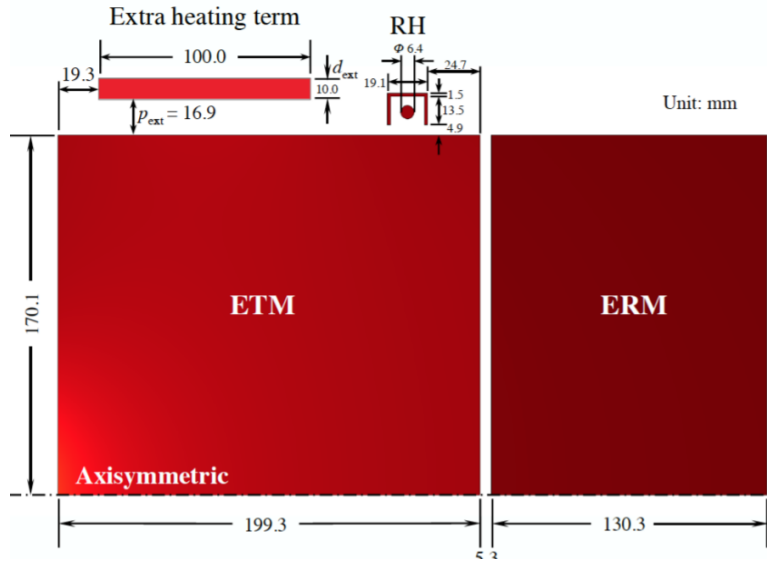


Figure 8: This shows the full dimensions of the 2DAS model, including scattering contribution (Extra Heating Term) Term [1].

Material Property	Symbol	Fused Silica	Unit
Density	$\rho$	2203	$\frac{kg}{m^3}$
Poisson's Ratio	$\nu$	0.17	1
Young's Modulus	E	$(7.3 \times 10^{10})(\Delta T)$	Pa
Thermal Conductivity	$\kappa$	1.38	$\frac{W}{mK}$
Thermal Expansion Coefficient	$\alpha$	$5.5 \times 10^{-7}$	$\frac{1}{K}$
Heat Capacity	$C_p$	740	$\frac{J}{kgK}$
Relative Permittivity	$\epsilon$	3.8	1
Electrical Conductivity	$\sigma$	$1 \times 10^{-14}$	$\frac{S}{m}$
Surface Emissivity	$\epsilon_e$	0.93	1

I conducted a thermal modelling simulation by adding a theoretical heating load on one of the faces of the test mass. I then conducted a time dependent study to observe how the model heated up by the Boundary Heat Source. When an object is vibrating with a certain eigenfrequency (natural frequency), the structure of the object deforms into a corresponding shape. The COMSOL three dimensional visual of the mode shape is an exaggeration of the deformation, but it provides us a geometrical understanding of the eigenmode. An

eigenfrequency analysis provides the shape of the mode, but the actual size of the deformation can only be determined with the knowledge of an actual excitation with damping properties. With the implementation of heat transfer physics to understand the effect the shooting of the high-power laser beam had on the model, I was able to model the thermal transient formed. To extract the eigenfrequency information from the time dependent thermal analysis output the COMSOL MATLAB interface was required.

The thermal model includes the end test mass (ETM), reaction mass (ERM), and the ring heater (RH). Figure 2, illustrates these components which form part of the TCS. The geometry of the model is simplified to be a 2-dimensional Axis-symmetric representation of the system in order to avoid high computational demands that would be present while running a realistic geometry. This axis-symmetric model required heating loads including the main interferometer beam being axis-symmetric which is adequate for the current analysis. Moreover, optical absorption of the coating and scattering from the surrounding suspensions can also be implemented without the burden of having them exponentially increasing the solution time. However, this model is restricted to only monitor circularly symmetric mechanical eigenmodes. There is historic LIGO data containing the eigenfrequencies of 5.9, 6.0 and 8kHz modes. These modes have been analyzed with a 3D geometry model. The 5.9 and 6.0kHz modes are not axis-symmetric. The 8kHz mode is axis-symmetric, as a result the rest of this report will focus on the 8 kHz mode.

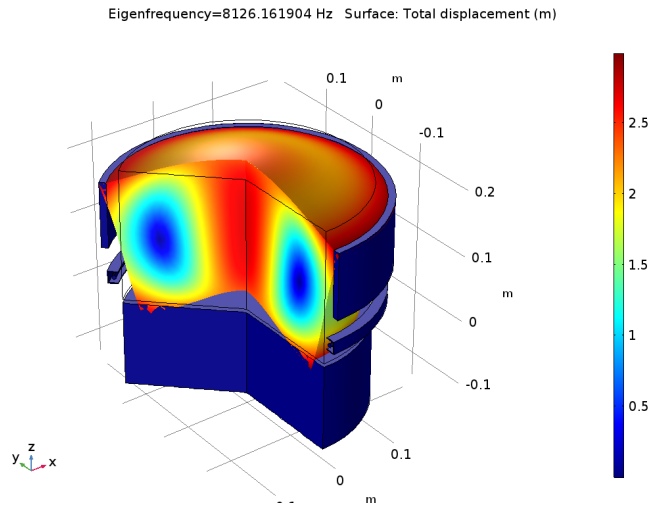


Figure 9: Mechanical mode shape of the 2DAS modelled ETM surrounded by the Ring Heater, ERM and an approximated suspension. [1].

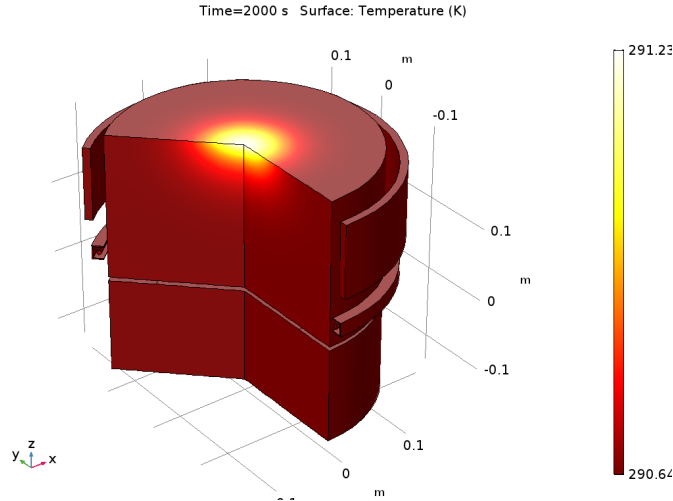


Figure 10: The change in the color has a radially outward changing direction which depicts the radial temperature gradient of the optic. [1].

An aLIGO interferometer's arm cavity contains a 750 kW Gaussian beam centered in the cavity. The equation below shows how the power absorbed by the optic,  $P_\alpha$ , is calculated. The  $\omega$  corresponds to the beam radius of the incident Gaussian light source and the value is 6.2 cm, the  $r$  refers to the distance from the center of the beam.  $P_{in}$  is the power input into the Fabry Perot cavity,  $k_{PRC}$  is the gain of the power recycling cavity, and  $k_{AC}$  is the gain from the arm cavity.[1]

$$P_\alpha = \frac{2(P_{in}k_{PRC}k_{AC})}{(\pi\omega^2)} \exp\left(\frac{-2r}{\omega}\right) \frac{1}{\alpha_c}$$

Equation 2. Power absorbed by the optic [1]

The  $\alpha_c$  is the absorption of the coating on the test mass and it has been measured to be approximately 1.4 ppm at 1064 nm by using a Hartman Wavefront Sensor (HWS) for ETMY.[1] Furthermore, while analyzing the model, I noticed how the shift in ETM mechanical mode frequency was observed with relation to the surrounding optics in over 40 hours. By adding more components to the simulation this produces an elevation in re-radiation effects from the addition of these bodies. As a result, there is an increase in the calculated change in frequency over time [1].

### 2.3 Model Parameterization

With a clear understanding and analysis of COMSOL, I derived the curve fitting exponential model described in S. Tait [1]. By using MATLAB to obtain the COMSOL data of the eigenfrequency shift of the specific eigenmode 8.126 kHz, one can obtain the modelled shift of the frequency. COMSOL provides us with an analytical model and my goal is to extract a good fit of this analytical model to obtain an exponential model that can be utilized for the Kalman Filter which will be explained in the next section.

The equation below is one of the fits for frequency change in temperature calculated with COMSOL. This is a single term exponent with respect to time. The  $A$  corresponds to

the total change in frequency,  $b_1$  is the model time constant, and  $c_1$  is the frequency at room temperature. This model can then be used to produce a simulated absorption estimate.

$$A(1 - e^{-b_1 t}) + c_1 \quad \text{Equation 3. Exponential Model [1]}$$

Before extracting the data through the MATLAB interface, I changed the COMSOL model by removing the frequency shift produced by heating and cooling and only kept the heating transient which would simulate a locked state. This heating transient shift to a cooling transient is a linear time invariant system. The model was clarified by adding descriptions to the parameters and variables to understand how each one was derived. By removing the cooling transient, one could obtain a set of frequencies for a specific eigenmode. Simon's code produced an estimate for the change in temperature and frequency with self-heating from his pre-solved COMSOL model. Before extracting the eigenfrequencies, I extracted the temperature difference from the model for all the ETM points and plotted it to see if I observed a reasonable behavior representative of the heating. The graph below illustrates the heating transient formed for all four bodies modelled.

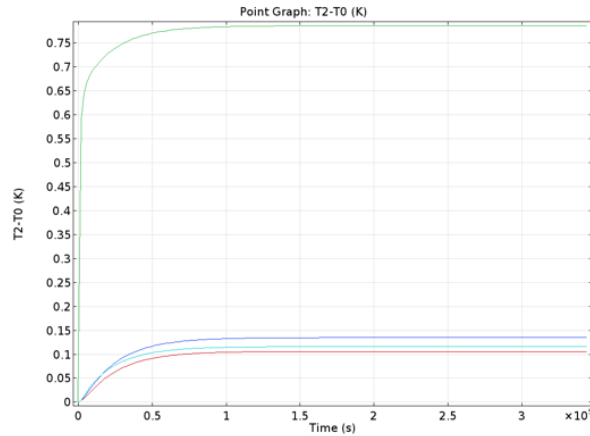


Figure 11: **Graph 1:** The graph above can be obtained by plotting on MATLAB the temperature over time of the end test mass and the other bodies to observe the behavior to model the heating of the masses. . [1].

Once I finished plotting Graph 1, I ran the loop through the data for the specific eigenmode. I then applied the curve fitting tool from MATLAB to work on fitting the analytical data to a second-degree exponential model to see if it provided a better fit to the data. As explained in section 2.2.2, thermal coupling must be taken into account when creating a thermal model. As a result, I decided to change the model to see if longer time constants are introduced. This could represent how some of the heat from scattered light radiatively couples to other elements.

$$A(1 - 2e^{-b_2t} + e^{-2c_2t}) + d_2 \quad \text{Equation 4. A Test Exponential Model}$$

Model

The gain  $A$  was approximately 0.09, the time constants  $b_2$  and  $c_2$  were 0.159 and 0.079, and  $d_2$  the eigenfrequency at room temperature was 8126. While fitting the model I made sure to keep the lower and upper limits large to ensure I was not forcing the data to fit a specific mold. Although the goodness-of-fit statistics obtained for both this model and the original model were very similar and the  $r$  squared was approximately 0.9936-0.9958, the first data point at time zero did not match the exponential model. By forcing  $d$  to be the  $y$ -intercept value of 8126.17, this would result in a less fit model of the function and would require a forcing of the variables  $b_2$  and  $c_2$ .

Below is a graph of the curve fitting model with the value obtained for  $b_2$  and  $c_2$  previously stated. Both the first order exponential model and the second order exponential model provide closely similar values for the goodness-of-fit statistics. In order to know which exponential model is best to apply for the Kalman Filter requires the testing of both in the implementation of the filter to analyze which assumed model provides the most reasonable coating absorptione estimates. I decided to apply the first order exponential model as an input for the mirror degradation monitor. Further work can be done by applying the second order exponential model instead to compare which curve fit model is best to apply for the monitor.

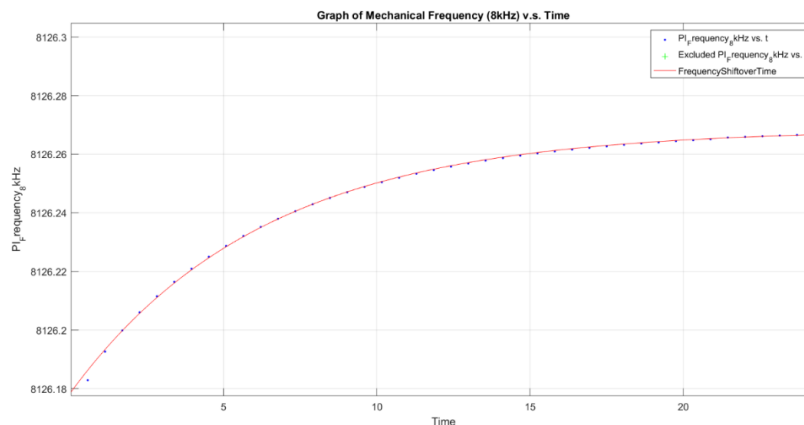


Figure 12: **Graph 2:** Exponential Model curve fitting of the analytical data from COMSOL. [1].

## 2.4 Calculated Coating Absorption

This section focuses on the results obtained by Simon for the period of January through September 2017 with estimates of the coating absorption.[1] The purpose of this section is to investigate the previous methods used to investigate coating absorption such that the Kalman Filter method can be contrasted and compared to this previous method.

The following graphs show Simon's results of calculated coating absorptions during this time period. Since the eigenfrequency data can only be measured from the OUTPUT mode cleaner during lock states, the unlocked state estimates were produced by applying the method of interpolation. This method constructs data points for the unlock state within the range of the discrete set of known eigenfrequency data and following the inverse behavior of a lock.

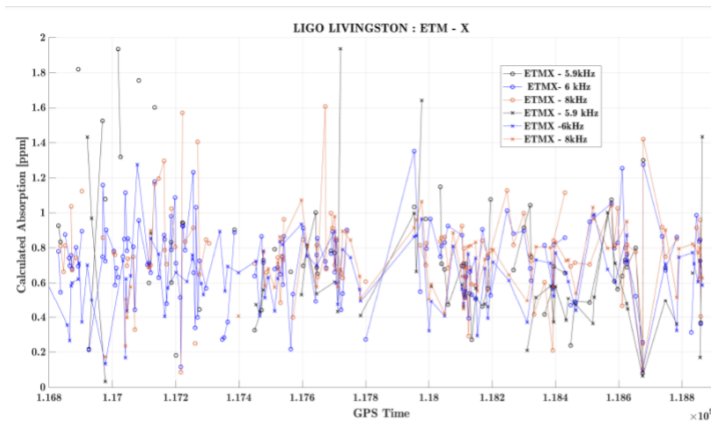


Figure 13: The calculated optical absorption during locked and unlocked states of ETMX between January and September of 2017. o refers to locked times and x is the interpolated estimate for the unlocked states. [1].

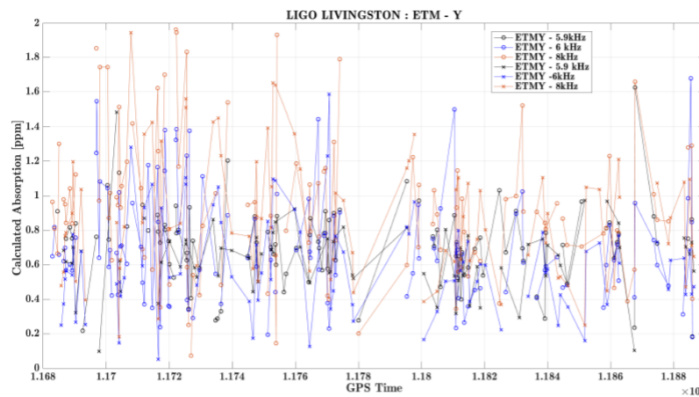


Figure 14: The calculated optical absorption during locked and unlocked states of ETMY between January and September of 2017. o refers to locked times and x is the interpolated estimate for the unlocked states. [1].

A prediction of the coating absorption was made, using the equation below,

$$\alpha_{coat} = \frac{\Delta f_{meas}}{\Delta f_{sim}} \alpha_{sim}$$

Equation 5. Coating Absorption [1]

where the numerator is the gradient of the measured frequency shift with time and it takes into account both locked and unlocked states. The denominator is the simulated change in frequency. The code to complete this coating absorption estimate was run for the month of July and the approximation was reasonable. It can be used to model both ITM and ETM. Although an understanding of this instantaneous coating absorption extraction technique is ideal for further comparisons, this project focuses on creating a different method of extracting the coating absorption through the implementation of a Kalman Filter.

## 2.5 Kalman Filter Implementation

### 2.5.1 Kalman Filter Theory

A Kalman Filter is a recursive algorithm that takes in as input a model and measurements to output less noisy and more accurate estimates. This filter is ideal for real time problems since it has the advantage of only taking in the previous state of the system and not requiring any other history. The figure below provides a clear overview of the mechanism behind the algorithm and its functionality. It is widely used in different areas such as navigational systems, guidance, and controlling vehicles. [5]

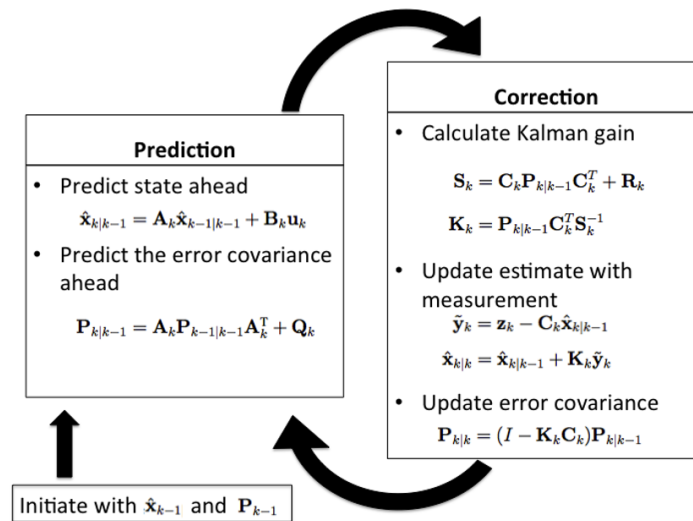


Figure 15: Kalman Filter thinking process. [5].

The application of a Kalman Filter for an internal system requires the use of the state-space representation of this system. The equation below illustrates how the present state is dependent on the previous state.

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \quad \text{Equation 6. [5]}$$

$\mathbf{A}_k$  represents the state transition model used to the previous state  $\mathbf{x}_{k-1}$  and  $\mathbf{B}_k$  represents the input-control model. The input-control model is applied to the control-vector  $\mathbf{u}_k$  and the state matrix is applied to the state-vector  $\mathbf{x}_{k-1}$ . The process noise is represented by  $w_k$  which in this case is a univariate normal distribution with covariance  $\mathbf{Q}_k$ . [5]

$$\mathbf{z}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{v}_k \quad \text{Equation 7. [5]}$$

Once the observation,  $\mathbf{z}_k$  (measurement), is taken representing the true state  $\mathbf{x}_k$ , as shown above,  $\mathbf{C}_k$  represents the observation model which is applied to the state vector at time k. Taking account of the measurement noise,  $\mathbf{v}_k$ , signifies the zero mean univariate normal distribution noise in this case with covariance  $\mathbf{R}_k$ . The observation is not affected by the control vector and as a result  $\mathbf{D}_k$ , the output-control model is equal to zero.

There are two variables that represent the state of the filter which are  $\hat{\mathbf{x}}_{k|k}$  known as the a posteriori with an estimate at time k, and  $\mathbf{P}_{k|k}$  which is the a posteriori error covariance matrix. The notation of these states can be described as  $\hat{\mathbf{x}}_{a|b}$  where a is the time of the estimate and b is the time of the observation and b is restricted to be less than or equal to a. Below I have divided the Kalman Filter into two sections as it is normally explained in two phases: prediction and correction. [5]

### Prediction

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \quad \text{Equation 8. [5]}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k \quad \text{Equation 9. [5]}$$

The prediction phase depends on the time-step's previous state as illustrated above. The predicted state  $\hat{\mathbf{x}}_{k|k-1}$  is known as the a priori state estimate as it depends on the observation at time k-1.  $\hat{\mathbf{P}}_{k|k-1}$  is the prior state covariance matrix. [5]

### Correction

Calculating Kalman Gain:

$$\mathbf{S}_k = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k \quad \text{Equation 10. [5]}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \mathbf{S}_k^{-1} \quad \text{Equation 11. [5]}$$

Updating Estimate with Measurement:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{C}_k^T \hat{\mathbf{x}}_{k|k-1} \quad \text{Equation 12. [5]}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad \text{Equation 13. [5]}$$

Update Error Covariance:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1} \quad \text{Equation 14. [5]}$$

The correction phase requires the calculation of the Kalman Gain as shown in equations 10



and 11 and then the updating of the estimate state by using the measurement. Moreover, the estimate state and the error covariance are updated by observing from the current time-step rather than the previous step.

### 2.5.2 Building a Kalman Filter

The following steps outline an approach towards building a Kalman Filter. These are the steps I have followed while moving forward in the project. [6]

1. **Understand the situation:** Look at the problem and break it down to the mathematical basics.
2. **Model the state process:** Starting with a basic model, it may not work effectively at first, but this can be refined later.
3. **Model the measurement process:** Analyze how you are going to measure the process.
4. **Model the noise:** This has to be completed for both the state and measurement process. The base Kalman filter assumes Gaussian (white) noise, so make the variance and covariance (error) meaningful (i.e., make sure that the error you model is suitable for the situation).
5. **Test the Filter:** Observe the behavior of the filter.
6. **Refine the Filter:** Try to change the noise parameters, if necessary go back and rethink the situation.

[6]

## 3 Results

### 3.1 Practice Kalman Filter

I started by going through a Kalman Filter tutorial to have a clearer understanding of the process before building the project monitor. I have implemented a practice Kalman Filter tracking the two-dimensional position of an object using Matlab. The following sections explain the process behind setting up the situation, state process, and analyzing the testing of the filter.

#### 3.1.1 State-Space Representation

I started by defining the position ( $x$ ), velocity ( $v$ ), and acceleration ( $a$ ) of the object and then creating the state space variables. The state space vector is composed of the four variables which were derived from following the four kinematic equations below. [12]

$$x(k) = x(k-1) + t_s v_x(k-1) + \frac{t_s^2}{2} a_x(k-1) \quad \text{Equation 15.}$$

$$y(k) = y(k-1) + t_s v_y(k-1) + \frac{t_s^2}{2} a_y(k-1) \quad \text{Equation 16.}$$

$$v_x(k) = v_x(k-1) + t_s a_x(k-1) \quad \text{Equation 17.}$$

$$v_y(k) = v_y(k-1) + t_s a_y(k-1) \quad \text{Equation 18.}$$

### State-vector, $X$

The state space is represented as:

$$X = \begin{pmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{pmatrix} \quad (19)$$

### State-propagation matrix, $A$

The state-transition model which is derived by following equations 15-18 and can be expressed as:

$$A = \begin{bmatrix} 1 & 0 & t_s & 0 \\ 0 & 1 & 0 & t_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

### Control-Input Matrix, $B$ , and Control vector, $u$

The control input matrix is derived from the acceleration of both the x and y directions and also applies the kinematic equations. The control vector is composed of both the x and y accelerations.

$$B = \begin{bmatrix} \frac{t_s^2}{2} & 0 \\ 0 & \frac{t_s^2}{2} \\ t_s & 0 \\ 0 & t_s \end{bmatrix} \quad u = \begin{bmatrix} a_x(k-1) \\ a_y(k-1) \end{bmatrix} \quad (21)$$

### Complete Process Model

The full state-space model for the tracking of a single object becomes: [12]

$$X(k) = \begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_s & 0 \\ 0 & 1 & 0 & t_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ y(k-1) \\ v_x(k-1) \\ v_y(k-1) \end{bmatrix} + \begin{bmatrix} \frac{t_s^2}{2} & 0 \\ 0 & \frac{t_s^2}{2} \\ t_s & 0 \\ 0 & t_s \end{bmatrix} \begin{bmatrix} a_x(k-1) \\ a_y(k-1) \end{bmatrix} \quad (22)$$

$$X(k) = AX(k-1) + Bu(k-1) + w \quad \text{Equation 23.}$$

### Observation Model, $C$ , and Observations, $Z$

The observation  $Z(k)$  comes from real measurements and  $C$  is the observation model. The prediction of the measurement is obtained by utilizing  $C$  which maps the state-space into the observation space.[8]

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (24)$$

$$Z(k) = CX(k) + v \quad \text{Equation 25.}$$

By following the state-space model outlined above and the Kalman Filter algorithm,

I created a Matlab script that generates the state space representation of the model and takes in simulated measurement data and control parameter data. It then applies a for loop to extract the Kalman estimate for a given time interval as it goes through both x and y measurements. [12]

Below are two graphs that distinguish the model and the measurements and the measurements with the Kalman Estimation. The model being applied is an assumed model and is not in fact the real model. This is analogous to my project's Kalman filter since I will be utilizing an assumed model that does not behave identically as the real model. A real model is not available to us and as a result we have to parametrize to find the closest match. By using an assumed model and actual measurements, a better estimate of the behavior of the system can be modelled. However, it is crucial to apply the correct process variance as explained in the following section.

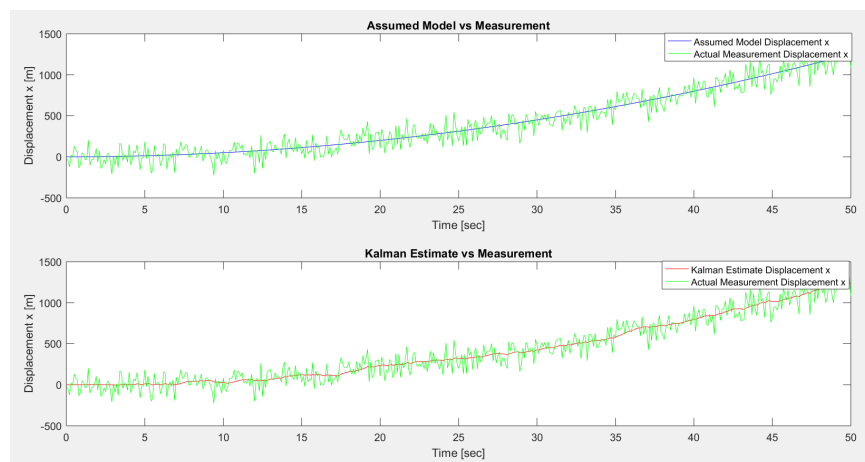


Figure 16: The top graph is a comparison of the assumed object's x-displacement and the actual x-measurement. The bottom graph is a comparison of the object's measured x-displacement and the estimated displacement.

### 3.1.2 Tuning the Practice Kalman Filter

When setting the Kalman Filter, it is imperative to carefully choose the values for the process variance  $Q$  and the measurement variance  $R$ . The variance corresponds to the fluctuations of the data and the choice of  $Q$  and  $R$  all depends on the trust one has on the data. [5]

The figure below illustrates an incorrect process variance known as overfitting where the Kalman estimates confide too much in the measurement data over the assumed model. This problem takes place when  $Q$  is significantly greater than  $R$ . It results in estimates that contain the measurement noise and that do not provide a better estimate of the behavior of the system.

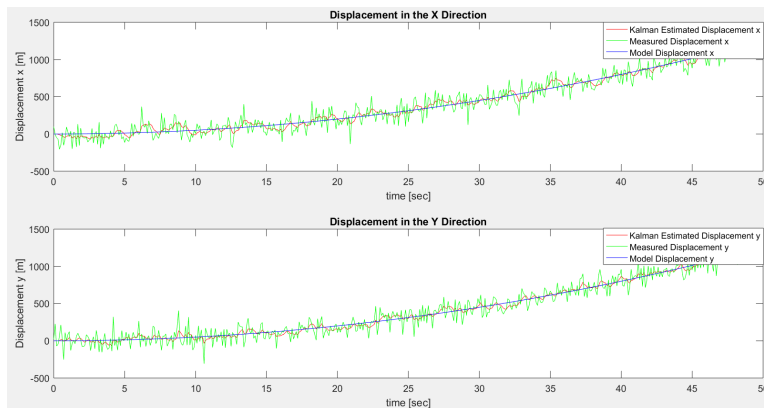


Figure 17: Incorrect process variance leading to an **overfitting** as shown above. The Kalman Estimate trusts the measurements over the model.

Figure 18 depicts another incorrect process variance known as underfitting where the Kalman estimates confide too much in the assumed model data over the actual measurements. This problem takes place when  $R$  is significantly greater than  $Q$ . This results in estimates that are dominated by the assumptions of the model.

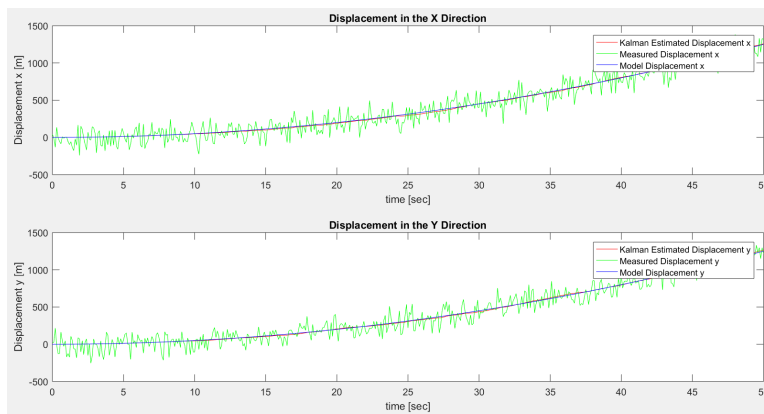


Figure 18: Incorrect process variance leading to an **underfitting** as shown above. The Kalman Estimate trusts the assumed model over the model.

The figure below proves a better fit as it confides in both the measurements and the assumed

model. When setting up a Kalman Filter, it is crucial to carefully decide on the process variance as it controls the behavior of the Kalman estimates.

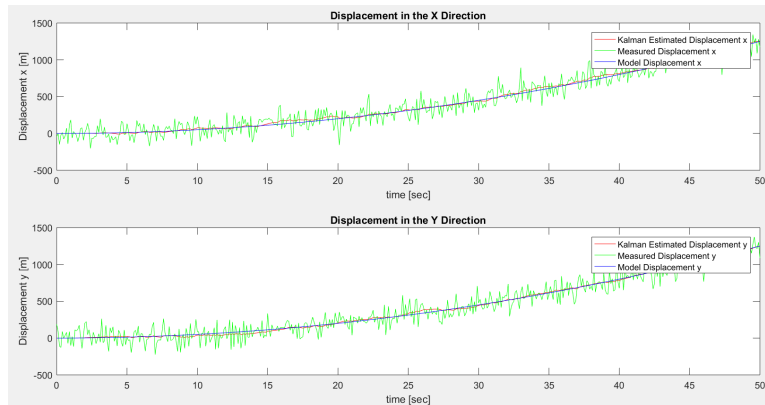


Figure 19: Correct process variance leading to an **good fitting** as shown above. The Kalman Estimate has a balanced trust in both the model and the measurements.

### 3.1.3 Extracting the Velocity from the Kalman Filter

After completing this practice Kalman Filter, the velocity state space variables for both the x and y direction were extracted. The purpose of this exercise is to see how a state parameter that is not a direct observable, such as coating absorption, might be extracted from the Kalman Filter. The graph below illustrates both velocity graphs of the two directions. Theoretically, if the behavior of the following results were based on the assumed model of the displacement, the velocity would be linear with a positive slope. This is since the displacement was parabolic and concave up. The graphs below are rising and linear regression can be applied to them to obtain the linear equations that best fit the data, but my goal was to transition into working on the project Kalman Filter. These plots were extracted by simply taking the third and fourth state space variables of the state-space representation of the model.

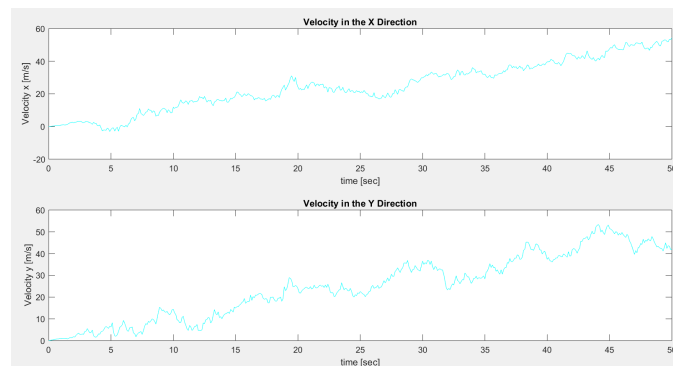


Figure 20: The extraction of both the x and y direction velocities from the state vector of the state-space representation of the system.

## 3.2 Coating Absorption Estimate Kalman Filter

After completing a practice Kalman Filter and gaining deeper insight into the steps involved within the recursive algorithm, I am now working on the implementation of a mirror degradation monitor. The ultimate goal is to create a real-time monitor that extracts estimates of the coating absorption while applying Kalman Filter theory.

### 3.2.1 Approach

With the first order exponential model described in section 2.3 and the detector's eigen-frequency measurements for a set period of time, the approach is to first create a Kalman Filter with simulated data and then test it. The model below is the normalized exponential that will be used for the state-space representation. After completing the testing of the filter, I will work on the extraction of the coating absorption and transition into the actual implementation of real data.

$$f(t) = \mu(1 - e^{-\frac{t}{\tau}}) \quad \text{Equation 26.}$$

### 3.2.2 State-Space Representation

To create the state-space representation one must first understand that we are taking a physical system as a set of input, output, and state variables. These components are related by first-order differential equations which I will find by starting with the unit step response of the system given by the model above. [5]

$$\begin{aligned} \mathbf{f}(s) &= \mathcal{L} \{ \mu(1 - e^{-\frac{t}{\tau}}) \} \\ &= \frac{N}{s^2 + Ds} \end{aligned} \quad \text{Equation 27.}$$

I took the Laplace transform operator  $\mathcal{L}$  and applied it to the analytical model and I obtained the expression above where the coefficients N and D correspond to the following variables,

$$N = \frac{\mu}{\tau} \quad D = \frac{1}{\tau} \quad \text{Equations 28 and 29.}$$

The transfer function of the system is then given by taking,

$$\frac{\mathbf{f}(s)}{P(s)} = \mathbf{s}\mathbf{f}(s) = \frac{N}{s+D} \quad \text{Equation 30.}$$

I then took the inverse Laplace transform of the transfer function which ultimately results in giving us the differential equation.

$$\begin{aligned} \mathcal{L}^{-1}\{\mathbf{f}(s)(s+D) = P(s)N\} \\ \dot{f}(t) + Df(t) = Np(t) \end{aligned} \quad \text{Equation 31.}$$

I then applied the differential equation above to show the state-space representation of the system as depicted below where  $\Delta k$  corresponds to the sampling period and  $z_k$  is the observation measurement and is a variable of interest. [5] The model below illustrates how I am working with a one degree of freedom system.

$$[f_k] = [1 - D\Delta k][f_{k-1}] + [N\Delta k][P_k] \quad (32)$$

**State-Matrix:**

$$A = [1-D\Delta k]$$

**Measurement-Matrix:**

$$C = [1]$$

**Input-Control Matrix:**

$$B = [N\Delta k]$$

**Observation Model:**

$$z_k = f_k \quad (33)$$

### 3.2.3 Parameters and Coefficients Estimation

With the state-space representation above, the following table shows the values of the parameters and the coefficients. The parameters were obtained as I outlined in the previous section by creating a parameterization of the COMSOL data through Matlab.

Gain( $\mu$ )	0.8114
Tau ( $\tau$ )	6.289
N	0.129
D	0.159

Table. 2.

### 3.2.4 Simulated Simple Kalman Filter Results

After completing the Matlab script with the given parameters, coefficients, and the state-space model described in section 3.2.2, I started by simulating a locked state. The control-input laser power is acting as a step function providing 0.1 W of power to the system. The graph below illustrates the shift in eigenfrequencies during the simulated lock with a comparison between the analytical model derived in section 2 with noisy eigenfrequency measurements.

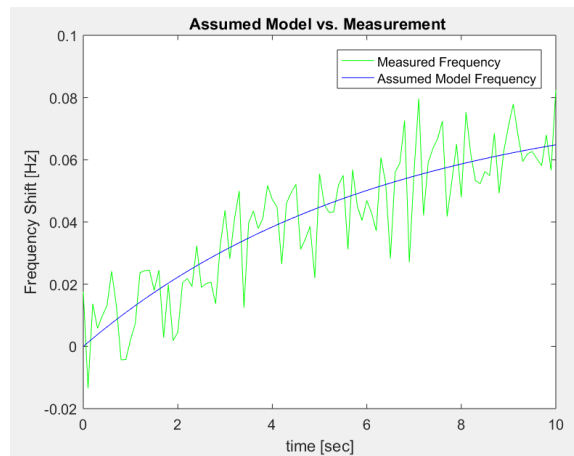


Figure 21: Simulated Lock State with the analytical model and simulated noisy measurements.

Once the Kalman Filter runs through the time interval and goes through the noisy measurements comparing them to the analytical model, the following graph depicts how the Kalman estimates compare to the measurements.

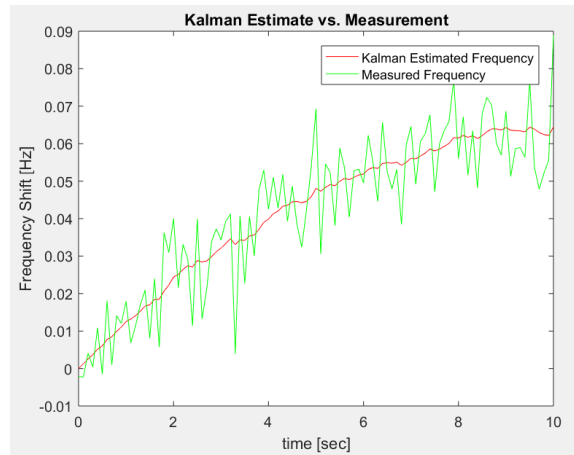


Figure 22: Simulated Lock State with the Kalman estimates from the MATLAB code and simulated noisy measurements.

This is how the monitor should ideally function as it runs through real-time eigenfrequency measurements, but the goal of the project is to extract estimates of the coating absorption. I reached a problem when realizing that this extraction would not be as efficient as first intended to.

### 3.2.5 Problem Extracting Coating Absorption

After completing the Kalman Filter, I arrived at a problem when reasoning through my strategy of extracting the coating absorption. Some questions I should have asked myself during the state-space representation process and while analyzing the situation include,

- Is there a parameter in the state space model that is proportional to the coating absorption?

In our case, the parameter associated with the coating absorption is the gain found in the exponential model defined as  $\mu$ . With this parameter, I then need to review the following questions.

- Is it possible to obtain the state-space representation of the gain? [5]
- Can I implement a Kalman Filter with the state-space of the gain? [5]

In order to obtain the state-space representation of the gain, this parameter must have a linear relationship to the system. The current state vector is one degree of freedom and



even though if I were to change it to two degrees of freedom it cannot include the gain. The gain needs to be linearized in order to implement the Kalman Filter. The common Kalman Filter only applies for linear systems. However, there is a version of a nonlinear Kalman Filter known as the Extended Kalman Filter which linearizes around the current mean and covariance. The issue with this type of filter is that it can result in high computational costs. [5]

The problem with the current Kalman Filter is that the Kalman Estimate Gain does not update itself. When simulating a noise in the measurements that would be representative of the change in coating absorption, the Kalman estimate does not update. This is due to the fact that the gain is found within the input-control matrix,  $B$ , as outlined in section 3.2.2. The input-control matrix in the Kalman Filter algorithm remains unchanged. During a lock state, when the laser power is nonzero at approximately 0.1 W, the coating absorption is correlated to the laser power. Due to this relationship which is described in the COMSOL section 2.2.3, the actual eigenfrequency measurements for the 8 kHz eigenmode are assumed to contain this coating absorption noise. However, the Kalman estimate does not update itself due to this noise in the measurements.

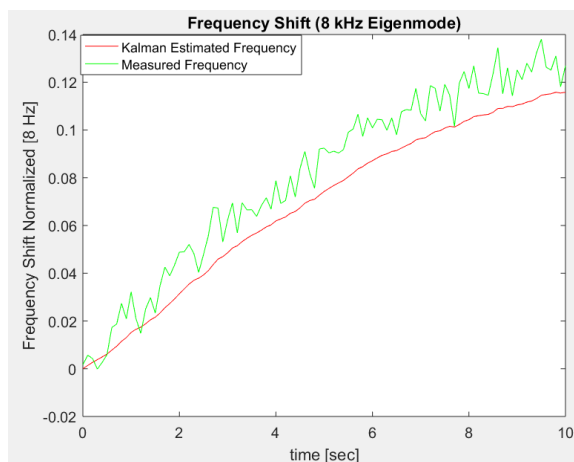


Figure 23: Simulated Lock State where the actual noisy measurements have a factor of 2 multiplied to the input laser power and the Kalman Estimate is not updated.

The following two graphs illustrate the problem. Figure 23 depicts an example where the measurements have an extra factor added to the input laser power. It changes the gain coefficient in the measurements as this noise increases the laser power's effect and can be analyzed as the coating absorption's effect. The problem is that the Kalman Estimate's gain coefficient remains unchanged with this measurement noise. Figure 24 below applies the same analysis, but in this case the Kalman Estimate's gain is greater than the measurements. The measurements have a noise simulating an effect where there is a factor reducing the laser power's effect. Once again, the Kalman Estimate does not experience a reduction in gain to match the behavior of the noisy measurements. Although both the Kalman Estimate and the noisy measurements graphs eventually converge to the same value during a lock, the difference in the early behavior of a lock is a problem since the coating absorption extraction is dependent on analyzing this time interval.

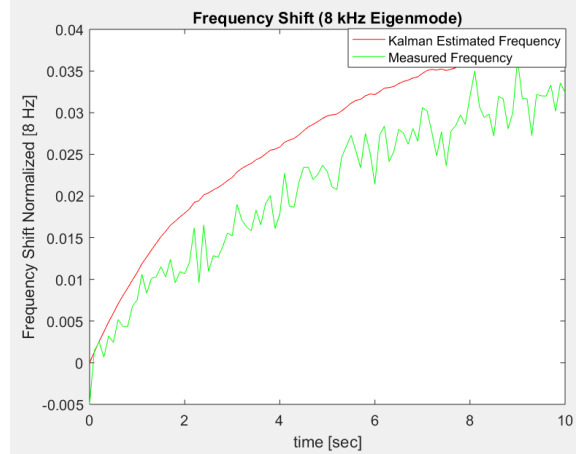


Figure 24: Simulated Lock State where the actual noisy measurements have a factor of .5 to the input laser power and the Kalman Estimate is not updated.

### 3.2.6 Creating a Nested Kalman Filter

The approach I took in order to monitor and change the gain of the Kalman Estimate was centered on creating a nested Kalman Filter. Although the gain can be linearized, due to the time constraint of the project, this measure will be postponed for later.

Applying the original Kalman Filter I created for this system as the basis of the nested filter, the strategy behind this more complex monitor involves updating the input-control model,  $B$ , and the process covariance at the end of each lock state. Since the process covariance is directly proportional to the product of  $B$  and its transpose, by updating  $B$  the process covariance changes as well. As I explained in the previous section, the gain is directly related to  $B$ . At the end of a lock state, the Kalman Estimate will either be above or below the measurement data as illustrated in figures 23 and 24. By measuring the average residual during this time frame between the measurement and the Kalman Estimate, this is then factored to the input-control model. The outer Kalman Filter is a simplified Kalman Filter where the model noise standard deviation,  $w$ , and the measurement noise standard deviation,  $v$ , are set at a value with no time evolution. As a result, I currently cannot provide estimates of the process covariance,  $Q$ , and the measurement covariance,  $R$ .

$$B_{updated} = B_{initial} + (\text{average residual}) * B_{initial} \quad \text{Equation 34.}$$

The equation above explains how the input-control model is updated at the end of each lock period. The gain is then updated at the end of each lock period and graphed separately to illustrate the change as the detector alternates between locked and unlocked states. With a larger sampling time, the gain can be updated to the point where the Kalman estimates' gain is closer to the actual measurement gain. After completing the Matlab code which implements this simplified nested Kalman Filter, I tested it with a simulated time frame of alternating locked and unlocked states to test two cases.

#### Simulation Results:

The first case involves encountering noisy measurements where the control parameter,

laser power, has an unknown factor greater than 1 resulting in a higher gain coefficient in the measurement data. As shown below, the nested Kalman Filter updates itself at the end of locked states to update its gain coefficient. This results in an increase in the gain and this change in gain is proportional to the coating absorption.

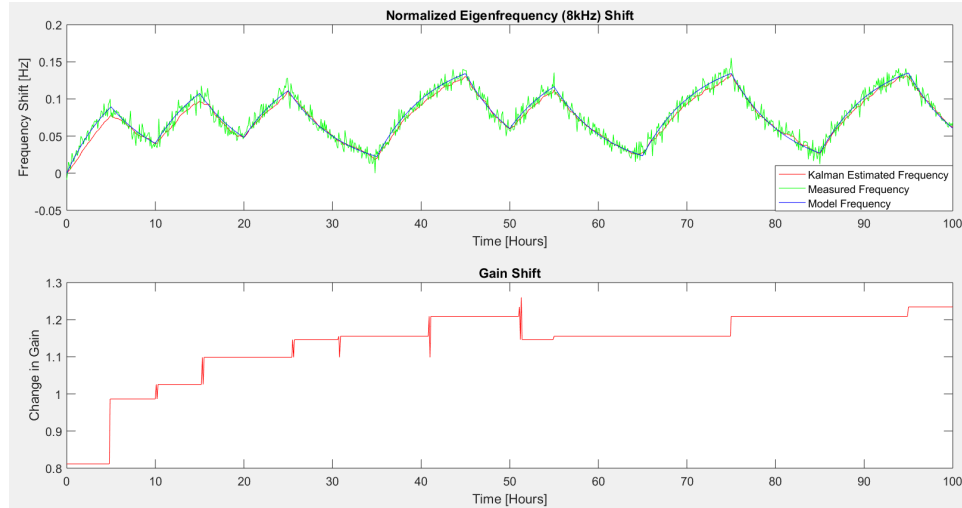


Figure 25: Simulated period of locked and unlocked states where the noisy measurements have a factor  $>1$  applied to the input laser power and the nested Kalman Filter updates itself in a bias towards this noise's behavior.

The second case involves encountering noisy measurements where the control parameter, laser power, has an unknown factor less than 1 resulting in a lower gain coefficient in the measurement data. As shown below, the nested Kalman Filter updates itself at the end of locked states to update its gain coefficient. This results in a decrease in the gain and this change in gain is proportional to the coating absorption.

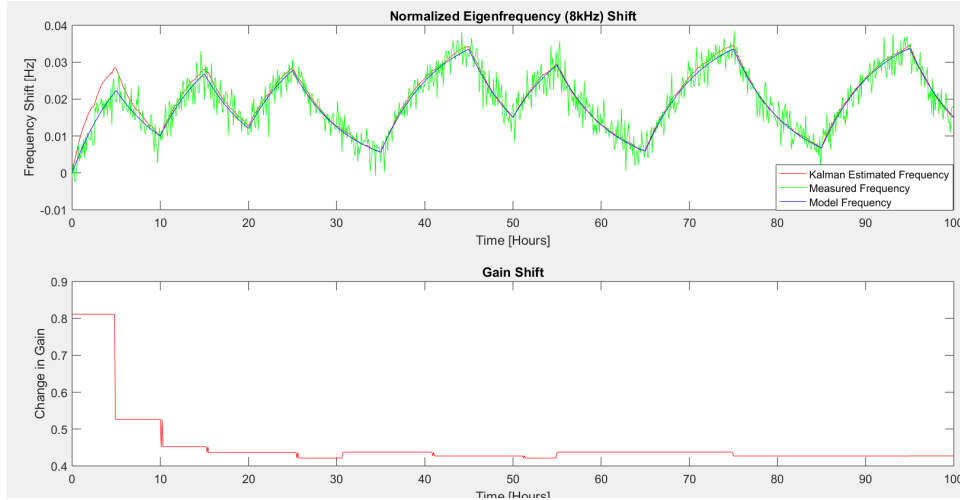


Figure 26: Simulated period of locked and unlocked states where the noisy measurements have a factor  $<1$  applied to the input laser power and the nested Kalman Filter updates itself in a bias towards this noise's behavior.

## 4 Future Work

Moving forward, these last three weeks I plan on testing my nested Kalman Filter with actual measurement data. IWAVE is a frequency tracking package that will be used to estimate eigenfrequencies in real time. I will be integrating IWAVE plot data provided by the researcher Timesh Mistry and focusing on analyzing the measurement noise to further understand its behavior. As described in the last section, depending on the behavior of the measurement noise, I can extract and analyze if there is a change in the coating absorption. Furthermore, while adapting the nested Kalman Filter to an actual data set of the month of July 2017 including both locked and unlocked periods, I will work on analyzing these results.

After obtaining results with actual eigenfrequency data, the next steps that need to be taken with this project revolve around improving the model of the system. The extraction of the estimated coating absorption can be completed by applying the linearization technique briefly discussed in section 3.2.5. Furthermore, the analytical model can be improved by running further studies to analyze higher order exponential models. By adding several other factors such as ambient temperature to the mirror degradation monitor, the monitor can be refined and further improved.

## 5 Conclusion

Overall, these past seven weeks have helped me learn different skills and have taught me how to manage my time. Although I have met challenges along the way, I have a clearer view on what I need to do to accomplish my goals these next few weeks. Most of my challenges have been technical while trying to navigate and write Matlab code. I have also met the

problem that the initial purpose of using the Kalman Filter to extract coating absorption is not directly applicable. With the guidance of my mentor I have been able to navigate around this problem through the implementation of a nested Kalman filter. My main goal is to complete the testing of the nested Kalman Filter on real data in order to see if this technique is a plausible mirror degradation monitor that can be utilized in the future.

I anticipate to continue working on my analytical skills while I test my nested Kalman Filter on actual eigenfrequency data from a specific period extracted from IWAVE plots. I plan on using my last weeks wisely to obtain and analyze results, while also working on setting a thorough list of further work that can be completed by someone who would like to continue working on this project. Challenges that I may face when testing the filter with real data can include having too noisy measurements. If I come across having a filter that is either over or under fitted, I will follow the tuning guidelines in section 3.1.2 that I learned while completing the practice Kalman Filter. Furthermore, I plan on testing it as soon as possible to quickly mitigate any technical issues I may come across and to transition into the analysis of the data smoothly these last weeks. Hence, I plan on dedicating the final week on transitioning into improving the model and providing through my final report a clear outline of what I have worked on for someone interested in conducting future work.

## References

- [1] S. Tait, *An Instantaneous Absorption Estimate of aLIGO Test Masses*. pgs. 1-26 (2018).
- [2] H. Wang, C. Blair, M. Dovalé Alvarez, A. Brooks, M. F. Kasprzack, J. Ramette, P. M. Meyers<sup>6</sup>, S. Kaufer, B. O'Reilly, C. M. Mow-Lowry, A. Freise. , *Thermal modelling of Advanced LIGO test masses*.. Class. LIGO Document. 26 Apr (2017).
- [3] S.C. Tait, I.W. Martin, C. Blair, R. Jones Z. Tornasi, A. Bell, J. Steinlechner, J. Hough,S. Rowan. *Optical Absorption of Ion Plated Coatings and Instantaneous absorption at LLO*. <https://dcc.ligo.org/DocDB/0150/G1800531/001/LVC2018.pdf>(2018).
- [4] S. Tait, *Thermal Modelling*. LIGO COMSOL models.
- [5] G. Valdes, *Data Analysis Techniques for LIGO Detector Characterization*. University of Texas at San Antonio (2017).
- [6] Kalman Filter Tutorial, *Water Tank Sample Tutorial*. University of North Carolina (2018).
- [7] COMSOL, *The Heat Transfer Branch*. 709745 ISSN 02608774 (2012).
- [8] A. Brooks, *Pure state space model of thermal effects*. LIGO-T16xxxx-v1-D (2018).
- [9] Mathworks, *Kalman Filter Example*. (2018).
- [10] E. Black, *Notes on the Pound-Drever-Hall technique*. LIGO-T980045- 00- D (1998).
- [11] Abbot B.P., and LIGO Science Collaboration. “Calibration of the Advanced LIGO detectors for the discovery of the binary black-hole merger GW150914.” *Phys. Rev. D*95, 062003 (2017). P1500248
- [12] M. Al-Ahdab, “2D Object Tracking Tutorial with Kalman Filter.” Aalborg University (2016).

## 6 Code

### 6.1 Practice Position Tracking Kalman Filter

```

clear
clc
close all
t_i = 0.1; %Defining the sample time

%The State Matrix Below
A = [1 0 t_i 0; 0 1 0 t_i; 0 0 1 0; 0 0 0 1];
%Output Matrix
C = [1 0 0 0; 0 1 0 0];
%B is the Input Matrix
B = [0.5*t_i^2 0; 0 0.5*t_i^2; t_i 0; 0 t_i];
x = [0;0;0;0]; % Initial Conditions

%Defining a system to generate model data
system = ss(A, B, eye(4), [], t_i);

t = 0:t_i:50 %The Time interval

%Below we are assuming the uncertainties in acceleration are all equal
stdax = 10; %standard deviation ax
stday = 10; %standard deviation ay

%In practice (below), these values are determined experimentally

%Defining the input accelerations:
inputax = ones(1, length(t));
inputay = ones(1, length(t));
u = [inputax; inputay];
x0 = [0;0;0;0];
%Now generating the state space representation of the model
Xmodel = lsim(system, u, t, x0);
xmodel = Xmodel(:, 1);
ymodel = Xmodel(:, 2);
velocity_xmodel = Xmodel(:, 3);
velocity_ymodel = Xmodel(:, 4);

%Defining V:
measurement_noise_v = [75.^2 0; 0 75.^2];

%Generating measurement data by adding noise to the modelled data: (This is
%simulation (actual measurement data already has noise))
xm = xmodel + normrnd(0, 100, length(xmodel), 1);
ym = ymodel + normrnd(0, 100, length(ymodel), 1);

%Initializing the matrices for the for loop
Xest = zeros(4, length(t));
Xest(:, 1) = x0;

%Defining R and Q
R = measurement_noise_v*C'*C'; %Measurement Noise covariance
Q = [stdax.^2 0; 0 stday.^2] %Process Noise covariance

%Initializing P
P = B'*Q*B';
for (i = 2:length(t));
    %Prediction Phase
    %The line below is predicting the error covariance ahead
    P = A'*P*A' + B'*Q*B'; %Predicting the covariance matrix
    Xest(:, i) = A*Xest(:, i-1) + B*u(:, i-1); %Predicting the State ahead
    %Correction Phase
    S = C'*P*C' + R
    K = (P*C')/S; % Calculating the Kalman Gain
    %Below I am updating the estimate with the measurement
    Xest(:, i) = Xest(:, i) + K*([xm(i); ym(i)] - C*Xest(:, i)); %Estimating the
    P = (eye(4) - K*C)*P; %This is updating the error covariance which then I
end;

```

### 6.2 Coating Absorption Simple Kalman Filter

```

clear
clc
close all

t_i = 0.1; % Sample Time



---


%% The Matrix Below is the State Transition Model
Gain= 0.1136/0.14;
Tau = 6.289;
c = 8126;
N = Gain/Tau;
D = 1/Tau;
A = [ 1-D*t_i];



---


%% C is the output matrix: It is the observation model that maps the true
%state-space into the observed space
C = [ 1 ];



---


%% B is the input-control matrix:
B = [ N*t_i ];
f = [0]; %Initial Conditions



---


%% Defining the State-Space Model
system = ss(A, B, eye(1), [], t_i);

t = 0:t_i:10; %The Time Interval (change)



---


%%
%The standard deviation in the control variable
std_laserpower = 0.07;
%% In practice (below), these values are determined experimentally
%Defining the input pow:
inputpower= 0.1*ones(1,length(t)) ;%normrnd(0, std_laserpower, 1, length(t))
u = [inputpower];
f0 = [0];
%% Now generating the state space representation of the model
Fmodel = lsim(system, 2*u, t, f0);
fmodel = Fmodel(:, 1);
%Defining V:
measurement_noise_v= [0.01.^2 ];



---


%% Generating measurement data by adding noise to the modelled data: (This
%simulation (actual measurement data already has noise))
fm = fmodel + normrnd(0, 0.006, length(fmodel), 1);
%dfm = dfmodel + normrnd(0, 50, length(dfmodel), 1);

%Initializing the matrices for the for loop
Fest = zeros(1,length(t));
Fest(:, 1) = f0;



---


%% Defining R and Q
R = measurement_noise_v*C*C'; %Measurement Noise covariance
Q = [std_laserpower.^2 ]; %Process Noise covariance



---


%% Starting Kalman Loop
P= B*Q*B';
for (i =2:1:length(t));
    %Prediction Phase
    %The line below is predicting the error covariance ahead
    P = A*P*A' + B*Q*B'; %Predicting the covariance matrix
    Fest(:,i) =A*Fest(:, i-1)+ B*u(:, i-1); %Predicting the State ahead
    %Correction Phase
    S = C*P*C' + R ;
    K = (P*C')/S; % Calculating the Kalman Gain
    %Below I am updating the estimate with the measurement
    Fest(:, i) = Fest(:,i) + K*(fm(i)-C*Fest(:,i)); %Estimating the State
    P = (eye(1)-K*C)*P; %This is updating the error covariance which then 1
end;

%When setting the input laser power as zero, there should be no change in
%the eigenfrequency since no heat is being applied (unlocked-cooling), when
%in locked state the eigenfrequency experiences an upward shift
subplot(111);
plot(t, Fest(1,:), 'r', t, fm, 'g', t, fmodel, 'b');
xlabel('time [sec]');
ylabel('Frequency Shift Normalized [8 Hz]');
title('Frequency Shift (8 kHz Eigenmode)');
legend('Kalman Estimated Frequency', 'Measured Frequency', 'Assumed Model F

```