Draft version July 29, 2022
Second Interim Report for Caltech LIGO SURF 2022

GW strain calibration of LIGO detectors with high precision and low latency

Naomi Shechter,[1]
Mentors: Ethan Payne,[2] and Dr. Alan Weinstein[2]

[1] *Department of Physics and Astrophysics, DePaul University*
[2] *Caltech LIGO*

ABSTRACT

The detection of gravitational waves (GW) has opened a new era of astrophysical observation, allowing scientists to view and analyze previously unseen phenomena. This process hinges upon measuring the strain $\Delta L/L$ of space over 4 km long baselines, which change by a differential arm (DARM) length on the order of 10^{-20} meters when a GW passes. From this information-rich time series, a wealth of astrophysical information may be deduced. In order to produce a reliable estimate of the strain, the Laser Interferometer Gravitational Wave Observatory (LIGO) detectors must be precisely calibrated. Furthermore, the calibration pipeline must produce an associated calibration uncertainty estimate with which to characterize the strain. While uncertainty estimates can currently be produced with low latency, it takes months to investigate the sources of error and verify the quality of calibration. Producing a high precision, low latency uncertainty estimate and a diagnostic monitoring software is therefore crucial for LIGO's fourth observing run (O4). It is this task that forms the basis of this research project. The uncertainty estimation and monitoring software are to be produced using pyDARM, a python package which implements the DARM control loop model and is currently under development. The software must reliably output uncertainty estimates and a suite of diagnostic plots on the timescale of an hour. It will actively be of use in O4.

I. PRINCIPLES OF OPERATION AND CALIBRATION FOR THE LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

1.1 - Operation of Laser Interferometer Gravitational Wave Observatory

The Laser Interferometer Gravitational Wave Observatory (LIGO) at Hanford and Livingston is among the most sensitive detectors ever built. This makes it all the more remarkable that LIGO functions similarly to the archetypal Michelson Interferometer. Due to this high sensitivity, the detector must be well calibrated to produce meaningful, unbiased results. However, calibration process cannot be adequately described without first explaining how LIGO operates. Laser light enters the detector and encounters the beam splitter, which sends it into two perpendicular, 4 km long arms (Fig.1). Each arm includes a Fabry-Perot cavity. The first mirror in the cavity is 99% reflective, so only 1% of the incident light enters the cavity. It builds in power significantly due to the nearly 100% reflective test mass at the other end of the arm. However, rather than produce a pattern of constructive interference after the split laser beams of the interferometer recombine, the light reflected by the arm cavities largely cancels the fresh light emitted by the laser. In other words, LIGO is an active null instrument that relies on destructive interference to maintain a baseline output from which deviations may be measured. When a gravitational wave (GW) passes, a transverse strain $\Delta L/L$ is induced in arms, which causes a minuscule differential arm (DARM) length change on the order of 10^{-20} meters. This causes a small phase shift in the light recombining at the beamsplitter, and thus a small amount of light exits the beamsplitter in the direction of the readout port.

1.2 - DARM Control Pipeline and Calibration Pipeline

A feedback control loop constantly attempts to maintain a null position of the interferometer. ΔL_{free}, the actual change of length in the arms, is measured in the presence of the feedback control loop which actively suppresses it through actuation. The DARM length ΔL_{res} is passed to the sensing function, which produces the error signal d_{err}. This is the input for the DARM control pipeline's digital filters and actuation function, which help maintain the

nominal positions of the test masses and resonance in the arm cavities (Fig.2). The d_{err} signal is also fed into the calibration pipeline, which uses a detailed reconstruction of the $1/C$ and A models to produce the gravitational wave strain. It does this by comparing the model to the measured response over different time epochs (2). A new epoch begins and new models are made whenever large scale changes are made to the detector. From the resultant response function and the outputs of various statistical algorithms, one may characterize the complete detector response along with the calibrated strain time series $h(t)$. The strain is comprised of GW data, displacement noise and sensing noise. To more fully understand the quality of $h(t)$, we also produce the strain uncertainty estimate, which is dependent solely on GW information.

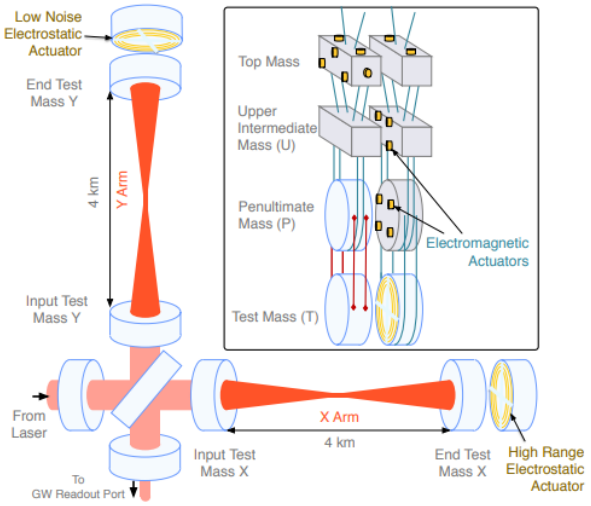

Figure 1. A simplified figure of the LIGO detector (1). Laser light enters from the left and encounters the beam-splitter. The test masses (inset) are constantly actuated to keep the detector on resonance and nearly complete destructive interference at the GW Readout Port (bottom left). The differential arm length is detected from the readout port as a change in intensity.

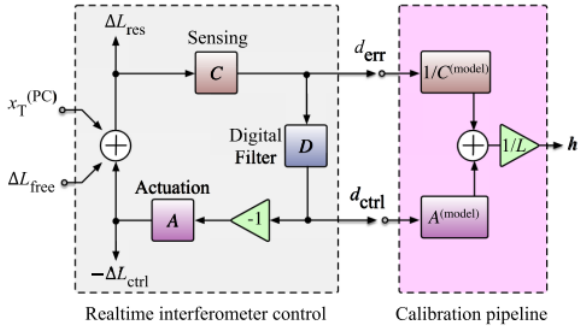

Figure 2. The LIGO control loop and the LIGO calibration pipeline (1). The sensing block C measures the change in length as the error signal d_{err} which the digital filter block D converts into d_{ctrl}. The control signal is then fed to actuation block A to actuate the test masses by the control length ΔL_{ctrl}. This ensures that the mirrors remain in their nominal positions. The calibration pipeline passes these same signals through filters $1/C^{model}$ and A^{model} which are fit by MCMC and GPR methods that compare the measurements to a model. The output of the calibration pipeline is the reconstructed strain h(t) and the uncertainty estimate.

1.3 - The Importance of Precision Low Latency Calibration and Uncertainty Estimates

Calibration is a crucial process to ensure that the aforementioned pipeline provides accurate data, allowing us to more fully understand GW sources. Given how sensitive LIGO must be to measure GW strain, it must also be as accurate as possible. Currently, a code called pyDARM is being utilized to model the DARM control loop and strain uncertainty estimation in a more accessible manner than has been done in O3 (2). Testing and supplementing pyDARM in order to develop a reliable low latency calibration uncertainty estimate for O4 forms the basis of my research.

As of the third observing run (O3) which lasted from April 2019 to March 2020, our calibration uncertainty within the sensitive band of 20-2000 Hz was known to $\sim 2\%$ in magnitude and $\sim 2°$ in phase (1). An uncertainty estimation graph illustrating this is given in Figure 3. There are several reasons why we continue to seek higher precision, although all are motivated by the prevention of biased astrophysical results (3, 4, 5, 6, 7, 8). Concerning tests of general relativity in the strong regime, only evidence collected with extremely reliable calibration would be capable of shedding doubt on the theory. For future observation runs such as O5, we expect an increase in sensitivity so significant that uncertainty must be lowered to 1% levels in order to reliably reconstruct signals of SNR greater than 100. Additionally, calibration uncertainty directly affects the parameter estimation uncertainty for both GW-based and multi-messenger astronomy(1). To summarize, many data analysis pipelines and research projects rely on high quality detector calibration. This makes it not only a crucial piece of LIGO's intricate construction, but one which must be well-understood, precise and efficient to produce reliable data.

Understandably, sub-optimal calibration introduces many issues; verifying calibration during O3 was a particularly time-intensive process (3). The $\sim 2\%$ level of uncertainty was only released after months of verifying the initial uncertainty estimate which was earlier estimated to be of order 5%. Furthermore, by the time the calibration model

C01 was completed, much of the parameter estimation had already occurred with the less precise C00 model (which actually lacked a reliable uncertainty estimate). In preparation for O4, the calibration group is working to expedite the process to verify the low latency strain uncertainty estimate. My research project will contribute to the realization of this goal.

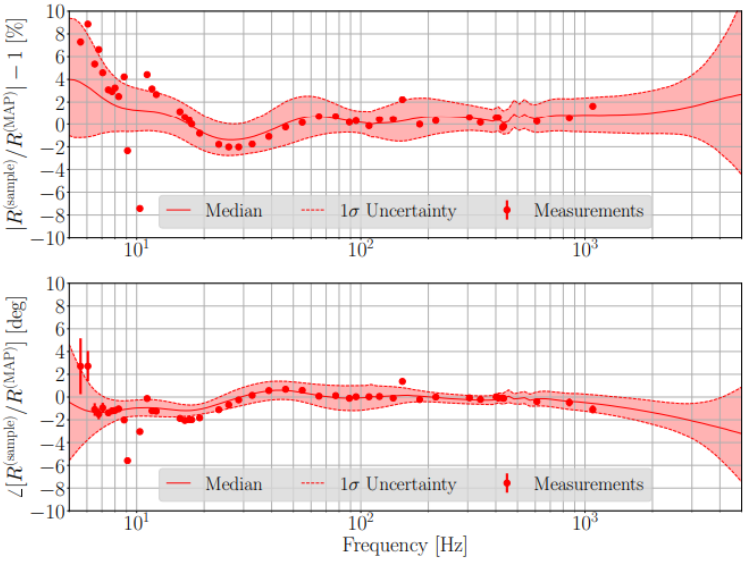

Figure 3. A graph of the combined error and uncertainty estimate from O3a at the Hanford detector. The data points are a variety of residuals between the modeled response function and the samples response function, compiled from many measurements. The solid line is the median of this relative response. It is the best estimate for the frequency dependent systematic error. The light red envelope outlines the $+1\sigma$ and -1σ boundaries on the uncertainty. While both the uncertainty and the standard deviation are large above 10 Hz, they even out significantly around 20 Hz. In the interval from 20 Hz to 1000 Hz, they barely pass positive or negative 2% in magnitude or $\sim 2^\circ$ in phase.

II. APPROACH AND OUTLINE FOR THE RESEARCH PROCESS

In order to expedite the process of generating calibrated strain uncertainty with high precision and low latency, I will be writing a monitoring program and producing a suite of diagnostic tools. The program will characterize the uncertainty output by pyDARM. Should it fall outside of satisfactory limits, the program will raise a specific warning as to what source of error must be further examined, along with plots of the calibration lines, time dependent correction factors (TDCFs) and other values which contribute to the overall uncertainty budget (1). The program must operate reliably, without crashing, produce these diagnostic tools with a latency of less than an hour. Once the monitoring program is able to correctly diagnose the issues found in O3, it will be automated with documentation in order to make it accessible to anyone seeking to understand the state of LIGO's calibration.

Developing a thorough understanding of pyDARM was necessary in order to produce a program which accurately characterizes uncertainty. Therefore, I spent the first week or so of this summer stepping through a Jupyter notebook by Evan Goetz and colleagues, which contains various demonstrations on sensing and actuation function models, as well as Markov Chain Monte Carlo (MCMC) and Gaussian Process Regression (GPR) algorithms. My understanding of the algorithms has been supplemented with several papers (9, 10). In the process of producing plots and encountering errors, I have become more familiar with the various pyDARM classes. I have also done some introductory learning about control systems and null instruments to more fully understand the LIGO control pipeline (11).

After acquiring this background information, I constructed several pyDARM-compatible model files using older files from O3. New models of the sensing, actuation and digital filter transfer functions are produced after significant changes to the detector, and demarcate periods called epochs. Once these were complete, I began producing the uncertainty via the previously described MCMC and GPR methods. The next step is to verify the uncertainty estimate by graphing it, after which I may move onto the next phase of my research project. During this stage, I will produce a monitoring program which will quantify the status of the reliability of the calibration uncertainty. It will

also raise specific warnings should any part of it be unreliable. Due to the emergent nature of this project, the exact manner in which the monitoring program will be constructed is as of yet unknown. However, it will likely include sampling the results of the GPR once an hour, and saving only as much data as is needed to produce a reliable model of the overall response. From this information, I will produce graphs which keep track of the time dependent behavior of the uncertainty.

III. PROGRESS IN REACHING PROJECT MILESTONES

Most of my first week and a half was spent reading papers, exploring the pyDARM code and building the background knowledge necessary to carry out my research. In this section, I will outline my understanding and share plots which are illustrative of the various subjects I have been introduced to thus far.

3.1 - Linear Time Invariant Systems, Control Systems and Null Instruments

Before interacting with pyDARM, I sought out resources about linear time-invariant (LTI) systems, control systems and null instruments (11). Consider a mass on a spring, to which you provide an impulse by hitting it suddenly. The function for driven dampened simple harmonic oscillators defines the impulse response, and the result would be a decaying sinusoid as the oscillating mass eventually comes to rest. If you model a more complicated input signal as a sum of impulses, the output of an LTI system will be the linear superposition of the impulse response from each individual impulse. In mathematical terms, the output is the convolution of the impulse response with the input. For LTI systems, convolution becomes multiplication after performing a Fourier transform, so it is simpler to work in frequency domain. Therefore, rather than the impulse response determining the output of the system, we use the Fourier transform of the impulse response: the transfer function. In the context of LIGO, rather an an impulse, the excitation signal is a swept sine stepping through a broad range of discrete frequencies, across the sensitive band of 20-2000 Hz.

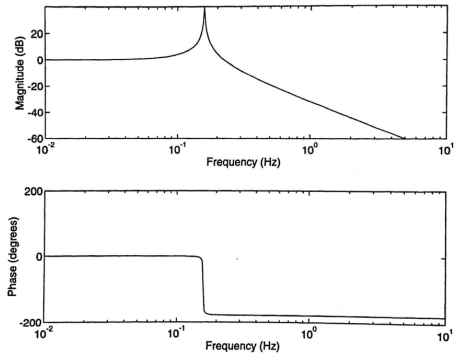

Figure 4. Bode plot of a mechanical oscillator from Fundamentals of Interferometric Gravitational Wave Detectors by Peter R. Saulson. The large peak indicates the resonant frequency, at which the phase passes through 90° (11).

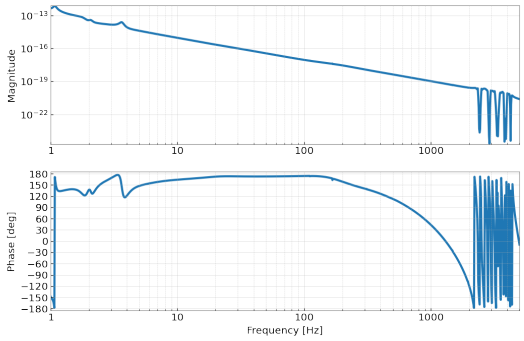

Figure 5. The complete actuation transfer function for the LIGO Hanford calibration model produced on April 16th 2019. The decreasing slope of the upper magnitude graph is similar to the decreasing slope which occurs in Figure 4 at frequencies higher than resonance.

The transfer function of a driven damped simple harmonic oscillator, such as a pendulum, is plotted over a range of frequencies in Figure 4. The large spike in the magnitude of oscillation is resonance, at which the driving frequency equals the natural frequency of the oscillator. As the two frequencies become completely out of phase, the magnitude becomes negligible. This means that there is a pole located at resonance, or that (should we act this transfer function upon a signal), frequencies around resonance would be amplified. There are also zeroes, at which the signal goes to zero and nearby frequencies are attenuated. In other words, poles and zeroes are a very useful method to quantify transfer functions, and allow us to construct filters. For example, actuation transfer function is modeled in such a way to reduce noise from the movement of the test masses (Fig.5). The upper intermediate mass and penultimate mass each contribute to the overall transfer function, and much like in Figure 4, they attenuate high frequencies. Effectively, the entire transfer function works as a digital low pass filter.

3.2 - Markov Chain Monte Carlo and Gaussian Process Regression

After I produced several plots of the the transfer functions in pyDARM, I moved on to understanding the way in which we compare and fit the modeled detector response with measured data. The first of these which I encountered was the Markov Chain Monte Carlo (MCMC) method (9). Given a prior distribution and a likelihood function of our model values and measured data, MCMC generates a distribution for the model parameters which are sampled for the maximum a posteriori (MAP) values. These MAP parameters correspond to the model which has the highest posterior probability, and they are used to construct the best fit curve. In the case of this project, I performed MCMC analysis on many individual measurements within each epoch. The file was set up such that a list of models and measurements could be provided as command line arguments, and each measurement object would be initialized with the corresponding model for the epoch it is in. However, as demonstrated in Figure 6, fitting with the MCMC code did not completely account for systematic error and does not adequately fit the data by itself. The residual systematic errors are handled by Gaussian Process Regression (GPR) in Figure 7. The GPR code takes the MAP parameter results of all measurements for a given epoch as input. It stacks these measurements, and then produces a probability distribution of many possible functions, each of which fits the systematic error (9). As more measurements are provided, the uncertainty envelope becomes more narrow. Compared to the graphs produced via the MCMC code, the variance from the GPR more fully captures the uncertainty between the model and measurements. This uncertainty arises both from an incomplete understanding of the detector and statistical errors in measurement.

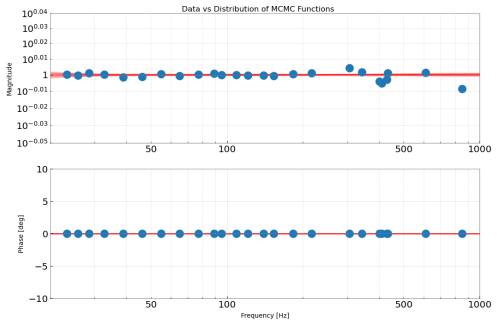

Figure 6. A graph of 100 possible fits to the residuals between the sensing function model for the 2019-4-17 epoch and measurements collected that same day. Produced by MCMC in the sensitive range from 20-2000 Hz. While the uncertainty in each individual data point is small, the deviation from the model is significant surrounding 500 Hz due to violin modes. The MCMC alone does not adequately fit the data or quantify the uncertainty between model and measurement. This is why we must employ Gaussian Process Regression.

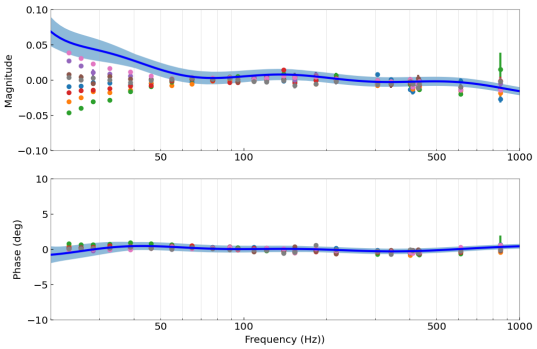

Figure 7. The corresponding GPR graph, in the sensitive region from 20-2000 Hz. The plotted points are all the measurements from the 2019-04-17 epoch. The envelope represents the standard deviation away from the blue line, which is the best fit curve. Deviation from the model is more reliably characterized across the frequency band, although issues remain in the magnitude plot below 50 Hz.

Once the full MCMC chain and the GPR posteriors were saved, I was able to create uncertainty configuration files for each measurement. These configuration files, along with the model configuration files, were used to compute a model of the DARM control loop and DARM response function uncertainty. While I have yet to produce a diagnostic plot of a specific measurement's uncertainty, there are more general details I can discuss. For example, number of quantities contribute to the uncertainty budget. These include time dependent correction factors such as $\kappa_c(t)$, $\kappa_{tst}(t)$ and $f_{cc}(t)$ which are the scalar gain factor of the sensing function and the test mass actuation function, and the coupled cavity pole frequency respectively. The final suite of diagnostic plots will provide graphs illustrating how the correction factors vary.

IV. CHALLENGES AND LEARNING EXPERIENCES

Most of the issues I have encountered so far have been about plotting or how certain pyDARM functions store data, and all have been learning experiences about paying attention to detail. For example, after running the MCMC,

you reset the parameters of the sensing function to the MAP values. However, this didn't take into account that one parameter (the detuned spring quality factor) should actually be inverse based on the equations for the sensing transfer function (Fig. 2). This was something which had to be fixed in order for the MCMC graphs to be accurate. An issue was also encountered with the GPR plots. The previously written code no longer seemed to work, and could have used a more efficient pyplot function to shade in the envelope. I produced a solution which was initially only valid on the left side of the graph and did not properly account for the increased uncertainty at higher frequencies. However, after discussing with my mentor Ethan, I was able to solve this issue by changing the frequencies over which I was fitting the data in both the MCMC and GPR files. Another longstanding issue did not have to do with the code I wrote at all, but rather an update to the package scikit-learn which caused the GPR uncertainty array to output with the incorrect shape. After my mentor Ethan's solution was merged into pyDARM, I received a small introduction to git. Currently, I am investigating graphs of the uncertainty estimate which I have produced for each measurement. I would also like to determine why the GPR in Figure 7 seems to fit the data more poorly than the MCMC code below 50 Hz, and fix this if it is indeed an issue.

The challenges I am anticipating involve the learning curve which is sure to become apparent as I begin writing the monitoring program. In particular, I am unfamiliar with automating operations such as sampling for values every hour. Given that the program must be reliable and robust, this is is a challenge which I will have to face. I can also expect to learn how to more fully interpret the code and data with which I have been making plots. For example, given a plot such as Figure 3, how does one determine when enough points are sufficiently uncertain to raise a warning? How does one quantify the various contributions to the hourly uncertainty estimate, or automate a page of diagnostic plots which updates on the same timescale? By learning the answers to questions such as these through experience, I believe I will be challenged in a meaningful, educational way.

IV. ACKNOWLEDGMENT

This report was funded by the National Science Foundation (NSF) and completed as part of the LIGO SURF Program at Caltech, in contribution to the LIGO Scientific Collaboration. Many thanks to my mentors Dr. Alan Weinstein and Ethan Payne.

REFERENCES

[1]Sun, L., Goetz, E., Kissel, J. S., et al. "Characterization of systematic error in Advanced LIGO calibration." Classical and Quantum Gravity, 37, 225008. doi:10.1088/1361-6382/abb14e (2020).

[2]PyDARM Gitlab Repo of the Laser Gravitational Wave Interferometer Calibration Group.

[3]Conversations with Mentors concerning background knowledge and the calibration process, during the first week of the LIGO SURF Program.

[4]Payne, E., Talbot, C., Lasky, P. D., et al. "Gravitational-wave astronomy with a physical calibration model." PhRvD, 102, 122004. doi:10.1103/PhysRevD.102.122004 (2020).

[5]Vitale, S., Del Pozzo, W., Li, T. G. F., et al. "Effect of calibration errors on Bayesian parameter estimation for gravitational wave signals from inspiral binary systems in the advanced detectors era." PhRvD, 85, 064034. doi:10.1103/PhysRevD.85.064034 (2012).

[6]Vitale, S., Haster, C.-J., Sun, L., et al. "physiCal: A physical approach to the marginalization of LIGO calibration uncertainties." PhRvD, 103, 063016. doi:10.1103/PhysRevD.103.063016 (2021).

[7]Essick, R. & Holz, D. E. "Calibrating gravitational-wave detectors with GW170817." Classical and Quantum Gravity, 36, 125002. doi:10.1088/1361-6382/ab2142 (2019).

[8]Huang Y., Chen H.-Y., Haster C.-J., Sun L., Vitale S., Kissel J. "Impact of calibration uncertainties on Hubble constant measurements from gravitational-wave sources." arXiv:2204.03614 (2022).

[9]Cahillane C. et al. "Calibration uncertainty for Advanced LIGO's first and second observing runs." Physical Review D 96.10 102001. (2017).

[10]Wang, J. "An intuitive tutorial to Gaussian processes regression." arXiv:2009.10862 (2020).

[11]Saulson, Peter R. "Fundamentals of Interferometric Gravitational Wave Detectors". Singapore: World Scientific (2017).