

Implementing Real-Time Calibration in Advanced LIGO Control Software

Dane Stocks and Joseph Betzwieser

LIGO Livingston Observatory

LIGO SURF Program

DCC T1700318

July 12, 2017

1 Introduction

Advanced LIGO (Laser Interferometric Gravitational-Wave Observatory) is a system of enhanced Michelson interferometers designed to measure relative changes in the spacetime interval between two sets of test masses. The arms of a detector contain 4 km-long Fabry-Perot cavities, which are filled with light from an Nd:YAG 1064 nm laser (Figure 1a). Laser light continuously resonates in these cavities, forming extremely stable beams which destructively interfere at the anti-symmetric port of the detector. A passing gravitational wave produces a differential change in the X and Y arm lengths, ΔL_{ext} , which in turn causes a phase shift in the beams to be recorded at the photodetector. When the cavities are held on resonance, or on “lock,” the laser power fluctuations measured at the photodetector are proportional to gravitational wave strain.

In practice, the strain incident upon the interferometer is due to various sources of noise, and not just the astrophysical signals of interest. Multiple design measures have been taken to reduce noise and help reach the necessary detector sensitivity. The mirrors at the end of each Fabry-Perot cavity are the bottom stages of quadruple-pendulum systems [6], and the interferometer components rest on active seismic isolation platforms, which assist in keeping lock at low frequencies by reducing motion to the level of 10^{-11} – 10^{-12} m / $\text{Hz}^{\frac{1}{2}}$ between 1 and 10 Hz [4].

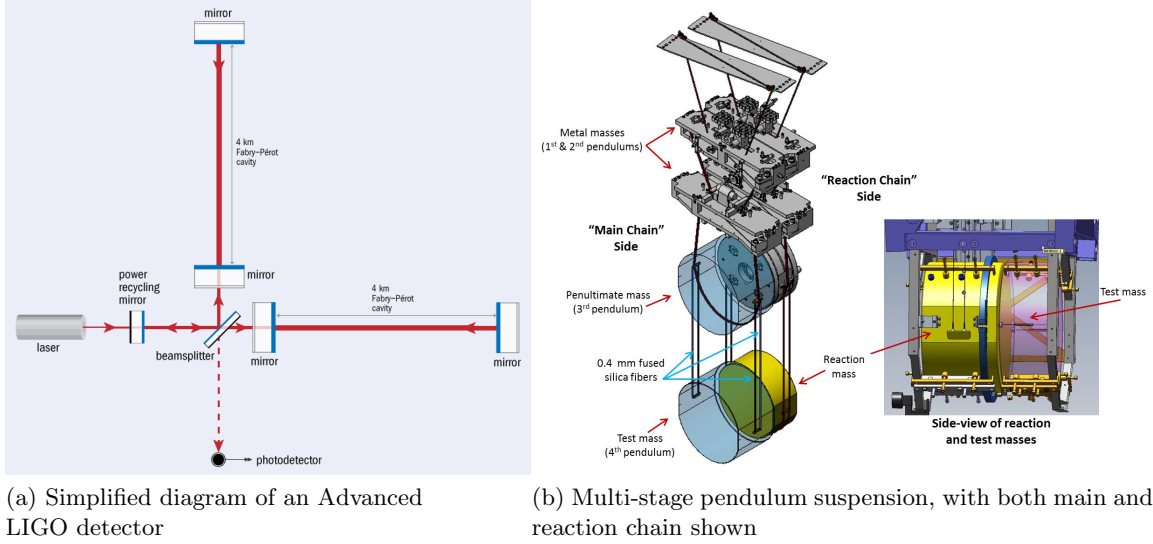


Figure 1: Basic detector infrastructure

Unwanted residual displacement remains, however, and additional methods are needed to keep the interferometer in resonance: in Advanced LIGO, this is largely accomplished through the use of five length sensing control loops. The differential arm (DARM) length control loop is the main focus of calibration efforts, as the final gravitational wave time-series $h(t)$ is produced from DARM length

$$\Delta L_{ext}(t) = L_x(t) - L_y(t) = h_{ext}(t)L, \quad (1)$$

where $L_x(t)$ and $L_y(t)$ are the distances from the input test mass to the end test mass of the X and Y arms at time t , $h_{ext}(t)$ is strain due to gravitational waves and external sources of noise, and $L = 3994.5$ m is the mean arm cavity length [2].

The outline of this paper is as follows: Section 2 provides an overview of the DARM loop, Section 3 explains how parameters of the DARM loop are determined through calibration, Section 4 details the structure of the current calibration pipeline, and Section 5 introduces a new pipeline, designed to run in the front end computers, and describes the progress made in constructing this model and results of its testing thus far.

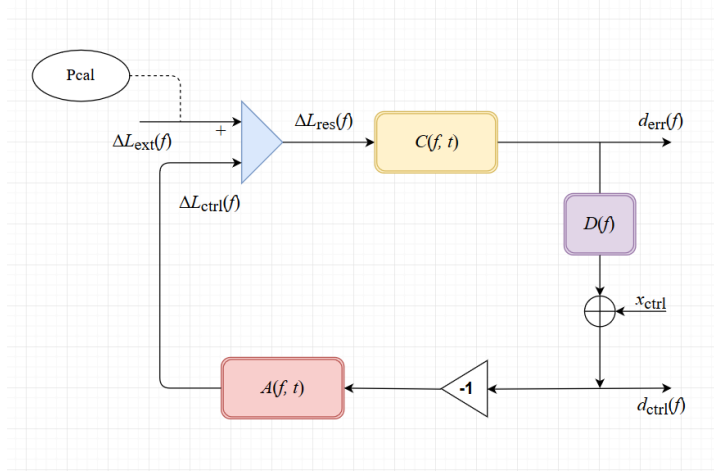


Figure 2: The DARM control loop. P_{cal} and x_{ctrl} are injection points that are used for determining the values of the time-dependent κ_C , κ_T , and κ_{PU} .

2 The DARM loop

A real-time control system, the CDS (Control and Data System), operates in the front end computers of the interferometer. Sensor and actuator electronics are placed throughout the detector, and PCIe input/output chassis stream data from these electronics to the front end clusters. These computers perform control algorithms and operate the DARM loop in real-time.

This feedback system contains three transfer functions, each of which is a complex-valued function of frequency (subject to time-dependent corrections).

2.1 Sensing function

The length sensing function, $C(f, t)$, describes the interferometer's optical response to $\Delta L_{res}(f)$. That is, it transforms residual displacement into a digitized error signal, d_{err} , which represents laser power fluctuations at the photodetector. The constituent parts of $C(f, t)$ include:

- $\kappa_C(t)$ – optical gain, dependent upon the amount of light stored in the Fabry-Perot cavities.
- $(1 + if/f_c(t))^{-1}$ – coupled-cavity pole between signal-recycling and arm cavities; f_c is the pole frequency (388 Hz at LLO [2]). Like $\kappa_C(t)$, the

coupled-cavity pole varies with time because of thermal fluctuations in the test mirrors.

- $Q(f)$ – time-independent part of sensing function, including detector response to laser power, frequency response of the ADC electronics in the sensing chain, and the time correction τ_C due to light travel in the cavities L/c as well as digitization and computational delays in the electronics.

We can write

$$C(f, t) = \frac{\kappa_C(t)}{1 + if/f_c(t)} Q(f) = \frac{d_{err}}{\Delta L_{res}(f)}. \quad (2)$$

2.2 Digital filters

A set digital filters transforms the digital error signal into a digital control signal in double-point precision [7]:

$$d_{ctrl}(f) = D(f)d_{err}(f). \quad (3)$$

These filters are known exactly, and unlike the sensing and actuation models, introduce no uncertainty into the overall frequency response function of the interferometer, $R(f)$.

2.3 Actuation function

The main-chains of the suspended cascading pendula possess twin reaction-chains that are suspended in parallel (Figure 1b). Actuation is provided separately to each stage of the main-chain via the corresponding mass of the reaction-chain. The digital control signal, d_{ctrl} , is distributed to each actuator in the form of voltage, and a reaction force displaces each test mass at the ends of the detector: $\Delta L_{ctrl}(f) = -A(f, t)d_{ctrl}(f)$, where the minus sign is included by convention. On the upper intermediate (U) and penultimate (P) stages, digital-to-analog converters drive electromagnets to induce a magnetic force on the masses of the main-chain. On the test (T) stage, a DAC converter drives an electrostatic system which produces dipole-dipole interactions between the test mass and a series of electrodes attached to the corresponding reaction-chain stage [3]. This system is called the electrostatic drive (ESD).

The actuation response of the test mass stage is known to vary over time. Charge accrues on the test masses during observation runs and alters the

strength of the electrostatic drives [5]. While the strength of the actuators of the upper intermediate and penultimate stages are not expected to fluctuate with time, they are tracked in the event of unexpected equipment failure. Hence, the time-dependency of the actuation function is tracked via two variables: 1) $\kappa_T(t)$, the test stage actuation scaling factor, and 2) $\kappa_{PU}(t)$, the combined actuation scaling factor for the P and U stages.

Explicitly, d_{ctrl} is distributed to each actuator and multiplied by a series of digital filters, $F_i(f)$, where $i = T, PU$ to represent the test and combined penultimate and upper intermediate stages, respectively. If we define $A_T(f)$ and $A_{PU}(f)$ as the frequency dependence of each stage's actuator, then the total actuation is written as

$$A(f, t) = \left[\kappa_T(t)F_T(f)A_T(f) + \kappa_{PU}(t)F_{PU}(f)A_{PU}(f) \right] e^{-2\pi i f \tau_A} \quad (4)$$

where τ_A is the delay involved with digital-to-analog conversions between d_{ctrl} and the voltages across the actuators.

2.4 Response function

The purpose of the DARM loop is to provide a complete and accurate model of the interferometer's response to external strain at a wide range of frequencies. Abstractly, we need to characterize the detector's response, $R(f)$, to infinitesimal external displacements ΔL_{ext} to be able to extract information about incident GW signals:

$$\Delta L_{ext}(f) = R(f)d_{err}(f). \quad (5)$$

Inspection of Figure 2 shows that

$$d_{err} = d_{err}D(f)(-1)A(f)C(f) + \Delta L_{ext}(f)C(f)$$

$$d_{err}(1 + D(f)A(f)C(f)) = \Delta L_{ext}(f)C(f)$$

$$d_{err} = \frac{\Delta L_{ext}(f)C(f)}{1 + G(f)}$$

Where for simplicity we have approximated $C(f, t)$ as $C(f)$ and ignored its time dependent factors (same for $A(f, t)$). Additionally, we define the open loop transfer function $G(f) = D(f)A(f)C(f)$. From (5) it follows

$$\Delta L_{ext}(f) = R(f) \frac{\Delta L_{ext}(f) C(f)}{1 + G(f)}$$

$$R(f) = \frac{1 + G(f)}{C(f)}.$$

We can re-write (5) using (1) to explicitly solve for h :

$$h(f) = \frac{1}{L} \left(\frac{1 + G(f)}{C(f)} \right) \cdot d_{err}.$$

Notice from (3) that d_{err} can be distributed as

$$h(f) = \frac{1}{L} \left(C(f)^{-1} \cdot d_{err} + D(f)A(f) \cdot \frac{d_{ctrl}}{D(f)} \right)$$

$$h(f) = \frac{1}{L} \left(C(f)^{-1} \cdot d_{err} + A(f) \cdot d_{ctrl} \right). \quad (6)$$

$h(t)$ is calculated in the calibration pipeline according to (6), with convolution replacing direct multiplication as the operation between the DARM signals and the transfer function models (as both are converted to the time domain for computation speed).

3 Modeling loop through calibration

Minimizing the uncertainty present in $h(t)$ involves making precise measurements of the parameters in the actuation function A and sensing function C which are not known absolutely.

One method used to make such measurements involves a radiation pressure actuator called a *photon calibrator* (Pcal). A Pcal device placed near each end test mass (ETM) shoots 1047 nm light from a power-modulated Nd³⁺:YLF laser onto the front surface of the mirror [3], where it reflects and is recorded by an auxiliary photodetector. Photon calibration can produce signals at specific frequencies, $x_T^{PC}(f)$, that are much larger than ΔL_{ext} at that frequency. These signals are known as calibration “lines” after the way they appear in frequency plots of DARM response, and they facilitate high-precision measurements of DARM suppression.

Similarly, digital excitations x_{ctrl} can be injected into the DARM loop to quantize certain parameters of the sensing and actuation functions. A table describing the use of Pcal and digital excitations appears below [8].

| Calibration lines at LLO | | | | |
|--------------------------|-------------|------------|------------|------------------------|
| Num. | Sym. | Freq. (Hz) | Type | Purpose |
| 1 | f_{tst} | 35.9 | ETM ESD | TST + PU actuation |
| 2 | f_{pcal} | 34.7 | Pcal | DARM control actuation |
| 3 | f_{ctrl} | 33.7 | x_{ctrl} | DARM control actuation |
| 4 | f_{pcal2} | 331.3 | Pcal | DARM sensing |
| 5 | f_{pcal3} | 1083.1 | Pcal | Calibration check |
| 6 | f_{pcal4} | 3001.1 | Pcal | Calibration check |

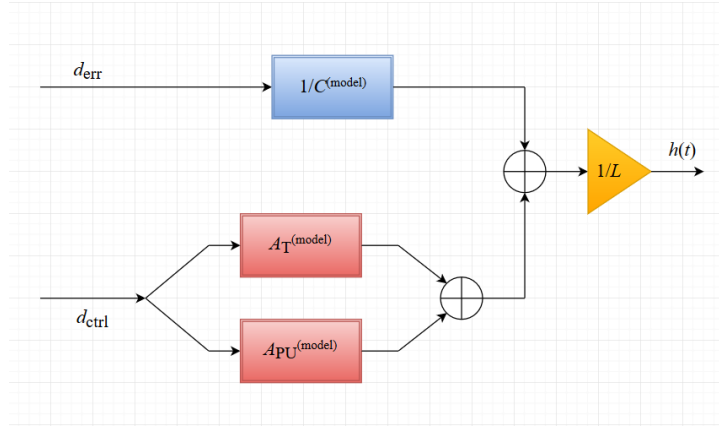


Figure 3: The DCS calibration pipeline uses models of the inverse sensing and actuation functions to calculate $h(t)$, the output strain of the detector.

3.1 Actuation model calibration

In the future I will describe methods used in calculating the different components of $A^{(model)}$.

3.2 Sensing model calibration

In the future I will describe methods used in calculating the different components of $C^{(\text{model})}$.

4 Current calibration model

For its first and second observation runs (O1 and O2), Advanced LIGO reproduced $h(t)$ using a dual calibration system composed of the Global Diagnostic System (GDS) and DCS search pipelines. Calibration begins in the front end computers, where real-time processors perform data conditioning through IIR filter modules. These modules are designed and implemented according to the below scheme:

- An operator designs control applications using the CDS Real-Time Code Generator (RCG). Most commonly, this is done through the graphical editor Simulink. This program allows the user to design empty filter banks and create all connections between the code blocks of a calibration model. Essentially, Simulink is used solely as a convenient way of allowing the user to describe how the model's many different parts should communicate.
- RCG Perl scripts read and parse the Simulink model files, creating a real-time executable, an EPICS database for communication between the CDS network and this specific user application, as well as an EPICS executable that allows the application to communicate with the EPICS database [1]. The real-time executable contains the user application as defined by the Simulink model and written in C code, as well as IIR filtering software.
- The individual filters are designed in either Matlab or LIGO's Foton program and installed in the real-time application as second order section filters. For a complete description of the Standard Filtering Module (SFM) software which implements these IIR filters, see LIGO-T0900606-v2.
- The real-time executable is installed as a kernel object in the operating system of the interferometer's front end computers.

After passing through the CALCS model, partially calibrated data is broadcast to the Data Monitoring Tool (DMT) via the Data Acquisition System (DAQ) Data Concentrator and stored in shared memory. The GDS

pipeline, written in `gstlal`¹ codebase, locates this shared memory location and reads data which include the following [7]:

- (a) DARM error signal, d_{err} , written in frame format
- (b) DARM control signal, d_{ctrl} , written in frame format
- (c) Excitation channel into which lines are driven, x_{ctrl} , written in frames
- (d) Frequency of line used to compute $\kappa_C(t)$, the time-varying gain of the sensing function, stored in standard file format
- (e) Time domain actuation, inverse sensing, and d_{ctrl} whitening filters determined from frequency domain calibration model
- (f) The Online Detector Characterization state vector representing the state of the detector, stored as frames

The pipeline produces as output a time series for $h_{ext}(t)$ at 16384 Hz.

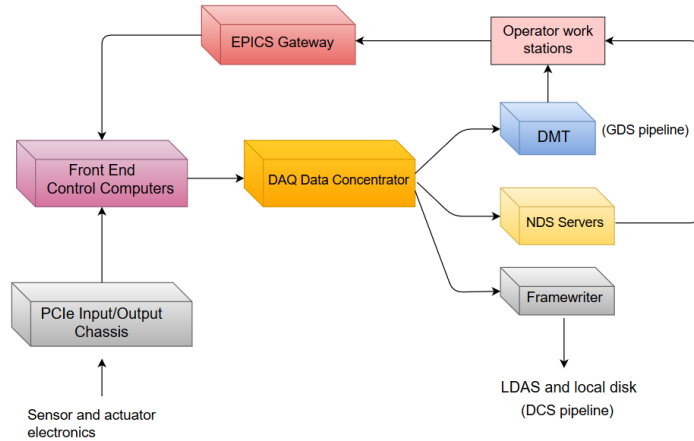


Figure 4: Flowchart of aLIGO CDS hardware. Network Data Servers (NDS) provide live and stored data to users at work stations, and EPICS communications allow for real-time interaction with control software executables.

Online calibration is prone to error, however, and a method is needed to refine calculations after signals from astrophysical events have passed.

¹A combination of LIGO Algorithm Library (LAL) tools wrapped in GStreamer

For instance, in LIGO’s fifth science run, the entire observation data set had to be re-calibrated four times [7]. In Advanced LIGO, uncalibrated measurements of d_{err} and d_{ctrl} from the front end computers are sent to the DAQ Framewriter and written to disk. The DCS pipeline functions offline as a self-contained system which transforms DARM error and control signals into external strain $h_{ext}(t)$. A view of Advanced LIGO’s entire calibration scheme for O2 can be found here: LIGO-G1501518-v15.

5 Front end calibration pipeline

The goal of calibration is swift and accurate reporting of $h(t)$: this gives astronomers the maximum response period to perform electromagnetic follow-up of gravitational wave events. A search pipeline that runs in the front end computers of the interferometer would provide operators in the control room calibrated strain with extremely low-latency. When implemented, such a system will yield $h(t)$ as a raw data product, similar to the way d_{err} and d_{ctrl} are currently produced. We construct and test a front end model according to the following procedure:

1. Write Python scripts for FIR filters that model transfer functions
2. Build Simulink models for new pipeline based on DCS structure
3. Compare calibration model’s output with one currently in place

Generating scripts

The inverse sensing function is currently modeled using a 1 second FIR filter at the full sample rate of 16348 Hz [7]. The FIR actuation filter lasts 6 seconds and is sampled at the lower rate of 2048 Hz; however, the code for these filters is in a computer running at 16384 Hz. For computational efficiency (to avoid the same value being sampled 8 times), we split the actuation filtering “across time:” we divide the computations across 8 distinct 16384 Hz cycles. Both filters are computed by taking the inverse Fourier transform of their corresponding frequency domain models.

The inverse sensing filter in the front end model caused run-time errors. Additional code was written to handle this case: the filter was split “across space,” meaning that the computation for the inverse sensing transfer function was delegated to two computer cores. In practice, this meant that an additional Simulink model for half of the inverse sensing function was built, and its output fed back in to the main calibration model at a later time.

Simulink model flow

Appendix A displays the two Simulink models used to build the front end pipeline. d_{err} and d_{ctrl} are injected into the front end model. d_{err} is split immediately and sent to a second auxiliary model, `x2calcs2`, which runs in parallel. All inter-model communications take one sample computation time, meaning that a unit delay must be introduced in the main d_{err} path. A similar delay is introduced in a logically consistent location in the path of d_{ctrl} . A ring buffer is used in the auxiliary model to advance d_{err} by 8192 samples, or half the inverse sensing filter length.

A second unit delay is applied to the paths of the filtered signals d_{err} and d_{ctrl} to account for shipping the second half of inverse sensing back to the main model. To make the signals time synchronous, a ring buffer is applied to the total inverse sensing filter signal. Each actuation filter is 6 seconds long, whereas the inverse sensing filter is 1 second long. Both filters are time stamped in the middle, meaning that the actuation filters are delayed by 3 seconds while the inverse sensing filter has a 0.5 second delay. The last ring buffer thus delays the inverse sensing signal by 2.5 seconds, or 40960 samples. The signals are then summed to produce $h(t)$.

Results

To date, only the output of the inverse sensing function has been compared to the current model in place. A swept sine function with unity amplitude and frequencies ranging from 10 to 7000 Hz was applied to the filter with the following results:

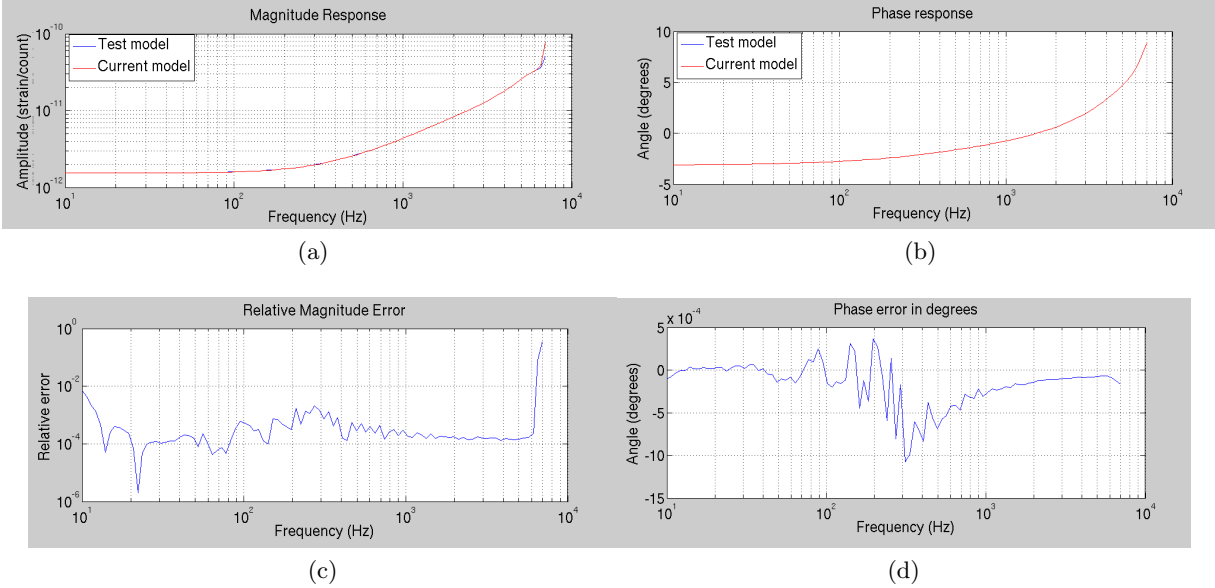


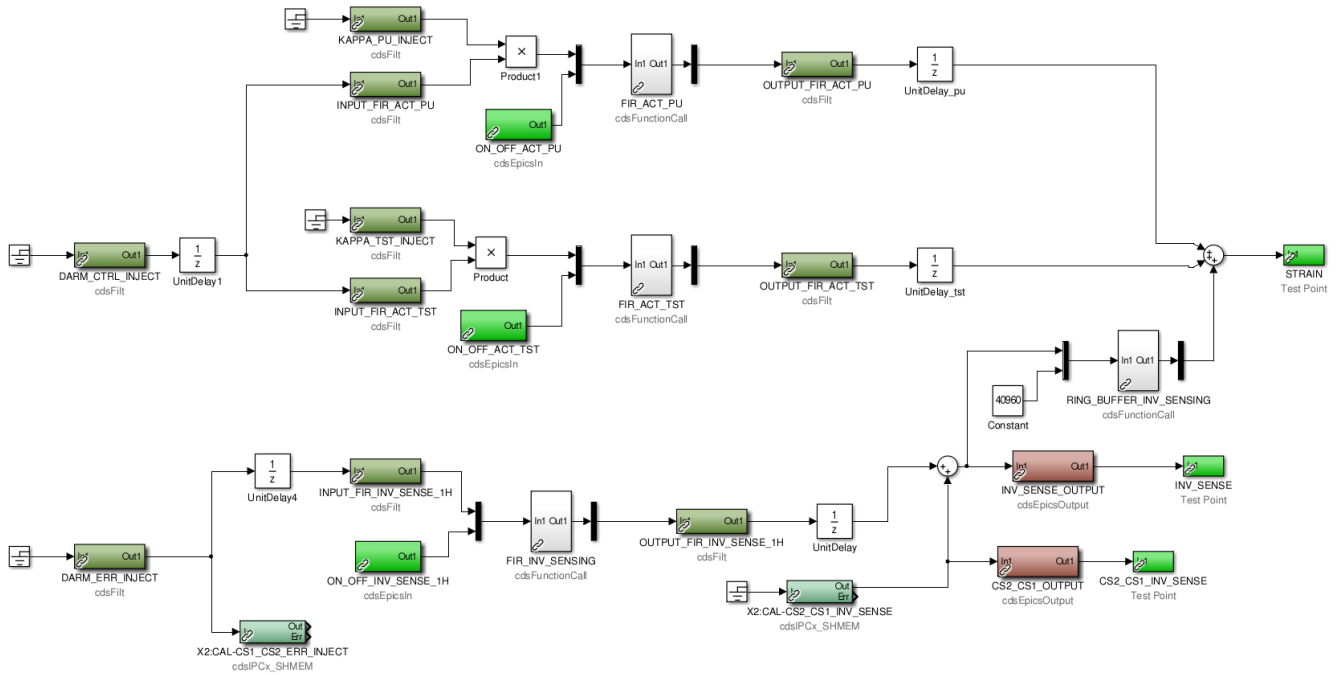
Figure 5: (a) Inverse sensing magnitude response, (b) Inverse sensing phase response, (c) Relative error present in the front end model, and (d) Phase error present in the front end model

We report that at its current state, the inverse sensing FIR filter has maximum error on the order of 10^{-2} in magnitude and 10^{-4} in phase at relevant frequencies (< 5000 Hz). We suspect that the large error in magnitude response at high frequencies involves issues with nearing the filter’s Nyquist range, and we will conduct more tests to verify this.

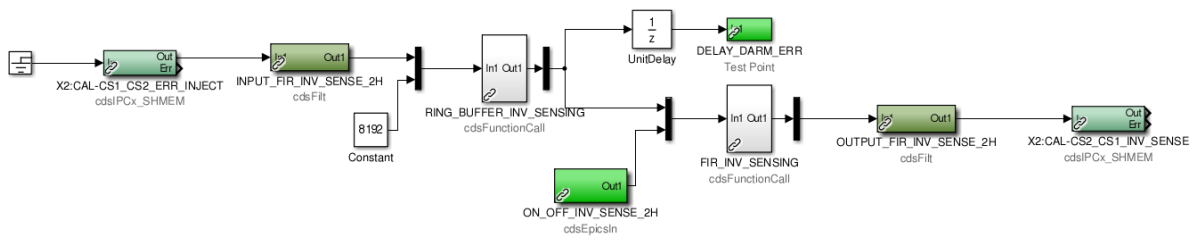
Goals

In the coming weeks, we plan to begin injecting large series of data into the front end model beginning at 10 minutes in duration. We will compare the output of this model with the data calibrated via the `gstlal` pipeline and begin troubleshooting discrepancies. Additionally, we will characterize the actuation response functions of each FIR filter (as done with the inverse sensing) and report the results of these comparisons. If time allows, and assuming a functioning front end model on the X2 test-stand, we will move to the real interferometer and implement the pipeline there.

A Simulink models



(a) x2calcs1



(b) x2calcs2

References

- [1] Ralph Bork. AdvLIGO CDS Design Overview. Dec. 2009, revised Oct. 2010.
- [2] B.P. Abbott *et al.* Calibration of the Advanced LIGO detectors for the discovery of the binary black-hole merger GW150914. Feb. 2016, revised Feb. 2017.
- [3] Craig Cahillane *et al.* Calibration Uncertainty for Advanced LIGO's First and Second Observation Runs. Apr. 2016, revised July 2017.
- [4] F. Matichard *et al.* Advanced LIGO two-stage twelve axis vibration isolation and positioning platform. Part 1: Design and production overview. *Precision Engineering*, 40:273–286, Apr. 2015.
- [5] M. Hewitson *et al.* Charge measurement and mitigation for the main test masses of the GEO 600 gravitational wave observatory. *Classical and Quantum Gravity*, 24(24):6379, Nov. 2007.
- [6] S. M. Aston *et al.* Update on quadruple suspension design for advanced ligo. *Classical and Quantum Gravity*, 29(23):235004, 2012.
- [7] Xavier Siemens *et al.* Time Domain Calibration in Advanced LIGO. (LIGO-T1400256-v17), Apr. 2015.
- [8] D. Tuyenbayev and the LSC Calibration Team. Tracking temporal variations in the DARM calibration parameters. July 2015, revised Sept. 2015.