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

Technical Note **LIGO-T1400256-v1** 2015/11/30

Time Domain Calibration in Advanced LIGO

Calibration Team

Distribution of this document:

LIGO Scientific Collaboration

Draft

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

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

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

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

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

Contents

1 Introduction

The starting point of all LIGO gravitational-wave searches is the strain—the fractional change in the length of the detector arms—caused by external sources including seismic and other noise along with gravitational-wave signals. The goal of the calibration is to provide an accurate reconstruction of the strain on the interferometer in the form of a time series $h_{\mathrm{ext}}(t)$,

$$h_{\mathrm{ext}}(t) = \frac{\Delta L_{\mathrm{ext}}(t)}{L} \tag{1}$$

where $\Delta L_{\mathrm{ext}}(t) = L_x(t) - L_y(t)$ is the length change of the x and y arms caused by external sources, and L is the effective length of the unperturbed arms. Calibration is the essential first step in the analysis of gravitational-wave data. We have developed a calibration procedure that produces $h_{\mathrm{ext}}(t)$ in low-latency, which is a crucial component in LIGO's ability to rapidly send gravitational-wave event alerts to the astronomical community, and so for LIGO to fully participate as a facility for transient astronomy [1].

The LIGO interferometers are kept in control through a set of multiple-input-mulitple-output (MIMO) digital feedback loops, including five length control loops: MICH, PRC, SRC, CARM, and DARM. Calibration focuses on the DARM control loop and approximates it as single-input-single-output (SISO).[1] TA major calibration effort is performing measurements to construct accurate frequency-domain models for the various sub-systems within the DARM feedback loop. These tasks are often referred to as the "frequency-domain" calibration. The tasks involved in the "time-domain" calibration are the construction of digital filters from the frequency-domain responses and the deployment and maintenance of a pipeline that applies the appropriate filters to the error and control signals of the DARM loop to produce a time series for h_{ext}.

This document is outlined as follows. Section 2 gives a brief description of the LIGO length sensing and control systems and discusses its applications to calibration models. Section 4 provides details about how the time-dependent variations in the calibration models can be tracked. Section 5 discusses how the frequency domain calibration models can be used to produce time domain FIR filters. Section 6 outlines the design standards for two different methods of time domain calibration. Section 7 discusses the involvement of calibration in LIGO's engineering runs so far, and Section 8 briefly summarizes the time domain calibration process.

2 A Brief Description: LIGO Length Sensing and Control System

The LIGO interferometers are controlled by a feedback loop in order to keep the instrument cavities in resonance. A diagram of the DARM feedback loop is shown in Fig. 1. At low frequencies, the noise in the external strain is dominated by seismic activity. A control

[1] This inherently neglects any cross-couplings between loops that might affect the detector response to GWs. The calibration group will measure and document these cross-couplings in order to establish the accuracy of this approximation.

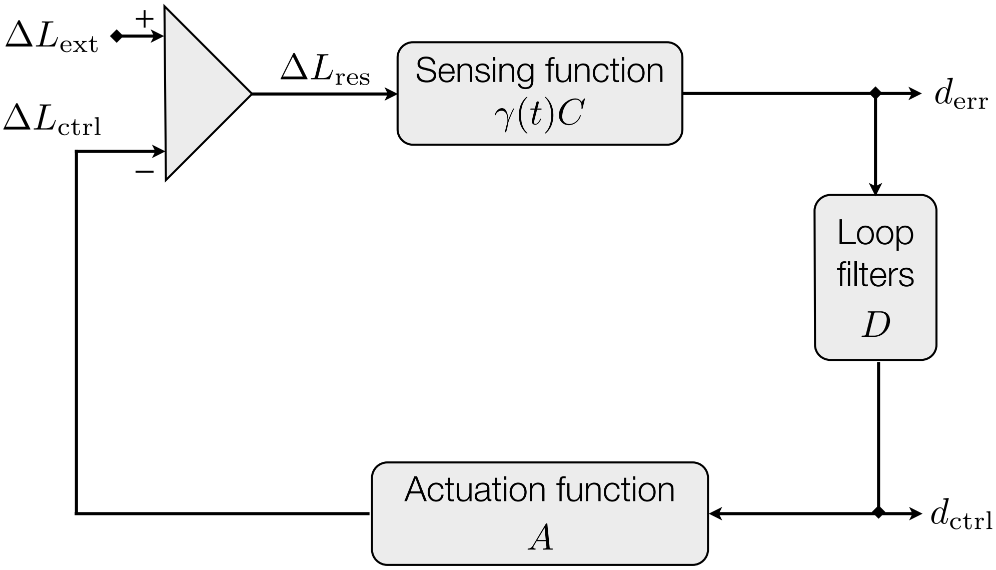

Figure 1: The LIGO interferometers are kept in control through multiple digital feedback loops. A simplified diagram of the DARM length control loop, which is used for calibration, is shown here.

displacement $\Delta L_{\rm ctrl}$ is applied to physically move the test masses in opposition of the external displacement $\Delta L_{\rm ext}$ at low frequencies. Active feedback at the low frequencies is necessary to ensure that the interferometer stays in alignment. A residual high-frequency displacement $\Delta L_{\rm res}$ remains.

The sensing function $\gamma(t)C$ turns the residual displacement into the digital error signal $d_{\rm err}$. This function contains a slowly-varying time-dependent gain $\gamma(t)$ which depends on the amount of light stored in the Fabry-Perot cavities in each of the arms. The calculation of $\gamma(t)$ is discussed in Sec. 4. The sensing function is the frequency response of the Fabry-Perot cavity. The Fabry-Perot cavity is sensitive to oscillations in the cavity length with a period greater than the average light storage time in the cavity. The frequency dependence of the sensing function is given approximately by a real pole at around 400 Hz known as the cavity pole. The cavity pole is inversely proportional to the light storage time in the Fabry-Perot cavity. Within the feedback loop, a set of known digital filters D is applied to the error signal in order to produce the digital control signal $d_{\rm ctrl}$. All digital filtering performed in the DARM loop is done in double floating-point precision. The actuation function A converts the digital control signal into the physical control displacement applied to the mirrors at the end of each arm. Each end test mass is suspended at the bottom of a quadruple pendulum and the control displacement is applied through actuators at each stage of the quadruple pendulum suspension system. The frequency dependence of the actuation function is given by the frequency response of the pendulum suspension systems. For more details on the sensing function, digital feedback filters, and actuation function, see [2].

The external displacement $\Delta L_{\rm ext}$ is constructed from both the digital error signal $d_{\rm err}$ and the digital control signal $d_{\rm ctrl}$. The result is then divided by the mean interferometer arm length L to obtain $h_{\rm ext}$. In fact, $\Delta L_{\rm ext}$ can be constructed from the digital error signal alone, but it turns out to be useful to to construct $\Delta L_{\rm ext}$ from both the error and control signals for reasons discussed in Sec. 4. Using the feedback loop shown in Fig. 1, we find

$$\begin{aligned} \Delta L_{\rm res} &= \Delta L_{\rm ext} - \Delta L_{\rm ctrl} \\ d_{\rm err} &= \gamma(t) C \ \Delta L_{\rm res} \\ \Delta L_{\rm ctrl} &= A \ d_{\rm ctrl} \ . \end{aligned}$$

Combining the above equations gives,

$$\Delta L_{\rm ext} = \frac{d_{\rm err}}{\gamma(t) C} + A \ d_{\rm ctrl} \ . \qquad (2)$$

In the frequency domain, Eq. (2) is a simple multiplication, up to the time-varying $\gamma(t)$. In the time domain, the time series $d_{\rm err}$ and $d_{\rm ctrl}$ need to be convolved with the appropriate digital filters. The sensing gain $\gamma(t)$ is more slowly varying than the decay time of the digital filter $1/C$, so we can just digitize the reference inverse sensing function $1/C$, convolve it with $d_{\rm err}$, and divide the resulting time series by $\gamma(t)$. Written explicitly,

$$\Delta L_{\rm ext}(t) \approx \frac{1}{\gamma(t)} \left[\frac{1}{C} * d_{\rm err} \right](t) + \left[A * d_{\rm ctrl} \right](t), \qquad (3)$$

where A is the digitized actuation filter, and the convolution of the filter F with the time series $x(t)$ is defined as

$$\left[F * x \right](t) = \int x(t') F(t - t') dt' \ . \qquad (4)$$

The digitized filters A and $1/C$ are kept fixed for as long as they are deemed to accurately represent the appropriate components of the DARM loop. When the various instrument subsystems are no longer accurately represented by the reference measurements for the sensing and actuation filters, updated measurements are used to produce new digital filters.

In summary, frequency-domain calibration activities include performing careful hardware measurements to accurately determine the sensing function C and the actuation A. Time-domain calibration activities include accurate digitization of the actuation and the inverse sensing frequency-domain models, tracking of the time-dependent gain of the sensing function, and applying the digitized filters along with the time-dependent sensing gain to the interferometer outputs to generate the external strain incident on the interferometer.

3 Injections and $h(t)$ Construction

In order to commission the interferometer and test our ability to detect gravitational waves in the data, we often inject simulated signals into the interferometer. For example, we simulate gravitational-wave events appearing in our interferometer by introducing a strain modeled after a real gravitational-wave signal directly into the instrument. These are known as "hardware injections" and are used to study our ability to find simulated signals with our search pipelines [3]. We also perform other types of injections, such as sinusoidal injections at a set frequency, in order to study the interferometer response to a known input for

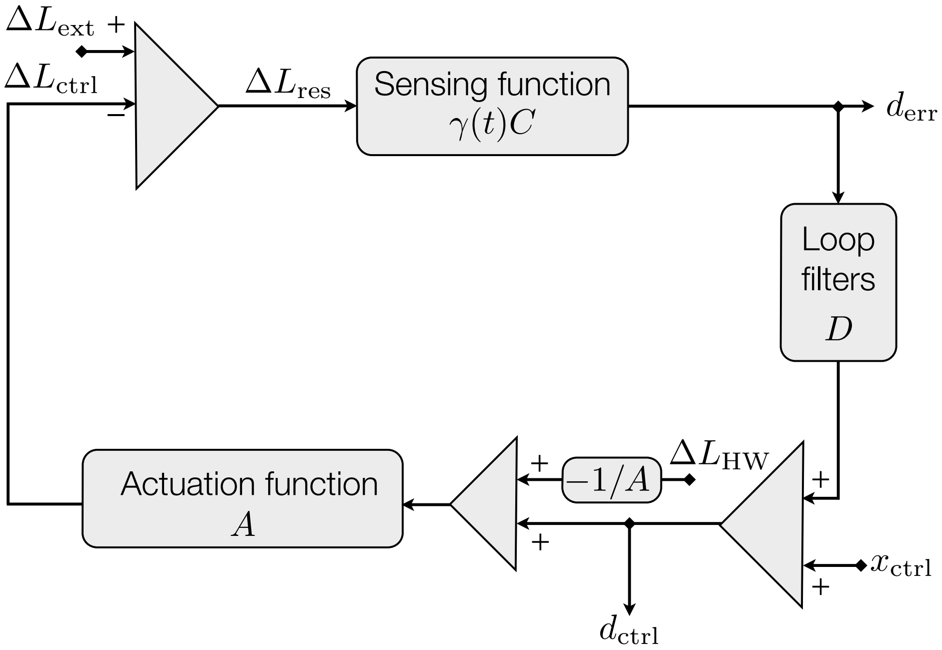

Figure 2: Hardware injections are injected after the digital error signal is read out, since hardware injections should appear in the final product ΔL_{ext}. However, injections that should not appear in the final product ΔL_{ext}, such as with the injections used to calculate $\gamma(t)$, should be injected before the digital error signal is read out.

commissioning. These types of injections are not of interest to data analysts when searching for gravitational-wave events in the external strain.

Injecting into the DARM loop requires careful thought about the desired result of the injection. There is flexibility as to whether injections are placed before or after the digital control signal d_{ctrl} is read out. If the injection should appear in the final result ΔL_{ext}, such as with hardware injections used to test our astrophysical search pipelines, the injection should be done after d_{ctrl} is read out. If the injection should not appear in the final result ΔL_{ext}, such as the excitation channel x_{ctrl} used to calculate $\gamma(t)$ for calibration (see Sec. 4), it needs to be injected prior to where d_{ctrl} is read out.

The reason this flexibility exists is because the digital control signal d_{ctrl} is used in the construction of ΔL_{ext}. The details become clear when the excitation signal x_{ctrl} and the hardware injection signal ΔL_{HW} are included in the feedback loop (Fig. 2). The injected excitation signal x_{ctrl} is automatically incorporated appropriately into the calculation of ΔL_{ext} (Eq. (2)), since the digital control signal d_{ctrl} includes the excitation signal x_{ctrl} within it. However, the hardware injection signal ΔL_{HW} is not contained within d_{ctrl} since it is injected after d_{ctrl} is read out. Using Fig. 2, the true differential arm length caused by external sources should be

$$\begin{aligned}\Delta L_{\mathrm{ext,true}} &= \frac{1}{\gamma(t)C}\, d_{\mathrm{err}} + A\, d_{\mathrm{ctrl}} + A\left(-\frac{1}{A}\Delta L_{\mathrm{HW}}\right)\\ &= \frac{1}{\gamma(t)C}\, d_{\mathrm{err}} + A\, d_{\mathrm{ctrl}} - \Delta L_{\mathrm{HW}}\ .\end{aligned}$$

Since Eq. (2) is used instead for ΔL_{ext} construction, the value reported for external displacement contains the hardware injection signal

$$\begin{aligned}\Delta L_{\mathrm{ext,reported}} &= \frac{1}{\gamma(t)C}\, d_{\mathrm{err}} + A\, d_{\mathrm{ctrl}}\\ &= \Delta L_{\mathrm{ext,true}} + \Delta L_{\mathrm{HW}}\ .\end{aligned}$$

This is the desired results, since hardware injections are meant to appear in the external strain sent to data analysis pipelines.

The use of Eq. (2) for the construction of ΔL_{ext} allows any control signal, such as x_{ctrl}, that is injected prior to the digital control signal d_{ctrl} to be automatically accounted for in the determination of ΔL_{ext}. Any signal that should appear as part of the reported ΔL_{ext} should be injected after the digital control signal is read out.

4 Tracking Slowly-Varying, Time-Dependent Changes in Calibration, $\gamma(t)$

The time-dependent gain of the sensing function $\gamma(t)$ can be measured by adding digital excitation signals x_{ctrl} into the DARM control loop. These excitations take the form of large amplitude sinusoids and are usually referred to as "calibration lines," because of how they appear in the spectra of the DARM error and control signals.

From Fig. 2, it is possible to derive an expression for the time-dependent gain of the sensing function $\gamma(t)$ in terms of outputs of the interferometer. For this derivation, the hardware injections are ignored. The digital control signal can be expressed in terms of the digital error signal and the excitation signal,

$$d_{\mathrm{ctrl}} = D\ d_{\mathrm{err}} + x_{\mathrm{ctrl}}\ .$$

At the injected line frequency f_{c}, the amplitude of the injected excitation is set to be much larger than the external displacement, meaning in the frequency domain the residual displacement is dominated by the injected signal $\Delta L_{\mathrm{res}}(f_{\mathrm{c}}) \approx -\Delta L_{\mathrm{ctrl}}(f_{\mathrm{c}}) = -A(f_{\mathrm{c}})d_{\mathrm{ctrl}}(f_{\mathrm{c}})$. This means that for Fourier transforms of x_{ctrl} and d_{ctrl} centered at time t we can write

$$d_{\mathrm{ctrl}}(f_{\mathrm{c}}) \approx -D(f_{\mathrm{c}})\ \gamma(t)C(f_{\mathrm{c}})\ A(f_{\mathrm{c}})\ d_{\mathrm{ctrl}}(f_{\mathrm{c}}) + x_{\mathrm{ctrl}}(f_{\mathrm{c}})\ , \tag{5}$$

which we can use to solve for $\gamma(t)$

$$\gamma(t) \approx \frac{1}{G(f_{\mathrm{c}})}\left(\frac{x_{\mathrm{ctrl}}(f_{\mathrm{c}})}{d_{\mathrm{ctrl}}(f_{\mathrm{c}})} - 1\right). \tag{6}$$

Here $G(f_\mathrm{c}) \equiv D(f_\mathrm{c})C(f_\mathrm{c})A(f_\mathrm{c})$ is the open loop gain at the calibration line frequency. The sensing gain $\gamma(t)$ should always be real, up to noise contributions to ΔL_res. Departures of $\gamma(t)$ from being real-valued indicate that one or more of the DARM model components are not modeled correctly, and a check on the reality of $\gamma(t)$ is used to diagnose the fidelity of the calibration model.

The time-dependent gain of the sensing function $\gamma(t)$, also commonly known as the calibration factor, is calculated using Eq. (6). The digital control signal $d_\mathrm{ctrl}(t)$ and the excitation signal $x_\mathrm{ctrl}(t)$ are measured as time series. These time series need to be demodulated at the calibration line frequency of interest f_c. The demodulation is accomplished by first applying a Hann window to one second of time series data centered at time t_0 and then taking a Fourier transform of the data at frequency f_c. The value of $x_\mathrm{ctrl}(f_\mathrm{c}; t_0)$ is given by

$$x_\mathrm{ctrl}(f_\mathrm{c}; t_0) = \sum_{j=}^{N-1} w_j x_\mathrm{ctrl}\left(t_0 - \frac{1}{2}N\Delta t + j\Delta t\right) e^{-2\pi i j \Delta t f_\mathrm{c}} \tag{7}$$

where the Hann window

$$w_j = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi j}{N-1}\right)\right] \, , \tag{8}$$

$1/\Delta t$ is the sample rate, and $N\Delta t = 1$ second. The equations above are for the excitation signal, but the same transform is performed for the digital control signal. The open loop gain is measured as a function of frequency using swept sine measurements, and the open loop gain at the calibration line frequency is determined from these measurements. The digital control signal d_ctrl is whitened before being recorded as output of the interferometer. Therefore, within the calculation for $\gamma(t)$, the digital control signal must also be de-whitened at the calibration line frequency. The value of the whitening filter for d_ctrl is known. This provides all of the ingredients required to compute $\gamma(t_0)$. The Hann window is then shifted ahead in time by one sample and the calculation is repeated for $\gamma(t_0 + \Delta t)$. In this way $\gamma(t)$ is calculated on a sample-by-sample basis starting with a one second delay.

5 Constructing Time Domain FIR Filters from the Frequency Domain Models

We digitize the frequency domain models for the inverse sensing function $1/C$ and the actuation function A in order to compute $\Delta L_\mathrm{ext}(t)$ (Eq. (3)). A lot of effort must first be put towards making accurate measurements of A and C in the frequency domain in order to produce reliable values for the external displacement. For more details on the frequency domain models for the sensing function and the actuation function see [2]. The impulse response for the inverse sensing and actuation functions are nicely approximated with finite impulse response (FIR) filters. The rationale for using FIR filters in the reconstruction of the strain, as opposed to infinite impulse response (IIR) filters, is two-fold. First, calibrated data often need to be re-calibrated with improved models. For example, in LIGO's fifth science run there were four revisions to the calibration which required re-calibration of the entire data set. Unlike IIR filters, FIR filters don't require storing a history, making re-calibration of the data set trivially parallelizable: all the data can be calibrated at once on a computing

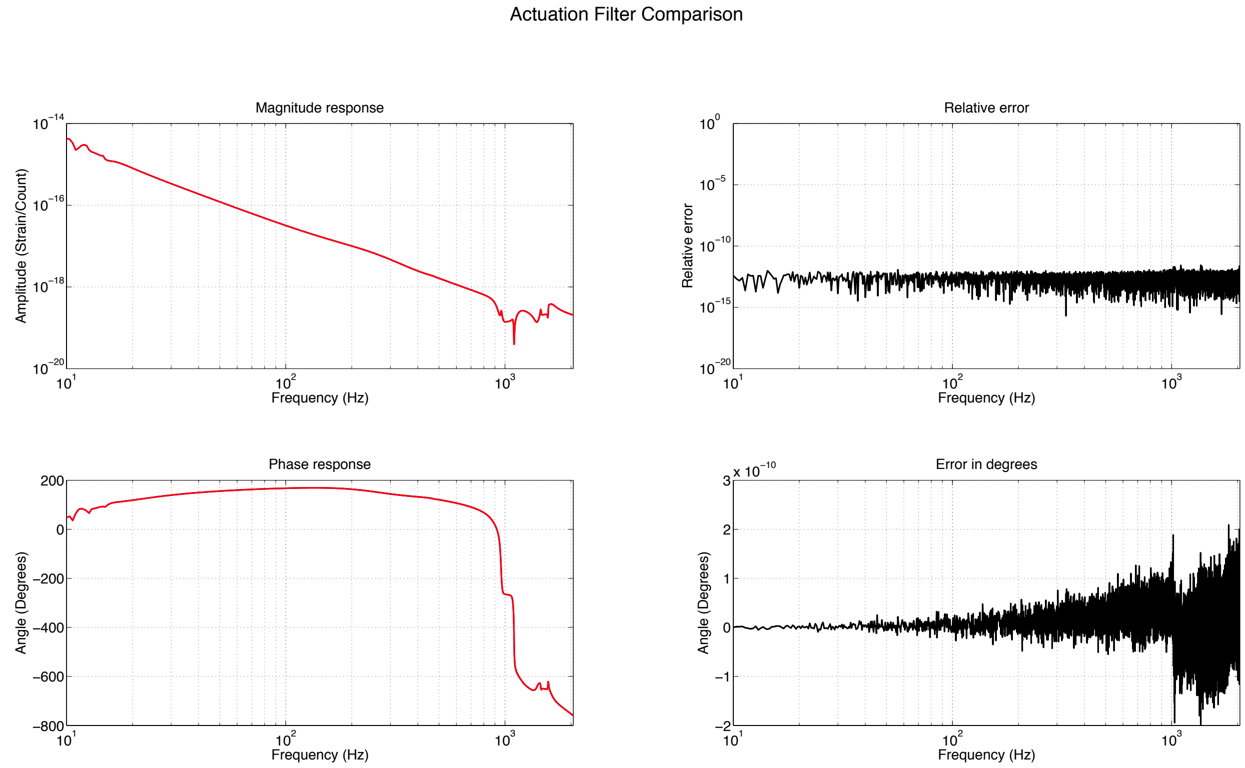

Figure 3: The magnitude (top) and phase (bottom) response of the actuation FIR filter (blue) and the true frequency domain actuation model (red). The plots on the right indicate the relative error between the two models. The errors in magnitude and phase are essentially zero across all relevant frequencies, [10 Hz, 2048 Hz].

cluster provided the data is divided into segments with sufficiently long overlaps. Second, when data drop-outs occur, filling in the gaps of missing data is trivial with FIR filters: again, we just need sufficient overlap with existing calibrated data. To fill in a few second gap, say, of missing data with IIR filters would require re-calibrating much larger quantities of data because of the need for filter histories.

The inverse sensing function is nicely approximated using a short, one second filter, and it is calculated using the full data sample rate of 16384 Hz. However, the actuation function requires a longer ∼a few seconds filter to capture the behavior appropriately. This is due to narrow frequency bandwidth notch filters, such as digital notch filters at the violin mode frequencies of the suspensions, in the actuation function that must be resolved using a long integration time. Since the actuation is only relevant for low frequencies and since it is a long filter, for computational speed-up the FIR filter for the actuation is computed at a lower sample rate of 4096 Hz. For both of these filters, we incorporate a high pass filter with a cutoff frequency of 9 Hz.

Frequency-domain calibration provides the transfer function for the sensing and the actuation. The FIR filters for inverse sensing and actuation are computed by simply taking the inverse Fourier transform of the frequency response of these transfer functions. To produce the actuation FIR filter, we

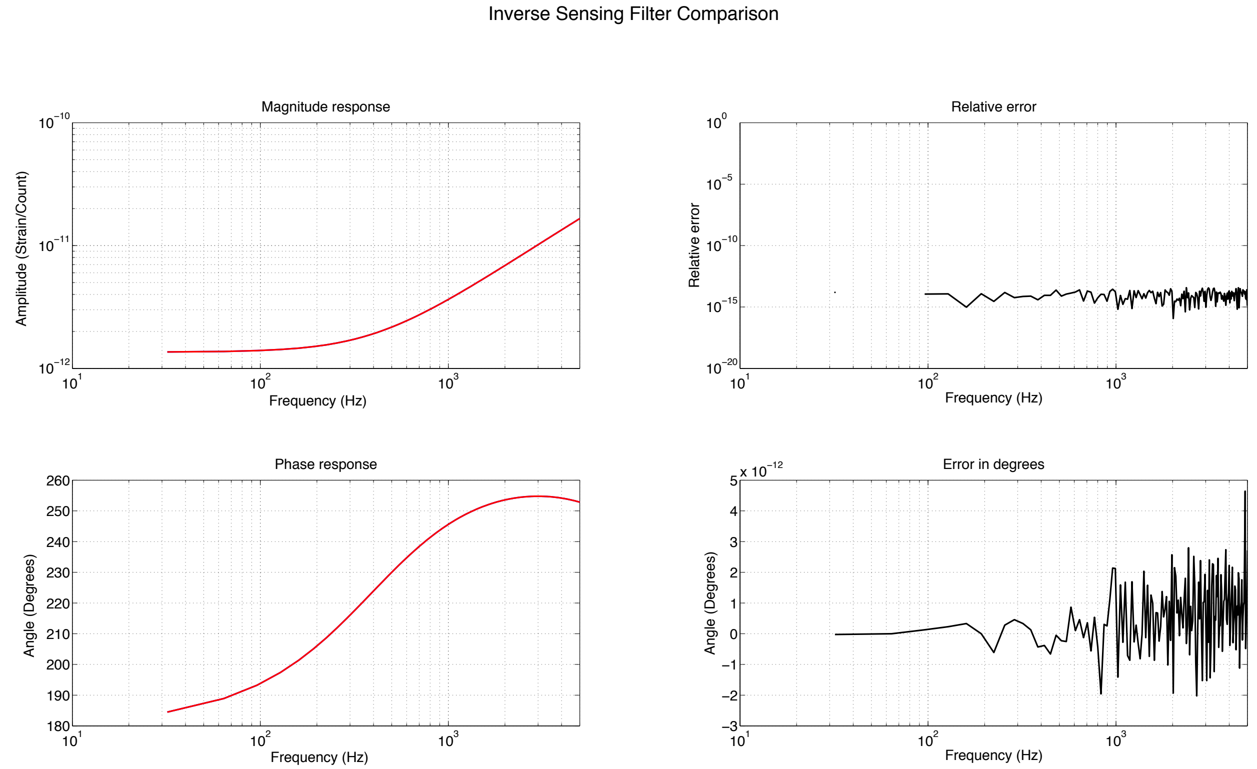

Figure 4: The magnitude (top) and phase (bottom) response of the inverse sensing FIR filter (blue) and the true frequency domain inverse sensing model (red). The plots on the right indicate the relative error between the two models. The errors in magnitude and phase are less than 10^{-12} across the entire calibration frequency band, [10 Hz, 5000 Hz].

1. Find the frequency response of the actuation transfer function using a frequency vector that corresponds to the desired properties of the final FIR filter. For example, the frequency spacing is $df = 1/T$ where T is the duration of the FIR filter and the Nyquist is chosen to be 2048 Hz.

2. Apply half of a Hann window that rolls off frequencies below 10 Hz to the frequency response.

3. Zero-out the Nyquist component of the frequency response.

4. Compute the inverse Fourier transform of the frequency response. The result is the desired FIR filter.

The resulting FIR filter from the above procedure is split across time, as shown in Fig. 5. We add an artificial delay to center the FIR filter in time. The resulting filter is shown in Fig. 6. This artificially delay must be undone by applying an advance during the filtering procedure.

The inverse sensing function FIR filter is computed in a similar manner. To produce the inverse sensing FIR filter, we

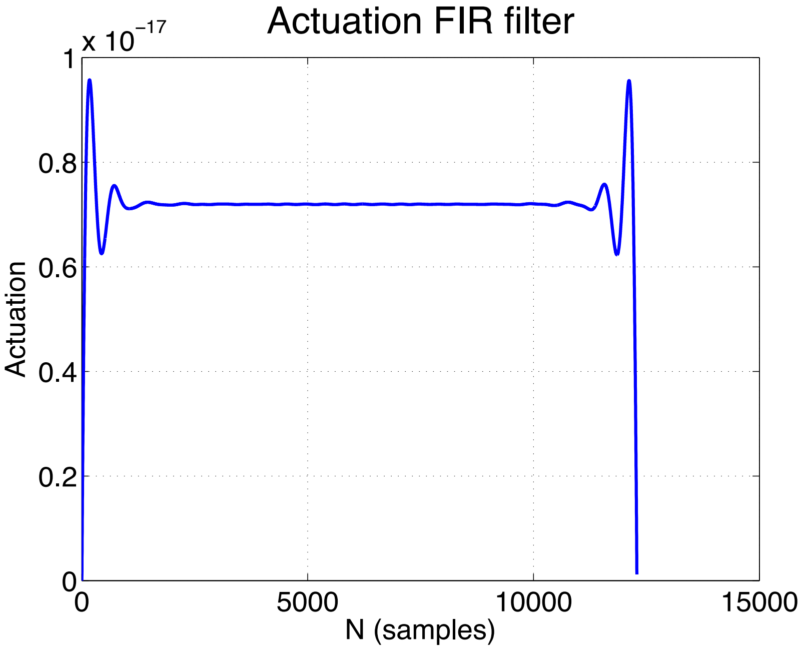

Figure 5: The actuation FIR filter when no artificial delay is added. The filter is split across time.

1. Find the frequency response of the sensing transfer function using a frequency vector that corresponds to the desired properties of the final FIR filter. Here, the Nyquist is chosen to be 8192 Hz, since $d_{\rm err}$ is filtered at the full sample rate of 16384 Hz.
2. Take the inverse of the frequency response to obtain the frequency response of the inverse sensing function.
3. Apply half of a Hann window that rolls off frequencies below 10 Hz.
4. Apply half of a Hann window that rolls off high frequencies (above ∼6 kHz).
5. Zero-out the Nyquist component of the frequency response.
6. Compute the inverse Fourier transform of the frequency response. The result is the dies red FIR filter.

Just as with the actuation FIR filter, we add an artificial delay to center the filter in time. The inverse sensing function requires that an additional artificial delay be added in order to produce a physical filter. There is a delay in the sensing function due in part to the light-travel time in the cavities. When inverting the sensing function, this turns into an advance. Since an advance is unphysical, we must add an artificial delay to the inverse sensing function in order to construct the FIR filter. All of the artificial delays are removed by advancing the time series when the filter is applied to the error signal in the calibration pipeline.

The error and the control signals have a large dynamic range, and in order to avoid floating point precision errors when recording these channels, we whiten both the error and control

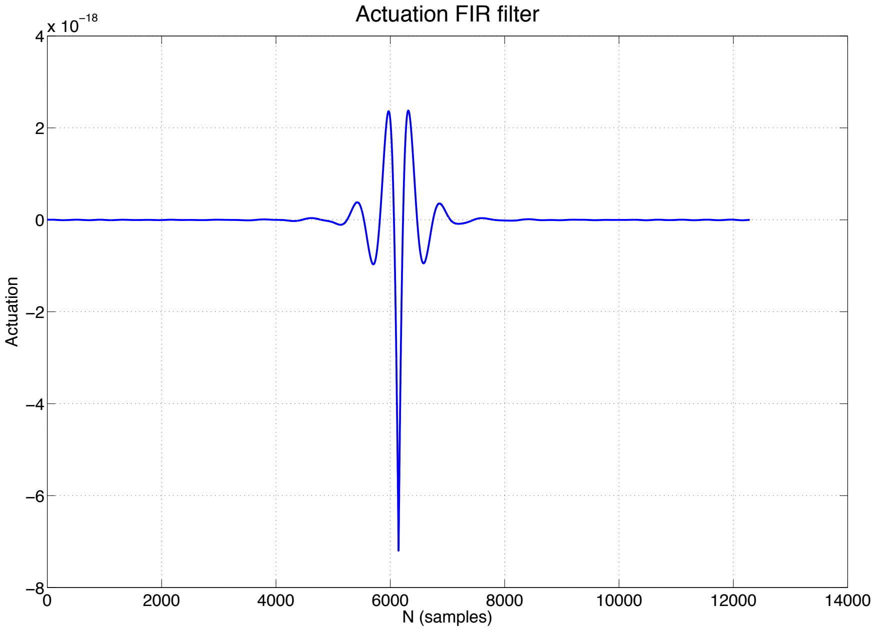

Figure 6: The actuation FIR filter with an artificial delay added. The filter is centered in time.

signals before recording them. This whitening needs to be undone in the calibration pipeline. I produce FIR de-whitening filters in the same manner as described above. I digitize a zero, pole, gain model for the de-whitening filters by taking the inverse Fourier transform of the frequency response of the model.

Fig. 3 shows a comparison of the magnitude and phase response of the computed actuation FIR filter to the true frequency response of the actuation function. The errors in magnitude and phase are essentially zero across all relevant frequencies, [10 Hz, 2048 Hz]. Fig. 4 shows the same comparison in magnitude and phase for the inverse sensing filter. The errors in magnitude and phase are lower than 10^{-12} across the entire calibration frequency band, which is [10 Hz, 5000 Hz].

Figure 3 shows a comparison of the magnitude and phase response of the computed actuation FIR filter to the true frequency response of the actuation function. At low frequencies, where actuation is most relevant, the error in magnitude is lower than 10^{-10} and the error in phase is essentially zero. Figure 4 shows the same comparison in magnitude and phase for the inverse sensing filter. At high frequencies, where inverse sensing is most relevant, the error in magnitude is lower than 10^{-2} and the error in phase is less than one degree. Finally, Figure **??** is the magnitude and phase response comparison for the response function. The response function is the combination of A, D, and $\gamma(t)C$ that must be applied to the error signal in order to compute ΔL_{ext}, assuming no calibration lines are injected (Equation (**??**)). The error in magnitude is lower than 10^{-3} across the whole frequency band and the error in phase remains close to zero until the highest frequencies, where it still remains less than one degree.

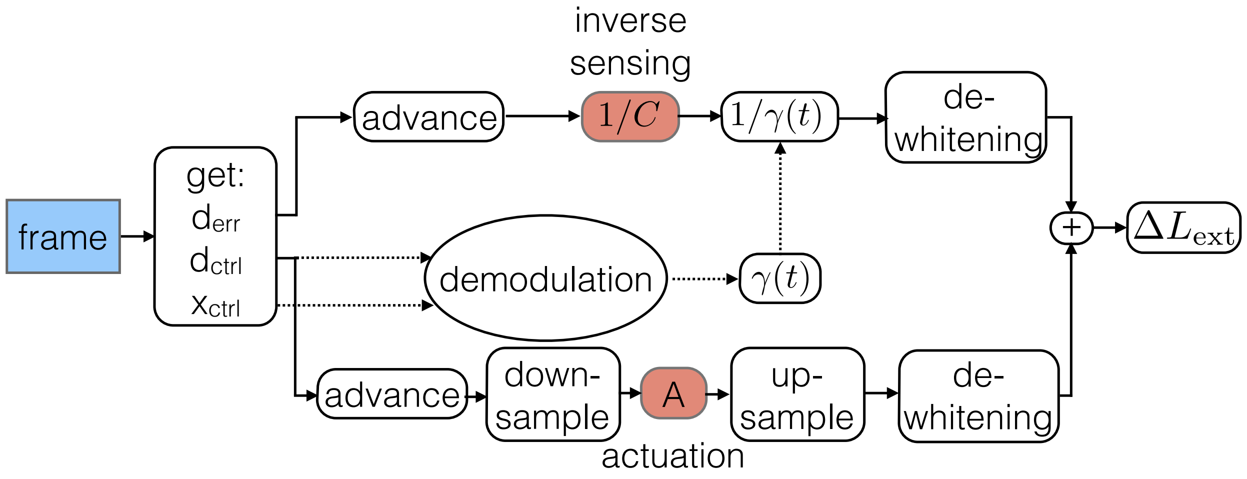

Figure 7: Basic schematic of $h_{\mathrm{ext}}(t)$ generation for the low-latency pipeline operated within the DMT.

6 Time Domain Calibration Software Design Standards

For Advanced LIGO, two methods are being developed to construct $h_{\mathrm{ext}}(t)$ in the time domain. The first method involves a low-latency pipeline that will be operated within the Data Monitoring Tool (DMT) at both the Hanford and Livingston interferometer sites. This method is fairly mature and most closely mimics what was done in Initial LIGO. The second method, which is currently in the early stages of development, would be implemented directly in the front-end of each interferometer. When this method is implemented, $h_{\mathrm{ext}}(t)$ will be available as a raw data product in the same way the error and control signals are currently available. Both methods are discussed in more detail in this section.

6.1 GStreamer Based Pipeline

Our goal is to produce $h_{\mathrm{ext}}(t)$ in very low-latency. The low-latency pipeline that is operated within the DMT is written using a codebase known as `gstlal` [4], which is a number of LIGO Algorithm Library (`LAL`) [5] tools wrapped with `GStreamer`. `Gstreamer` [6] is a common audio and video streaming codebase which is optimized for the manipulation of data streams that need to be processed in real time. The primary motivation for low-latency generation of the strain is prompt data analysis and reporting of gravitational-wave events (from, say, binary neutron star mergers) for electromagnetic follow-up [1].

Fig. 8 shows how the calibration pipeline fits into the current low-latency data flow. The front-end of the instrument collects data that is broadcasted to the DMT in the "frame" packaging format. These frames are synced up to a shared memory partition on the DMT machine from which the calibration pipeline reads. The calibration pipeline writes the calibrated data back into frames into a new shared memory partition. This data is then sent to the various LIGO Data Grid (LDG) computing clusters. Full documentation of the low-latency data network can be found in [7].

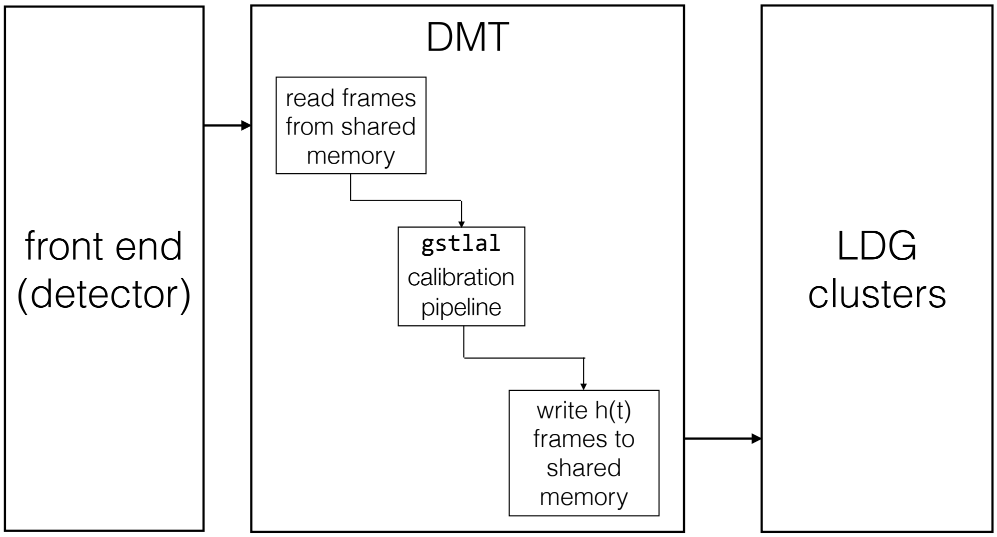
front end (detector)
DMT
read frames from shared memory
gstlal calibration pipeline
write h(t) frames to shared memory
LDG clusters

Figure 8: Diagram of how the calibration pipeline fits into the DMT schematic.

The procedure for computing $h_{\rm ext}(t)$ in the time domain is shown in Fig. 7. The pipeline begins by reading in data from a frame and extracting the necessary data channels: the error signal $d_{\rm err}$, the control signal $d_{\rm ctrl}$, and the excitation signal $x_{\rm ctrl}$. The error signal $d_{\rm err}$ is first advanced to compensate for the delay that was added to the inverse sensing FIR filter (see Sec. 5), and then convolved with the inverse sensing filter. In parallel, $\gamma(t)$ is calculated as described in Sec. 4. The error signal is then divided by the real part of $\gamma(t)$ to produce the residual displacement. Also in parallel, the control signal $d_{\rm ctrl}$ is first advanced to compensate for the delay that was added to the actuation FIR filter (see Sec. 5), and then down-sampled from 16384 Hz to 4096 Hz. The actuation FIR filter can be long (several seconds) and down sampling is performed for computational efficiency. Since the control signal only contains the low frequency part of $h_{\rm ext}(t)$, downsampling it and losing the high frequency information is not problematic. The down-sampled control signal is then convolved with the actuation FIR filter and resampled back up to 16384 Hz. The control and error signals in the DARM loop have large dynamic ranges and need to be whitened prior to being recorded in single floating-point precision. For the calibration, the control and error signals must therefore be de-whitened. This is done after the control signal has been filtered through the actuation and after the error signal has been filtered through the inverse sensing and multiplied by $1/\gamma(t)$. Finally, the filtered error and control signals are added together, producing $\Delta L_{\rm ext}(t)$, and divided by the mean detector arm length L to form the external strain $h_{\rm ext}(t)$. All of the filtering in this process is performed in the time domain to avoid latencies in $h_{\rm ext}$ production.

6.1.1 Calibration State Vector

In addition to producing $h_{\rm ext}(t)$, the `gstlal` calibration pipeline also produces calibration state information. The pipeline outputs a bit-wise calibration state vector known as the `GDS-CALIB_STATE_VECTOR` where the populated bits are defined as follows:

0 : HOFT_OK: $h(t)$ was successfully computed

1 : OBSERVATION_INTENT: “science intent” button is pushed

2 : OBSERVATION_READY: science quality data is available

3 : HOFT_PROD: $h(t)$ is produced by the calibration pipeline

4 : FILTERS_OK: filters settled in

5 : NO_STOCHASTIC_HW_INJ: No stochastic HW injection

6 : NO_CBC_HW_INJ: No CBC HW injection

7 : NO_BURST_HW_INJ: No burst HW injection

8 : NO_DETCHAR_HW_INJ: No DetChar HW injection

9 : KAPPA_A_OK: κ_A in expected range

10 : KAPPA_PU_OK: κ_{PU} in expected range

11 : KAPPA_TST_OK: κ_{TST} in expected range

12 : KAPPA_C_OK: κ_C in expected range

13 : FCC_OK: f_{cc} in expected range

14 : NO_GAP: input data was not a gap

A value of zero for a given bit means that the bit is “off” and a value of one means the bit is “on.” For example, if the zeroth bit is one then the calibration is considered okay.

Bit 14 being on indicates that there is no gap in the incoming data. Gaps can occur in incoming data when the calibration is done online.

Bits 9-13 are related to the slowly changing time-varying calibration correction factors, known as κ's. For a derivation of each κ and the coupled cavity pole frequency $f_{\rm cc}$, see [**?**]. If bits 9-13 are on, this indicates the calibration factors are in the expected range.

Bits 5-8 indicate whether or not hardware injections are on. This information is copied from the Online Detector Characterization (ODC) state vector [8].

Bit 4 indicates whether or not the filters have been given an appropriate amount of time to settle in after OBSERVATION_INTENT mode has been entered. If the detector has just entered OBSERVATION_INTENT mode, then the data for the first N seconds, where N is the length of the filter, should not be analyzed. This is because any glitchy data or data contaminated by, for example, calibration sweeps in the non-intent mode time period would be contaminating the first N seconds of $h(t)$. If the FILTERS_OK bit is turned on, that means the filters have been given the appropriate amount of time to settle in. Included in the filter settling time are the calibration filters and the smoothing filters for the κ's calculation.

Bit 3 is turned on if any output for $h(t)$ is produced by the pipeline.

The value of bit 2, OBSERVATION_READY, is determined from the ODC state vector. If the ODC state vector indicates that the instrument is producing observation ready data (which is a separate statement from the “observation intent” button being turn on) then bit 2 of `GDS-CALIB_STATE_VECTOR` will be turned on.

Bit 1, OBSERVATION_INTENT is also determined from the ODC state vector. The “observation intent” bit is a human-determined state of the instrument that indicates the instrument is in a mode that should be producing science-quality data.

The calibration is considered okay, and hence the zeroth bit would be recorded as “on”, if the criteria for bits 2-4 are met. If bit 14 is off, then bits 1-2 at the very least will also be off, which will in turn shut off bit 0. Bits 9-13, being off does not imply the calibration is not acceptable. For example, glitches can cause temporary bad values of the κ’s, but $h(t)$ is still acceptable. Smoothing algorithms are in place to try and ensure bad κ values caused by glitches do not infect $h(t)$. However, bits 9-13 look at the raw, unsmoothed κ’s and the smoothed κ’s to make an overall statement about whether or not each κ value is okay. Note that in order for bit 4 to be on, the detector must be in “observation intent” mode. Therefore, bit 1 is coupled to bit 0 through bit 4. However, bit 1 is left as a separate bit so that the statement about when we are in “observation intent” mode is clear from the GDS-CALIB_STATE_VECTOR.

6.1.2 Pipeline Inputs, Outputs, and Post-Processing

The `gstlal` calibration pipeline requires the following inputs in order to operate during advanced LIGO:

- The differential arm length error signal d_{err} distributed to the DMT in the form of frames.
- The differential arm length control signal d_{ctrl} distributed to the DMT in the form of frames.
- The time domain actuation, inverse sensing, and d_{ctrl} whitening filters computed from pre-determined calibration models. These are computed in advance and stored in a standard file format on the DMT.
- The ODC state vector that represent the state of the instrument (i.e. science mode) distributed to the DMT in the form of frames.
- The frequency of the calibration line that will be used to compute the time-dependent gain on the sensing function $\gamma(t)$ stored in a standard file format on the DMT.
- The excitation channel into which the calibration lines are injected x_{ctrl} distributed to the DMT in the form of frames.
- At least the real and imaginary parts of open loop gain at the calibration line frequency, but a full frequency series is also acceptable. This is pre-determined and stored in a standard file format on the DMT machine.
- The real and imaginary parts of the d_{ctrl} whitening filter at the calibration line frequency. This is pre-determined and stored in a standard file format on the DMT machine.

The `gstlal` calibration pipeline produces the following outputs:

- A copy of the ODC state vector (`ODC-MASTER_CHANNEL_OUT_DQ`) that was used to determine science mode.
- The `GDS-CALIB_STATE_VECTOR`, as discussed above.
- The real (`GDS-GAMMA_REAL`) and imaginary (`GDS-GAMMA_IMAGINARY`) parts of the computed calibration factors $\gamma(t)$ as a time series at 16384 Hz.
- The calibrated strain (`GDS-CALIB_STRAIN`) of the instrument $h_{\text{ext}}(t)$ as a time series at 16384 Hz.

In the current infrastructure, the `gstlal` calibration pipeline can read and write frames either from/to frame file images within shared memory or from/to frame files in the standard `.gwf` format.

6.2 Front-End Pipeline

A second method for time domain calibration is under development. This method will compute the external strain $h(t)$ directly in the front-end of the interferometer. When this method is implemented, $h(t)$ will become available in the raw frames. Although the code will be required to be different than the `gstlal` pipeline due to the nature of front-end code, the methods used to calibrate will remain as close to eachother as possible. Once the front-end and `gstlal` pipelines are verified to produce equivalent outputs, then the front-end pipeline can become the low-latency, streaming mechanism for computing $h(t)$ and the `gstlal` pipeline can be used for offline regeneration of $h(t)$ as necessary.

The front-end calibration model is currently under development, and the details of the methods and implementations will be updated in this document as development continues.

7 Engineering Runs

In order to prepare for the advanced detector era, the LIGO Scientific Collaboration has been conducting “engineering runs” that aim to test various aspects of the aLIGO infrastructure. These engineering runs begin with the production of h_{ext}, whether it be fake or real, and test the aLIGO infrastructure from transferring data to the appropriate computing sites all the way through searching the data for gravitational wave signals and preparing electromagnetic follow-up events.

Before the DARM loop was commissioned in the aLIGO interferometers, the `gstlal` calibration pipeline was exercised by producing fake strain data during the early aLIGO engineering runs, specifically during engineering runs 3 through 5 (ER3-ER5). During these earlier ERs the calibration procedure described throughout this document could not be implemented. However, the infrastructure surrounding the `gstlal` calibration pipeline was built-up and tested within the DMT machines during this time. In the most recent engineering run (ER6), a full interferometer was commissioned at the LIGO Livingston Observatory (LLO). We were able to implement the actual calibration procedure described in Sec. 6 during ER6.

7.1 Early engineering runs (ER3-ER5)

During the early engineering runs for which the `gstlal` calibration pipeline participated (ER3-ER5), the pipeline produced fake strain data by recoloring data from one of the aLIGO subsystems. The infrastructure surrounding the `gstlal` calibration pipeline was built up and tested during these runs.

The recoloring process is outlined in the following manner:

1. The aLIGO subsystem data is read-in from frames and whitened.
    (a) During ER3, this whitening was done adaptively. This means that the power spectral density (PSD) that was used to whiten the data was measured and updated on the fly while the pipeline was running. This process hoped to remove undesirable line features in the PSD of the recolored data. However, this process was found to cause data quality issues, specifically the introduction of anti-line features in the recolored data PSD.
    (b) Due to the data quality issues discussed above, the whitening scheme was adjusted for ER4 and ER5. In ER4 and ER5, the data was whitened using a fixed reference PSD that was taken during a known stable time for the relevant subsystem. The fixed whitening spectrum was smoothed to remove line features by taking the median of 32 samples on either side of each point in the PSD. Since the data is whitened using a fixed PSD, the data quality of the recolored fake strain can vary, sometimes significantly, when the real-time PSD of the relevant subsystem strays significantly from the measured reference PSD.
2. The whitened data was resampled from its original sample rate (32768 Hz) to 16384 Hz.
3. The whitened data was recolored to match the aLIGO zero-detuned high power (ZDHP) PSD [9]. This was done by creating an FIR filter out of the inverse aLIGO ZDHP PSD and filtering the whitened data in the frequency domain.
4. The recolored data was labeled with the channel name `FAKE-STRAIN` and written to frames at a sample rate of 16384 Hz.

During ER3 and ER4, the aLIGO subsystem that was recolored was the pre-stabilized laser (PSL). Specifically, the recolored channels were `H1:PSL-ISS_PDA_OUT_DQ` and `L1:PSL-ISS_PDB_OUT_DQ`. During ER5, the aLIGO subsystem that was recolored was the input mode cleaner (IMC). Specifically the recolored channels were `H1:IMC-F_OUT_DQ` and `L1:IMC-F_OUT_DQ`.

In addition to producing fake strain data, the `gstlal` calibration pipeline also produced a data quality vector at a rate of 1 Hz known as the `LLD-DQ_VECTOR`. This is not the `GDS-CALIB_STATE_VECTOR` described in Section 6.1.1. However, the `LLD-DQ_VECTOR` did serve as a data quality vector which informed the data analysis groups about the analyzability of the data. A definition of each of the bits for the LLD-DQ_VECTOR can be found in [10]. The gstlal calibration pipeline only set the first three bits of this state vector, which are defined as

0 : SCIENCE

1 : UP

2 : HOFT-OK .

Each bit is turned "on" (i.e. given a value of one) if the bit definition is true. For example, if the calibration pipeline produced non-zero fake strain for an entire second, then the third bit is set to one for the corresponding second. The only check that went into determining the $h(t)$-OK bit was that non-zero output was produced. There was no data quality information that was used to determine this bit.

The state vector that was used to determine the value of bits 1 and 2 was the Online Detector Characterization (ODC) state vector for the relevant subsystem. If the summary bit of the ODC state vector for the relevant subsystem indicated that the subsystem was in a good state, then both bits 1 and 2 were turned on. Since there was no "science mode" button during engineering runs, SCIENCE was taken to be true if UP was true.

There is one exception to this process. During ER3 and ER4, the first three bits could not be set independently due to the way the LLD-DQ_VECTOR was constructed within the gstlal calibration pipeline. Therefore, all three bits were turned off if either the ODC state vector indicated that the relevant subsystem was in a bad state or if the gstlal calibration pipeline failed to produce output. This means that there were times when the ODC state vector indicated that the subsystem was in a good state, but the LLD-DQ_VECTOR had both the SCIENCE and UP bits turned off. By ER5, the construction of the LLD-DQ_VECTOR was improved, and the three bits could be set independently. Although, in practice, bits 1 and 2 were still set together as described above.

During ER3 and ER4, the gstlal calibration pipeline only produced fake strain data when the ODC state vector indicated that the relevant subsystem was in a good state. Otherwise, the gstlal calibration pipeline just pushed out zeroes. This was also changed in ER5. Starting with ER5, the gstlal calibration pipeline attempted to recolor all of the data, no matter the state of the subsystem. This means that the third bit in the LLD-DQ_VECTOR could be turned on while the first two bits were off.

During these early engineering runs the reliability and stability of the gstlal pipeline infrastructure was tested and improved. By ER5, the pipeline produced fake strain data in very low latency more than 99.9% of the time for the entire run.

7.2 Recent engineering run (ER6)

In the most recent engineering run (ER6), there was a full interferometer commissioned at LLO. Since the LIGO Hanford Observatory (LHO) did not yet have a full interferometer, we still produced fake strain data as the LHO data stream. The gstlal pipeline produced real,

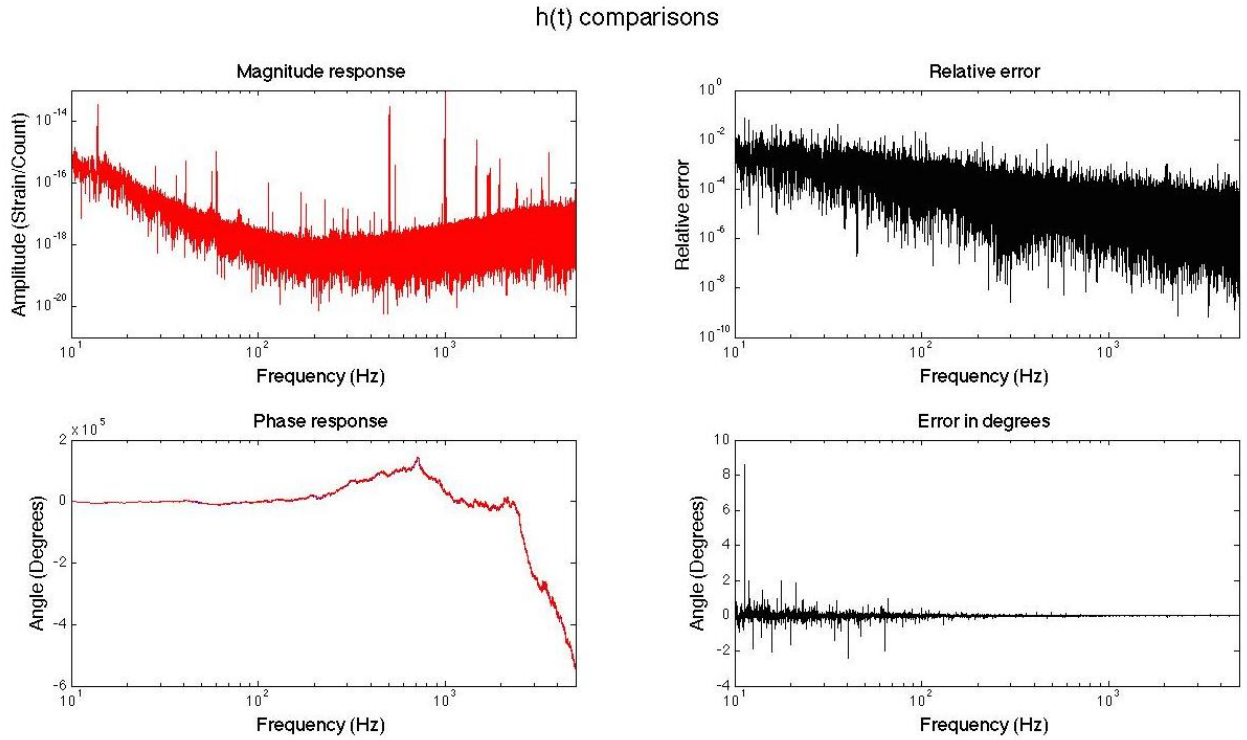

Figure 9: Comparison in magnitude (top plots) and phase (bottom plots) of the `gstlal` calibration pipeline to a frequency domain calibration (Eq. (9)).

calibrated $h_{\rm ext}(t)$ data during ER6 at LLO. The procedure used for calibration is exactly that described in Sec. **??**, except we artificially set $\gamma(t) = 1$ for the entire engineering run. We were able to produce calibrated data in near-real-time (~ 10 second latency) throughout the entire engineering run. During ER6 we were able to identify several errors in the FIR filter generation procedure and the `gstlal` calibration pipeline. Upon fixing these errors, the time domain calibration agreed to better than 1% in amplitude and 2 degrees in phase with a frequency domain calibration. The frequency domain calibration was computed by Fourier transforming a stretch of $d_{\rm err}$ and $d_{\rm ctrl}$ data, multiplying these by the inverse sensing and actuation function frequency responses, respectively, adding the result of each multiplication, and dividing by L,

$$h_{\rm ext}(f) = \left(d_{\rm ctrl}(f)\ A(f) + \frac{d_{\rm err}(f)}{C(f)} \right) \frac{1}{L}\ , \tag{9}$$

taking $\gamma(t) = 1$. Fig. 9 shows a comparison of the frequency domain calibration to the `gstlal` calibration pipeline output. The segment of the strain produced by the `gstlal` calibration pipeline is Fourier transformed for comparison to the frequency domain calibration. If the frequency domain calibration is taken to be the true calibration, the time domain calibration procedure introduces errors of less than 1% in amplitude and 2 degrees in phase across the whole frequency band. This is well within the promised errors for calibration for ER6. However, the complete calibration errors include both the time-domain errors and the errors accrued by inaccuracies of the frequency-domain models. The calculation of the frequency-domain model errors is part of the frequency-domain calibration effort.

8 Summary

Time-domain calibration provides the transition step between data acquisition at the instrument level and analysis of this data for astrophysical signals. The time-domain calibration process takes in raw instrument data and models of the instrument behavior to produce a time series of the external strain incident on the detector. The first step in the calibration procedure is to make accurate measurements of the relevant instrument parameters and to model each of the relevant subsystems. Detailed discussions of measurement techniques and required models can be found in [2], and a brief discussion of the models can also be found in Sec. 2. In addition to modeling the relevant subsystems, the calibration process must track the time-dependent gain of the sensing function, as discussed in Sec. 4.

Once measurements are made and models are developed, then the time-domain calibration process can begin. The first step in the time-domain calibration process is to make FIR filters out of the actuation and sensing function models. These FIR filters are quite accurate to the true models, as shown in Sec. 5. The time-domain calibration is currently performed with a `gstlal` based pipeline that is discussed in Sec. 6.1. However, a method to perform time domain calibration in the front-end of the interferometer is also under development.

The calibration team has been heavily involved in the LIGO engineering runs, and the `gstlal` calibration pipeline has produced the fake strain data for the third through fifth engineering runs and real calibrated strain data at LLO during the sixth engineering run. We have improved the `gstlal` calibration low-latency infrastructure to better than 99.9% reliability. During the sixth engineering run we were able to test the full accuracy of the time domain calibration for the first time at LLO, and we found accuracies of better than 1% in amplitude and 2 degrees in phase. The errors do not account for the errors associated with the frequency-domain models.

Looking forward to the seventh engineering run and the first official observing runs, we will continue to investigate the major sources of error in the time-domain calibration procedure. We will commission and test the implementation of the $\gamma(t)$ calculation in the `gstlal` calibration pipeline, and we will develop an infrastructure for ensuring the FIR filters for the actuation and inverse sensing are automatically updated in real time when models within the front-end are adjusted.

References

[1] L. Singer, L. Price, B. Farr, A. Urban, C. Pankow, S. Vitale, J. Veitch, W. M. Farr, C. Hanna, K. Cannon, T. Downes, P. Graff, C.-J. Haster, I. Mandel, T. Sidery, and A. Vecchio. The first two years of electromagnetic follow-up with advanced ligo and virgo. *ApJ*, 2014.

[2] Interferometer Calibration Preliminary Design Document. https://dcc.ligo.org/LIGO-T1300088, 2014. LIGO Document T1300088.

[3] Documentation of the Advanced LIGO Hardware Injection Infrastructure. https://dcc.ligo.org/DocDB/0113/T1400349/002/hwinjections.pdf, 2014.

[4] K. Cannon, C. Hanna, D. Keppel, P. Ajith, R. Cariou, M. Crispin-Ortuzar, B. Daudert, N. Fotopoulos, M. Frei, S. Hooper, A. Mercer, C. Pankow, S. Privitera, A. Searle, L. Singer, and M. Wade. gstlal, 2014.

[5] LSC. LSC Algorithm Library (LAL), 2015.

[6] Gstreamer. Gstreamer open source multimedia framework, 2015.

[7] Low Latency Data Distribution System Design. https://dcc.ligo.org/DocDB/0113/T1400343/002/lowl
2014.

[8] Online Detector Characterization System Overview. https://dcc.ligo.org/LIGO-T1200323, 2012. LIGO Document T1200323.

[9] Advanced LIGO anticipated sensitivity curves. https://dcc.ligo.org/LIGO-T0900288, 2009. LIGO Document T0900288.

[10] Low Latency Data Quality Vector Specification. https://wiki.ligo.org/viewauth/DAC/ER4#Data_Ger
2013. LIGO ER4 DAC Wiki.