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

Technical Note **LIGO-T080155-00-R** 2008/09/26

Adaptive Noise Cancellation - 40m Mode Cleaner

Sharon Rapoport
Mentors: Rana Adhikari, Alan Weinstein

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/

Abstract

A variety of fundamental and technical noise sources impact the ability of the Laser Interferometer Gravitational-Wave Observatory (LIGO) to directly detect gravitational radiation. Noteworthy examples include Newtonian gravity gradient, seismic, acoustic, thermal and photon shot noise. These are the obstacles which must be overcome by the planned upgrades to the LIGO detectors - Enhanced and subsequently, Advanced LIGO. To achieve improved sensitivity, significant improvements of LIGO's hardware must be paralleled by equivalent changes to the digital control system. Using adaptive filtering techniques, it is possible to cancel noise from known sources. We present results showing successful cancellation of low frequency seismic noise in the input Mode Cleaner at the Caltech 40m prototype interferometer using commonly available sensors and standard FIR filters.

Contents

1 Background

The Laser Interferometer Gravitational-Wave Observatory (LIGO) is based on a Michelson interferometer with Fabry-Perot cavities in its two perpendicular arms. In addition LIGO also employs power recycling to resonantly enhance the input laser power (D. McClelland) [1].

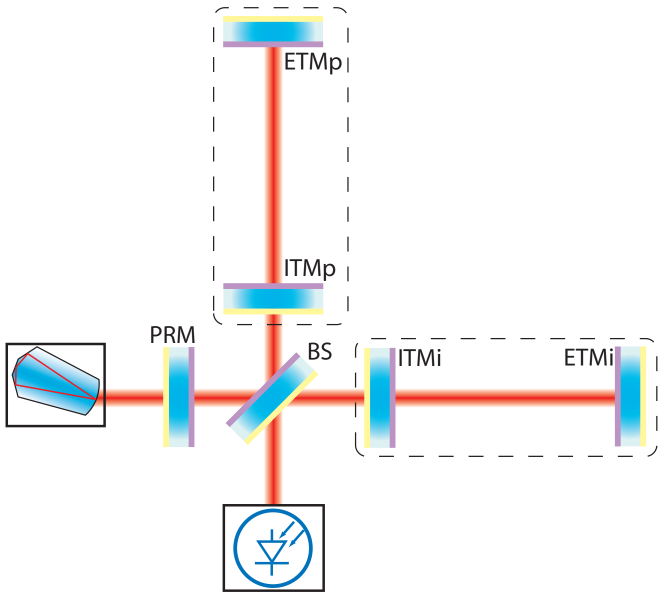

Figure 1: A schematic of LIGO, a power recycled Fabry-Perot Michelson interferometer, showing input laser, power recycling mirror (PRM), beam splitter (BS), Fabry-Perot arm cavities (encircled by dashed line) and readout photodiode.

With reference to figure 1 we briefly describe the initial LIGO detector. The laser enters the interferometer through the partially transmissive power recycling mirror and is subsequently split between the two arms at the beam splitter. Each arm comprises a Fabry-Perot optical resonator, the purpose of which is to enhance the phase response of the detector to differential arm motion. All optics are hung from pendulums which are suspended from isolation stacks to attenuate seismic disturbances. Due to the near unity reflectivity of the end mirrors of the Fabry-Perot cavities most of the incident laser light is returned to the beam splitter where it recombines. The Michelson is nominally held at a dark fringe (to reduce laser noise couplings) so the vast majority of light is transmitted to the PRM which forms a resonator with the arms to effectively increase the input laser power. Any differential arm motion (e.g. a gravitational wave) causes light to leak to the readout photodiode where the signal is detected.

2 Motivation

LIGO is designed to detect gravitational waves (GW). However, the interferometer is only sensitive to GW when it is in a particular state. The LIGO control system is tasked with maintaining the interferometer in such a state. At present there are over 100 individual control loops which together hold the interferometer at its operating point. Improving our control loops will result in a higher signal to noise ratio, increasing the probability of detection.

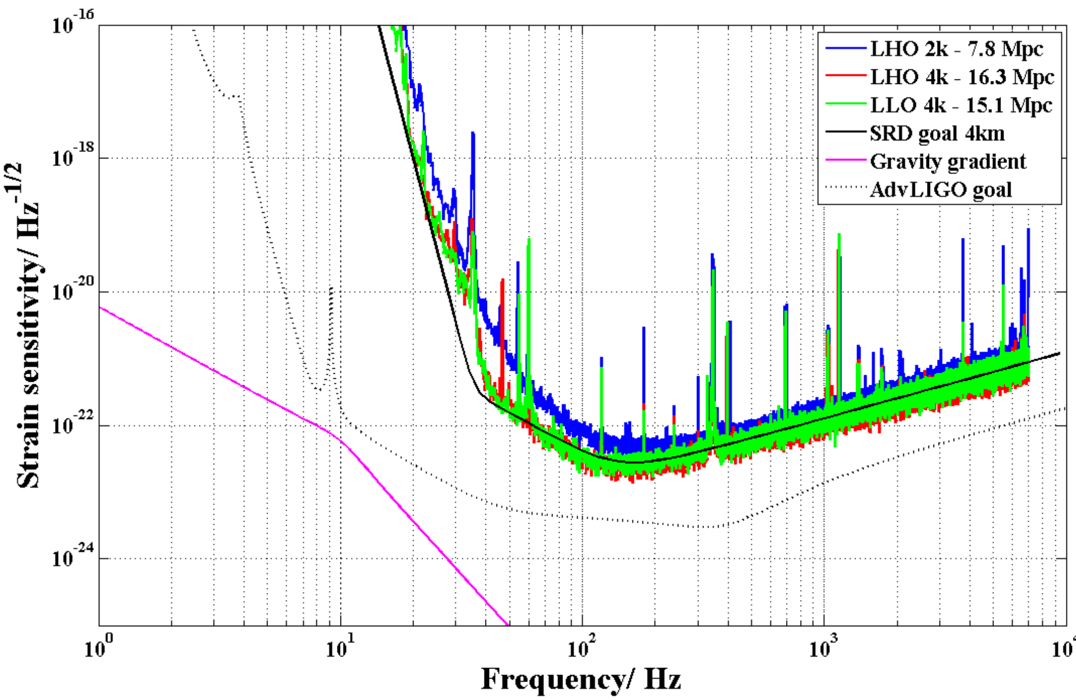

Figure 2: LIGO interferometer sensitivity curves from the S5 run (May 2007). Here we see the poor sensitivity at low frequencies ($<\sim$ 30 Hz) due to seismic noise. In addition to seismic noise Advanced LIGO will also be affected by Newtonian gravity gradients at the low end of its frequency band. http://www.ligo.caltech.edu/~jzweizig/distribution/LSC_Data/.

Adaptive filtering shall become increasingly useful as future upgrades to LIGO increase sensitvity and uncover hitherto unseen noise sources. Amongst other phenomena, Advanced LIGO will search for binary inspirals of high mass black holes. Such sources emit the majority of their GW at low frequencies, with higher mass binaries emitting at lower frequencies. Below $\sim$10 Hz any detectable signal is swamped by fundamental Newtonian gravity gradient noise and seismic disturbances (see figure 2). Using adaptive filters, it is possible to increase sensitivity in this low frequency band and allow detection of systems with one fifth of the mass. This will permit LIGO to make confident detections at redshifts of up to $z = 5$, when z is defined as $z = \frac{f_{emitted} - f_{observed}}{f_{observed}}$.

In this paper I will present recent advances in the application of adaptive filter theory to noise cancellation in LIGO. Our goal is to remove the effects of known noise sources from the gravitational wave channel (or auxiliary channels which affect our ability to sense gravitational signals). I will focus my discussion on the development of a multi-channel-multi-rate system which has been shown to mitigate one of the most problematic noise sources - low frequency seismic motion.

3 Filtering methods

The adaptive filtering techniques discussed below rely on a working knowledge of digital filtering. This section provides the necessary background.

3.1 Finite Impulse Response (FIR) - Non-Adaptive

Finite impulse response (FIR) filters are commonly used in digital signal processing. This type of filter is referred to as finite since it does not include a feedback path; every signal impulse incident upon the filter will eventually result in an output of zero. The filter is composed of a finite number of constant poles and zeros, where the poles are located at the origin of the Z-plane[1]. An FIR filter can be written in the following way:

$$y[n] = \sum_{i=0}^{N} b_i x[n-i] \tag{1}$$

Another characteristic property of FIR filters is the number of taps (N) they employ. The number of taps corresponds to the number of poles and zeros. A longer filter is more computationally expensive but results in more aggressive filtering, facilitating sharper cutoffs and a better defined bandpass filters (J. Proakis) [3]. Another important property of FIR filters, and a principal reason for their widespread use in LIGO, is that they are unconditionally stable. As mentioned earlier, all incoming signals will result in an output of zero after a well-defined time has elapsed. This can be explained by noticing that all the filter's poles are located at the origin of the Z - plane. For example, using the *zpk* function of *MATLAB*, we can plot the frequency response of a filter with 2 poles at zero, and zeros at -1 and -10 rad/sec for $s = jw$:

$$F = \frac{(s+1)(s+10)}{s^2} \tag{2}$$

3.2 FIR - Adaptive

An adaptive FIR filter is an FIR filter with varying coefficients. The adaptive algorithm aims to minimize the error between some target signal (from which we wish to remove noise) and noise signals (which provide a valid indicator of the disturbance to be removed), taking into consideration their constant online changes. The noise signals are filtered by the ever-changing adaptive FIR filter, such that they emulate the desired output signal. Subtracting the desired known signal from the imitated one provides an error signal which is then used to modify the FIR coefficients. There are several different methods for constructing

[1] The Z plane represents real or complex time-domain inputs, in the complex frequency-domain using a generalized discrete-time Fourier transform.

adaptive filters, however, they are based on the same general principles. Looking at the final representation of an adaptive LMS filter, it is quite simple to understand the significance of the variables that characterize a filter:

$$W(n+1) = W(n) + \mu\epsilon X(n) \tag{3}$$

In (3), W represents the set of FIR coefficients, ϵ is the input from the error signal. X is the input noise signal vector, which has the same length as the number of taps. μ, the step size, is also referred to as the adaptation rate. Since ϵ, W and X are fixed, the key variable in the equation is the adaptation rate. This step size has to be chosen carefully. By setting μ large, we can neglect instantaneous changes, however, at the same time, this can cause the filter to diverge. Finding the balance between a too small μ and a too large one will maximize the efficiency of the adaptive filter. Below is an example of a SISO (single-input signal-output) adaptive filter. The noise signal and the error signal are used to modify the coefficients of the FIR filter.

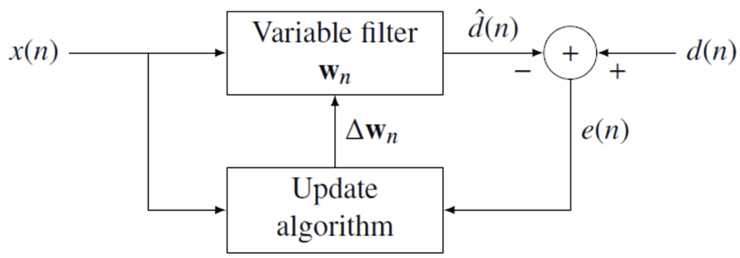

Figure 3: Adaptive filter block diagram. Can be implemented with a Least-Mean-Square (LMS) algorithm to predict the target signal will the incoming noise signal (S. Haykin, Adaptive Filter Theory) [5]

3.3 Infinite Impulse Response (IIR)

Unlike FIR filters, an IIR filter will not yield zero output after an infinite amount of time. This is due to their having poles away from the origin of the Z-plane. The IIR filter has both feedforward and feedback components. The feedback element leads to a dependence of the output on the previous output results. An IIR filter can be written in the following way:

$$y[n] = \sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j] \tag{4}$$

Although they are less stable, IIR filters are sometimes preferable over FIR filters. Due to the feedback component, an IIR filter results in a much better bandpass cutoff than an FIR filter of the same number of taps.

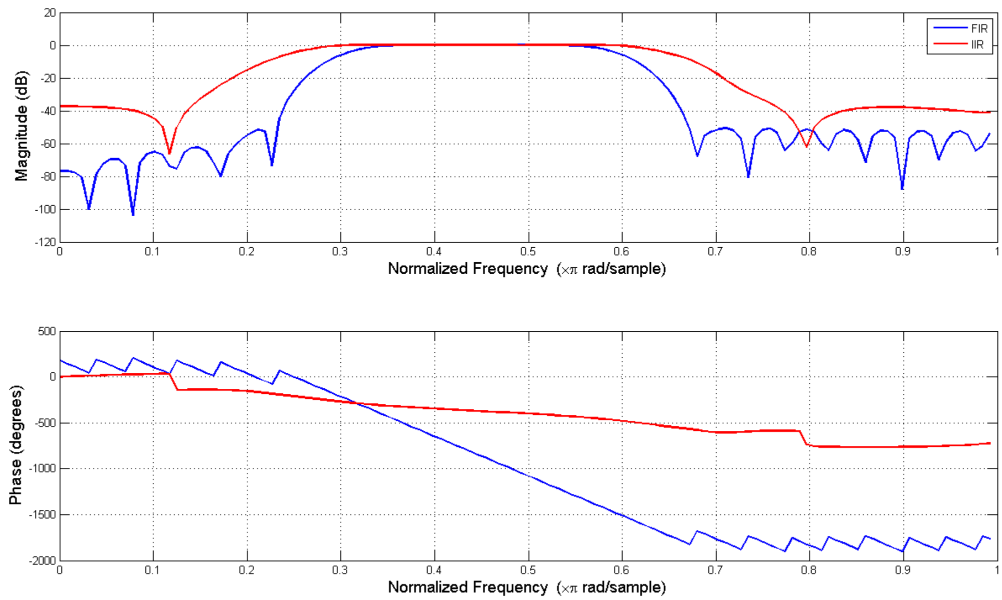

Figure 4: Bode plot comparing FIR and IIR bandpass filters each with 128 coefficients.

4 Active noise suppression using adaptive filters

In this section we present results of work to actively suppress seismic disturbances in the input Mode Cleaner of LIGO's 40m prototype interferometer. Before doing so we discuss whether adaptive techniques are strictly necessary in this system. We also discuss the choice, calibration and placement of the noise witness sensors used.

4.1 The Mode Cleaner (MC)

The MC is a triangular ring cavity with two flat mirrors (MC1 & MC3) and one spherical mirror at the apex (MC2). The laser is incident on MC1 and exits via MC3. The cavity is close to critically coupled and provides excellent transmission to the interferometer. The job of the mode cleaner is to remove undesirable characteristics from the beam exiting the PSL before it enters the interferometer[?]. These characteristics could be unsuppressed higher order modes, amplitude or frequency noise, beam jitter or spurious polarization components.

The length of the cavity and the frequency of the laser are controlled by two separate servos. At low frequencies (approximately below 100 Hz), the system actuates on mirror 2 in order to keep the cavity length commensurate to the laser's frequency. At higher frequencies, where such mechanical movement is hard to control, the laser frequency is adjusted such that the incident light is resonant in the mode cleaner cavity (N. Jamal) [4]. The Mode Cleaner is an integral part of LIGO, any instability of this cavity has negative repercussions on the entire interferometer. As such this key component represents an ideal setting in which to test adaptive filtering.

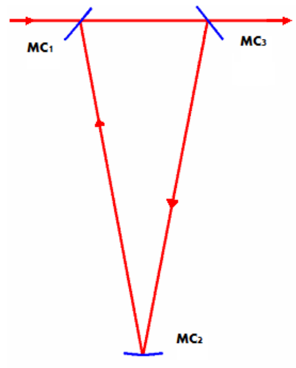

Figure 5: Mode Cleaner

4.2 Is adaptive filtering necessary?

Since we are concerned with dynamic systems, susceptible to many internal and external forces, an adaptive code can always contribute to the reduction of noise. However, while an adaptive filter may always be more accurate than a stationary FIR filter, the improvements might not justify the intensive calculations it requires. An adaptive filter is most useful when the transfer function between the target signal and noise signals are not constant. Therefore, to test whether adaptive techniques were necessary for use on the Mode Cleaner, I used several hours' data to examine the changes of the transfer function in this system. Using *MATLAB* code written by Matt Evans, I plotted the transfer functions between the target signal and the modified noise signals. The code employs Wiener filter approximation. Wiener's algorithm generates a filter that minimizes the LMS error between the target and noise signals. The following plots show the transfer functions between the mode cleaner's length and the noise signals after 13, 26 and 39 minutes.

We see that over the timescales studied here vast changes can be observed in the transfer functions, hence we must conclude that it is not possible to imitate the transfer function with a simple constant filter. In addition it has been demonstrated during LIGO's science runs that robust operation of the interferometer can be achieved for periods far exceeding these times, and as such the the system can benefit from the implementation of adaptive filters.

4.3 Low frequency control

An FIR adaptive filter has been implemented on the input Mode Cleaner at LIGO's 40m prototype interferometer. The main focus of this work was to better understand the adaptive algorithms used and determine how such techniques could be optimized for use in full scale detectors. The code (Appendix A), written in C, is designed to ingest input noise signals (up to a maximum of 32) and one target signal, and implement a Least Mean Square (LMS) algorithm to find the next set of the FIR coefficients. The code includes several user-controlled parameters. τ - Decay rate, is used to suppress the old set of FIR coefficients. Every cycle, the old coefficients are multiplied by $(1-\tau)$ to avoid runaway instabilities. μ, as mentioned previously, is the step size which controls the weight of the latest measurement in

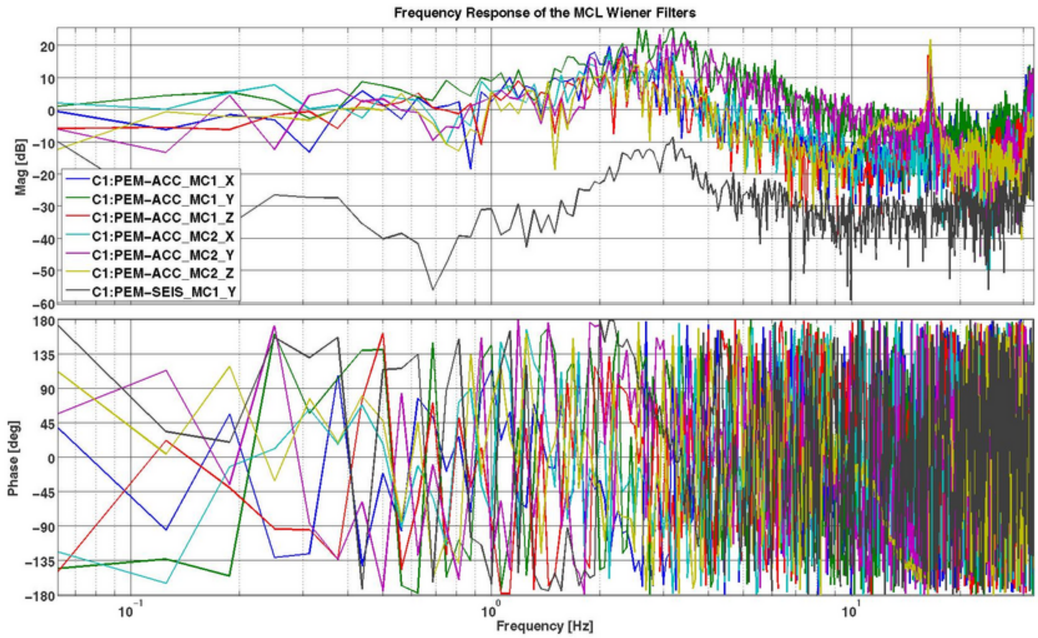

Figure 6: Transfer Function between MC_L and the different noise channels for 13 minutes of data

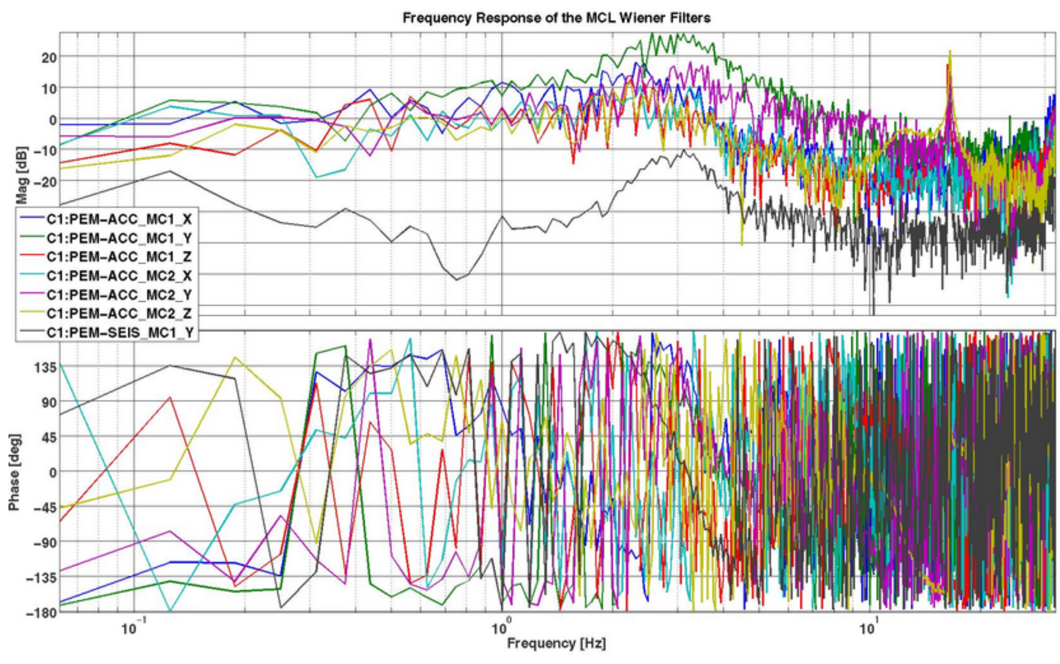

Figure 7: Transfer Function between MC_L and the different noise channels for 26 minutes of data

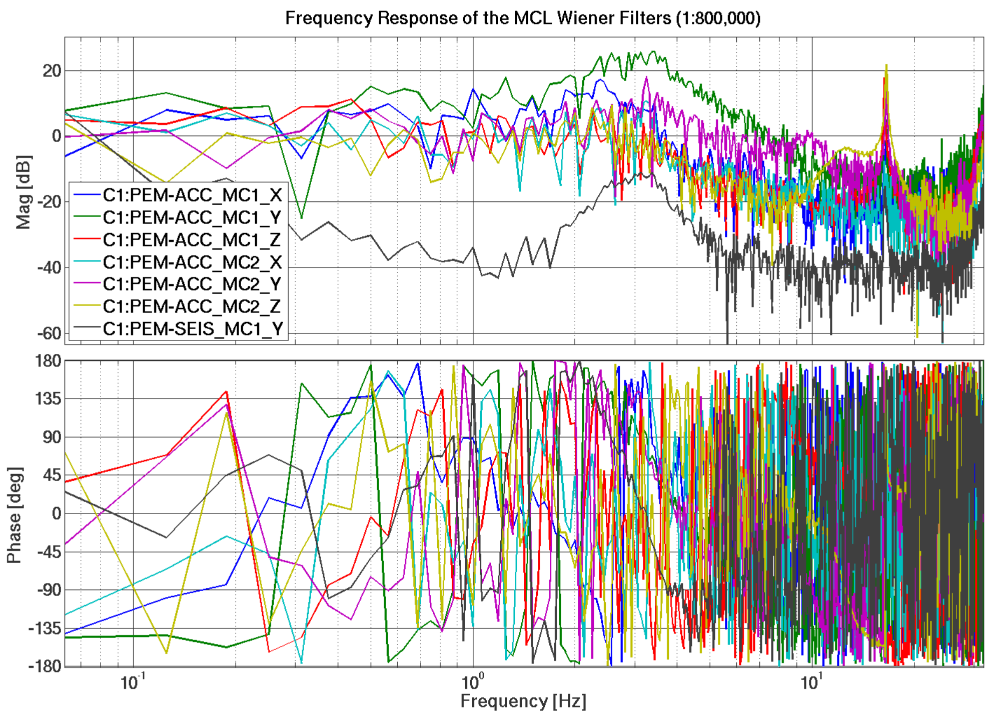

Figure 8: Transfer Function between MC_L and the different noise channels for 39 minutes of data

the calculation of the new set of coefficients. N, the number of taps, determines the length of the filter. Below is the block diagram of the high-gain loop:

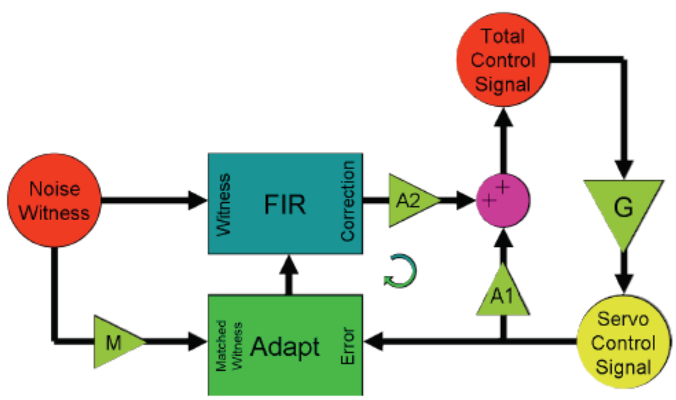

Figure 9: High-gain loop designed for LIGO (T080151-00) [6]

When the adaptive code is not running, the mode cleaner's length is controlled by actuating on mirror 2. This signal is proportional to the change in the length of the cavity relative to the laser frequency. With the adaptive code operational, this actuation on mirror 2 is used as the target signal. The adaptive code actuates on mirror 1 to null the round trip length change of the cavity.

The block diagram below represents the code which has been used in this investigation. Since the various channel are not recorded at equal sampling rates, and since the real time operation of the system could be compromised by large volumes of data, the signals need to be downsampled before entering the adaptive code. $D\downarrow$ and $U\uparrow$ signify the downsampling and upsampling in the code.

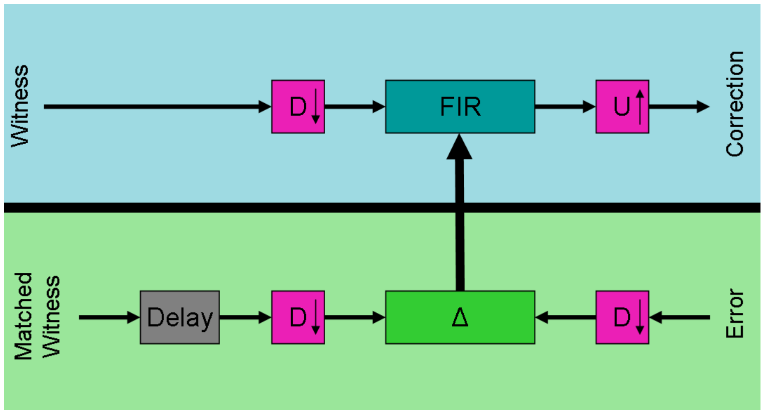

Figure 10: Online adaptive filter currently running on the Mode Cleaner, the adaptive part needs to be modulated [6]

4.4 Witness sensors

In order to cancel the acceleration noise we utilized 6 accelerometers and a seismometer. Three mutually orthogonal accelerometers, labeled X,Y and Z, were placed at each end of the cavity (see figure 11) with the seismometer at its mid point. These sensor were chosen to give good coverage across the frequency band of interest. The seismometer operates most effectively at low frequencies whilst the accelerometers are more useful above a few Hz.

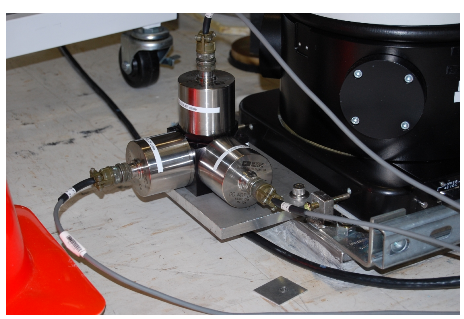

Figure 11: Accelerometer's setting. The accelerometers are Wilcoxon accelerometers, model 731A with a sensitivity of 9.9-10.3 V/g.

4.5 Accelerometer calibration

In order to ensure that each accelerometer provided the same contribution to the adaptive filter, we normalized their responses, making certain that the readout of all accelerometers will be the same for a specific input. This was accomplished by removing the accelerometers from their usual positions and placing them, facing in the same direction, in two groups. Two distinct groups were necessitated by cabling constraints. Having two groups of accelerometers X, Y and Z, we used accelerometer X of each group to calibrate the others. After finding the apposite gains for Y and Z, we placed the X accelerometers from each group in close proximity and repeated the measurement. The results of this stage allowed us to calibrate the relative gains of each group. Once found the gains were entered in to our adaptive code to maintain coherence between the 6 accelerometers.

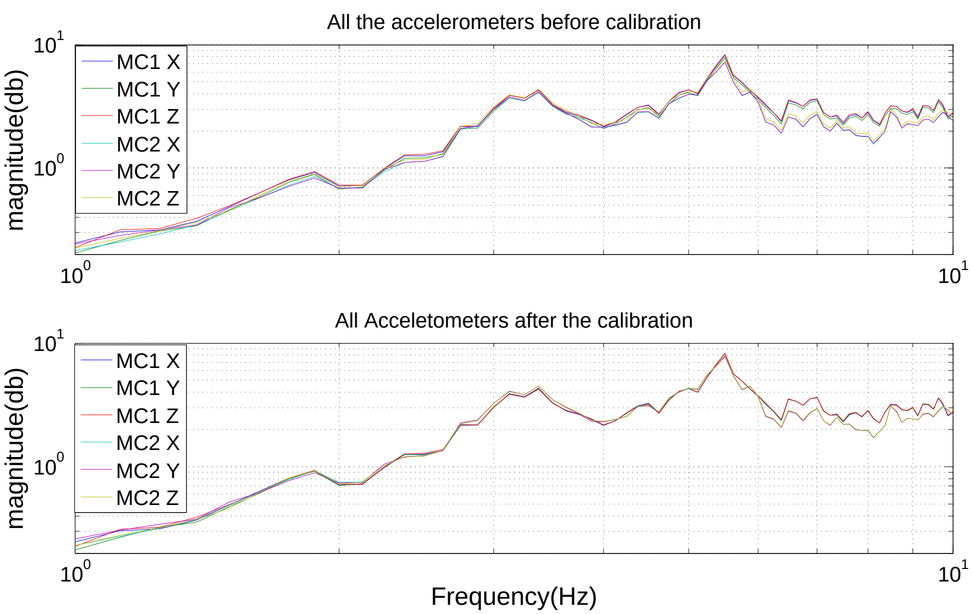

Figure 12: The power spectrum of the accelerometers before and after the calibration

4.6 Adaptive code optimization

The power spectrum of the control signal applied to mirror 2 reveals shows two dominant features. Around 1 Hz there is a resonance of the pendulum and at 16 Hz we see the bounce resonance of the mirror suspension. Reduction in the height of these peaks was the goal of this work and also the metric which informed the choice of parameters in the adaptive code.

To gain some intuition as to the effect of varying the available parameters (τ, μ, N) each one was varied in turn whilst keeping the remainder fixed. We immediately observed the expected correlation between μ and τ. τ may be thought of as the weighting of the old set of coefficients; with μ being the weight of the new values. By making μ larger, τ has to be smaller to maintain approximately the same results. We also observed only weak dependence on the number of taps N. No significant changes were witnessed on changing from 500 to 1000 taps.

$$W(n+1) = W(n)(1-\tau) + \mu\epsilon X(n) \quad (5)$$

This initial data was downsampled by a factor of 16, from 2048 Hz to 128 Hz. Since the Nyquist frequency is $Fs/2 = 64$ Hz, this theoretically allows noise to be actively suppressed up to 64 Hz. However, at such frequencies acoustic noise is dominant and therefore disturbances in the target signal at these frequencies will show poor correlation with our sensors and be troublesome to suppress. In order to focus on the frequency band exhibiting high acceleration noise, we reduced the sampling rate to 64 Hz, giving a Nyquist frequency of 32 Hz. Noticing a significant improvement after this change it was maintained for the following investigation.

4.7 Suppression of small optic suspension (SOS) bounce mode

Using only 4 of the six accelerometers we attempted to suppress the bounce mode of the Mode Cleaner Suspension. These particular accelerometers were chosen as their spatial orientation was well suited to canceling a disturbance in the vertical direction. Of the accelerometers chosen two were aligned with the cavity axis (Y direction) and two were aligned with the bounce mode of the optics in the vertical/ Z direction. Here follows (figures 13 and 14) a comparison between using the all 7 sensors with a downsampling factor of 16, and the most appropriate 4 sensors at a downsampling factor of 32.

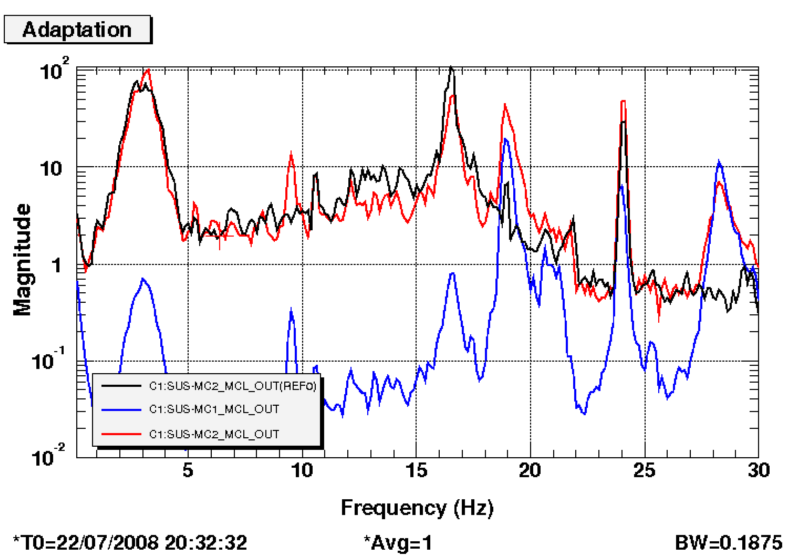

Figure 13: Adaptation with $\mu = 0.005, \tau = 0.01, N = 800$, Downsampling rate = 16. The black curve represents the output of actuation on MC2 when the adaptive code is not running. The red one is the MC2 output when it is running, and the blue is the MC1 output when the code is running. It is clear there is not much suppression. All the sensors and the seismometer were used.

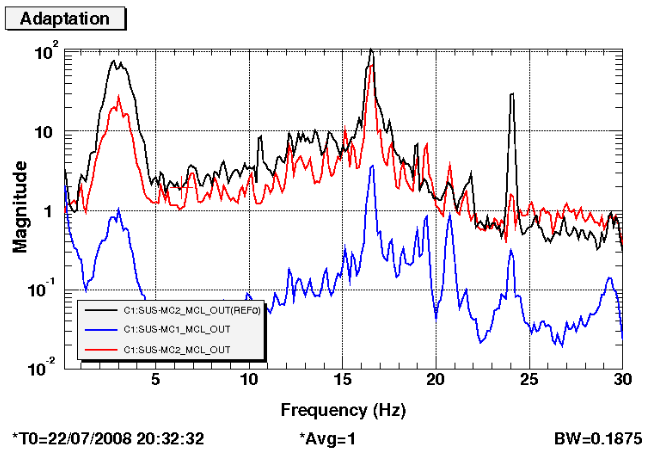

Figure 14: Adaptation with $\mu = 0.005, \tau = 0.01, N = 800$, Downsampling rate = 32. The black curve represents the output of actuation on MC2 when the adaptive code is not running. The red one is the MC2 output when it is running, and the blue is the MC1 output when the code is running. This time it is possible to notice some suppression under 16 Hz. Only the Y and Z accelerometers were used.

5 Future Work

We have successfully demonstrated the suppression of seismic disturbances in a fully suspended optical cavity using adaptive filtering. This result provides further evidence that adaptive filtering could benefit the LIGO detectors and improve their signal to noise ratio. When appropriately implemented, adaptive control techniques can reduce noise from seismic disturbances, acoustic couplings or any other noise for which there exists an appropriate witness sensor.

In order to fully exploit this filtering method, we should continue investigating the significance of the different parameters in the code to find the values which simultaneously optimize both stability and noise mitigation. Thereafter, we can expand our efforts to higher frequencies. This will likely entail the installation of an array of microphones as the noise witnesses and loudspeakers for performing injections.

Another area where adaptive filtering may be fruitfully applied is in the suppression of 60 Hz noise arising from the AC electrical supply. During initial LIGO no attempt was made to actively suppress this noise, rather the collaboration relied on good practice from its engineers.

Attenuation of the 60Hz noise is particularly important due to the 59.7 Hz radiation frequency of the Crab pulsar. During the recent science run LIGO was able to beat the classical spin-down limit for this pulsar by a factor of 3.4. Any mitigation of 60 Hz noise would increase this value and reduce the Crab limit still further.

Due to the concerns it raises for LIGO and the ubiquity of power line noise in other systems I have chosen to make suppression of 60 Hz using adaptive filtering the keystone of my senior thesis.(M. Landry) [9].

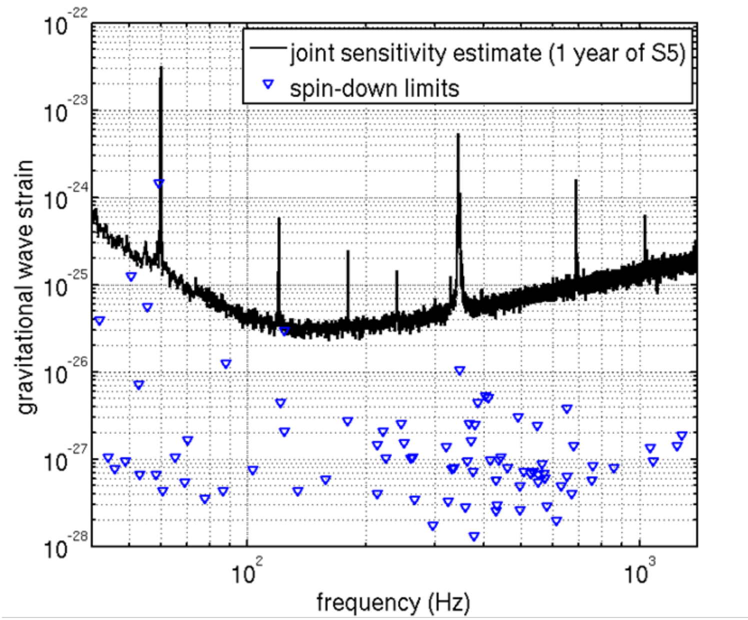

Figure 15: Search sensitivity during S5 run and the spin-down limits for pulsars in LIGO's sensitivity band [9]

The results of this adaptive filter work are promising, once this method has been completely characterized, it is likely that this knowledge will be ported to the sites and see widespread implementation.

References

[1] An Overview of Recycling in Laser Interferometric Gravitational Wave Detectors, D.E. McCelland, *Australian Journal of Physics*, Vol.48, 1995, P.953

[2] Sensitivity and Noise Analysis of 4 km Laser Interferometer Gravitational Wave Antennae, R. Adhikari, *http://www.ligo.caltech.edu/docs/P/P040032-00.pdf*

[3] Digital signal processing (3rd ed.): principles, algorithms, and applications, J. Proakis, D. Manilas, Prentice Hall, 1996, ISBN:0-13-373762-4

[4] Modelling of the Effects of Beam Fluctuations from LIGOs Mode Cleaner, N. Jamal, *http://www.ligo-la.caltech.edu/archive*

[5] Simon Haykin Adaptive Filter Theory, Prentice Hall, 2002, ISBN 0-13-048434-2

[6] Online Adaptive Filtering: The Bork-Space XFCODE, M. Evans, *http://www.ligo.caltech.edu/docs/T/T080151-00.pdf*

[7] The MathWorks, Inc.,R.A. Losada, Practical FIR Filter Design in MATLAB, *http://tagteamdbserver.mathworks.com/ttserverroot/Download/14883_firdesign.pdf*

[8] Optomechanical Alignment Instability in LIGO Mode Cleaners, J. Driggers, *http://www.ligo.caltech.edu/ cit40m/Docs/Driggers_Final_Paper_11Oct06.pdf*

[9] Beating the spin-down limit on gravitational wave emission from the Crab pulsar, M. Landry, 2008 APS *www.ligo.caltech.edu/docs/G/G080190-00/G080190-00.ppt*

A Adaptive C code

Written by Matt Evans, modified by Alex Ivanov
This is a summary, includes relevant adaptive algorithm of the code

```
/* adaptive filter algorithm, based on NLMS (normalized least-mean-square)
* and filtered X-LMS algorithms.
* This code is intended to be used in the LIGO Bork-space environment,
* initially as an arbitrary C code block.  As such its interface to the
* world is through an input array and an output array.  The expected data
* order in these arrays are:
* datIn[0] = reset flag
*   change from previous value to any value other than zero to reset
* datIn[1] = number of FIR coeffients (on reset)
* datIn[2] = downsample ratio (on reset)
* datIn[3] = corr to error extra delay (on reset)
* datIn[4] = number of aux channels (on reset)
* datIn[5] = adaptation gain
* datIn[6] = decay rate (fraction per downsampled cycle)
* datIn[7] = not used
* datIn[8] = diagnostic request 1
* datIn[9] = diagnostic request 2
* datIn[10] = error signal
* datIn[11] = auxiliary channel 1, signal path
* datIn[12] = auxiliary channel 1, adaptation path
* datIn[13] = auxiliary channel 2, signal path
* datIn[14] = auxiliary channel 2, adaptation path
* ... more auxiliary channel pairs ...
* datOut[0] = correction signal
* datOut[1] = diagnostic signal 1
* datOut[2] = diagnostic signal 2
// Coefficient storage
  // pointers into FIR and adaptation buffers
  // Servo enable flag
  // make sure there are enough signals
  // ================ RESET
    // reset with new parameters
    // add sample-and-hold delay, and one sample turn around delay
    // ======== Check for Sanity
    // clear the delay buffer
  // Load coeffs from storage
  // Only reload when pLocalEpics->ass.TOP_SUS_ENABLE goes from 0 to 1
    // Only if the number of coeffs didn't change
  // Save servo enable flag
  // update reset flag, nDelay, etc.
  state.resetFlag = datIn[0];
```

```
  state.nDelay = (int) datIn[3];    // extra aux delay
  adaptGain = datIn[5];
  decayRate = datIn[6];
  // add sample-and-hold delay, and one sample turn around delay
  // ================ Update State
  // downsample counter
  // delay counter
  // reset wait counter
    // increment wait counter
    // modify adaptation gain for soft start
  // ================ Process Signals
  // filter error signal (or just copy)
  // process auxiliar signals to make correction signal
  // Save the number of coeffs
  // Save the number of Aux inputs
    // pointer for easy access
    // correction path input
    // adaptation path input, with circular delay buffer
    // ======== Downsampled Operations
      // compute the adaptation change
      if( thisAux->prevNorm != 0.0 )
    delta = adaptGain * sigErr / thisAux->prevNorm;
      else
    delta = 0.0;
      // put sigCorr and sigAdapt at end of buffers, after HP filtering
      // computer FIR output and perform adaptation
      norm = 0.0;
      for( j = 0; j < state.nFIR; j++ )
      {
    // FIR
    *pBufFIR = *(pBufFIR + 1);        // shift the buffer
    corr += (*pCoefFIR) * (*pBufFIR);    // add to the output
    // Adaptation
    *pBufAdapt = *(pBufAdapt + 1);        // shift the buffer
    *pCoefFIR *= (1.0 - decayRate);        // decay FIR coef
    *pCoefFIR += delta * (*pBufAdapt);    // update FIR coef
    norm += (*pBufAdapt) * (*pBufAdapt);    // add to the norm
    // Store
    if (st) cstor[2 + i * state.nFIR + j] = *pCoefFIR;
    // move pointers
      // keep the norm for next time
}
```

B FIR Wiener Filter Algorithm

```
function [W,R,P] = miso_firwiener(N,X,y)
%MISO_FIRWIENER Optimal FIR Wiener filter for multiple inputs.
%   MISO_FIRWIENER(N,X,Y) computes the optimal FIR Wiener filter of order
%   N, given any number of (stationary) random input signals as the columns
%   of matrix X, and one output signal in column vector Y.
%   Author: Keenan Pepper
%   Last modified: 2007/08/02
%   References:
%     [1] Y. Huang, J. Benesty, and J. Chen, Acoustic MIMO Signal
%     Processing, Springer-Verlag, 2006, page 48
% Number of input channels.
M = size(X,2);

% Input covariance matrix.
R = zeros(M*(N+1),M*(N+1));
for m = 1:M
    for i = m:M
        rmi = xcorr(X(:,m),X(:,i),N);
        Rmi = toeplitz(flipud(rmi(1:N+1)),rmi(N+1:2*N+1));
        top = (m-1)*(N+1)+1;
        bottom = m*(N+1);
        left = (i-1)*(N+1)+1;
        right = i*(N+1);
        R(top:bottom,left:right) = Rmi;
        if i ~= m
            R(left:right,top:bottom) = Rmi';  % Take advantage of hermiticity.
        end
    end
end

% Cross-correlation vector.
P = zeros(1,M*(N+1));
for i = 1:M
    top = (i-1)*(N+1)+1;
    bottom = i*(N+1);
    p = xcorr(y,X(:,i),N);
    P(top:bottom) = p(N+1:2*N+1)';
end

% The following step is very inefficient because it fails to exploit the
% block Toeplitz structure of R. It's done the same way in the built-in
% function "firwiener".
W = P/R;}
```

C Least-Mean-Square (LMS) Algorithm

N = filter length

μ = step-size

$$\mathbf{u(n)} = [u(n), u(n-1), ..., u(n-N+1)]^T \tag{6}$$

For $\mathbf{u(n)}$ being the input noise signal.

$$d(n) = W(n)^H \mathbf{u(n)} \tag{7}$$

error signal:

$$\epsilon(n) = d(n) - y(n) \tag{8}$$

For ϵ being the error between the signal output $y(n)$ and the desired signal $d(n)$.

$$W(n+1) = W(n) + \mu\epsilon\mathbf{u(n)} \tag{9}$$

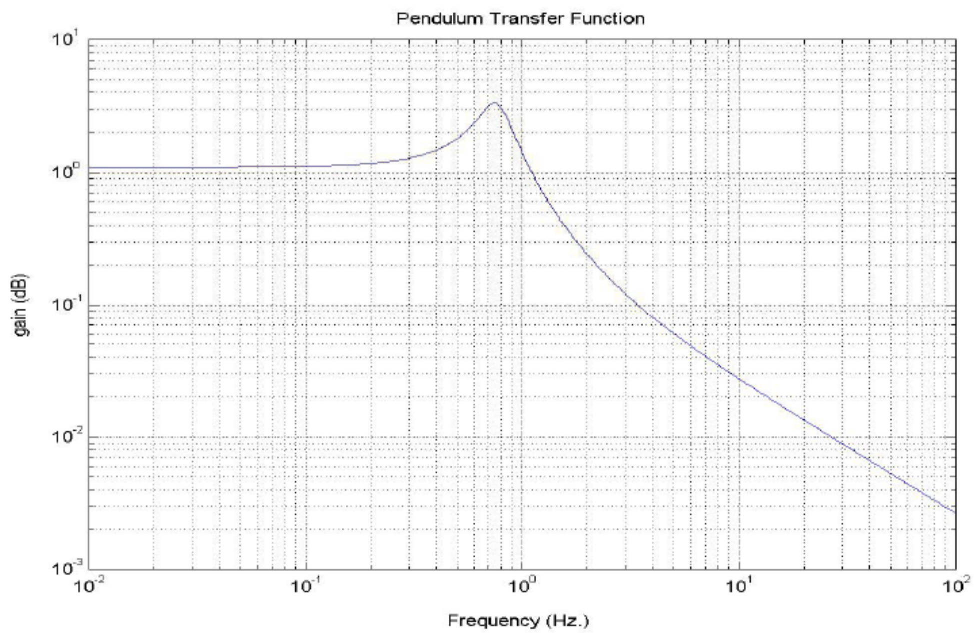

Figure 16: Pendulum transfer function

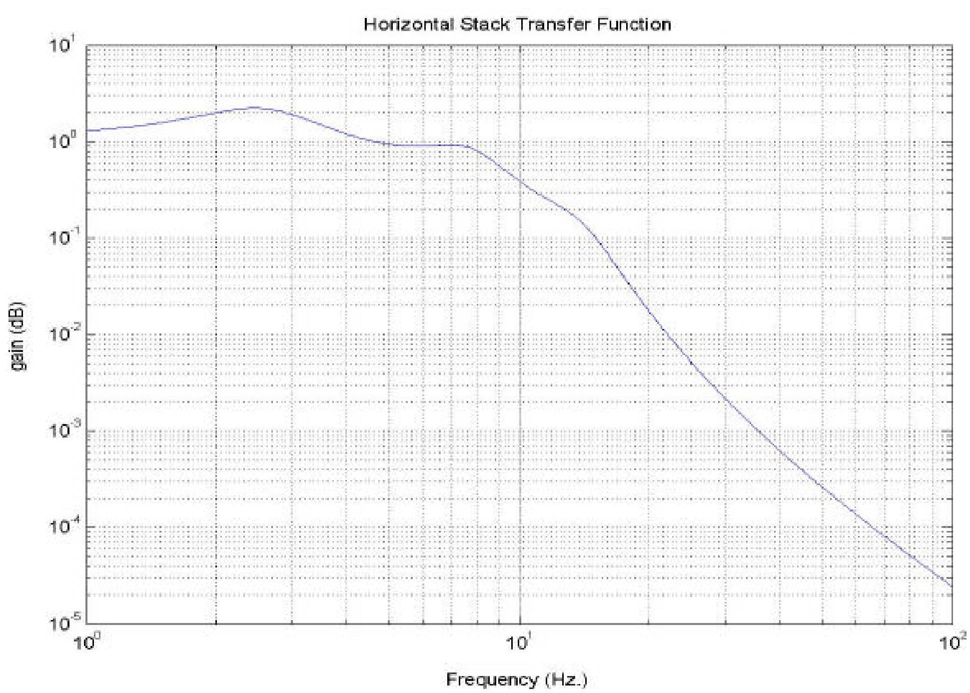

Figure 17: Stack transfer function