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

Technical Note **LIGO-T1400634–v1** 2014/09/30

Designing a frequency offset locking loop for the 40m prototype Arm Length Stabilization System

SURF Student: Sai Akhil Reddy K., Mentors: Manasadevi P Thirugnanasambandam, Eric Quintero , Koji Arai

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

The primary goal of LIGO is to directly detect gravitational waves which would open doors for ground-breaking observations in the field of astrophysics and give us a better understanding about the universe. For such a huge detector, there is a dire need of automated systems to control the state of the interferometer in a very efficient and precise manner. The goal of my project is to design a digital PID loop to lock the beat frequency between the Pre-stabilized Laser(PSL) and the Auxiliary (AUX) laser, within the working range (<100 MHz) of the Arm-Length Stabilization(ALS) system. To design such a loop, it is necessary to estimate the transfer functions and noise of all the control blocks like sensors, actuators, transducers inside the loop such as the frequency counter and the temperature actuator. Also,the interfacing between the electronics and the control computers is required as a part of PID loop design and a remote frequency beat-note readout. The readout has been tested and verified with the green laser beat note and the PID controller has been designed to actuate on the frequency of the AUX laser via a Piezo-Electric Transducer (PZT).

Acknowledgements

I would like to sincerely thank my mentor, Dr. Manasadevi Thirugnanasambandam, for all the insight and guidance offered during the course of this project. I would also like to thank my co-mentors: Eric Quintero for guiding me and putting me along the right path all the time, and Koji Arai for bearing with my questions and solving puzzles that cracked my head throughout. I would also like to thank Prof. Rana Adhikari for consistently helping me set project goals and also improve the quality of my plots I generated. I would like to thank my co-intern and project member Harry Hall for helping me make the hardware FOL box required for the project. I also had the privilege of working with a large number of people involved with the LIGO project, both at the 40m and at West Bridge. I would like to thank all of them (Dr. Jameson Rollins in particular) for their help and advice. The summer of 2014 was a great experience working with a gravitational wave detector and a highly motivating project for someone to work on the engineering applications in pure sciences.

Finally, I would like to thank the SURF community, the Student Faculty Programs Office, LIGO, the NSF, and IIT Patna for giving me the opportunity to work at Caltech for 10 weeks. It was one of the most enriching learning experiences that I have had in my three years of bachelor studies.

Contents

List of Figures

1 Introduction

1.1 Gravitational waves and Detectors

Gravitational Wave (GW) is a ripple of space-time and propagates at the speed of light. On the basis of the general theory of relativity , GW was predicted by A. Einstein in 1918. A gravitational wave can be treated as a distortion or perturbation in space time, expressed by a 4- dimensional metric which follows Einstein's equation. This also means that measurement of this strain in space-time gives us the knowledge of the gravitational wave. However, direct measurement of gravitational strain is not that trivial. From the derivation of the GW from Einstein's equation [1] , the strain of a GW is greatly diminished due to the ratio of gravitational constant,G to the fourth power of speed of light c.

$$h \approx G/c^4 \approx 8.23 \times 10^{-45} m^{-1} kg^{-1} s^2 \ . \ . \ . \ . \ . \ .(1)$$

However, characteristics of lasers such as the large coherence length and high output intensity coupled with low divergence, enables us to measure small and precise length measurements using interferometric techniques. To serve the purpose, the so-called 1st generation(1G) interferometers were built and operated over the past years and conducted several science runs. However none of them could really detect GWs owing to the limitations in the sensitivity of the detectors. Now the 2nd generation interferometers are being upgraded or constructed to increase the sensitivity by a factor of approximately 10 compared with that of the 1st generation interferometers. The 2nd generation(2G) interferometer typically consists of a

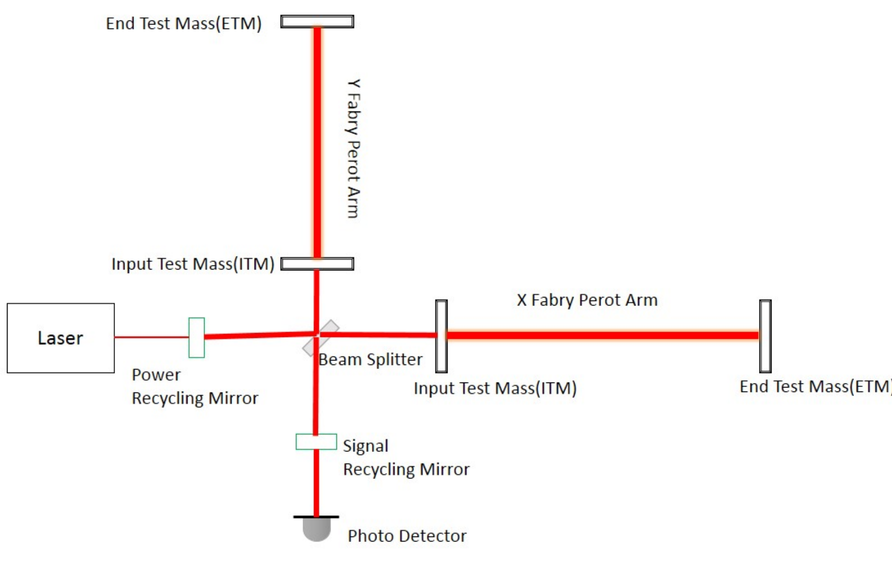

Figure 1: Dual Recycled Fabry Perot Michelson Interferometer.

Michelson interferometer [2] enhanced by the addition of Fabry-Perot (FP) arms and recycling cavities at the input and output sides of the interferometer, forming the Dual-Recycled Fabry-Perot Michelson interferometer (DRFPMI) as in Figure 1.

A major challenge in using this interferometer is in bringing the interferometer to a quiescent operating point in order to achieve the desired interference and make precise measurements. This iterative progression where the interferometer is brought to the final state, where all the optical lengths are brought to the operating point from an initial state of arbitrary lengths and random interference, is generally called as Lock Acquisition [3].

1.2 Arm Length Stabilization System and Length Sensing

To mitigate the problems in 2G interferometers such as higher degrees of freedom due to coupling of cavities, non-linearities in length measurement and noise, a scheme which uses multicolour laser interferometry technique [4], known as Arm Length Stabilization(ALS) is adopted. But for such a scheme, we need to have the length information first. The length information of an optical cavity is extracted by phase modulation of the laser field and synchronous demodulation by a scheme called Pound-Drever-Hall(PDH) technique [5]. The modulation frequencies are usually in the RF range to minimize laser noises. The phase modulated signal gets affected by the cavity via reflection or transmission, which means after photodetection and demodulation of the photo current, a linear signal containing the length information can be obtained.

The cavity arm length can be stabilised independent of the rest of the interferometer. This scheme uses frequency doubled Non Planar Ring Oscillator (NPRO) of wavelength 532nm in order to sense the arm lengths. The frequency control of this auxiliary(AUX) laser can be achieved either by piezoelectric transducer or by temperature actuator [2]. This AUX laser is injected through the End Test Mass(ETM) and the dichroic mirrors ensure that the AUX laser couples only with the laser cavity by PDH Locking. The transmitted light from the arm cavity propagates to the vertex of the interferometer and then combines with the frequency doubled PSL thus forming a beat note [4]. As an auxiliary tracker, a photodetector (PD) is placed at the transmission side of the cavity to record the aforementioned beat note for driving the cavity length. This frequency of the beat note obtained at photodetector by interfering the AUX laser transmitted from one end of the arm cavity, and the PSL must be checked in order to verify the working and robustness of the prototype by generating a certain voltage error depending on the sign of the DFD(delay line frequency discriminator) signal. This error signal is then fed back to the coil magnet actuator of the ETM in order to stabilize the length of the cavity using a digital servo called Arm Length Stabilization(ALS) [4]. However, this ALS scheme requires a progressive sequence since the stabilization of arm length does not necessarily imply that the PSL is fully resonant. For this the arm length has to be brought to a desired point every time the arm length is stabilized.

2 The Problem, Project Objectives and the Approach

2.1 The Problem

But there are few problems associated with the ALS technique. The frequencies of two lasers are different because they are locked to different references (could vary by few GHz) . The PSL is locked to the mode cleaner and AUX is locked to the arm cavity using PDH scheme. The PDH control loop of the AUX laser allows its frequency to follow the motion of the cavity length. The frequency of the AUX laser can be altered either by a piezoelectric transducer or by a temperature actuator.The actuator response of the laser cavity actuation is very small(5 MHz/V) and the frequency shift of the AUX laser for approximately 1 micrometer arm cavity displacement is around 7 MHz. So for this reason, to improve the control parameters, the temperature feedback is added so that the offset in the laser cavity feedback path is loaded to the temperature actuator with a much larger response(1 GHz/V). So in operation, if the beat-note is not within the detectable range, one needs to change the temperature of the AUX laser to bring the beat-note within the efficient working range of the ALS servo. Changing the temperature unlocks the other AUX laser frequency control systems like the cavity length actuator, however, it quickly reverts because the control bandwidth is fast. For this, we try to build a control loop for the AUX laser to bring and keep track of the beat note frequency such that it is always within the efficient working range of ALS (typically a beat frequency of < 100 MHz) by designing a slow feedback servo which by changing the temperature of the AUX laser, can serve the purpose.

2.2 Project Objectives

The main objective of my project is to design a digital PID Loop [6] to lock the frequency of AUX laser with the main Pre-Stabilized Laser using the beat note frequency detected at a RF photodiode, making two lasers interfere in order to sense the difference of the frequencies between two lasers. This loop will serve as a servo by driving the temperature of the auxiliary NPRO according to the desired set frequency difference between the two lasers. An Integro-differential loop has to be constructed which satisfies the condition that the beat note frequency must be in the working range of ALS(< 100 MHz).

2.3 The Approach

The Proposed design of PID loop typically consists of the following configuration: First the frequency difference obtained from the beat note, is fed into a frequency counter which counts the frequency change according to the offset. This set point frequency error signal is sent into the digital world via an ADC. Now we have a digital signal which is passed into the PID(constant-integro-differential) loop. This is processed and then used for the temperature control of the laser through a DAC. The configuration with the setup is shown in figure 2 . The steps involved in this design are:

- Make an optical setup to detect the beat frequency between the AUX laser and PSL at an RF photodiode.
- Measure the the beat note frequency at a frequency counter.

- Filter this error and digitize it(for digital control).
- Design a PID control loop.
- Actuate on the temperature of the AUX NPRO via a PID Loop control signal.
- Lock the loop such that the beat note frequency is measurable(< 100 MHz).

In the previous sections, a conceptual summary of the project was defined. In the subsequent sections, the aspects of implementation of the PID loop, the progress made thus far in this respect, and the plan of action for the coming month will be discussed.

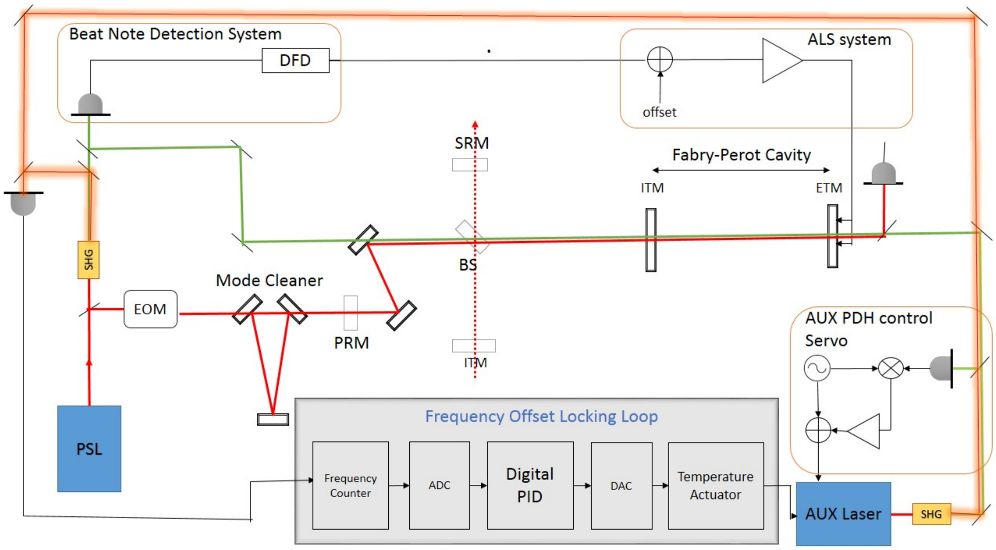

Figure 2: Schematic of interferometer with the proposed frequency offset locking loop.

3 Implementation of the Frequency Offset Locking PID Loop

For designing the required control loop, it is essential to characterize the plant(NPRO), transducers(Frequency Counter) as well as actuators(Temperature Actuator) involved in the loop. The design of the PID controller involves knowledge of transfer functions of all the control units involved in the system. The characterization of these control blocks is done by giving in a test input and recording the response, and estimating the transfer function of a particular block. The noise in the loop such as Frequency Counter quantization noise, Temperature actuator noise are also calculated as a part of PID loop design. The interface of the devices with the user-end EPICS is also simultaneously carried in python-C ctypes interface.

3.1 Characterization of Frequency Counter

As a part of the the PID loop design, characterization of the RF Frequency Counter(FC) used in the loop is being done. The response of the FC has been measured and the gain vs

frequency plots have been plotted. Plotting the Bode plots would give response of the FC to modulation which is used to estimate:
1) Transfer Function of FC.
2) Quantization noise from Power Spectral Density(PSD) vs Hz.

Details of Measurement

A modulated signal can be represented as

$X(t) = Asin(2\Pi \times F_c t + Dsin(2F_m t + \Phi));$
where F_c and F_m are carrier and modulation frequencies respectively and D is the modulation depth.
This signal Y(t) is input to the FC and the output frequencies of the FC are recorded.
Let the output of the FC be

$Y(t) = A^{'} sin(2\Pi F_c^{'} t + D^{'} sin(2F_m^{'} t + \Phi^{'}));$

$$Gain = D'/D;$$
$$phase = \Phi^{'} - \Phi;$$

$D^{'}$ is calculated by subtracting the carrier frequency from the output frequency and calculating the amplitude of the resulting fitted sine wave.

The sampling rate of the RFFC is 0.1s. So to satisfy Nyquist criterion, the maximum modulation frequency is 5 Hz beyond which aliasing effects are seen.The gains plots are plotted for carrier frequencies of 5 MHz and 25 MHz as in Figure 3.

3.2 Determination of Measurement Error of the Frequency counter

The measurement statistics of the instrument would give knowledge about the error and tolerance in the measurement which will be helpful to negotiate the error when the counter is being used in the setup.

The output of FC was measured for different test input frequencies. The obtained histograms(for sampling time of 1s) at different test frequencies are attached in figure 5.

The measurement error of the UFC-6000 RF Frequency Counter changes with sampling time from 0.01 Hz-0.02 Hz. This error varies at different frequencies as inferred from figure 5. The error for different sampling times at input frequency of 1 MHz of the FC are also plotted 6.

3.3 Interfacing the Frequency counter with Raspberry Pi

The RF Frequency Counter(FC) is an electronic device which takes an analog RF input and gives the frequency of the RF signal digitally. The range of the RFFC is 1MHz - 6000MHz.

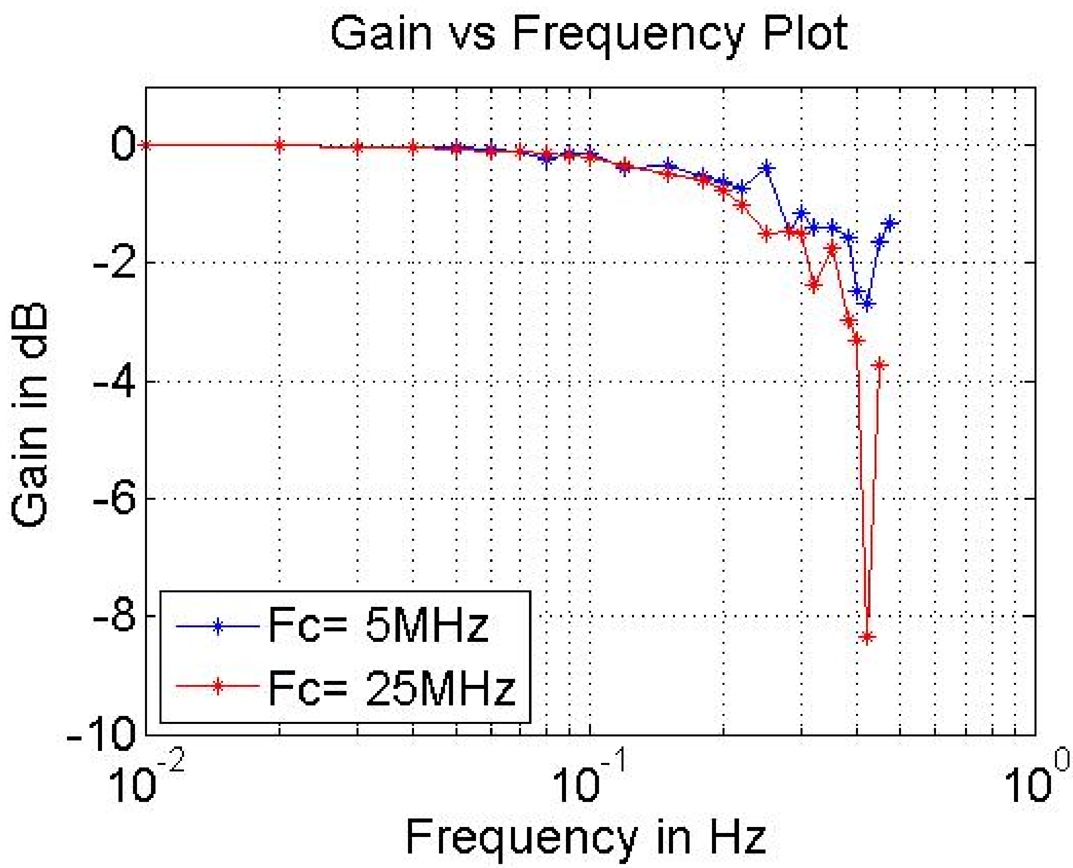

Figure 3: Gain vs Frequency Plot.

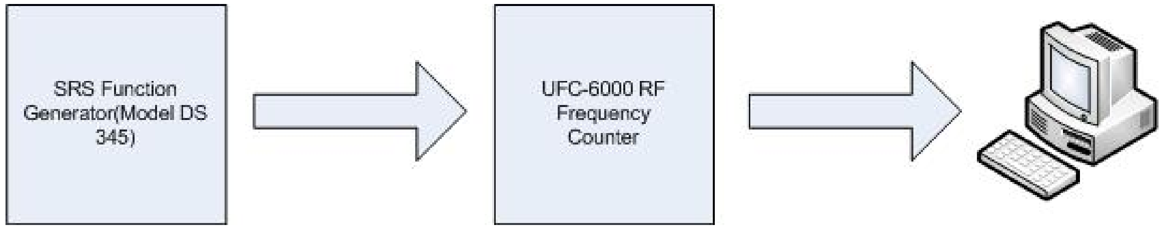

Figure 4: Frequency Readout Setup.

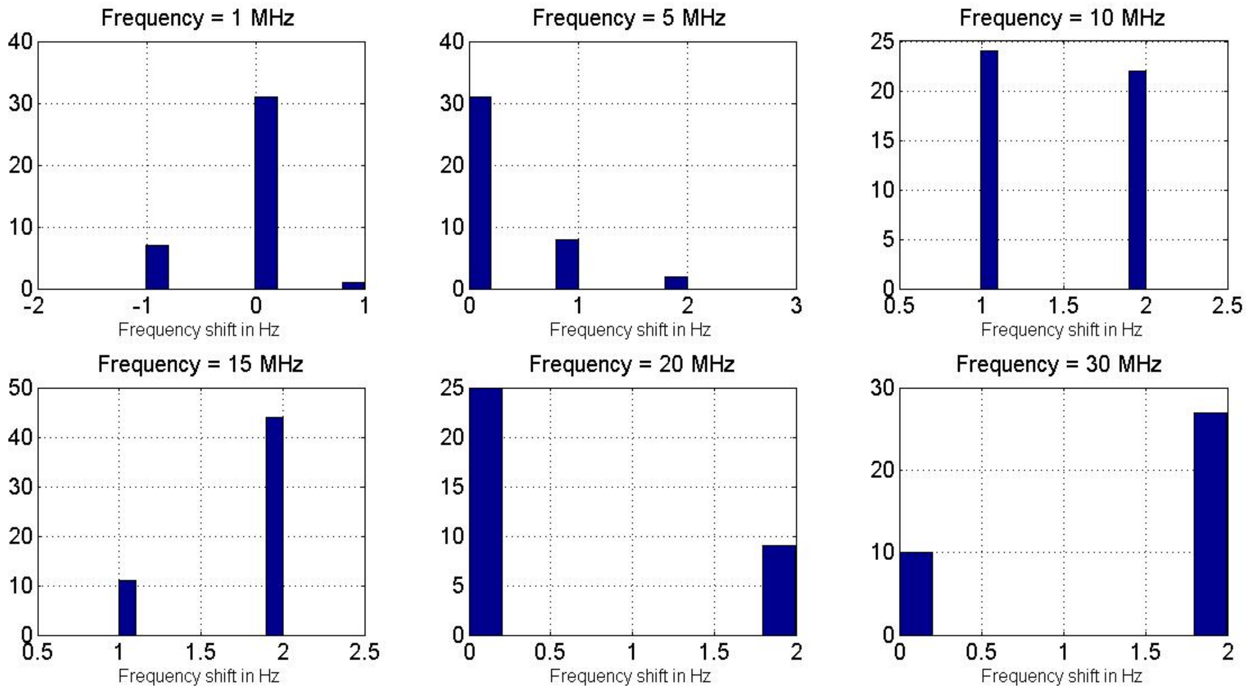

Figure 5: Frequency Counter Measurement Error for 1s Sampling time.

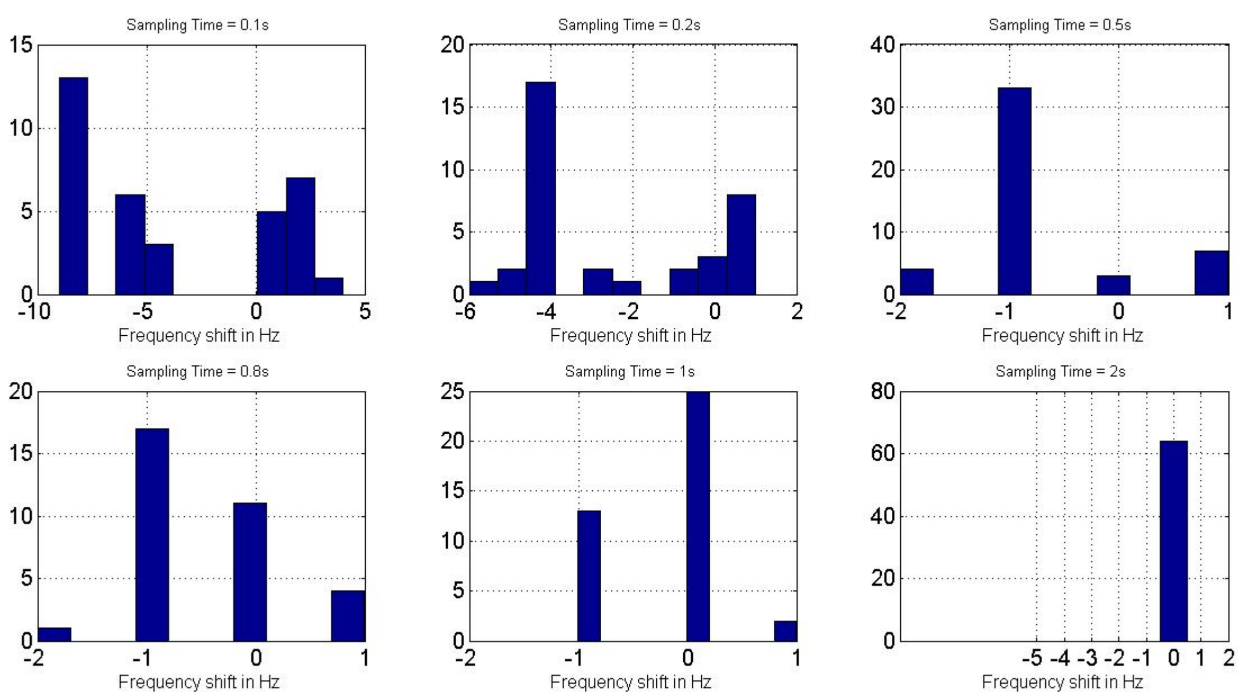

Figure 6: Measurement Error for different Sampling times for 1MHz.

This device is a USB-HID device which can talk to an EPICS network via a computer with Linux platform. A C code is written which can read and write byte codes into the device can be referred in
Interface Byte Code Description. The frequency output of the FC is available real-time for time-to-time beat note control. The following can be achieved via the Linux interface:
1) Read The Frequency
2) Set the Range
3) Set the Sample Time

3.4 Issues involved in Raspberry Pi and Frequency Counter Interface

3.4.1 Main Problem

The FC takes in an analog RF input(signal) and outputs the frequency of the signal(Ranging from 1 MHz- 6000 MHz) in the digital domain (into a processor). The FC samples the data with a given sample rate(user defined) which ranges from 0.1 s to 1 s(faced problems in fixing this initially). For data acquisition, we have been using a Raspberry Pi(as a processor) which is connected to the martian network and can communicate with the computers inside the 40m. Raspberry Pi will be running Raspbian which is a version of Linux, and not a RTOS. When sampling data at a certain frequency we want samples to occur at fixed time intervals corresponding to the sampling period. A normal operating system cannot provide us with this functionality, and there will be jitter (variation) in the time difference between consecutive samples. Whether this is an issue depends on how much jitter we have and what the specific application is. In our application(measuring phase and noise), the jitter has to be taken into consideration. Hence for data acquisition we need to sample with much more tightly defined sampling periods (reduced jitter) which can be done by providing an external timing standard(Like a square pulse of the frequency same as the sampling rate of the FC). The ultimate challenge which I faced(and knocked my head off for two weeks) is the synchronization of clocks between the Raspberry Pi and the FC i.e the clock which the FC uses to sample and dump data(every 'x' s) and the clock inside the raspberry pi(used in the loop to wait for a particular amount of time the frequency counter takes to dump successive data).

3.4.2 Steps Taken

To address this problem, first I added an external clock circuit which monitors the Raspberry Pi and the FC to dump and read data at a particular rate(which is equal to the sampling rate of the FC). While doing so, at first the level trigger algorithm was used which means that the external clock frequency was half as that of the reciprocal of the sampling rate and a trigger was seen every time the level shifts from +DC to -DC(of the external square wave). But this did not completely mitigate the issues and there were still few issues on how quickly the ADC reads the signal and R Pi processes it. To minimize these issues completely, an edge trigger algorithm which detects a posedge(rising) of the clock was used. The clock frequency is now equal to the reciprocal of the sampling rate. This algorithm showed better results and greatly minimized the drift of the sampling time. The setup used for timing synchronization is shown in figure 7

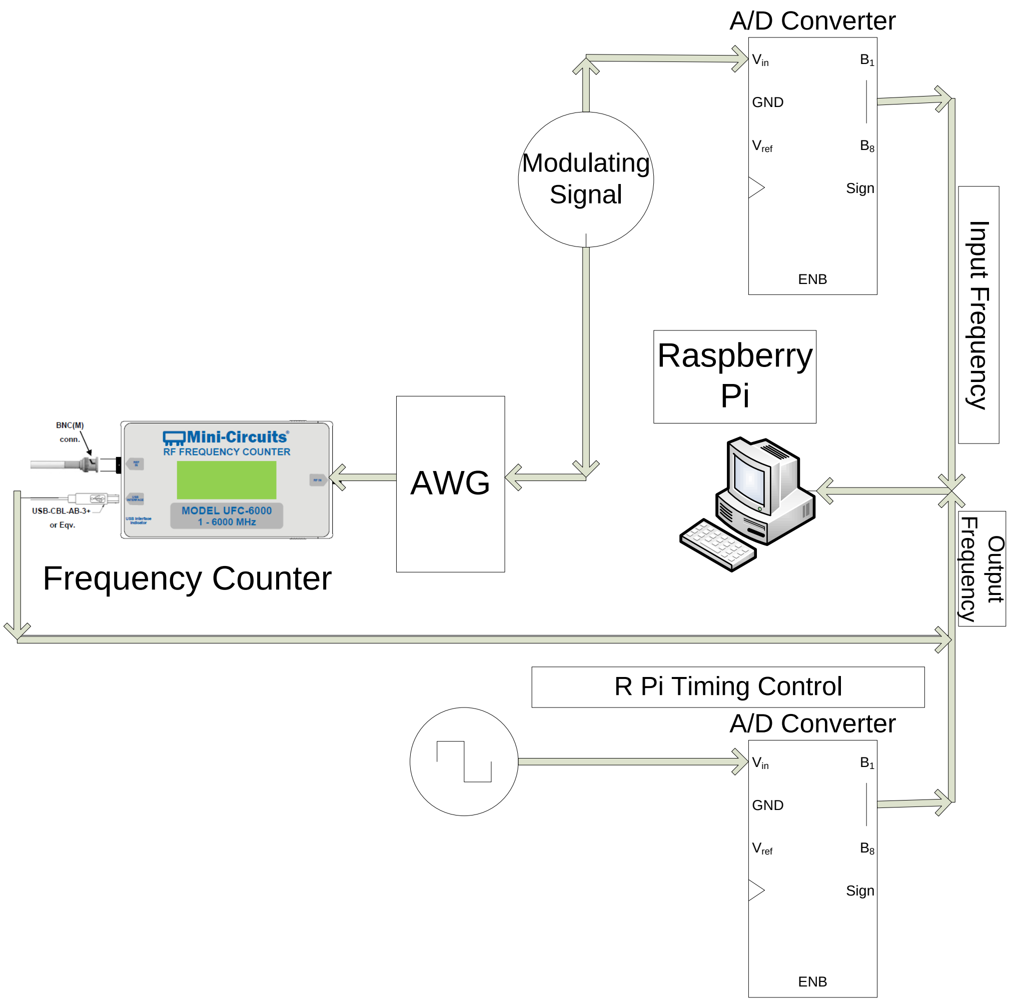

Figure 7: Frequency Counter Setup for Characterization.

3.4.3 Pseudo Code

open device : FC via USB-HID;
open device : ADC via I2C;
always(for t= recording time):
read data from ADC(external clock);
if pos edge detected:
read data from FC and store it in a register;
else read data from ADC;
end
write data stored in the register to a file(can be an Epics channel or a text file)

3.5 Characterization of FC with improved Timing

3.5.1 Gain Plots

The FC seems to behave like a Low pass filter with 3dB points of:
2.6 Hz for a sampling time of 0.1 s (Figure 3)
0.3 Hz for a sampling time of 1 s. (Figure 8)

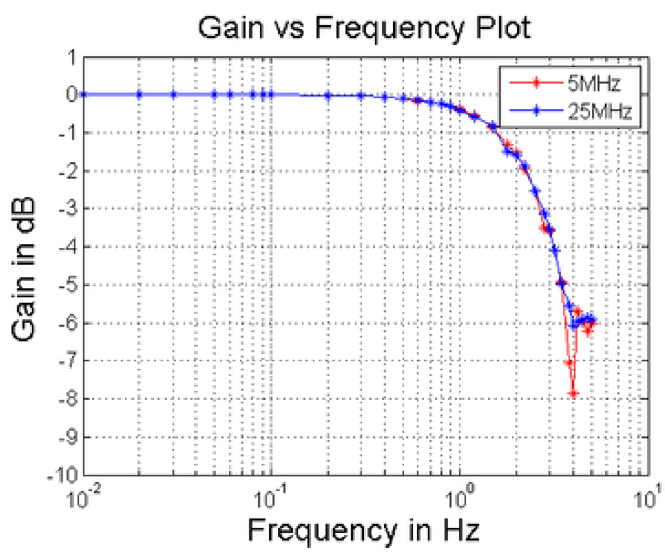

Figure 8: Gain at Range 1 Frequencies of FC.

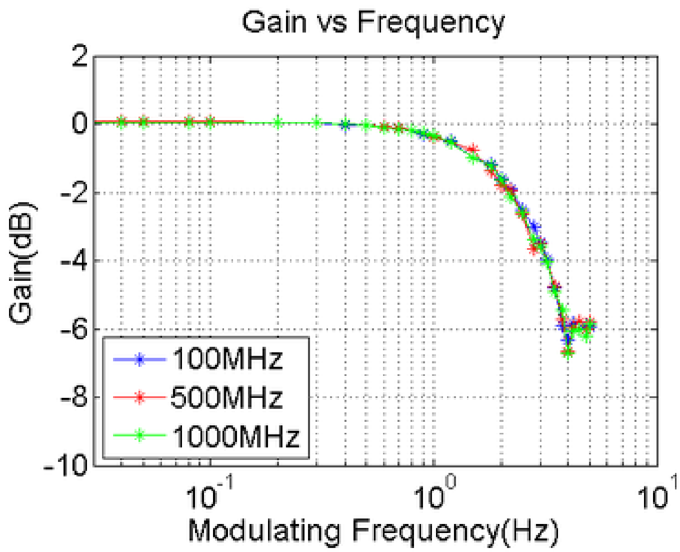

Figure 9: Gain at Range 2 and 3 Frequencies of FC.

3.5.2 Bode Plot of the FC

The steps I followed to generate these plots are:

Took the FFT of both FC out data(from FC) and Modulation input(from SRS via ADC). Estimated the phase angles at the particular modulation frequencies from the FFT data(in Matlab using angle(x) for phase at the frequency f(x); x: is the frequency bin) Then for the phase of the system at a particular modulation frequency, Phase(system) = Phase(FC Signal) - Phase(Input Signal)

From the plots its can be inferred that :the delay of the FC is almost 0 until the modulation of 0.1 Hz. Then there are phase shifts of ±180° showing that the system has multiple poles and zeroes.

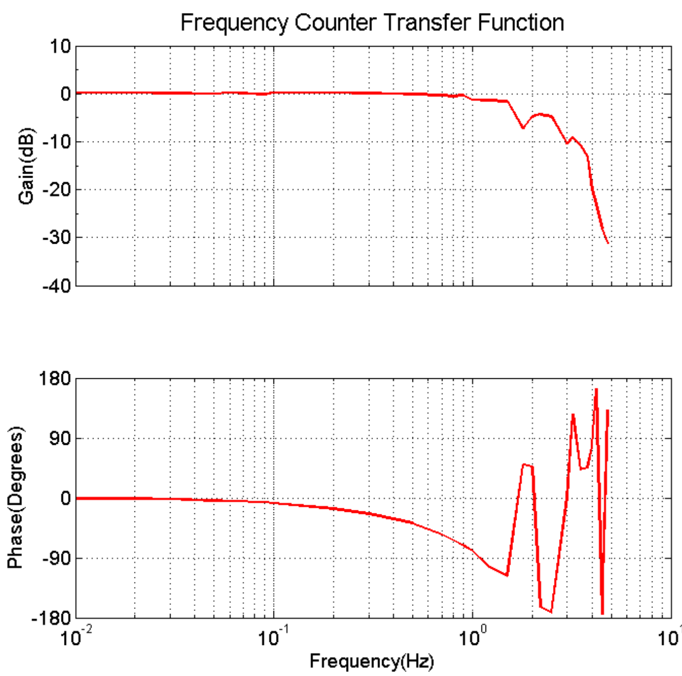

Figure 10: Transfer Function Characterization of FC.

3.5.3 Quantization Noise from Power Spectral Density Plots

The noise floor of the PSD plot(figure 11) represents the quantization noise of the FC. The quantization noise due to sampling is: 10^{-2} $Hz/\sqrt{Hz}$ for 1s sampling time and 10^{-4} $Hz/\sqrt{Hz}$ for 0.1 s sampling time.

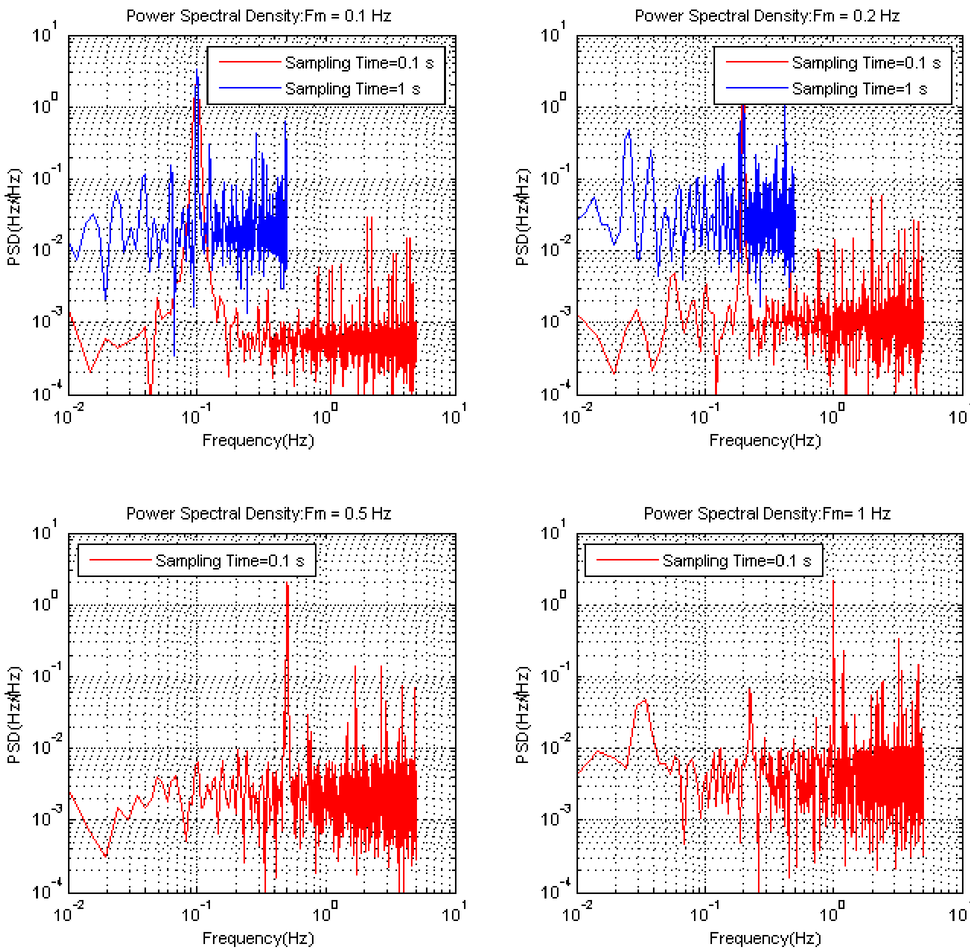

Figure 11: Power Spectral Density vs Modulation Frequency.

3.5.4 Comparison of Theoretical Estimate and Measured Frequency Counter Transfer Function

One of the standard technique used to improve the quality of the measurement(for example reduction of quantization noise) in devices like the Frequency counter is averaging.

There are two general use-cases for averaging . The first, successive sample averaging, takes a single acquisition and averages between its samples. The second, successive capture averaging, combines the corresponding samples of multiple captures to create a single capture.

Successive sample averaging is also called boxcar filtering or moving average filtering. In an implementation of this type of averaging each output sample represents the average value of M consecutive input samples. This type of averaging removes noise (low noise seen from figure 11) by decreasing the device's bandwidth(could be one of the reasons why the FC operates in 4 different frequency ranges). It applies an LPF function with a 3dB point approximated by $0.433 * s/M$, where M is the number of samples to be averaged, and s is the sample rate in samples per second.

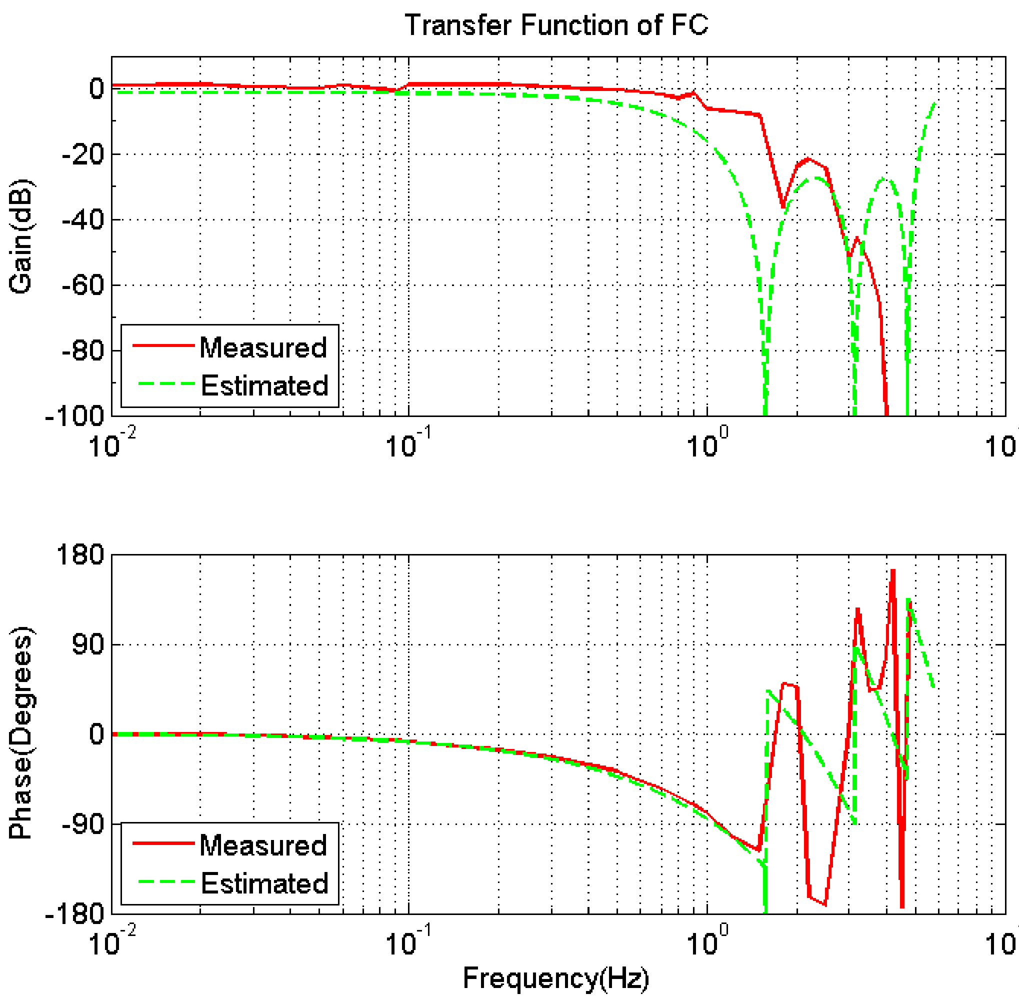

Figure 12: Comparison of Measured Frequency Response of FC with Predicted.

The 3 dB points in the measured gain plots :

For 1 s Sampling time : the 3 dB point for such a Boxcar filter should be at $0.433 * 1/M$. If we assume that it averages for 2 samples, M=2 which gives the 3dB point at 0.288 Hz but occurs somewhere between 0.3 and 0.4 Hz. (Figure 3)

For 0.1s Sampling time: the 3dB point should be at 2.17 Hz and in reality is 2.5 Hz(Figure 8).

Also, This type of filter will have very sharp nulls at frequencies corresponding to signals whose periods are integer sub-multiples of M/s. As seen in figure 10 there are sharp nulls at frequencies 0.4 Hz for 1S sampling time and at 1.5 Hz,3 Hz for 0.1 S sampling time as correctly predicted.

Therefore it can be concluded that this is a L-sample moving average FIR, with the frequency response as:

$$H(\omega) = (1/L)(1 - e^{-j\omega L})/(1 - e^{-j\omega})..$$

There is an overall delay of $(M - 1)/2$ samples from such a length-M causal FIR filter. The measured open loop Transfer Function of the FC is compared with the predicted TF of the boxcar filter as in figure 12

3.6 Characterization of Laser Thermal Actuator

The goal was to characterize the laser frequency thermal actuator that is a part of the FOL-PID loop. When the temperature of the AUX laser is altered, the lock of the arm cavity is lost, for an instant, when the Piezo-Electric Transducer(PZT) accordingly actuates on the ETM of the cavity, adjusting the length for lock acquisition. However, the direct transfer function between the AUX Laser frequency and the temperature offset of the AUX laser cannot be measured due to the fast PZ response, which brings back the laser frequency to the keep the arm locked. For this reason, indirect TF measurements for the thermal actuator were made as in figure 13 by tracking the Piezo-Electric Transducer(PZT) response by :
1) Arm cavity(End Test Mass ,Input Test Mass) displacement and
2) Temperature offset excitation.

Mathematically, $G_1 = TF_3/TF_1$ and $G_2 = TF_3/TF_2$ where,
TF_1= Temperature Offset (count)
TF_2= X and Y arm displacement(count)
TF_3= PZT response (count)

Then, $G_3 = G_2/G_1 = TF_2/TF_1$,which is the TF of Arm Cavity Displacement to the Temperature offset.Calibration of G_2 to count/m is done and finally, G_3 is calibrated to the desired Hz/Count using the equation:
dL/ L = dF /F where,
$F = c/\lambda$;
$\lambda = 532nm$;
L =
X arm length = 37.79 +/- 0.05 m
Y arm length = 37.81 +/- 0.01 m
Further details about calibration of the measurements can be seen at [7] . The final TFs obtained are the X and Y arm TFs for Laser frequency response vs Temperature offset in(HZ/count).The TF (poles and zeroes) of the actuator are used in estimating the type of controller required for the FOL.

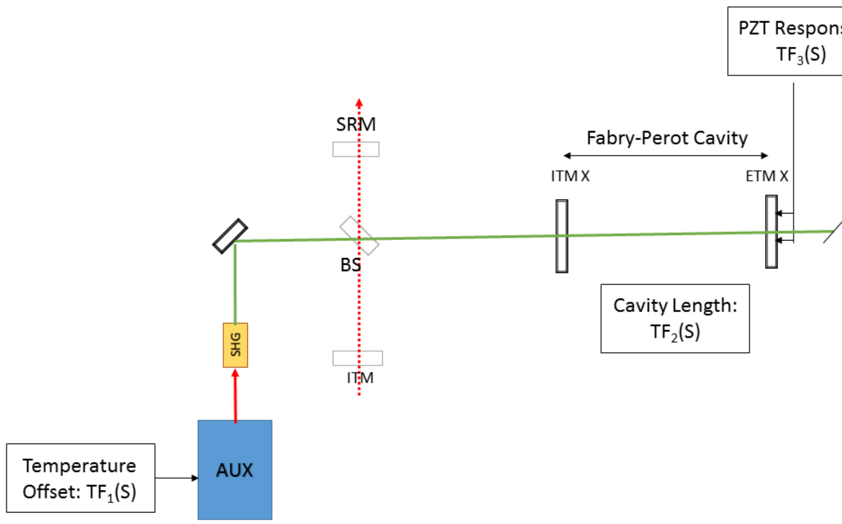

Figure 13: Thermal Actuator Characterization Setup

5th order TF fitted as shown in fig 14

Gain(k): 9000
Zeroes(z):
z1 = -0.9799;
z2 = 2.1655;
z3 = -2.9746- i * 3.7697
z4 = -2.9746+ i * 3.7697
z5 = 95.7703 + 0.0000i
Poles(p):
p1 = -0.0985- i* -0.0845
p2 = -0.0985+ i* -0.0845
p3 = -0.6673- i* -0.7084
p4 = -0.6673+ i* -0.7084
p5 = -8.7979.

$$G_3(s) = \frac{9000 * (s + 0.9799)(s - 2.166)(s^2 + 5.949s + 23.06)}{(s + 8.798)(s^2 + 0.197s + 0.01684)(s^2 + 1.335s + 0.9471)}$$

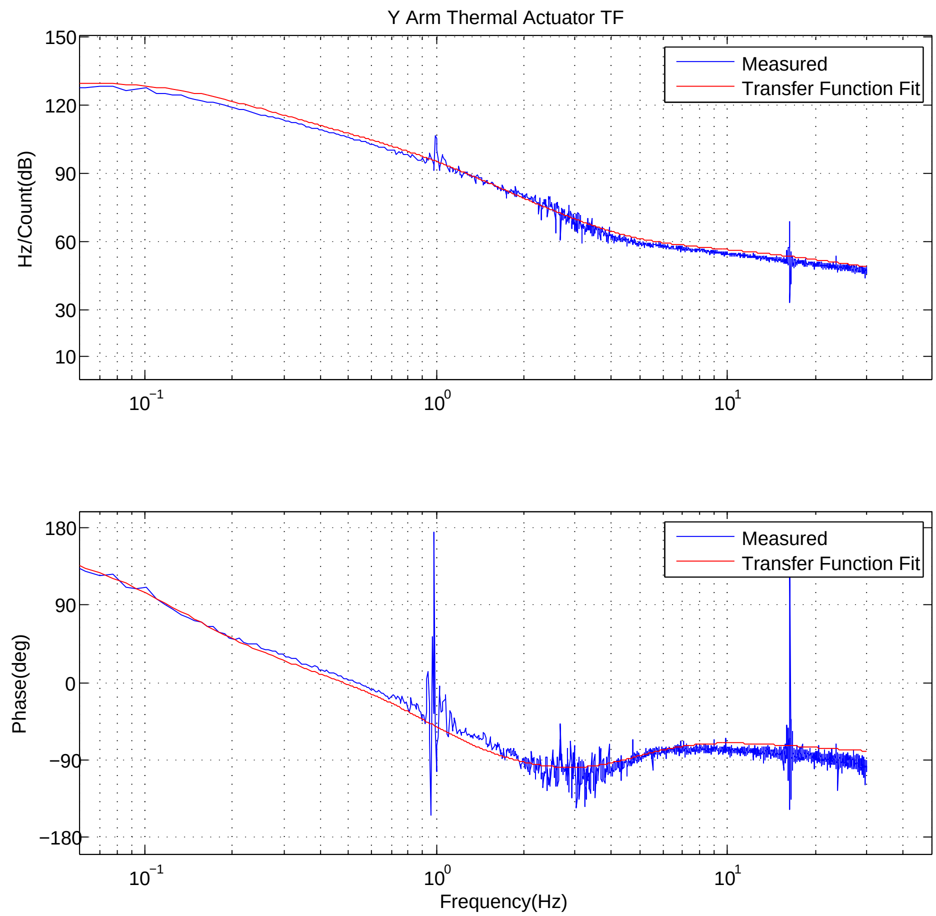

Figure 14: Thermal Actuator Transfer Function

3.7 Interface with EPICS

Experimental Physics and Industrial Control System(EPICS) is a set of Open Source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for instruments for scientific experiments.

Any system or subsystem that is built in the LIGO 40m interferometer should have an EPICS interface so that it is possible to remotely control the state of the interferometer from any of the control computers connected to a common network called as the Martian network.In an EPICS control system the Input Output Controller (IOC) is the name given to the piece of software which sits between the network stack and the hardware devices whose inputs and outputs are of interest. The larger control system sees it as a set of Channels which can be read or written to. Inside the IOC a Channel is revealed to be an a field of a Record in the Process Database, a Process Variable (PV). Records can be associated with each other by creating Links. This allows information to flow between Records when they are processed. For this the data from the I/O devices(like the FC or temperature Actuator)can be either passed onto one of the existing channels or a new channel access server can be created for the device so that the data can be accessed by multiple clients.

To build the EPICS database, I have written a dbd script which can ask the C code(A) to acquire data from the FC and create a SoftIOC for data access . Then a Python script provokes EPICS to write the data from the C code to a channel which is accessible to all the clients on the Martian network.Now this frequency data can be used as the input for the digital PID loop to be built. The methodology and the steps to create a channel access server and installation steps of EPICS on Raspberry Pi is updated on the LIGO 40m Wiki page.

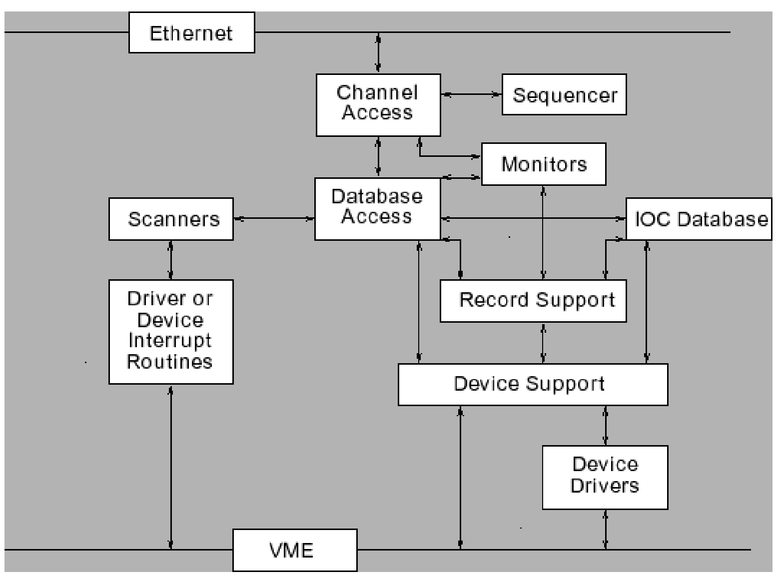

Figure 15: EPICS SoftIOC Components

3.8 PID Loop Design

The PID values were estimated using PID tuner in Simulink and the PID controller was realized using a PERL program and included in the control network. Subsequent tests were performed to check the credibility and working of the designed PID loop. This PID loop works in the following routine:

- First the beat note of the AUX laser and the PSL is checked.
- If this value exceeds 100 MHz, the PID loop is turned on.
- The PID controller operates till the set point value is met, with a given settling time and transient parameters.
- The error signal is fed back into the PZT to actuate on the AUX laser's frequency till the set point is achieved.
- Once the beat note is below 100 MHz, the PID loop turns off.
- This process is repeated iteratively with a given sampling time.

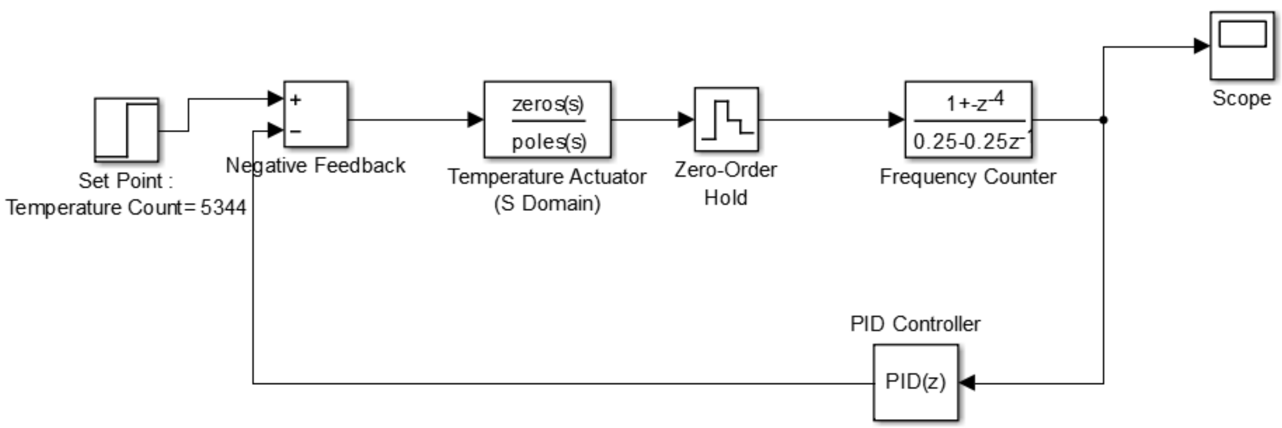

Figure 16: Implemented PID Loop

4 Concluding Remarks and Future Work

The frequency offset locking loop provides the automated control means of bringing the beat note of the AUX and PSL within the detectable range of the ALS. The flow chart of the control process is shown in fig 17. The actuation of the frequency of the AUX laser is carried out by actuating with the PZT of the laser. However, this is not an efficient way to actuate on the laser frequency due to effects like saturation of the PZT and uncontrollability at large frequency offsets. Hence the temperature actuation provides an efficient way of actuating on the laser frequency. The temperature actuation can be carried out once the infrared beat note is obtained from the PSL table.

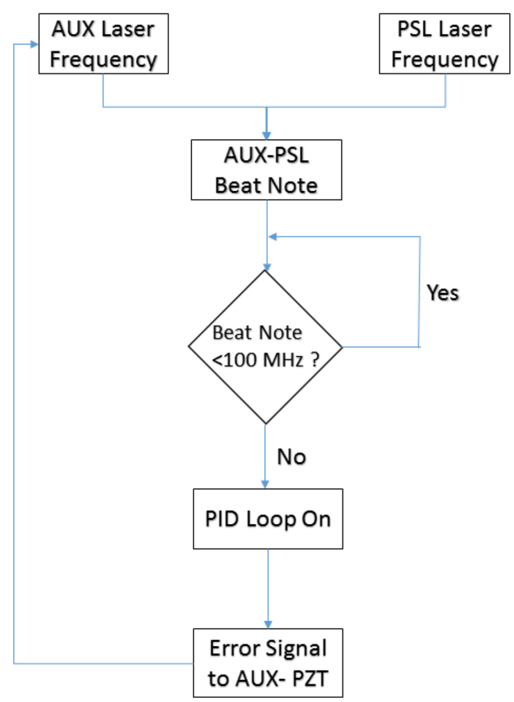

Figure 17: Flowchart of the Control Process

Some future work with regard to this locking loop are:

- Altering the PID parameters for the temperature actuator once the infrared beat note is obtained from the PSL table.
- Testing the loop with new PID parameters with the temperature actuator.
- Building a general device interface standalone EPICS server for future use in Advanced LIGO.

References

[1] Grishchuk L,2004 *AstrophysicsUpdate ed J W Mason*. (Berlin: Springer) p281 .

[2] K. Thyagarajan, A.K. Ghatak, *Lasers : Theory and Application*. Macmillan India (2008).

[3] William F. Egan, *Phase-Lock Basics*. Wiley (2007).

[4] Kiwamu Izumi, *Multi-Color Interferometry for Lock Acquisition of Laser Interferometric Gravitational-wave Detectors*. Thesis(2012).

[5] Eric D. Black, *An introduction to PoundDreverHall laser frequency stabilization*. Am. J. Phys. 69, 79 (2001).

[6] John Bechhoefer, *Feedback for Physicists: A Tutorial Essay on Control Reviews Of Modern Physics*. Volume 77, July 2005.

[7] http://nodus.ligo.caltech.edu:8080/40m/10298

[8] Charles Bishop,Catherine Kung *Effects of Averaging to Reject Unwanted Signals in Digital Sampling Oscilloscopes*

[9] *Sampling and Filtering of Continuous Measurements* $http://higheredbcs.wiley.com/legacy/college/seborg0471000779/dig_control/ch22.pdf$

[10] Jeffrey O.Hill *EPICS R3.14 Channel Access Reference Manual* http://www.aps.anl.gov/epics/base/R3-14/12-docs/CAref.html

[11] Python Recipe *Implementing C function callbacks using Python* http://code.activestate.com/recipes/65210-implementing-c-function-callbacks-using-python/

[12] Paul Caspi and Oded Maler *From Control Loops to Real-Time Programs* http://www-verimag.imag.fr/ maler/Papers/caspimaler.pdf

[13] *EPICS Manual:Hardware Links and Driver Support* https://pubweb.bnl.gov/ mdavidsaver/epics-doc/epics-ioc.pdf

[14] *PyEpics: Epics Channel Access for Python* http://cars9.uchicago.edu/software/python/pyepics3/pv.h

[15] MathWorks *PID Controller Design for Fast Reference Tracking* http://www.mathworks.com/help/control/getstart/pid-controller-design-for-fast-reference-tracking.html

[16] Fabrizio Padula, Antonio Visioli *Tuning rules for optimal PID Controllers*

A Interface Byte Code Description

Description of the Code:
HID USB Interfacing by sending byte Values.

1)Read The Freq or Range
Reading the Freq is done by reading the 1st and 2nd LCD of the Frequency counter.
1st line containing Range information, 2nd line is the Frequency result
The code should be send is 2
1st byte: 2
The returned 64 byte array is as follows:
1st byte: 2
2nd byte to Byte17 the ascii value of 16 characters of the 1st LCD line
Byte18 to Byte33 the ascii value of 16 characters of the 2nd LCD line

2) Set the Range
By default Freq Counter is in "AutoRange" mode.
To set the range manually send the code 4
1st byte: 4
2nd byte: the range value. can be any legal range value.
for auto range need to be 255.
the 64 byte array is:
1st byte: 4

3)Set the Sample Time
By default Freq Counter Sample Time is 1 sec.
You can set the sample time from 0.1 sec and up in step of 0.1 sec.To set the Sample Time send the code 3
1st byte: 3
2nd byte: the sample value in sec double 10.
For example: to set the sample time to 0.4 sec 2nd byte need to be: 4
The 64 byte array is:
1st byte: 3

These bytes can be changed by changing the values of buffer[0] and buffer[1] in function /*Send Report to the device */ in the main program.