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

Document Type	**LIGO-T010161-00-R**	**10/15/01**

LIGO Physics Environmental Monitoring at the 40-meter Prototype

Victor C. Tsai, Alan Weinstein, Dennis Ugolini

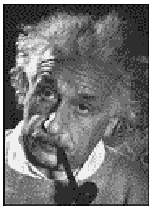
Distribution of this draft:

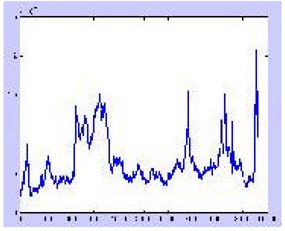
This is an internal working note
Of the LIGO Project

California Institute of Technology
LIGO Project – MS 51-33
Pasadena, CA 91125
Phone (626)395-2129
Fax (626)304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – MS 20B-145
Cambridge, MA 01239
Phone (617)253-4824
Fax (617)253-7014
E-mail: info@ligo.mit.edu

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





LIGO Physics Environmental Monitoring at the 40-meter Prototype

by

Victor C. Tsai

Mentor: Alan J. Weinstein

Co-mentor: Dennis Ugolini

Abstract

When Einstein formulated General Relativity, he made numerous predictions including the existence of gravitational waves. Until now, though, they have been impossible to detect. LIGO, the Laser Interferometer Gravitational-Wave Observatory, has been built to overcome this. Major difficulties arise as a result of the fact that gravitational waves are inherently weak; LIGO is expected to detect stretching on the order of 10^{-18} meters.

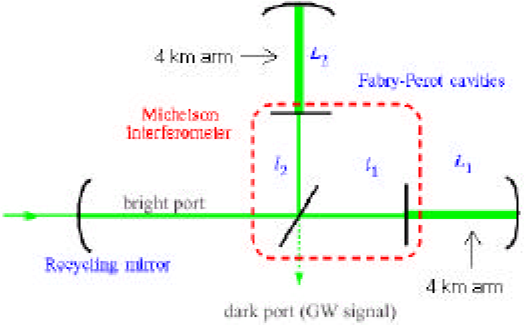
With the need for such precise measurements, a very large number of unwanted effects have to be minimized. Thus, physical environmental effects must be monitored with care and analyzed. Among the tools needed are a weather monitor, accelerometers and seismometers, and vacuum monitors. Each of these devices must be connected to the network and queried by the database, and the data coming from them must be analyzed. In order to accomplish all this, we must setup the hardware; write code to query each device and format the data; create GUIs to display the data; and design data analysis programs.

Such systems have been designed and built for the two LIGO observatory sites. In this project I implement a Physical Environmental Monitoring system for the Caltech 40-meter Interferometer Prototype Laboratory, and analyze the data obtained.

Introduction

When Einstein formulated General Relativity in the early 1900's, he made a number of predictions. Many of those have been confirmed, but one prediction has not; the existence of gravitational waves has never been directly detected. LIGO, the Laser Interferometer Gravitational-wave Observatory, is being built right now in hopes of being able to detect these inconspicuous waves. The detection of these gravitational waves would bring experimental physics into an entirely new world. For example, as of now, the only information we have is limited to events after the universe became transparent to photons. Before that, there is no experimental data. Since gravitational waves were not obstructed, one of the hopes of LIGO is that it will be able to probe into times far earlier than that. Thus, the success of LIGO is extremely important to further developments in cosmology and theoretical physics.

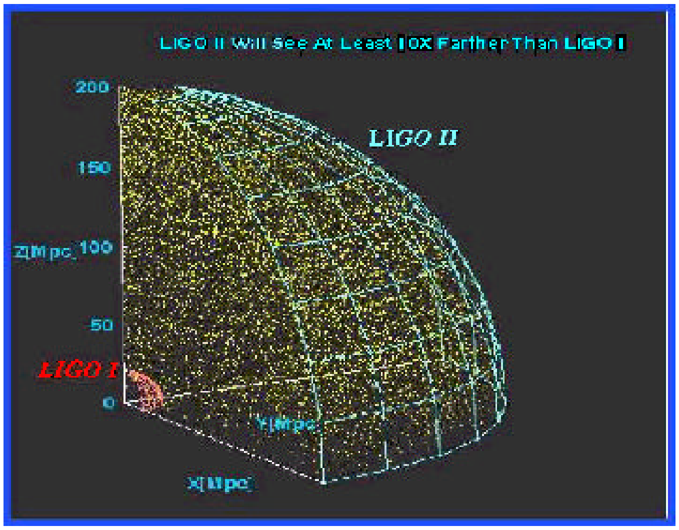
LIGO is essentially two long arms perpendicular to each other with a highly precise measuring system to detect the change in lengths of the two arms. More precisely, LIGO has a high-powered laser that emits infrared light that is split by a beam-splitter into two beams, each going down one arm of the detector. After bouncing back and forth through the arms many times, the beam makes its way back to the vertex where the two beams are allowed to interfere (see figure at right). Ordinarily, these beams should destructively interfere since the path lengths should be the same. However, if one arm should stretch the slightest bit, as would be expected from a gravitational wave, then the beams would no longer destructively interfere but show some response instead.



Layout of LIGO

The major difficulties in LIGO arise as a result of the fact that gravitational waves are inherently weak. LIGO is expected to detect stretching on the order of 10^{-18} meters. With the need for such precise measurements, a very large number of unwanted effects have to be minimized. LIGO has a near-vacuum chamber to reduce aberrations due to atoms interfering; its mirrors are hung from thin wires to isolate them from high frequency vibrations; the entire apparatus is in turn mounted on a system of seismic isolators (see figure at right) that reduce larger disturbances.

Seismic Isolators



Yet even with all these precautions, the environment still affects LIGO greatly. Some possible sources of environmental noise are earthquakes and lightning storms on the large scale to simply road traffic and weather. All of these effects must be monitored in order to be sure that false positives are not declared.

With the near completion of Initial LIGO (originally called LIGO I), plans have been set for Advanced LIGO (LIGO II), a version of LIGO that should see at least ten times farther than initial LIGO (see figure at left). Due to this, new models must be built and even more attention must be paid to the environmental effects that could disrupt the science done by LIGO. Before this summer, no such physical environment monitoring (PEM) system existed at the Caltech 40-meter prototype laboratory and so it has been my task to design and implement the PEM system as well as analyze the data from it.

My project consists of three primary parts:

1. Setting up Hardware and Control Systems
2. Operations and Maintenance of Devices and Code
3. Data Analysis

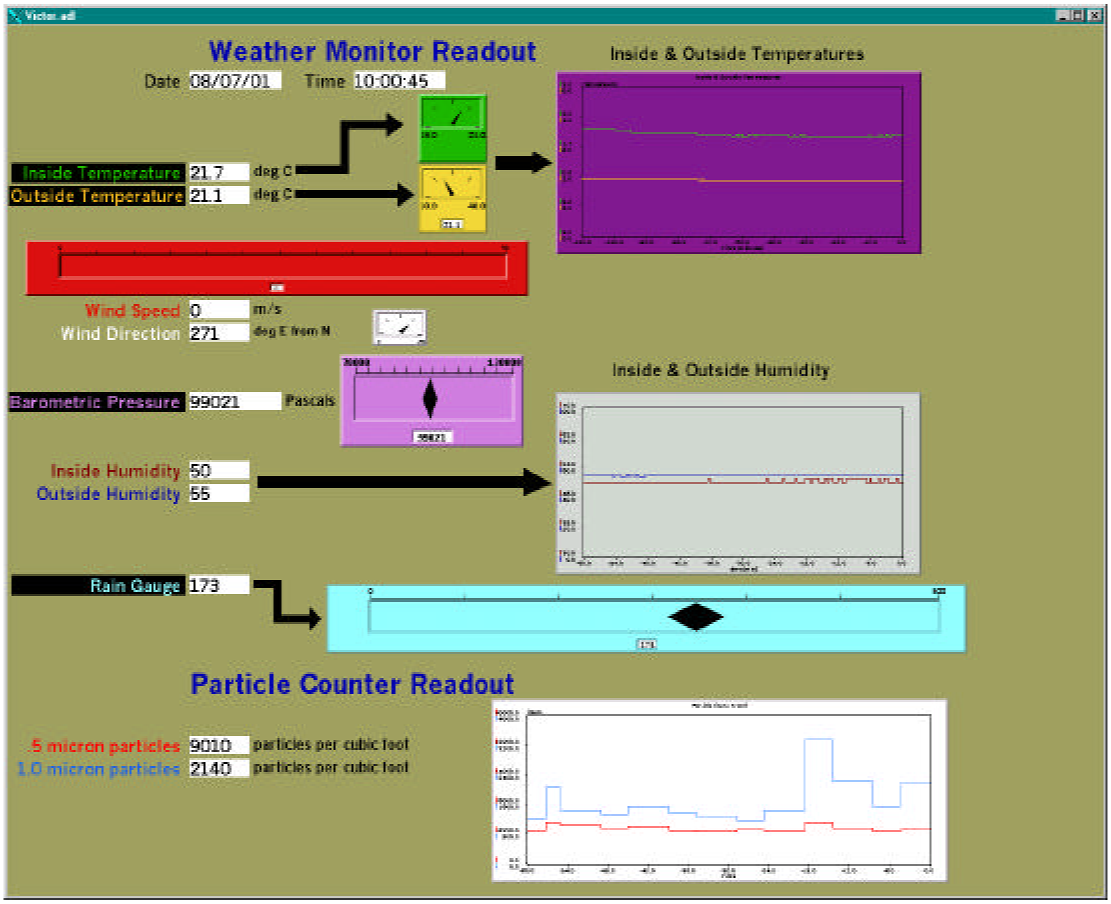
For the first part of my project, numerous devices had to be setup in the laboratory. These devices included a weather station, particle counter, accelerometer and STACIS devices (see figures below), most of which were already in place and ready to be used. In particular, only the weather station needed to be bought. In order to monitor these devices, though, all devices needed to be connected to the laboratory network of computers and controlled by commands sent out from the computers via code that we wrote. In addition to the controls code written, additional code was written for use with the database, saving the data in a prespecified form.

Weather Station

STACIS

Particle Counter

Accelerometer



Weather Monitor Readout

Date 08/07/01 Time 10:00:45

Inside Temperature 21.7 deg C
Outside Temperature 21.1 deg C

Wind Speed 0 m/s
Wind Direction 271 deg E from N

Barometric Pressure 99021 Pascals

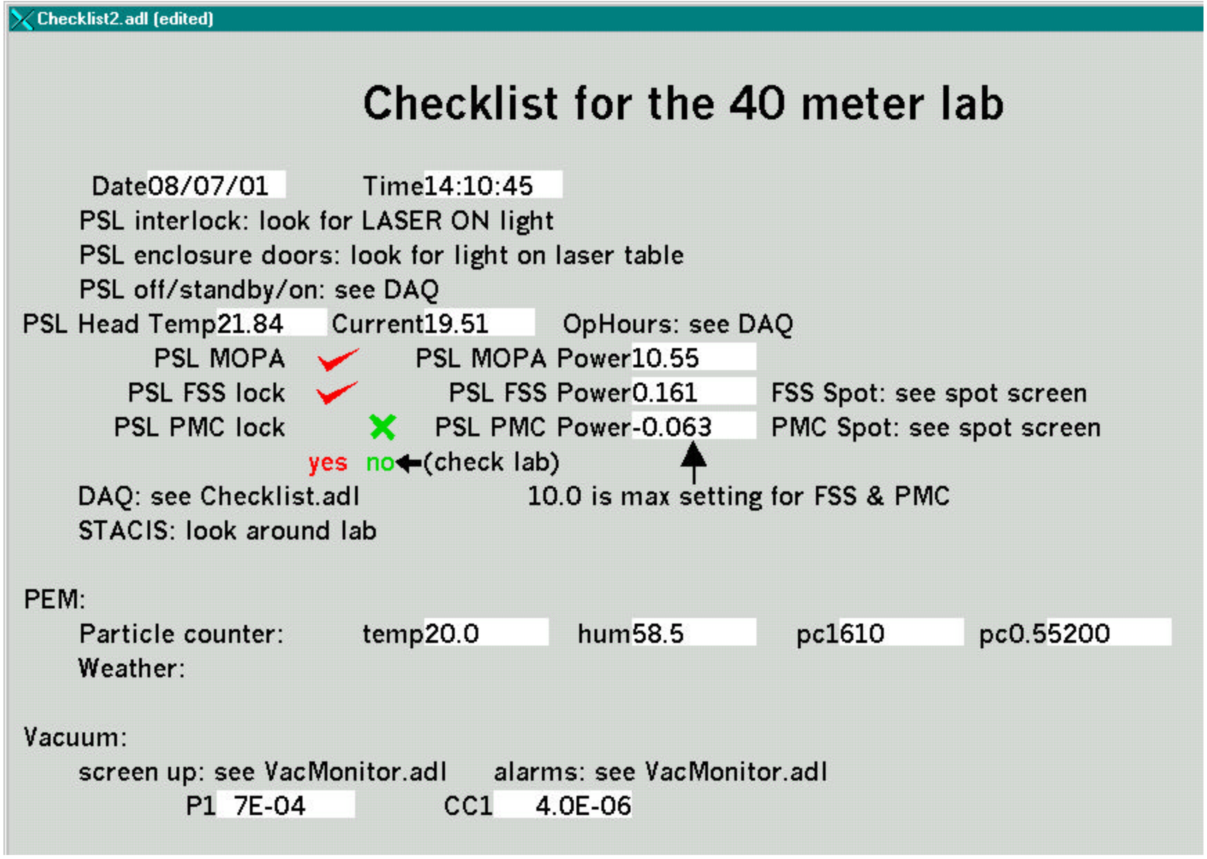
Inside Humidity 50
Outside Humidity 55

Rain Gauge 173

Particle Counter Readout

.5 micron particles 9010 particles per cubic foot
1.0 micron particles 2140 particles per cubic foot

The next section of my project involved creating graphical user interfaces (GUIs) for use in monitoring data in real-time on computer screens. I created one GUI to show weather monitor data as well as particle counter data (at right). This allows users to see the current data as well as the trends over the last hour or so. With this GUI the date, time, temperatures, wind speed, atmospheric pressure, humidity, rain, and particle count is displayed in an easily viewable fashion. In addition to this GUI, I created a checklist GUI to aid in the daily checklist that we performed in the laboratory (see below).



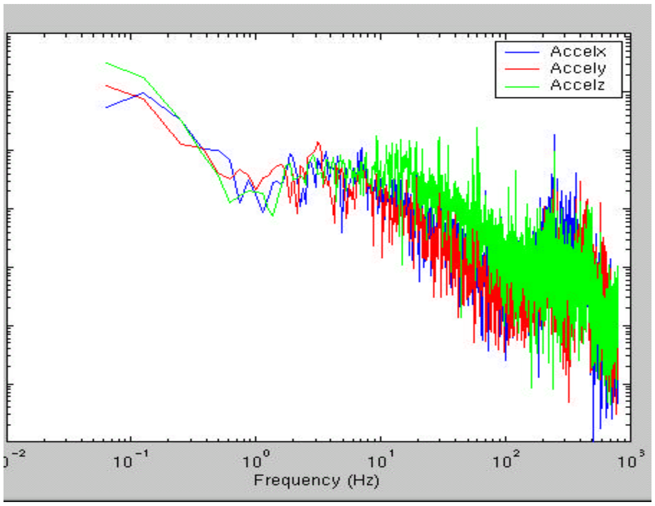
Checklist for the 40 meter lab

Date 08/07/01 Time 14:10:45
PSL interlock: look for LASER ON light
PSL enclosure doors: look for light on laser table
PSL off/standby/on: see DAQ
PSL Head Temp 21.84 Current 19.51 OpHours: see DAQ
PSL MOPA ✓ PSL MOPA Power 10.55
PSL FSS lock ✓ PSL FSS Power 0.161 FSS Spot: see spot screen
PSL PMC lock ✗ PSL PMC Power -0.063 PMC Spot: see spot screen
yes no ← (check lab)
DAQ: see Checklist.adl 10.0 is max setting for FSS & PMC
STACIS: look around lab

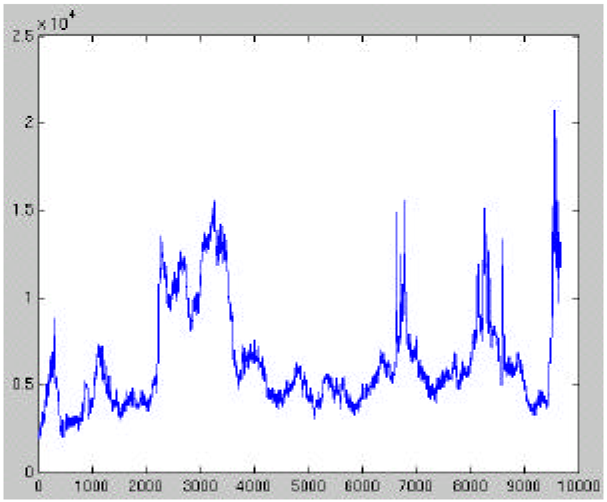
PEM:
Particle counter: temp 20.0 hum 58.5 pc1610 pc0.55200
Weather:

Vacuum:
screen up: see VacMonitor.adl alarms: see VacMonitor.adl
P1 7E-04 CC1 4.0E-06

My third major task consisted of analyzing the data that came from the various devices. In particular, I examined long-term trends, trends in the frequency domain, Gaussianity of data, and power spectra of data. Examples of the plots made can be seen below.

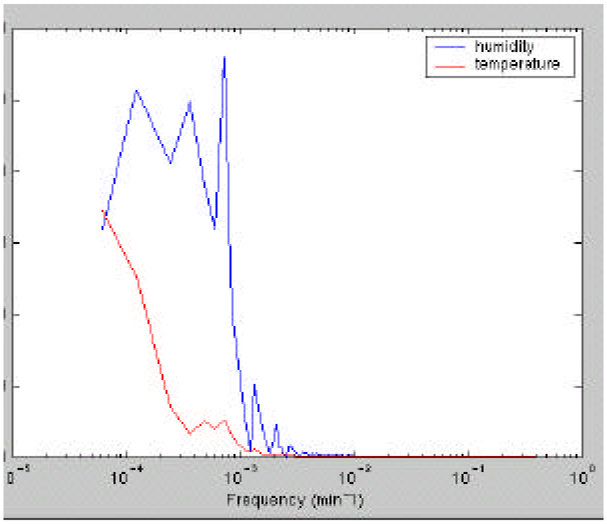


Accelerometer Displacement Power Spectrum

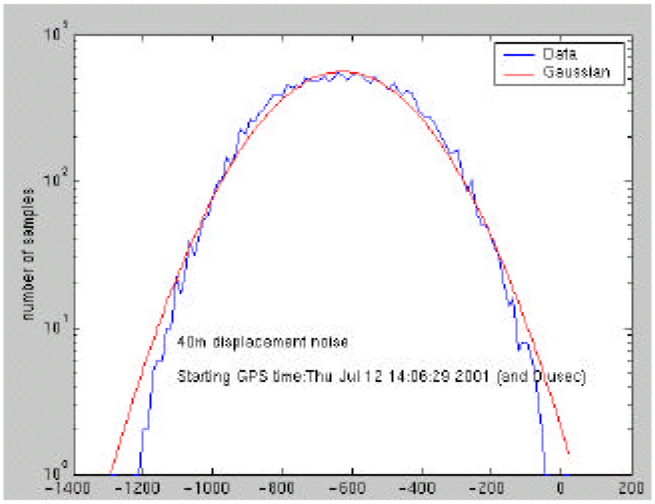


Minutes (Note: 1440 minutes = 1 day)

Long-term trends for .5 micron particles



Weather data as a function of frequency



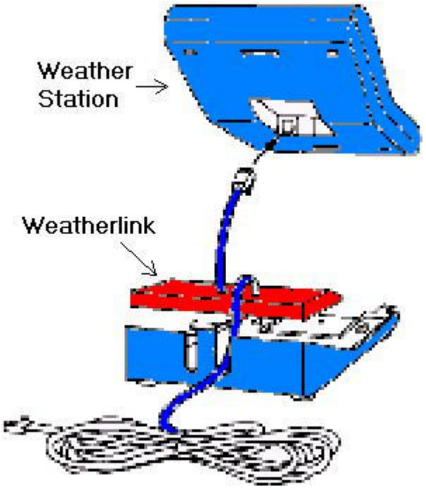
Histogram of Accelerometer data compared to a Gaussian Curve

With the conclusion of my project, I accomplished my major tasks as described above. Further conclusions could be reached if more extensive data analysis were done comparing PEM data with data from the laser and vacuum systems. However, this further analysis is beyond the goals of my project this summer. Since I had some data from the LIGO sites with respect to the PEM system, I was able to conclude that our devices were working as desired and achieving levels of noise comparable to the sites. This is a positive result, assuring us that the 40-meter prototype conforms to specifications in the PEM area.

Material and Methods

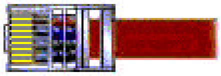
Hardware

The hardware for the PEM system was obtained through various sources. The weather station was bought from Davis Instruments[1], STACIS from TMC[2], and the particle counter from MetOne[3]. I took part in the purchasing of the weather monitor only. For this, we bought the main console with accessories which included the temperature/humidity sensor, anemometer and (optional) rain gauge. In addition, although not immediately obvious, the Weatherlink package had to be bought as a separate product in order for the weather station to be used properly in conjunction with any type of computer devices (in our case, the VME processors in the computer network of the 40-meter lab).



Once the hardware was assembled and set up, each device was to be connected to the VME crates so that data could be stored and later analyzed. All the devices had different connections. The particle counter had a standard DB-9 connector (see table below). This made the communications with the particle counter easy to deal with. The accelerometer was connected without any of my input. The weather station had problems though. Initially, we assumed that the weather station could be directly commanded from its 4-pin I/O port. We tried various combinations of possible pin arrangements for transmit, receive and ground, but none of them worked effectively. After reconsulting PEM experts at the Hanford site, we decided to buy the Weatherlink package that allowed communication by known commands[4]. After the working Weatherlink was connected, a simple switch of pins 2 and 3 gave the correct response from the Weather Station to our commands. After unsuccessfully trying to send and receive digital signals from the STACIS devices, we contacted the manufacturer. It turns out that no one knows how to communicate with the devices other than through the Windows-based program that has no documentation. Thus, STACIS never got monitored.

DB-9 pin assignments



DB-9 Pin	Corresponding DB-25 Pin	Signal	Function
2	3	RD	Received data
3	2	TD	Transmitted data
8	5	CTS	Clear to send
7	4	RTS	Request to send
6	6	DSR	Data set ready
5	7	SG	Signal ground
4	20	DTR	Data terminal ready
1	8	DCD	Data carrier detect

The other hardware implementation involved consisted of placing the Weather Station on the roof of the Physical Plant and sending cables down into the building. After a few mishaps, this was successfully accomplished.

Software Development

In order to do all my computer work, I worked in a UNIX environment on a PC. To do this, one opens an SSH (or telnet) session with the server to work on. In my case, these were limited to sargas, sirius, luna, rana, fb40m, cdssol6, and canopus. UNIX is a command-based operating system[5]. Among the applications it runs, two of the more important ones are vi and emacs, both of which are editors. More information about them can be gotten by typing ‘man vi’ and ‘man emacs’ respectively.

Certain specific tasks I had required convoluted processes that I will describe here.

Creating an executable for the VME processors to handle:

1. Edit the state code in [directory]
2. login luna as [user]
3. source ~barker/.cshrc [Note: barker’s cshrc file works best]
4. setup epics/release/r3.12.2 [Note: sets up the necessary epics files]
5. cd /opt/CDS/d/epics/apple/Caltech/40mWFS/scipe16/dev/src

6. cp ~[user]/[file.st] .
7. cd ../target/mv162/obj
8. make [Note: creates executable files]
9. cp [file.o] ~cit40m/[user]/.
10. cd ../../../src [Note: if doing multiple edits, repeat steps 6-10]
11. log on rana as controls
12. cd ~cit40m/[user]
13. scp [file.o] cdssol6:/export/home/40mPEM/.

Note: before step 8, the file ‘makefile’ must be in directory of step 5 with appropriate information as described in the standard ‘makefile’ file (in comments).

Steps to allow an executable to be run

1. login rana as controls
2. ssh cdssol6
3. cd /export/home/40mPEM
4. change files seq.load and seq.exe appropriately, using the editor vi
   a. copy the format already in seq.load: ‘ld</export/home/40mPEM/[file.o]’
   b. copy the format already in seq.exe: ‘seq &[file]’

Accessing frames data

1. login rana as controls
2. cd [frames dir]
3. ftp fb40m as controls
4. cd /usr1
5. cd frames or minute-trend-frames or trend-frames
6. binary
7. prompt
8. mget [filename] or multiple filenames (e.g. mget C-643235*)
9. exit completely
10. login rana as cit40m
11. scp /export/home/controls/[frames dir]/[filename] /export/home/cit40m/[frames dir]/

State code is the code that runs on the VME processors and is compiled by ‘make’. It is mostly C-code (which follows ‘%%’) with some vxWorks commands added to it. The following is the general format it follows:

```
program [program_name]

%%#include <[header.h]>

%{[C-code for structure declaration and function prototypes]}%

/* comments */
int [variable_name1];
float [variable_name2];
%% unsigned char [variable_name3]; /* for C variables */
```

```
ss [program_name]
{
    state [init_state] {
%%                                [C-code];
                                  [vxWorks code];
                                  state [cycle_state]
                       }
    state [cycle_state] {
                                  [code];
                        }
}

%{ [C-functions] }%
```

The programs that I wrote in state code are: Weather.st, Accel.st and Stacis.st. In the end, only Weather.st was used as a driver. Note that all useful files are saved in /home/cit40m/vtsai/

In conjunction with Weather.st, I wrote Weather.db, a file of database code used for specifying the weather station data and how it is formatted.

In addition to state code, I also wrote numerous matlab programs:

- FReader.m reads in data from frames and outputs the average displacement spectra of the input
- transfer.m reads in data from frames and outputs plots of average displacement spectra, power spectra, Hanford seismic spectrum, Gaussianity, isotropism, correlation between two inputs, best correlation of one input with all others, and the transfer function. Note that much of this code was taken from /home/ajw/transfer/transfer.m
- LongFreq.m reads in data from minute trend frames (i.e. long term trends) and calculates the same quantities as transfer.m except with a long time span
- trends.m reads in data from minute trend frames and outputs a plot of the long term trends

All of the programs that created outputs as a function of frequency made use of the Fast Fourier Transform (FFT)[6], an algorithm that takes a time series and converts it into a function of frequency. The general formulas that it relies on, the Fourier Integrals, are as follows:

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-2\pi f t i} \, dt \qquad h(t) = \int_{-\infty}^{\infty} H(f) e^{2\pi f t i} \, dt$$

Since I had previously not known about Fourier Analysis, I learned enough about it so that I could use it effectively.

I also wrote a C program, eq3.c, that reads in text from /home/vtsai/eq.out (which is copied from the source code of pasadena.usgs[7]). This program extracts the pertinent information from the file, saves it to in a structure, and displays the information. This program ended up not being useful because the data could not be taken from the internet into the Martian network in an automated manner.

In the first few weeks of research, I worked quite a bit to learn how to use ROOT, a C-interpreter developed by CERN for use in high data systems such as LIGO. However, this learning was wasted as I ended up using matlab for data analysis.

Operations and Maintenance

In order to view PEM data in a concise manner, I created an MEDM screen. MEDM stands for Motif-based Display Editor/Manager. It is a program that allows an easy, graphical way of displaying current data coming out of the DAQS (Data Acquisition System). As explained in the introduction, one MEDM GUI displayed PEM data. (See readout on page 3.) The creation of this GUI was easily accomplished, as the editing is purely graphically-based. One takes the following steps to open MEDM:

1. login rana as controls
2. ssh fb40m
3. medm
4. in MEDM, open the files /cvs/cds/caltech/medm/*.adl

The PEM dataviewer is entitled PEM.adl

Starting about halfway through the summer, we began performing a daily checklist of various quantities in the lab. To facilitate this process, I created another MEDM GUI as described on page 4. This checklist is entitled Checklist2.adl.

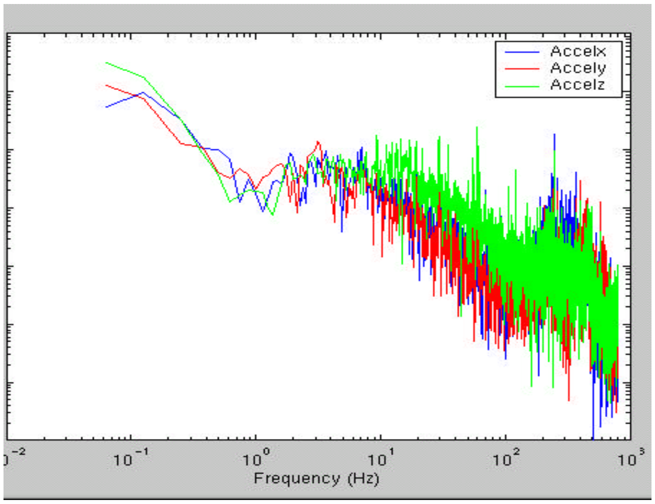
Results and Conclusions

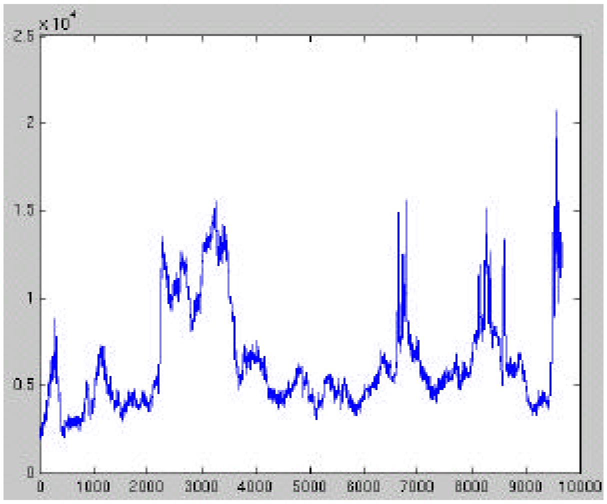
My project's main goal was to setup a physical environment monitoring system at the 40-meter prototype at Caltech similar to the ones already in place at both the Hanford and Livingston sites. Thus, due to the nature of my project, good data is data that agrees with existing data. Since my project was, on the whole, successful, the data that I have gotten generally conforms very nicely to similar data from the sites[8].

The sample data that I have represented on pages 4 and 5 give the basic results for a specific set of data. Basically all other sets of data gave similar results and so those 4 graphs shall act as archetypes for the rest of my data.

Accelerometer Displacement Noise Spectrum

The accelerometer used in the laboratory is sensitive in the range of approximately 1-1000 Hz. As can be seen in the graph, there is more noise at low frequencies, which agrees with the known spectrum of seismic noise. In addition, the plot agrees fairly well with the data from the other LIGO sites[8]. As a note, the plot is a log-log plot, as is standard with this type of plot.



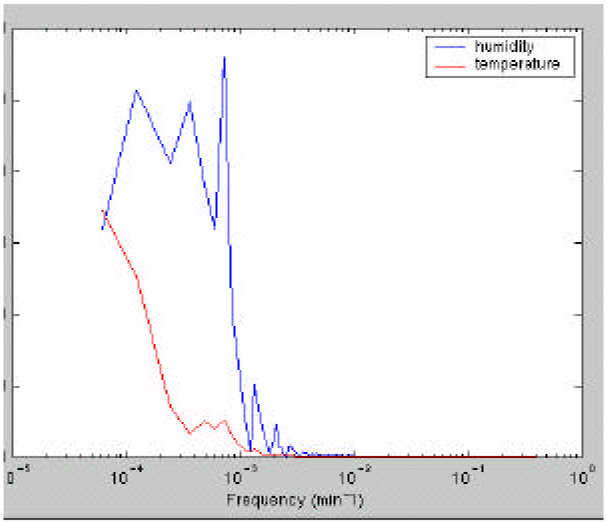


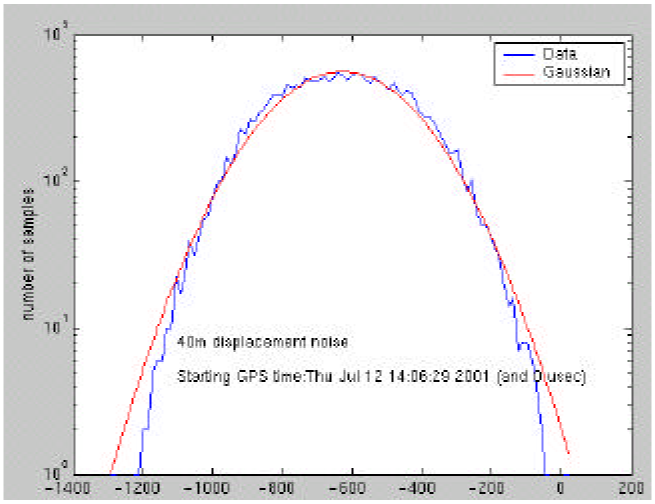
Long-term trends for .5 micron particles

This long term plot (over about six and a half days) shows how the level of .5 micron particles fluctuates over the days. Since the lab is a class 10000 clean room, the amount of dust in the air must be carefully monitored to assure the quality of air. As is easily seen in the trends graph (but is not immediately apparent in the FFT), each day (1440 minutes) there are peaks in the data corresponding to when people are working in the lab and bringing in dust.

Weather Data as a function of frequency

In this frequency plot of inside humidity and temperature, there is one interesting feature. Again, this is the peak at 1 day^{-1} or $7*10^{-4}$ min^{-1}. The humidity peak is much stronger than that for temperature. Both, however, point out the fact that despite the extreme measures LIGO takes to keep the lab's temperature and humidity constant, daily work still affects both of them to a measurable extent. This plot is a semi-log x plot (only the x axis is log plotted).





Histogram of Accelerometer Data compared to a Gaussian Curve

This plot shows that the data conforms to a standard Gaussian extremely well. What this means is that the displacement (arbitrarily at a mean of about –650) has noise that is roughly Gaussian. Because of the Central Limit Theorem, we would expect our data to be roughly Gaussian, and so this agreement is reassuring.

Thus, I have characterized the physical environment of the 40-meter LIGO prototype at Caltech. This data will be immensely valuable when the 40-meter interferometer has data to be analyzed. My work will help determine whether the data will be interpreted as environmental effects or gravitational waves.

For further study, the PEM data should be compared with all of the other data including (but not limited to) the PSL data. However, this requires intimate knowledge of the PSL which is beyond the scope of this project.

Finally, I would like to extend many thanks to Alan J. Weinstein, Dennis Ugolini, Ben Abbott, Steve Vass, Ken Libbrecht, NSF and SURF.

References

[1] Davis. http://www.davisnet.com Accessed Jan-Aug 2001.

[2] Technical Manufacturing Corporation. http://www.techmfg.com/index.htm Accessed Aug 2001.

[3] Met One. http://www.particle.com/metair.htm Accessed Aug 2001.

[4] “Davis Support: Weather Support.” http://www.davisnet.com/support/weather/ Accessed Jan-Aug 2001.

[5] “Commonly Used UNIX Commands” New Mexico Tech Computer Center http://www.nmt.edu/tcc/help/unix/unix_cmd.html Accessed June-Aug 2001.

[6] Brigham, E. Oran. The Fast Fourier Transform. Englewood Cliffs, N.J.: Prentice-Hall, 1974.

[7] Pasadena USGS. http://pasadena.wr.usgs.gov/recenteqs/Quakes/quakes.big.htm Accessed June-Aug 2001.

[8] Marin, A., D. Shoemaker, and M. Smith. “Physics Environment Monitoring Final Design Document.” Technical report, http://www.ligo.caltech.edu/docs/T/T970112-00.pdf, published 30 Sept 1997, accessed 11 Feb 2001.