

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

Technical Note	LIGO-T2000291-vX	2020/07/31
<b>LIGO SURF 1st Interim Report</b>		
Ella Buehner Gattis		

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

## 1 Introduction

The Laser Interferometry Gravitational Wave Observatory (LIGO) aims to detect gravitational waves predicted by Einstein's theory of relativity in 1916, which predicted that gravitational waves would cause ripples in space-time, which can be measured as contractions and expansions in space [1]. LIGO consists of two L-shaped detectors, with arms 4km in length. The basic concept they employ in order to detect changes in length caused by ripples from gravitational waves is the same as that for a Michelson interferometer. Each arm of the detector is an exact integer number of wavelengths long, so when monochromatic light travels down each detector arm and is reflected by the mirror at the end, the reflected beam is exactly in antiphase with the initial beam and no light returns to the detector. As such, any microscopic change in length of the LIGO detector arm will cause the transmitted and reflected beam to no longer be in antiphase, and the resulting change in length can then be measured by the detector.

However, seismic movement also causes movement in the detector arms, introducing noise into the detector readings [2]. We aim to create a heat map of seismic activity in the area surrounding LIGO to better understand and account for seismic activity in detector readings.

Seismometers measure ground movement and are usually placed 2 meters below the surface, where temperature is steady with very few or negligible fluctuations. However, the seismometers around LIGO are placed above ground in order to measure seismic activity at the surface. Because of this, they are subject to above-ground temperature fluctuations, which affect the seismometer readings and introduce noise into the seismic data. In order to minimise the effect of temperature fluctuations on seismometer readings, the seismometer is placed in a steel enclosure surrounded by a resistive heater and insulating foam.

## 2 Objectives

The aim of this project is to design a feedback system that will keep the temperature within the seismometer enclosure constant, to ensure there is as little noise in the seismic data due to temperature fluctuations as possible. The first step is to design a model of the system, and test the system response to different changes in temperature.

Once an accurate model of the system has been created, it will be possible to test the response of the system with linear control, such as PID (proportional-integral-derivative) control. In linear control, the output signal is proportional to the input signal supplied, which limits the flexibility of control that can be implemented to only be linear.

The main focus of my project is to test non-linear feedback systems - in particular, a machine learning mechanism using a neural network - for control. In non-linear control the output signal is not necessarily proportional to the input signal. The aim of this is to further reduce noise from temperature fluctuation by minimising overshoot and improving the system response to disturbances, which is sometimes a problem with linear control methods such as PID [3], and to reduce the time constant.

The efficacy of a linear vs nonlinear temperature control system will then be measured and

compared. Our final aim is to design a feedback system that can achieve and maintain temperature stability to within the sensitivity of the temperature sensor being used.

With the seismic data obtained, we aim to plot a "heat map" to visualise the intensity of seismic activity in the areas surrounding the LIGO detectors.

### 3 Approach

We will first model the seismometer in stainless steel system in Python, using the heat equation to model temperature changes as a function of applied power. This modelling will be done in the Laplace domain, and also in the time domain to model non-linearities.

Once the first step of modelling the system has been achieved, it will be possible to also model the effect of PID for temperature control, and to evaluate different methods for tuning the PID parameters.

The next step is to set up an environment in which to train a neural network for temperature control.

#### 3.1 Modelling the system

The physical system consists of a seismometer in a stainless steel enclosure wrapped in resistive heating mesh with a resistance of 30 Ohms, covered with an insulating foam. The temperature sensor is an AD590 temperature sensor.

The pulse-width modulation (PWM) circuit used to control the heating system consists of a MOSFET which is used to control current flow to the resistive heater

Using the heat equation, the system can then be modelled by

$$P_{in} = M_{ss}C_{ss} \frac{dT_{ss}}{dt} + P_{loss} \quad (1)$$

where  $P_{in}$  is the input power supplied as heat,  $M_{ss}$  is the mass of stainless steel,  $C_{ss}$  its heat capacity and  $T_{ss}$  its temperature at a given time  $t$ .  $P_{loss}$  is the power lost through conduction and radiation.

The power lost by conduction will be

$$P_{\text{conductive loss}} = \frac{k_{foam}A_{foam}}{t_{foam}}(T_{ss} - T_{amb}) \quad (2)$$

where  $k_{foam}$  is the thermal conductivity of the insulating foam,  $A_{foam}$  is the outward facing surface area of foam, and  $t_{foam}$  the thickness. The difference in temperature between the stainless steel enclosure and the ambient temperature of its surroundings is given by  $(T_{ss} - T_{amb})$ . Then, the power lost by radiation is given according to the Stefan-Boltzmann law by

$$P_{\text{radiative loss}} = \sigma A_{foam}(T_{ss}^4 - T_{amb}^4) \quad (3)$$

We will model this system in Python. To model the non-linearities such as radiative losses and other noise, it will be useful to model the system in the time domain rather than in the Laplace domain. Neural network training will also be done in the time domain.

### 3.2 Linear control- PID

In PID control, there are 3 parameters to be tuned, according to the PID equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (4)$$

There are a number of different methods with which to tune the PID parameters, some of which have been investigated in Rushabha's project.

### 3.3 Nonlinear control using a Neural Network

The role of the neural network will be to function as a nonlinear controller to regulate the duty cycle of the PWM circuit, thus controlling the power to the heating circuit to maintain temperature stability.

#### 3.3.1 Software

We will develop the neural network for temperature control in Python using TensorFlow, an existing open-source machine learning library distributed by Google. The OpenAI gym environments will also form the basis for the learning environment of the neural network.

#### 3.3.2 Supervised Learning

Supervised learning uses a given set of labelled data to learn a function that maps the input onto the output without changing the input space. The neural network will then learn to adjust parameters for a given set of input variables to get the desired output. This means we would need a labelled set of data for temperature of the steel enclosure and optimum power supplied to the heating circuit.

#### 3.3.3 Reinforcement Learning

Reinforcement Learning, in contrast to Supervised Learning, doesn't need a labelled data set, and uses trial and error with game-like scenarios operating in discrete time steps to find optimum control. RL interacts directly with its environment to create its own data, meaning it can use environments such as the openAI gym for training.

A Reinforcement Learning model takes much longer to train than Supervised Learning, so it will be useful to utilise SSH to use a LIGO computer to run the programs and train the network more quickly.

Because of the nature of the problem, the neural network has to interact with the input space (the temperature of the enclosure) to adjust its parameters to optimise the output

(power supplied to the heating circuit). This means that this network will use Reinforcement Learning for training.

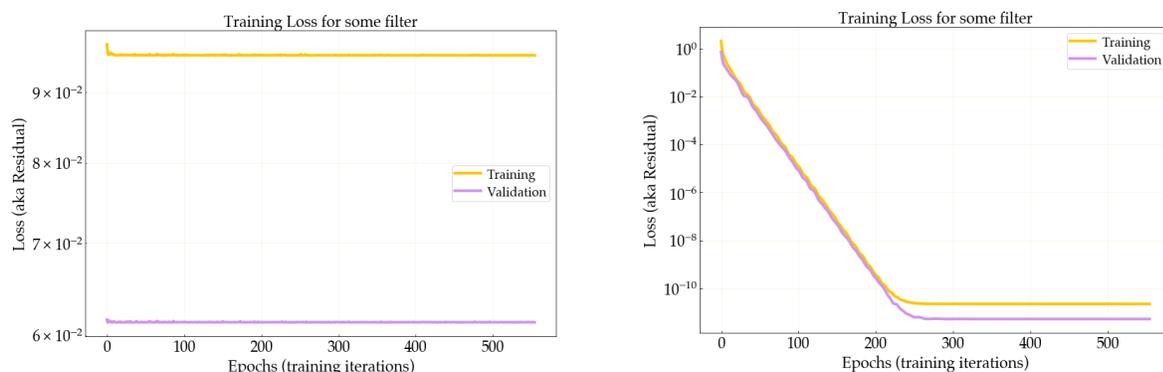
## 4 Existing Progress and Plan

The first step is to design a model of the system in Jupyter notebooks - this is the foundation for all subsequent things, and should take a few weeks. This has been done by Rushabha in the weeks before I began my project. His project also developed PID tuning methods and modelled the system response to changes in temperature.

The first task in my project has been to set up a Supervised Learning convolutional neural network to learn the behaviour of a simple electronic system - fitting the frequency response of an LC filter. This was done using the Keras Sequential model and a Conv1D layer. This formed a useful basis for better understanding signal processing and the principles of convolutional neural networks.

The learning set used was a large sample of noisy signal - random noise generated by the `numpy.randn` function - together with a filtered signal, which had been denoised using a butterworth lowpass filter. Later this was changed to a filter using second order sections. The job of the network then was to take in a sample of noisy signal and produce a denoised signal, thus imitating the output of the filter.

The main element needed for this to work was to specify the input shape of the data correctly, which took some time to figure out. The figure below illustrates the training loss during learning for the neural network in the two scenarios for differently specifying the input shape. Here, it is clear to see that in the first case (a), the neural network was not learning at all. This is because the way that the input shape had been specified meant that the network only read in one data point at a time, making it impossible to learn a pattern. The Keras Conv1D input layer requires the input shape to be specified in the format (batch size, number of steps, channels), as was done in the second case (b). Once this was correctly defined, it was possible for the neural network to learn to imitate the output of the filter.



(a) Input shape specified in format (shape of data, 1, 1) (b) Input shape specified in format (1, shape of data, 1)

Figure 1: Training loss for neural network over 500 epochs

The next step is to start setting up the neural network for temperature control which will ensure that the true temperature fluctuation of the seismometer enclosure is close to zero.

To do this, I will be adapting one of the classic control examples from the OpenAI gym, such as the problem of stabilising a pendulum, and substituting in the differential equations for our seismometer enclosure system. First, I will set up some more simple scenarios with which to train the neural network. Once these are done, it will be possible to build the complexity of the learning environment until it resembles the real system.

It is important for the neural network to recognise the pattern in ambient temperature fluctuation (which generally occurs over a 24 hour cycle). The ambient temperature data will be collected using the nds2 channel to receive temperature data in one-week chunks. This will also form part of the learning environment. The neural network should then be able to recognise the patterns of temperature fluctuation and account for them ahead of time, thus minimising overshoot and increasing the accuracy of the temperature control.

Once simulations of temperature control using both PID and neural network control are running it will be possible to compare the efficacy of the two. The overall target control is to hold the temperature of the seismometer constant to within the accuracy of the temperature sensor of the enclosure. At this point, the effect of temperature fluctuations on the seismometer readings should be negligible, producing the maximum accuracy for results with this setup.

## References

- [1] Albert Einstein. *The Foundation of the General Theory of Relativity*, Annalen der Physik, volume 49, 10.1002/andp.200590044, 1916
- [2] Macleod, D M, et al. *Reducing the effect of seismic noise in LIGO searches by targeted veto generation*, Classical and Quantum Gravity, vol. 29, no. 5, 10.1088/0264-9381/29/5/055006, 2012,
- [3] Yuvraj V. Parkale. *Comparison of ANN Controller and PID Controller for Industrial Water Bath Temperature Control System using MATLAB Environment*, International Journal of Computer Applications, volume 53, 10.5120/8390-1967, 2012