Contact: Jose Amador – joseamador2022@u.northwestern.edu

Title: High Cyclic Fatigue Machine Control Interface

Background: A custom load frame meeting many of the required specifications for in-situ material analysis in beamline 2BM at the Advanced Photon Source, Argonne National Laboratory was constructed. The custom load frame is a piezo-actuated system can deliver about 500 Hz cyclic frequency at 0.1% strain on miniaturized ASTM E606 test coupons. Force data is collected through a load cell mounted at the base of the machine, directly in line with the piezo actuator. It was developed to study high cyclic fatigue, a task that was difficult to do with the current mechanical screw machines that were being used. However, the current control interface and scheme are quite rudimentary.

This summer I was tasked with getting the fatigue machine to an operable state and ready to conduct real tests. I began the summer with learning about how fatigue works and the underlying mechanisms that dictate various material properties. This involved a lot of technical reading that was difficult to understand however this research was useful to understanding the goal of this project.

After I had a basic understanding of what was going to be tested, I had to understand how each component of the fatigue machine functioned and what its role was in conducting high cyclic fatigue testing. Throughout this process I learned a lot about reading through component documentation and design documents to find necessary and relevant information. Once I had a basic understanding of fatigue and the limits of the components I would be working with, I began to learn about LabVIEW programming environment and its applications in data acquisition devices.

I began by analyzing the previous rudimentary code and learning about ways to improve both the control scheme and the overall speed and style of the code. The original designer of the machine (Fengchun Li) coded in Chinese so understanding the function of each component of the code was quite difficult. I heavily relied on online documentation provided directly by National Instruments (instructional videos, VI documentation, etc.) and forums on the National Instruments website. After feeling prepared to tackle the problem, I quickly ran into my first obstacle. While Northwestern provided me with the LabVIEW software, Northwestern provides only the newest version of LabVIEW (2019) which is incompatible with the much older hardware used for this machine.

The important interface is between the NI-DAQmx software and the data acquisition card chassis. This machine uses the cDAQ-9172 chassis which was last supported in NI-DAQmx version 17.5 (http://www.ni.com/product-documentation/54233/en/) which rendered me unable to test any code/run the machine from my computer. However, I was able to install older versions of NI-DAQmx, but not all older versions of this software are compatible with the LabVIEW 2019 work environment and I was unable to write/view/edit any code involving data acquisition. I spent many weeks trying to resolve this compatibility issue, even going as far as installing LabVIEW 2011 off an old CD I borrowed from Nicolas Prieto who works in a different lab. The CD install still did not work which I believe was because I had LabVIEW 2019 still installed. I could have tried removing every trace of LabVIEW 2019 from my computer and trying again however the issue was resolved before I could try that solution. Nicolas had discovered an old computer in his lab that was not being used at the moment which had LabVIEW 2011 and a compatible NI-DAQmx version installed. After hooking it up to the machine, the code ran without an issue and I was finally able to get started on the task I was assigned to do from the beginning.

I began the code part of this project by first removing and learning how to replace any redundant/slow code, such as the DAQ Assistant, to improve the overall efficiency. After decreasing the redundancies I implemented a way to record the data that was being generated by the load cell since the current code had no method of doing so. I researched many virtual instruments (VI) that LabVIEW has to offer and chose the “write to measurement file” VI because of its simple yet versatile file writing abilities.

I split the code into three different loops: voltage generation, load cell data acquisition, and file writing. The voltage generation loop is setup in figure 1 and the loop itself is shown in figure 2. In figure 1 I started by creating the task and setting properties such as sample rate, and I finished by starting the task. However, this method would give an error (error -200462: Generation cannot be started because the output buffer is empty) if the code is run for the first time because when the code reaches the “start task” VI, it immediately tries to send an output signal which has yet to be generated since the signal generation is done inside the loop. This is the reason I included the 0 amplitude signal generation outside the loop, so the start task VI has something to send. There could be some better way to do this or avoid this problem however I was unable to solve it.

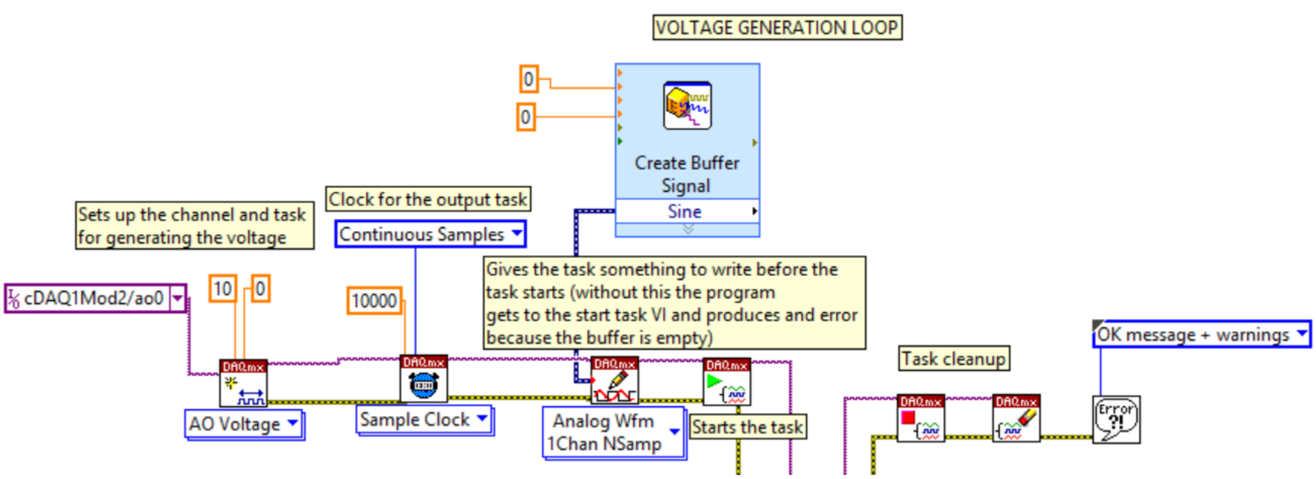

Figure 1: Setup for Output Voltage to Piezo

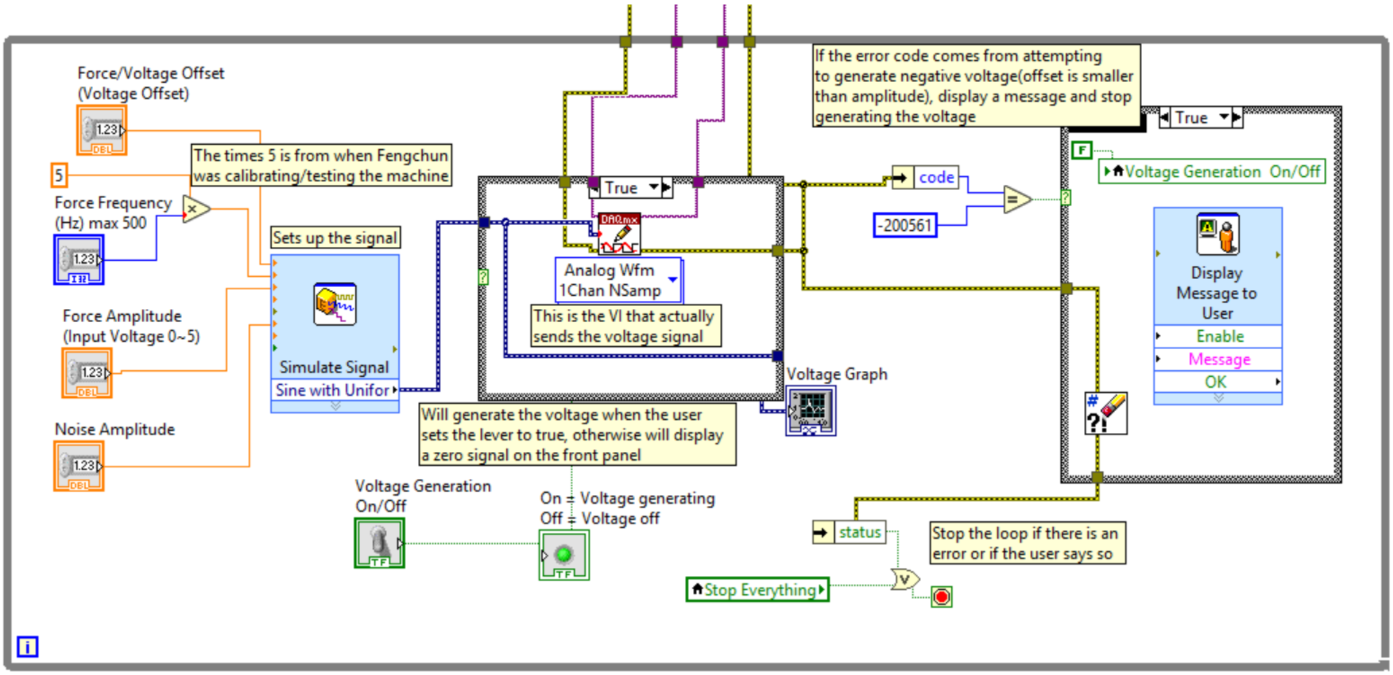

Figure 2: Voltage Generation Loop

In figure 2 the loop that generates the voltage signal is shown. It starts by taking in the input parameters from the front panel to generate a sine wave which is then send to the piezo and shown on a graph. There is a calibration factor that Fengchun determined for the frequency in order to output the proper voltage. This loop only sends a voltage signal to the piezo when a switch is flipped to the "on" position in the control panel. Furthermore, the piezo cannot handle negative voltages so the voltage offset always has to be greater than the voltage amplitude. If this condition is violated, the voltage signal generation is turned off to allow the user to correct their error.

The load cell data acquisition is shown in figures 3 and 4. In figure 3 the setup for the loop is shown where I started by creating the task, setting the timing parameters, and then starting the task.

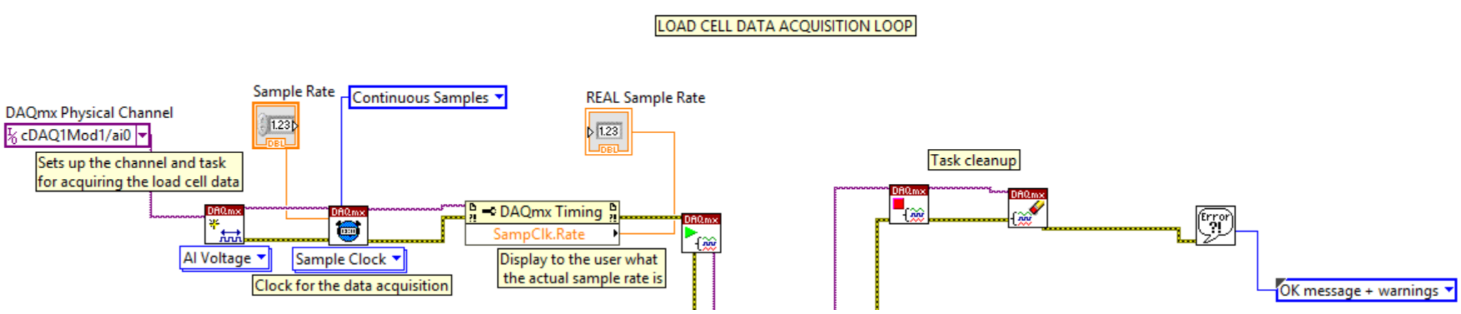

Figure 3: Setup for Load Cell Data Acquisition

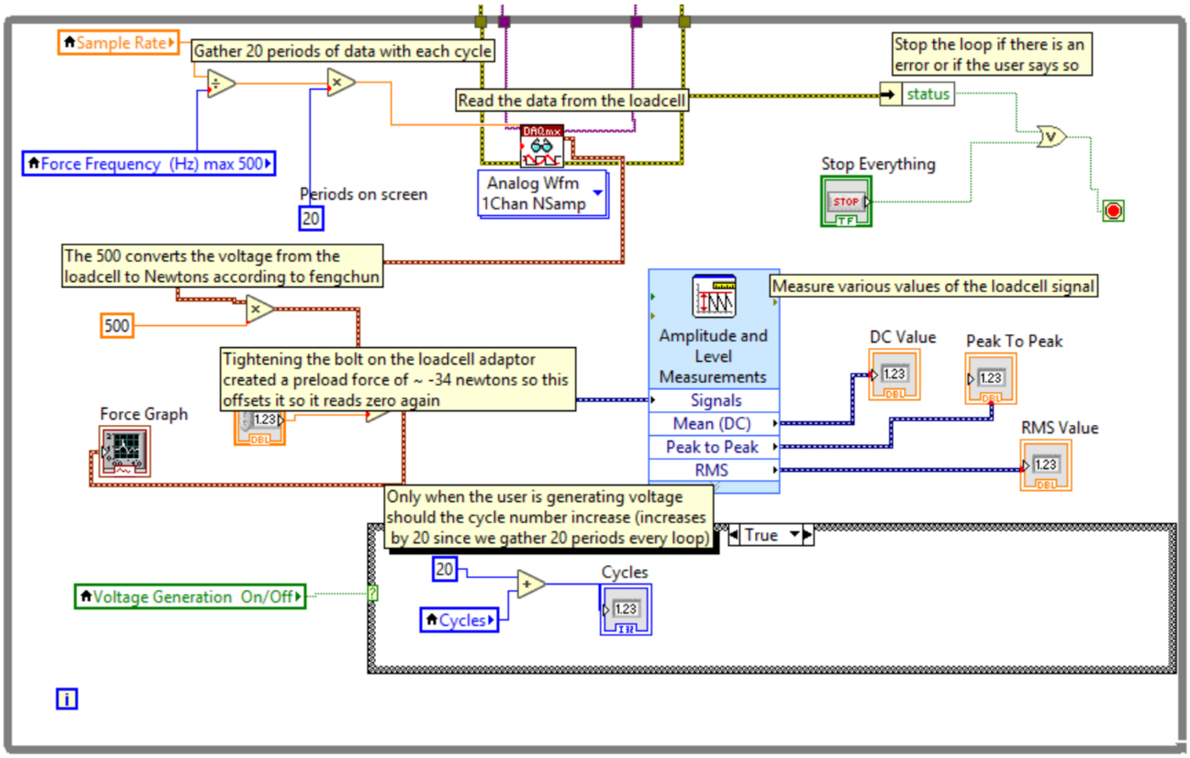

Figure 4: Data Acquisition Loop

Figure 4 shows the loop itself where I used Fengchun's math to accurately collect and display 20 periods of data at a time. This data is then multiplied by 500 to convert the voltage generated by the load cell into Newtons. This force data is then shown on a graph and various measurements are taken from the data such as peak to peak value and shown on the front panel. The number of cycles is also calculated, but only when we are sending a voltage to the piezo.

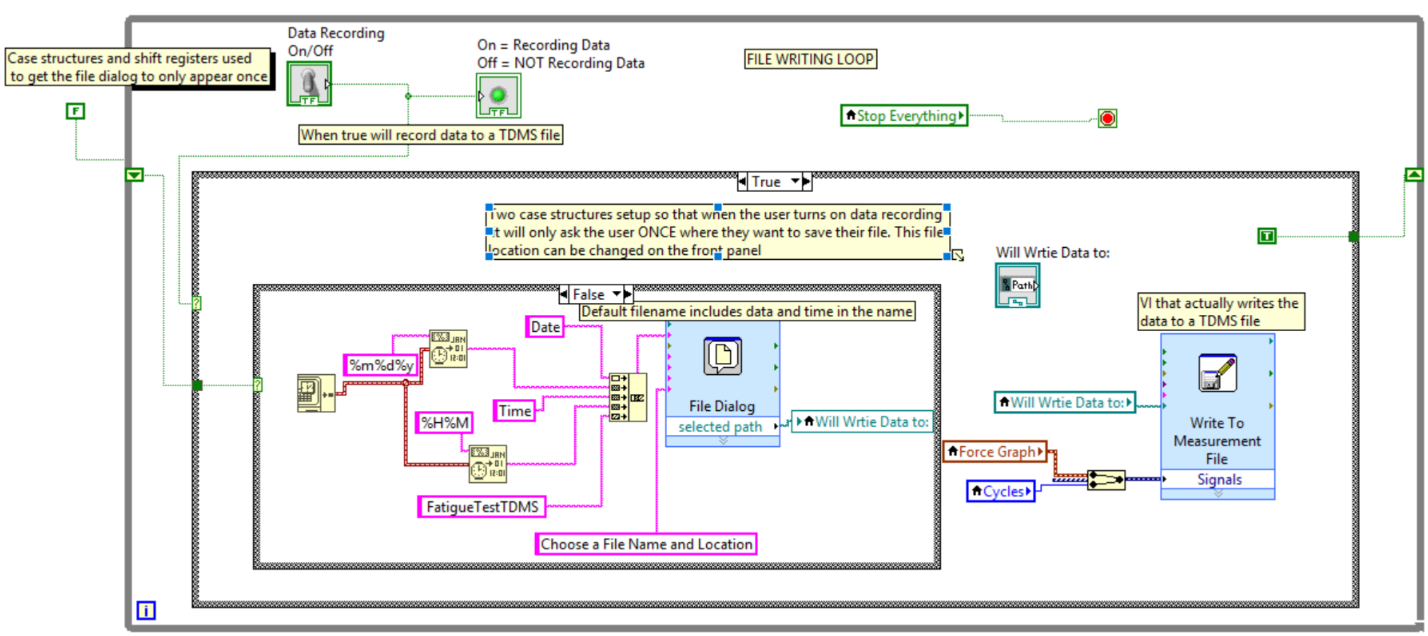

Figure 5: File Writing Loop

Figure 5 shows the loop that writes to .tdms file, the most compact of file formats available. It is setup to prompt the user where they would like to save the data when the data recording switch is flipped to "on" from the front panel. It only prompts the user when the switch is flipped for the first time and not again, even if the switch is turned off then back on. The location of the data writing is shown on the front panel and can even be changed while running.

After setting up all the code above, the machine needed some physical fixes to get up and running. The most major of the fixes was re-tapping the holes on the load cell adaptor because the current holes were stripped. The reason for the stripping of the holes was never discovered but it could possibly be from the cyclic force of the piezo actuator over a long period of time. Furthermore, while waiting for a test specimen to be made, I looked into the air cooling system which would be necessary for longer runs and designed some components to hold an air filter and pressure regulator. The cooling air was required to be at least class 4 according to ISO 8573.1 and pass through a dryer and a microfilter with 99.9999% efficiency. One such filter was found in the desk drawer along with fittings and tubing, none of which was setup. I found a pressure regulator in the drawers of miscellaneous components located in Tech BG21. After the machine had a working air cooling system it had to be setup and plugged into all of its components.

To set up the machine, Fengchun included setup instructions that are very informative but I will go over it again here. Firstly, you need to ensure that you have all the necessary electronics shown in figure 6 which includes:

1. Data acquisition chassis and input/output cards
2. Voltage amplifier
3. Power supply for the load cell
4. Load cell controller
5. (Optional) Oscilloscope

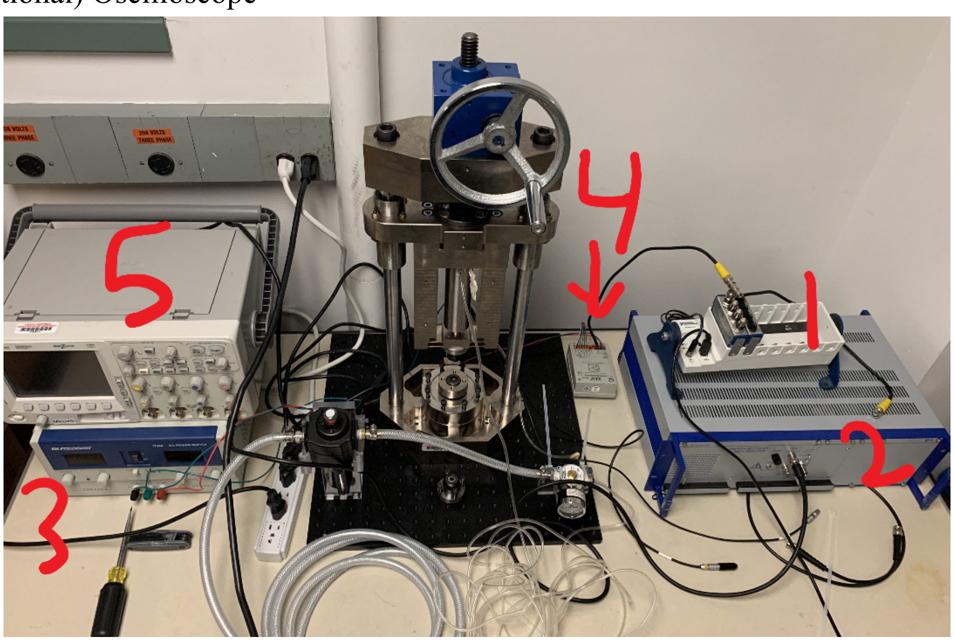

Figure 6: Full Machine Setup

Firstly ensure that the data acquisition cards are properly secured in the chassis. Plug in the load cell controller to the input DAQ card and the output DAQ card into the input of the amplifier (Control In) as shown in figure 7 and 8.

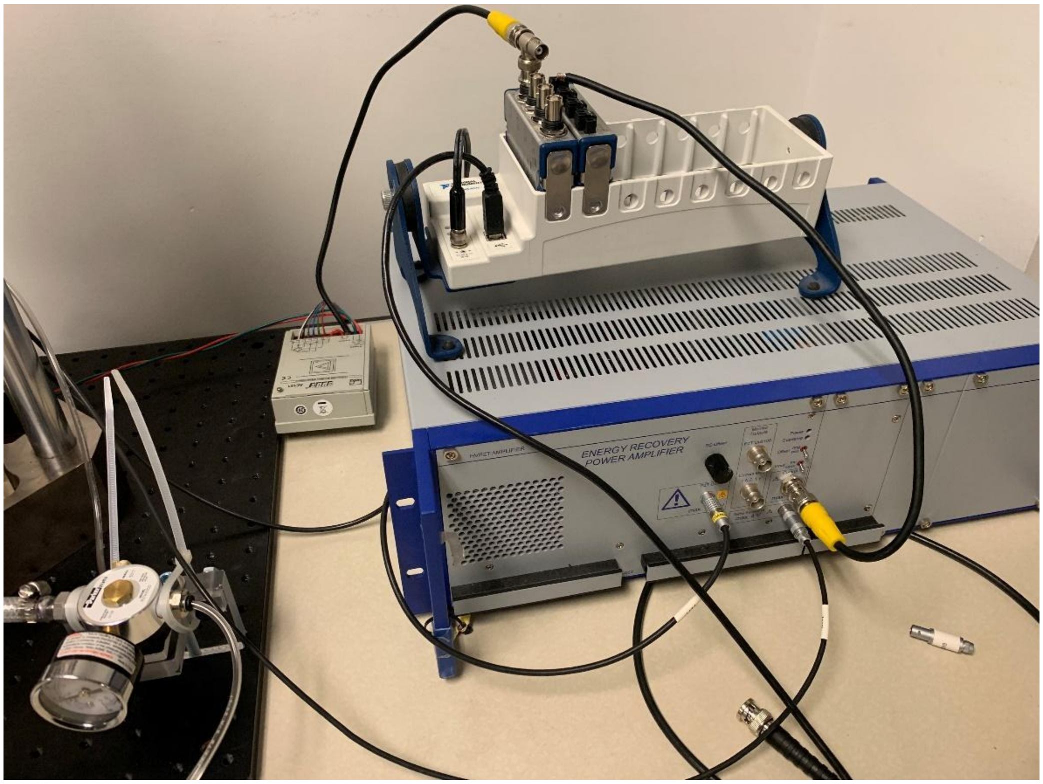

Figure 7: DAQ Card Setup

Figure 8: Amplifier Connections

Plug the input of the piezo actuator into the output of the amplifier (PZT Out) and be sure to plug the temperature sensor in as well, as shown in figure 8. Note: The DC offset knob on the amplifier should be set to zero.

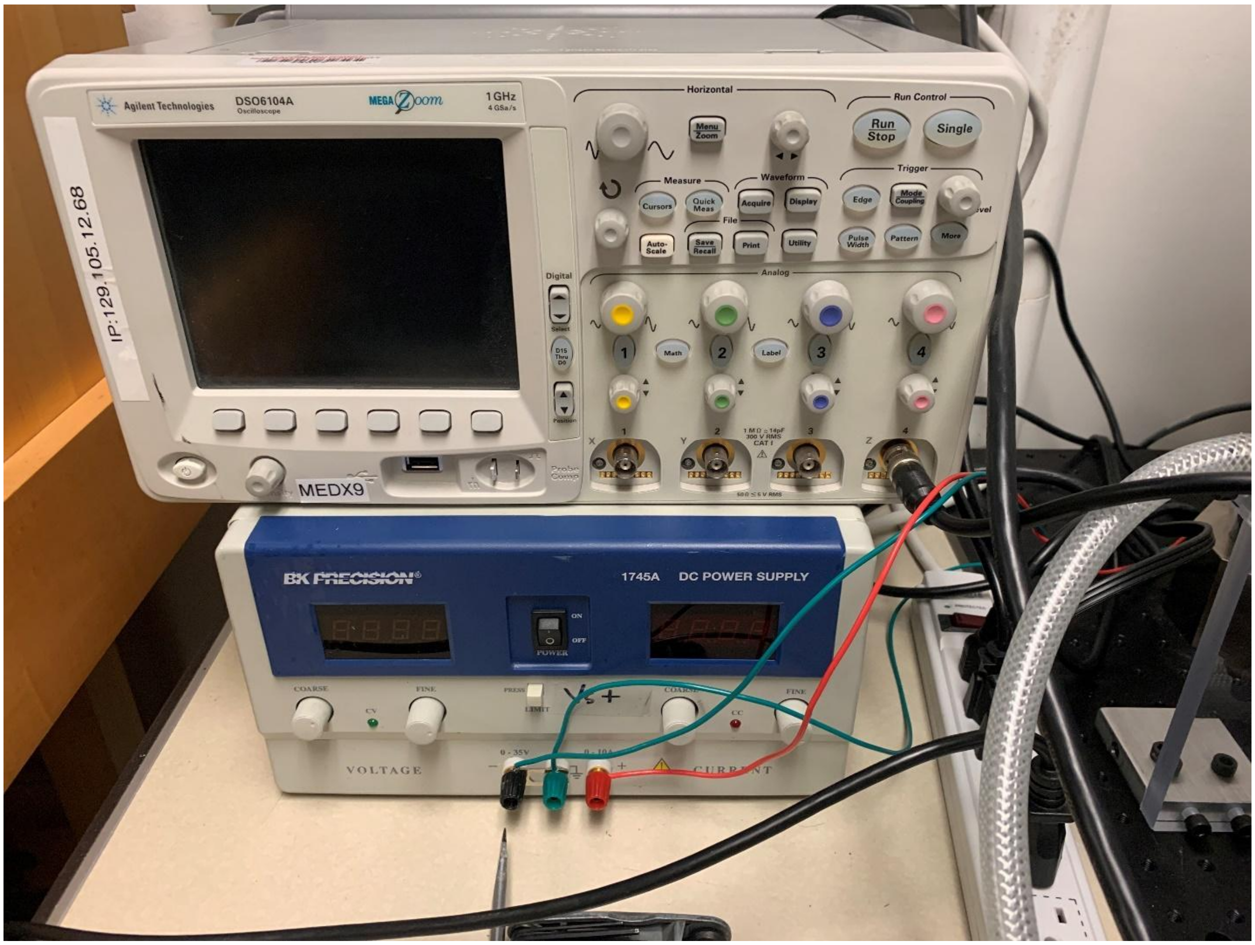

Figure 9: Power Supply Connections

Next plug the green wire from the load cell controller and the green wire from the piezo actuator to ground, which can be either the black or green knob on the power supply for the load cell. Plug the red wire from the load cell to the red knob on power supply. The power supply should ultimately be set to 24 volts.

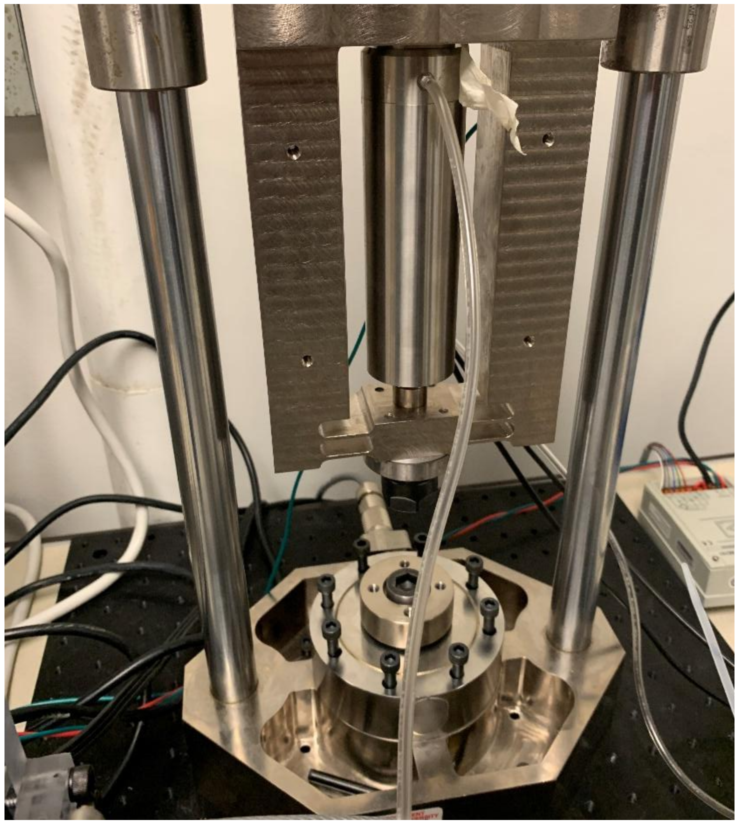

Figure 10: Air Hose Connection

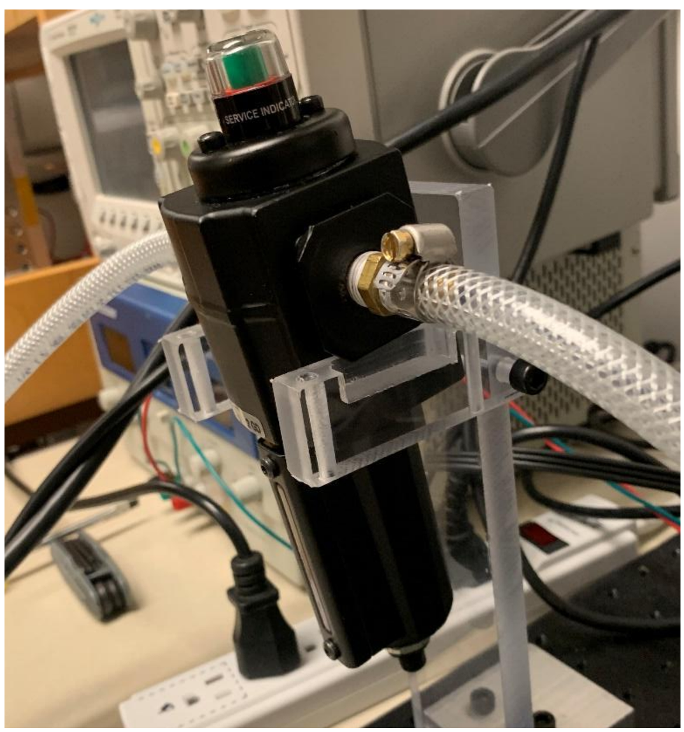

Figure 11: Air Filter

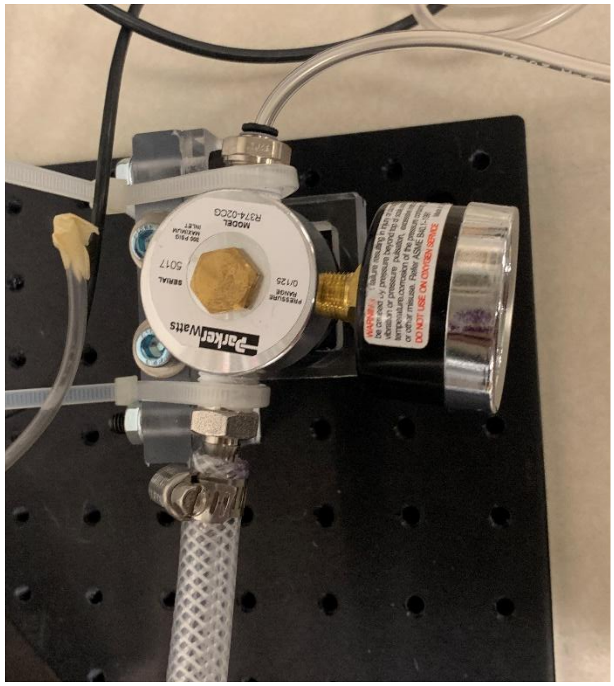

Figure 11: Pressure Regulator

Once all the electronics are plugged in, the air cooling must be setup as shown in figure 10. Note: The air pressure must not exceed 0.5 bar (7.3 psi). The air goes from source, through the filter (figure 11), through the pressure regulator (figure 12), and then to the piezo. Once each component is plugged in, the specimen can be put into the machine and a test can conducted to ensure that everything is functioning properly. Once again refer to Fengchun's instructions for a different, more detailed version.

The test coupon is made of tight tolerance highly corrosion-resistant 316 stainless steel rods (shown in figure 12) and is machine according to the drawing in figure 13 which must be done by the instrument shop professionals.

Tight-Tolerance Highly Corrosion-Resistant 316 Stainless Steel Rods

- Yield Strength: 30,000 psi
- Hardness: Rockwell B75 (Medium)
- Temper Rating: Softened (Annealed)
- Heat Treatable: No
- Specifications Met: ASTM A276, AMS 5648

The diameter of these precision-ground rods is held to a ±0.0005" tolerance. Made of 316/316L stainless steel, they contain less carbon than standard 316 for better weldability. The addition of molybdenum gives this material excellent corrosion resistance. Use it in a variety of marine and chemical-processing applications. This material maintains its corrosion resistance in temperatures up to 1500° F.

Dia.	Dia. Tolerance	Straightness Tolerance		1/2 ft. Lg.	1 ft. Lg.	2 ft. Lg.	3 ft. Lg.	6 ft. Lg.
1/4"	-0.0005" to 0.0005"	1/8" per 6 ft.	8936K3	$5.30	$8.16	$14.44	$18.83	$31.39

Product Detail

Figure 12: Coupon Material from McMaster

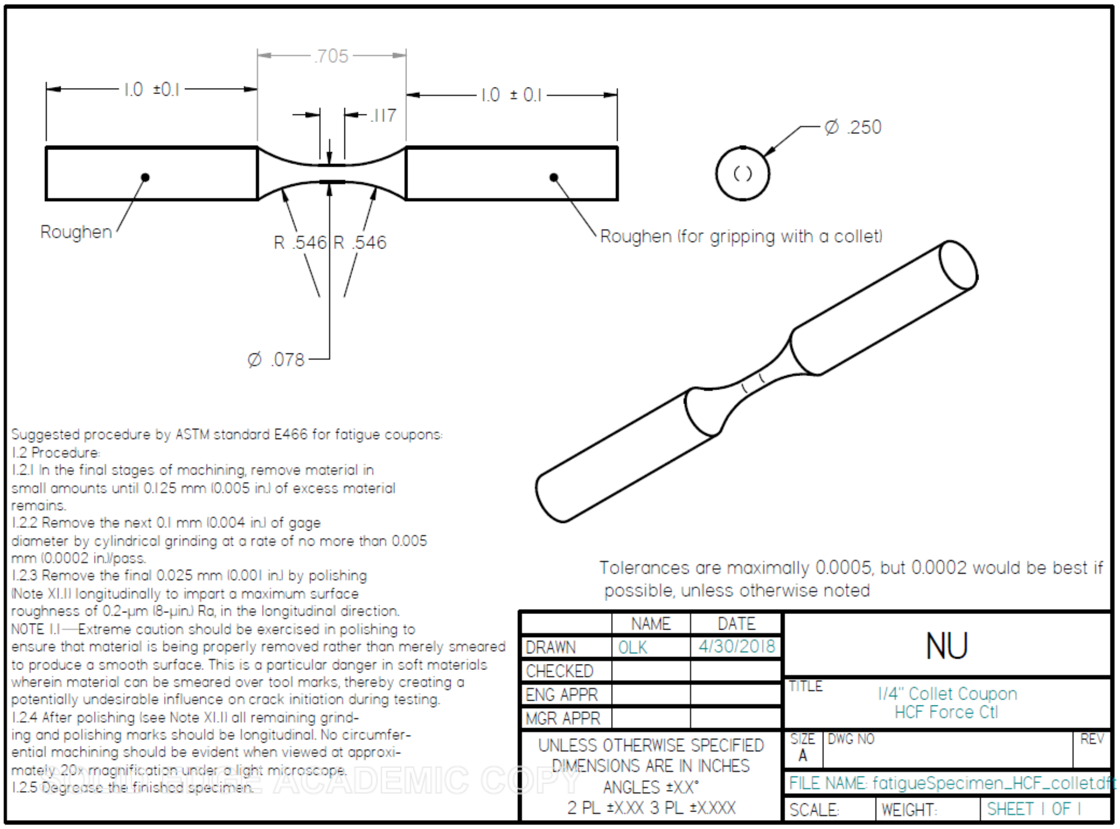

Figure 13: Coupon Specifications

That summarizes my work done in the 3 weeks after I finally got a computer with the proper software. Since my time was limited there are many improvements that could be done/were realized as I was conducting tests during the last week of work which will be listed below.

Firstly, the buffer size and sampling rate for voltage generation need to be properly configured. If you connect the voltage generation DAQ card directly to an oscilloscope you can see there are many discontinuities in the sine wave at low sampling rates. I tried various things like changing the timing clock settings, configuring the buffer (with the configure buffer VI), but I was unable to find a solution to this issue. However, while the voltage from the DAQ card has a lot of discontinuities, monitoring the voltage from the amplifier shows a smooth curve which indicates that the amplifier smooths out the curve in some way. There is no mention of this in the documentation for the amplifier and could be fine for real tests.

Secondly, a way to clear the output buffer when the voltage generation switch is turned to “off” needs to be implemented. Currently, when voltage generation is set to “off” the piezo will still take in voltage data that is within the output buffer and continue to actuate, even though no more new data is being added to the buffer. Resolving this issue would mean that the piezo immediately stops actuating as soon as the switch is set to “off” on the front panel.

Thirdly, the data logging system needs to be improved upon as a whole. Currently the code records data at a variable sample rate set by the user which is default 50,000 samples/sec. This sample rate is arbitrary and does not reflect how much data should be collected or is needed to achieve usable results. Furthermore, high sample rates, such as the default, record many gigabytes of data which can add up quickly to fill the hard drive on the computer and crash the program. The sample rate needs to be thought of carefully and implemented to gather useful data.

One change that will greatly affect the code is changing from graphs to charts in LabVIEW. This change was thought of at the end and makes a lot of sense because we want to see the real time evolution of the machine signals, which a graph does not accomplish. These two forms of output function very differently and have different requirements and can affect this data recording & sampling rate problem.

Fourthly, some method of measuring how the machine is behaving physically could prove to be essential. A safety switch of some sort that can detect if the piezo is exerting too much force, getting too hot, or doing something it shouldn't and respond appropriately by turning the machine off could prevent the machine from getting damaged. Additionally, measuring the force data and detecting when the specimen breaks to turn off the data recording could prevent hours of searching through a very large data set.

Fifthly, a way to measure strain would be necessary to create stress-strain plots for test coupons. No effective method has been decided to measure the strain for this machine. There are several possibilities such as an extensometer or a fast camera to optically record the strain. This is perhaps the biggest most time consuming improvement that can be made and will most likely involve purchasing a new device.

Lastly, purchasing a pressure regulator designed for much smaller pressures is necessary because the current regulator does not give a very accurate reading at the low pressures required by the piezo. Once this regulator is purchased redoing the air cooling system mounting would be beneficial. All holes on the baseplate are 25 mm apart and are tapped for an M6 bolt. Looking back I would 3D print something simple that mounts to the base rather than machine something overly complex.

Ultimately, I learned a lot about the research world and sophisticated devices used to answer complex questions. No matter how small the task or how thought out something is, you are likely to run into problems that have easy solutions and others that can take a lot of time to solve. How we deal with these setbacks is essential to the success of a researcher and is the most valuable lesson I learned this summer.