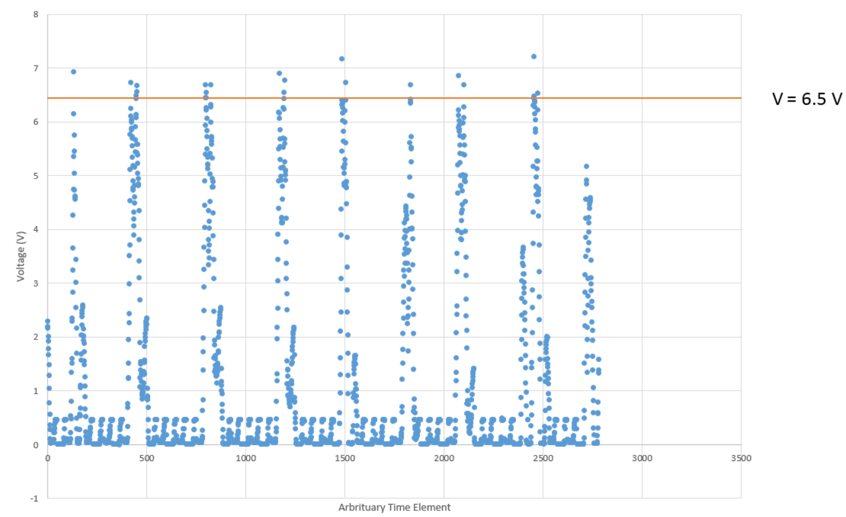
Miguel Garcia De Los Santos, Christina Nelson, Dmitriy Leonchik
Capstone Project Term #2
March 23, 2016

Data Analysis of New Current Sensor

Looking at the results on appendix 1, we can see that the new current sensor is functioning as expected. The average voltage is 0.447 V which is what we were expecting from initial testing on the oscilloscope. The greatest value discrepancy from the average is 0.94 % and the lowest value discrepancy is -0.98 % from the average. So we have succeeded in making a circuit with minimal noise and functioning properly and we know the circuit is linear based on earlier test.

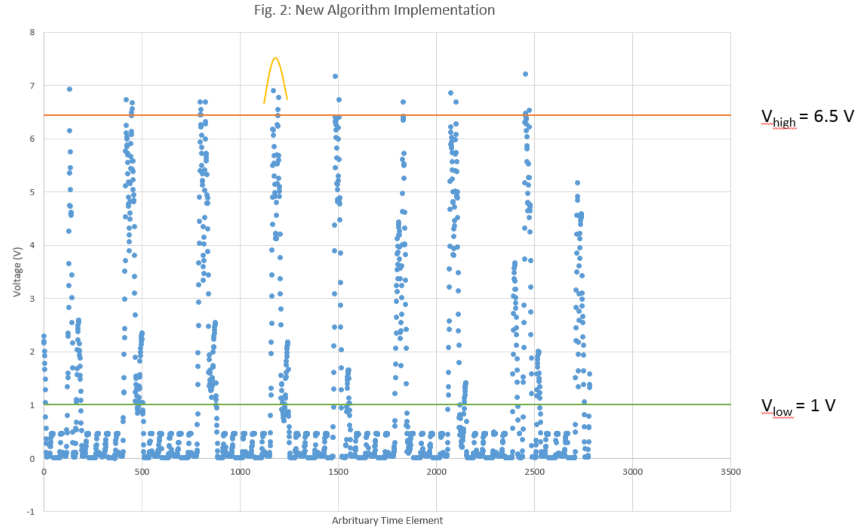
Before reading the report, please take a look at the appendix 2 found on pg. 3. Looking at the data, we can see that the current sensor output maxes out at 7 V. Theoretically, the sensor should be able to output to 9.4 V, since the maximum current that the current sensor can provide is 125 mA and the resistance is 75 Ohms. However no testing has been done to see if the current sensor is able to provide all the current that it specifies in the data sheet. It is also a possibility by looking at the data that welder provides a static current value and just increases the frequency of the spikes. There is also more possibilities, such as the current spikes move too fast for the current sensor to react or the sensory board does not sample the data fast enough to read the input properly. I do not doubt the current sensor since the data results show that it is linear and functioning properly. Looking at Fig. A and Fig. H in the appendix we can see that voltage spikes are the same height, but the frequency of the spikes have increased in Fig. H. Which leads me to believe that the welder outputs the same level of current, but the current setting has to do with the frequency of the spikes.

Fig. 1: Old Algorithm Implementation



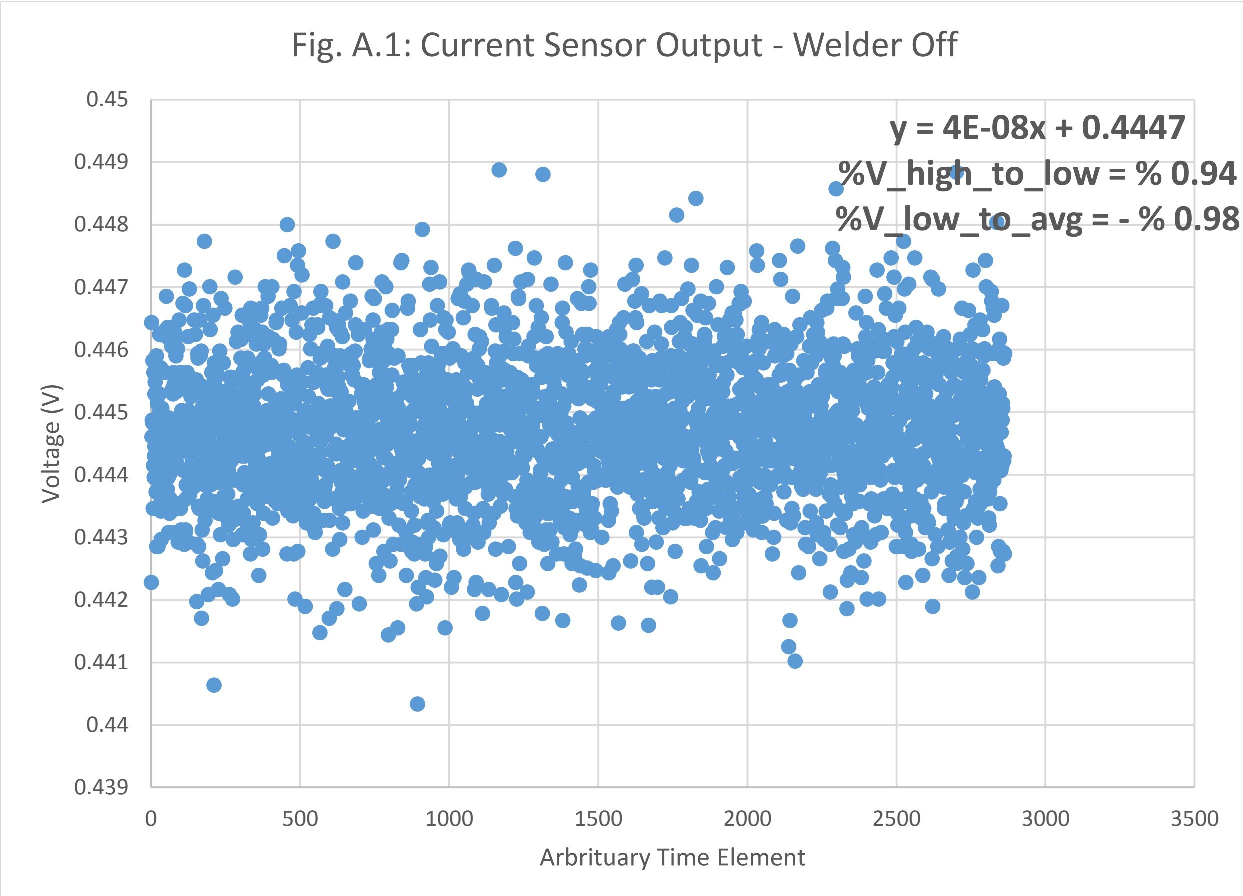
Now assuming we connect the sensor, and using the old algorithm of the droplet calculator, we are going to run into issues. In the old code, we consider a spike larger than 6.5 V to be a "spike." However, the program does not distinguish between a spike and data point. Looking at Fig. 1, we can see if we consider a point larger than 6.5 V it will count as a spike in the program. Looking at the second spike, we have 3 points larger than threshold; this means the program thinks there are 3 spikes, when in reality it is 1. This can also be why the machine is not functioning properly at the moment.

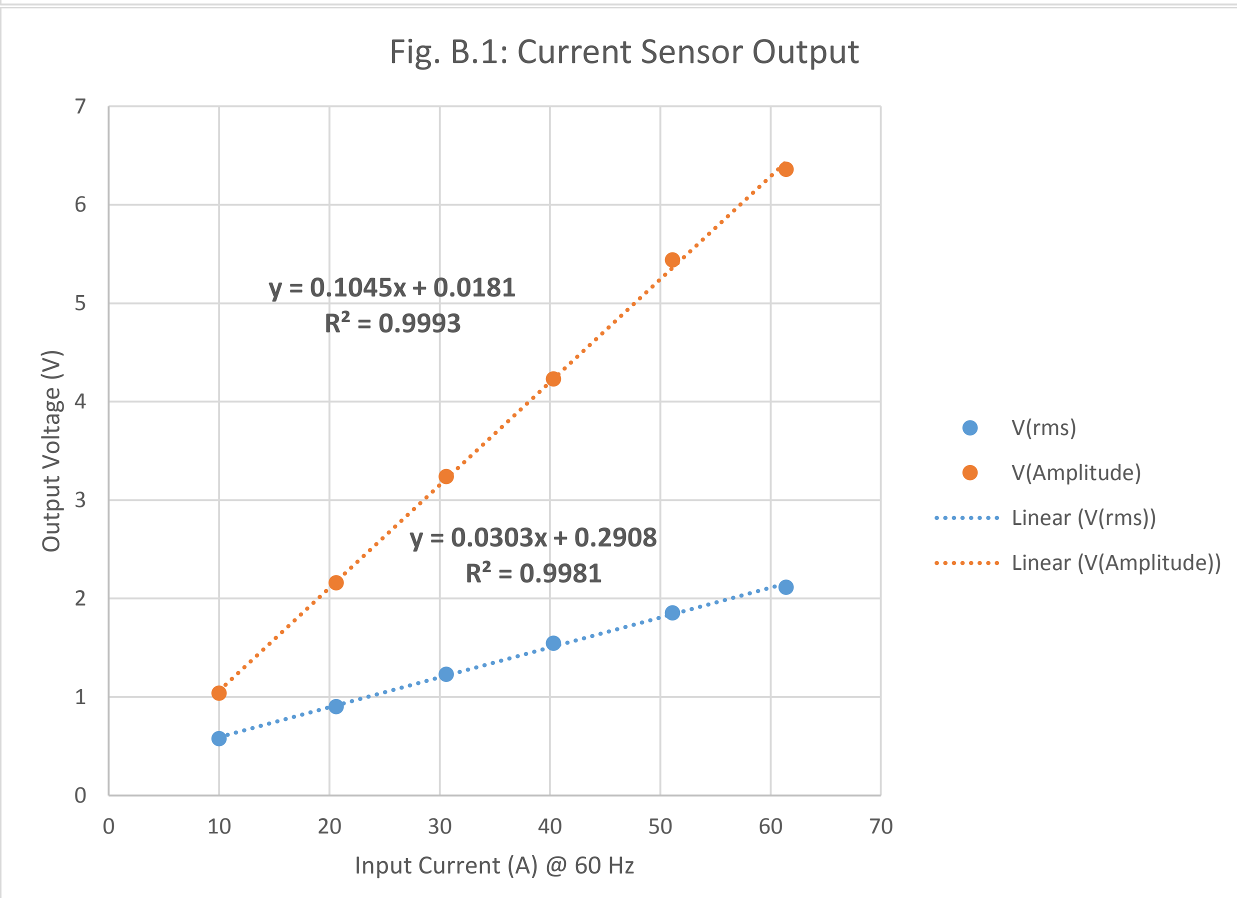
Fig. 2: New Algorithm Implementation



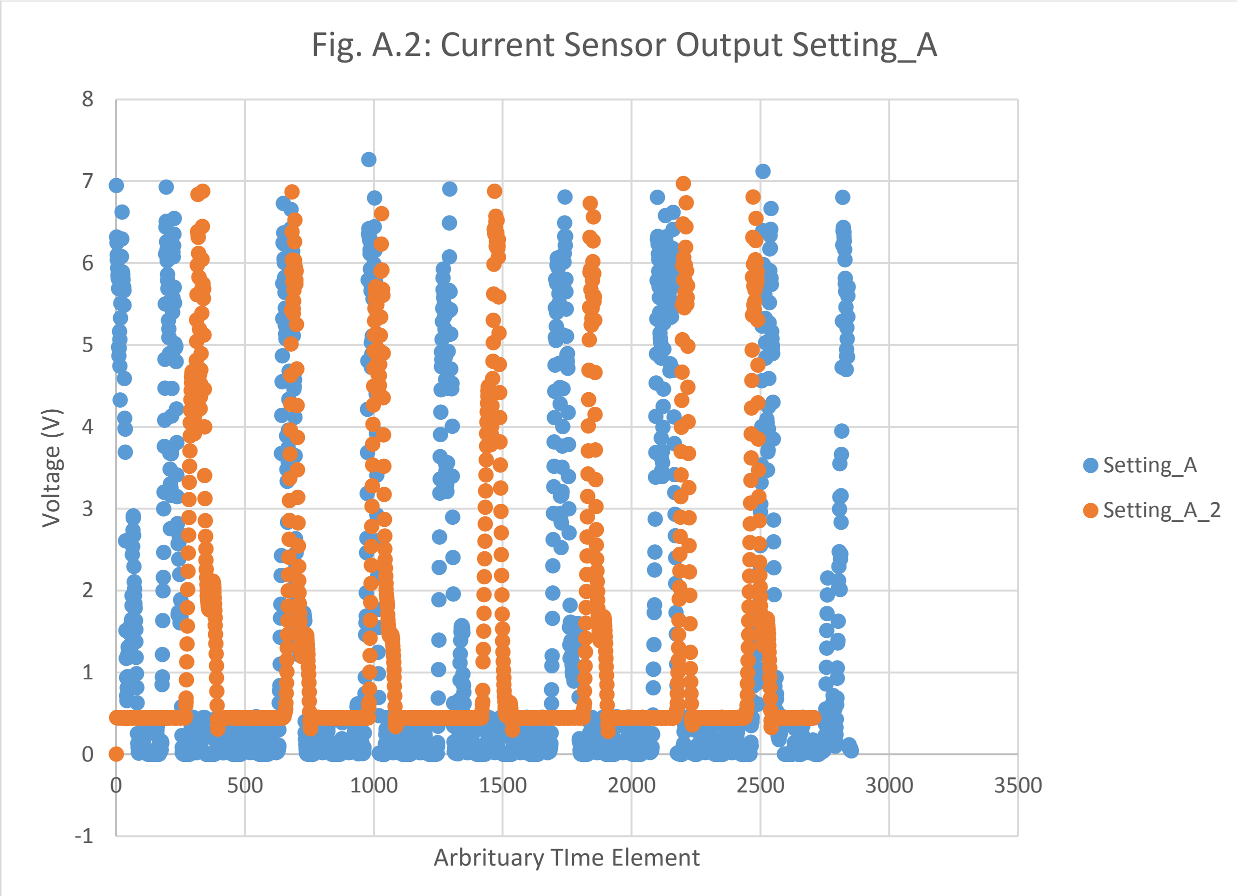
My proposal to fix this is simple. Let us look at Fig. 2. First, we set two parameters, V_{high} and V_{low}. The program will track points that are larger than V_{high} and will find the largest value. Once the program sees that the points are starting to decrease and reach V_{low}, the program will consider it a spike. These values are not set in stone and will be adjusted to future testing. They are for illustration purposes only.

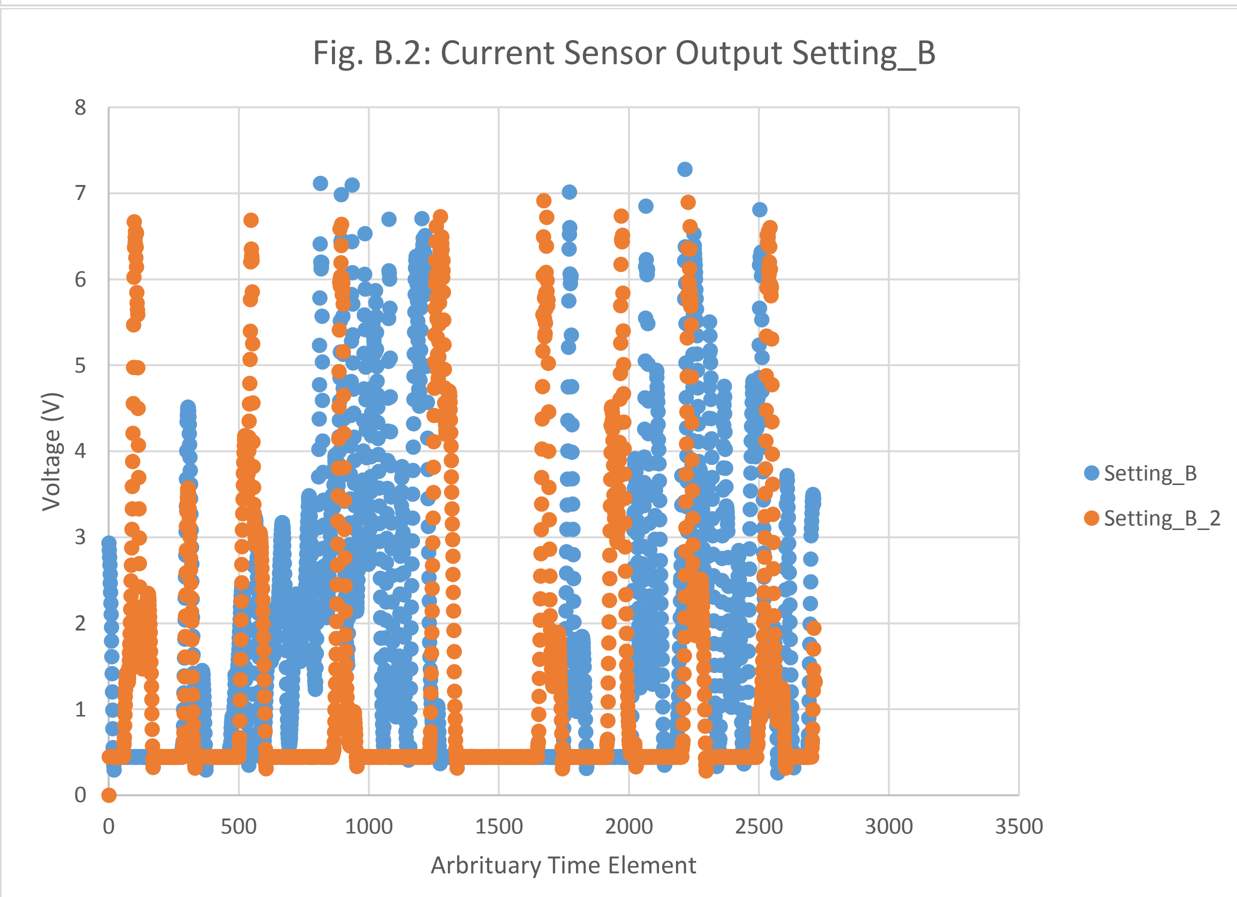
Appendix 1: Welder Off





Appendix 2: Welder On





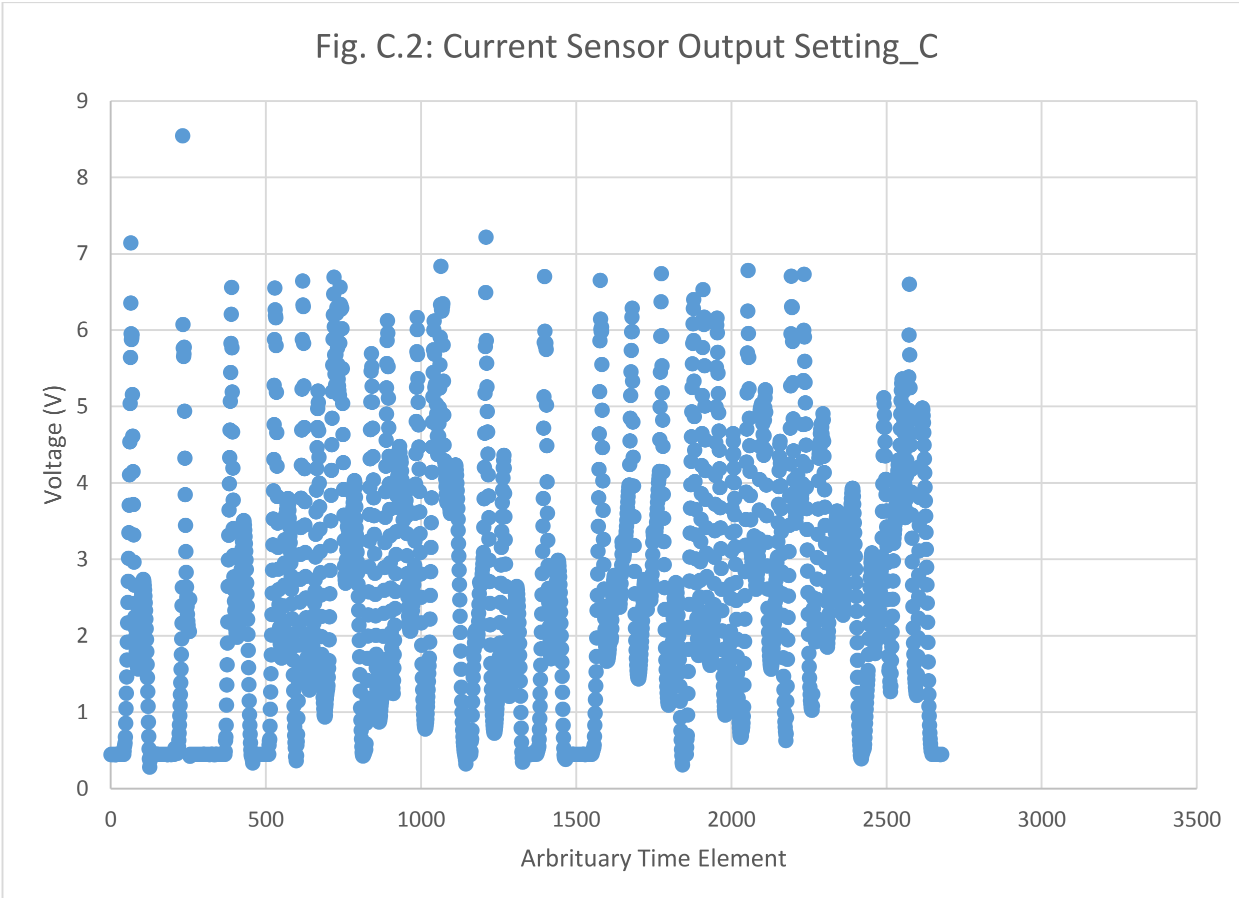


Fig. C.2: Current Sensor Output Setting_C
Voltage (V)
Arbrituary Time Element
0
1
2
3
4
5
6
7
8
9
0
500
1000
1500
2000
2500
3000
3500

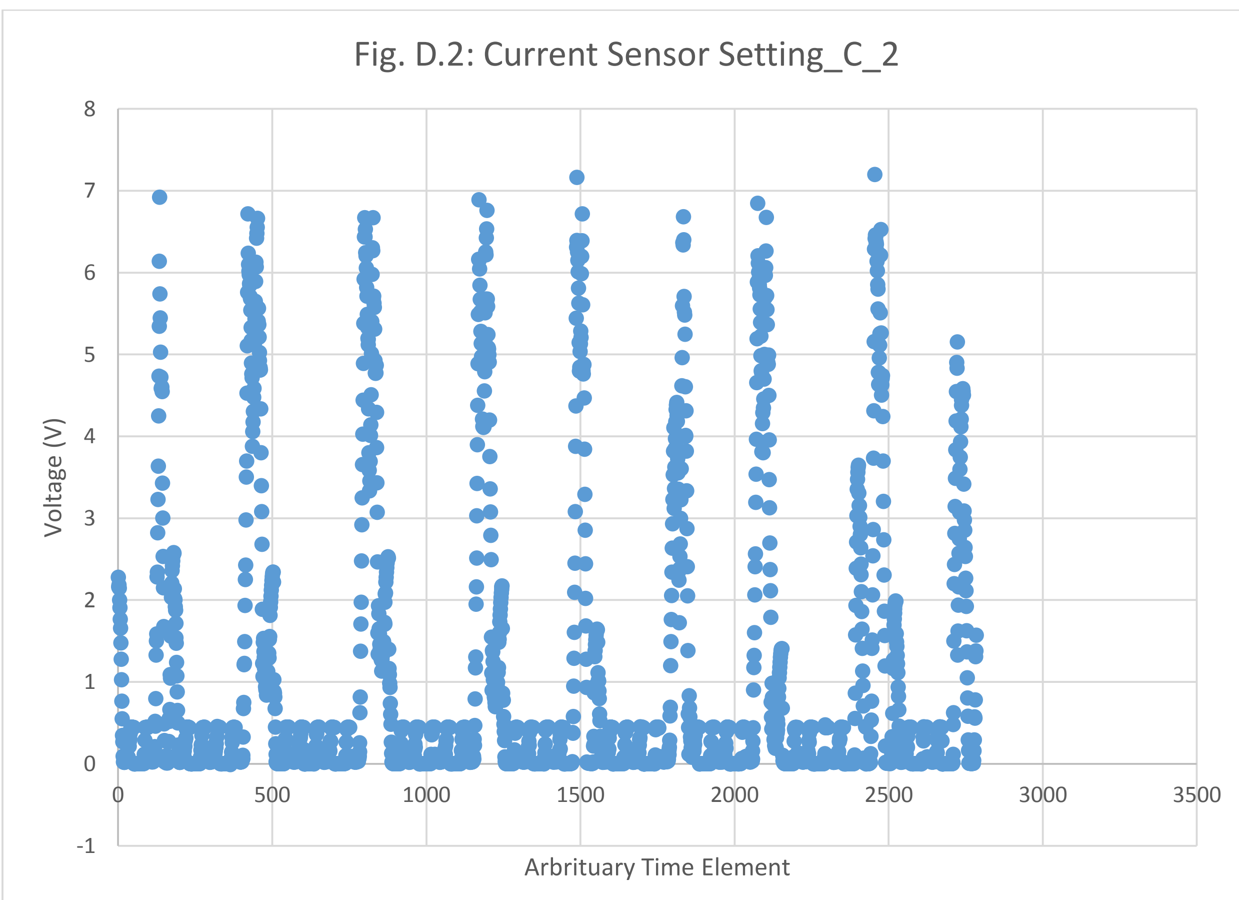


Fig. D.2: Current Sensor Setting_C_2
Voltage (V)
Arbrituary Time Element
-1
0
1
2
3
4
5
6
7
8
0
500
1000
1500
2000
2500
3000
3500

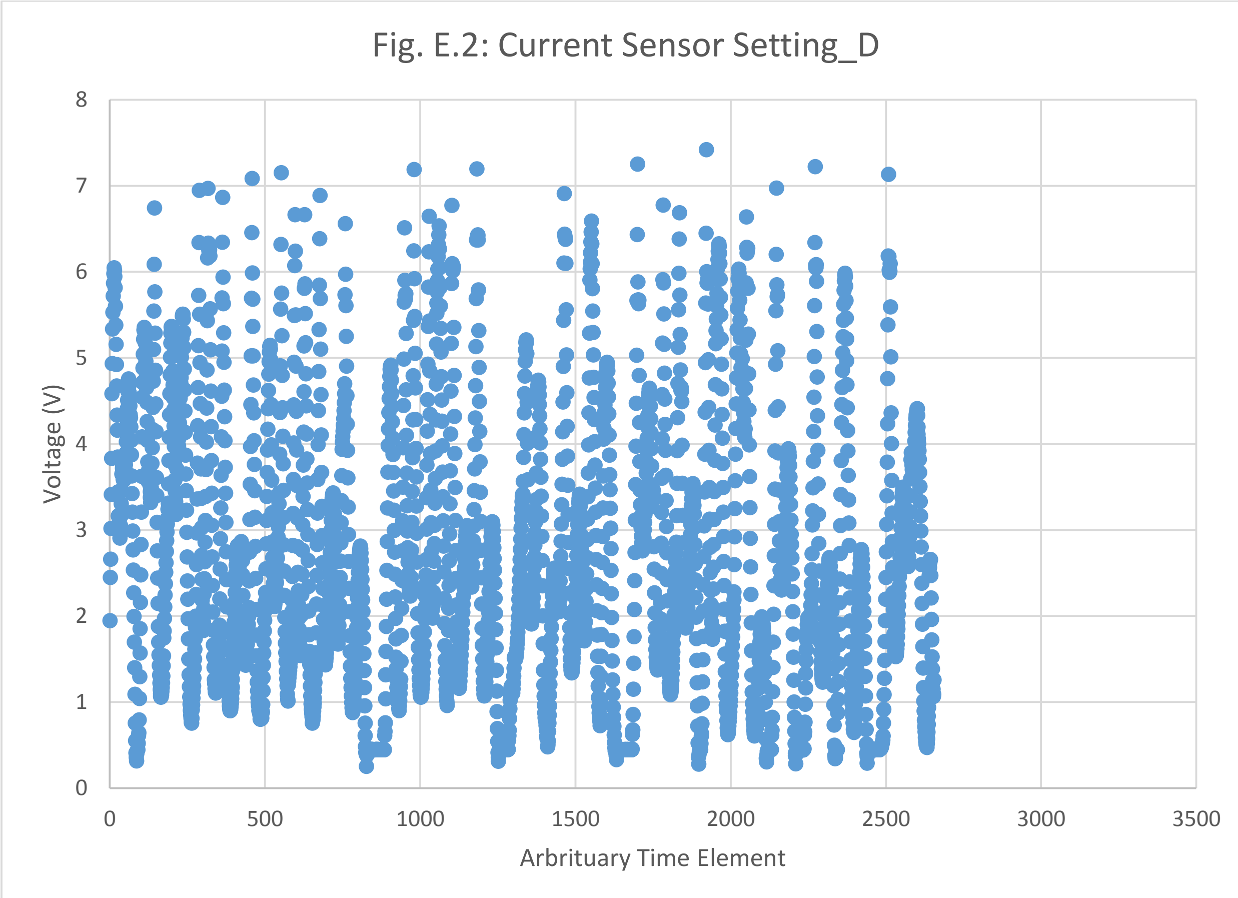


Fig. E.2: Current Sensor Setting_D
Voltage (V)
Arbituary Time Element
0
1
2
3
4
5
6
7
8
0
500
1000
1500
2000
2500
3000
3500

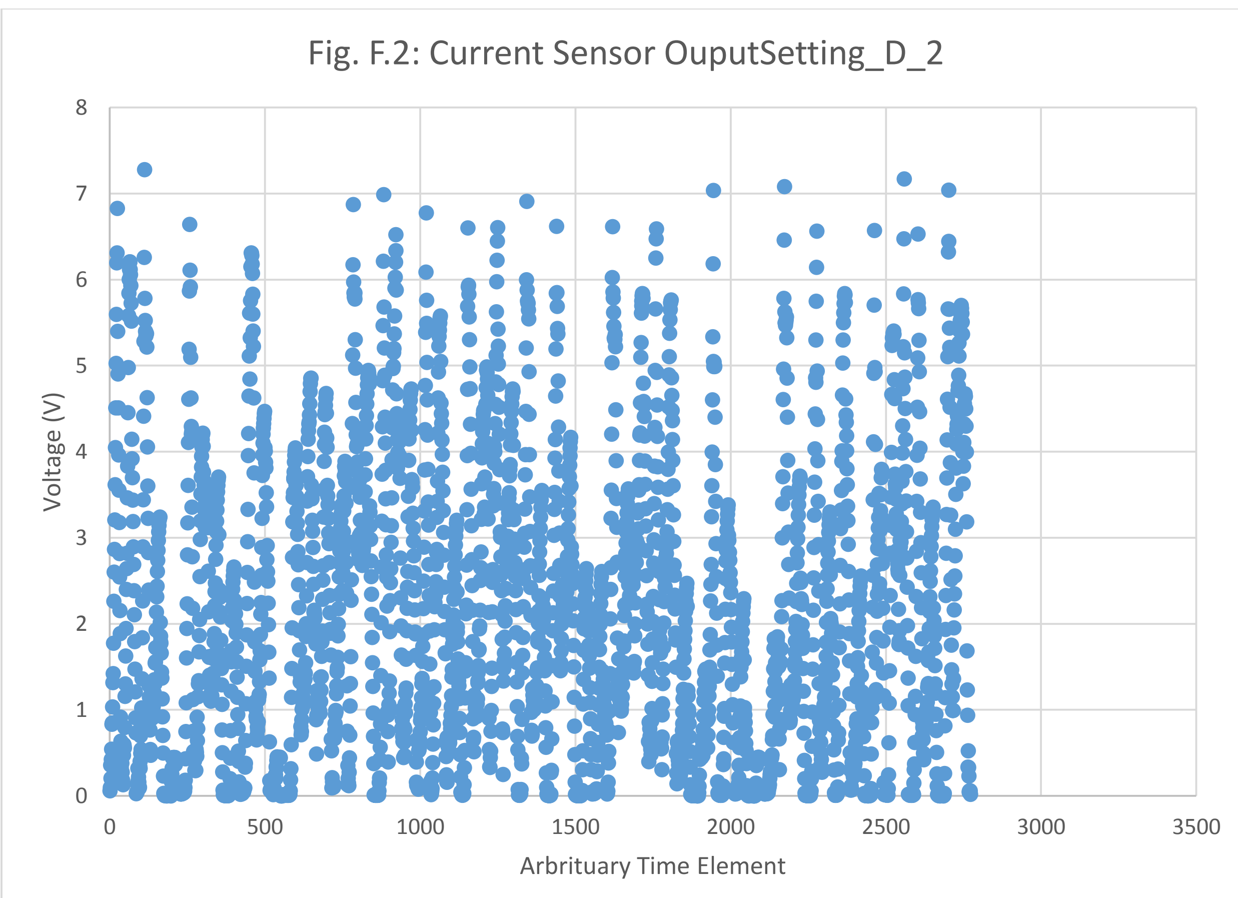


Fig. F.2: Current Sensor OuputSetting_D_2
Voltage (V)
Arbituary Time Element
0
1
2
3
4
5
6
7
8
0
500
1000
1500
2000
2500
3000
3500

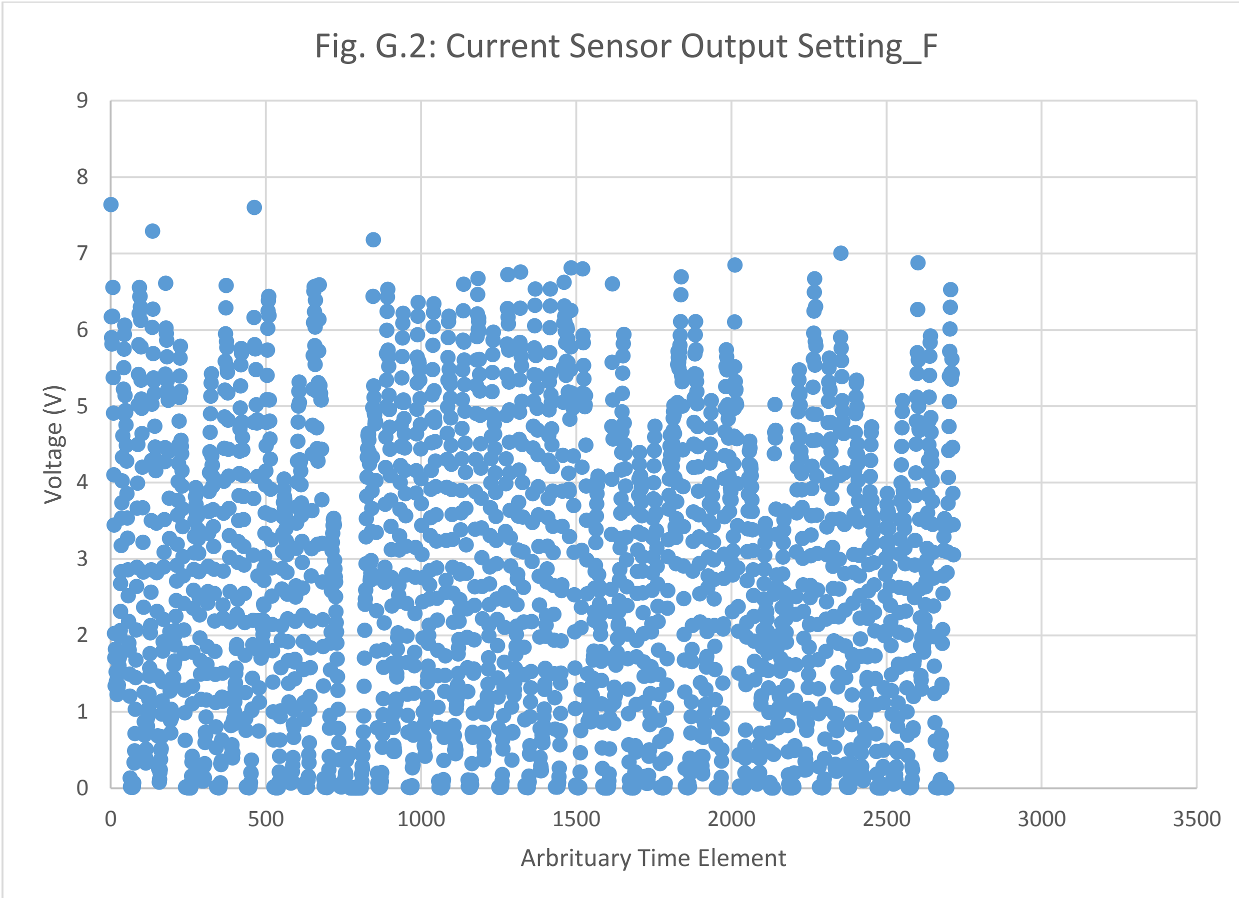


Fig. G.2: Current Sensor Output Setting_F
Voltage (V)
Arbrituary Time Element
0
1
2
3
4
5
6
7
8
9
0
500
1000
1500
2000
2500
3000
3500

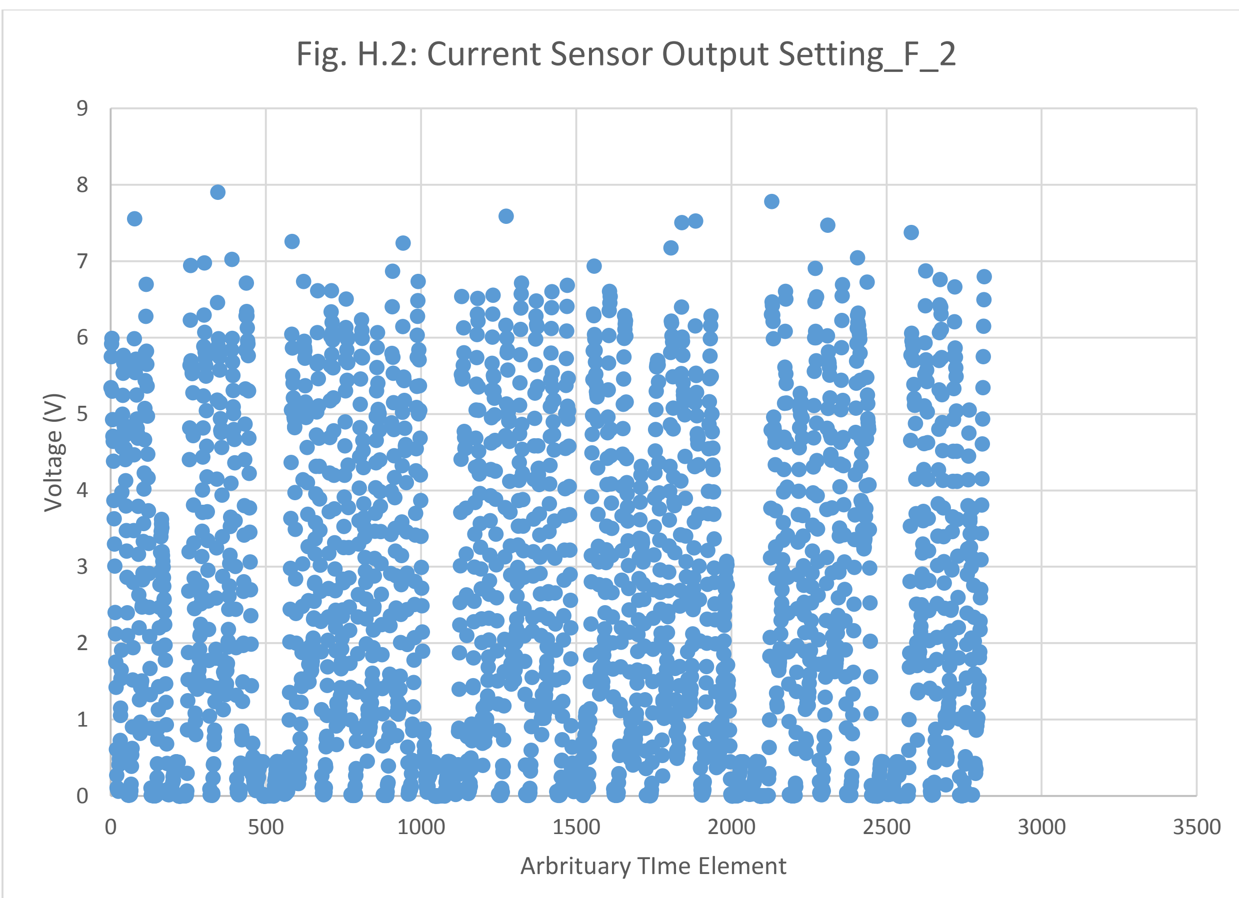


Fig. H.2: Current Sensor Output Setting_F_2
Voltage (V)
Arbrituary TIme Element
0
1
2
3
4
5
6
7
8
9
0
500
1000
1500
2000
2500
3000
3500