Lab 5 : Signal and Noise

Olaf Bach

1 Introduction

The arduino is a great tool however, because of how it creates voltage through PWM the output it gives is not smooth. This is especially visible through an LED running an analogWrite function. The point of this lab is to determine if we can remove the signal noise through averaging values, or smoothing.

2 Setup

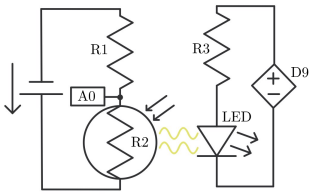

Figure 1: Circuit Diagram

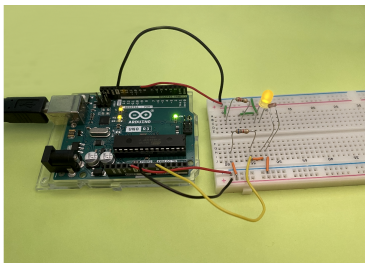

Figure 2: Photo of Breadboard

The Breadboard was set-up into two modules. The first was the LED which was controlled by the arduino, and the other was a photoresistor which was reading the LED from the first module.

3 Arduino Code

The arduino code is responsible for writing the sinewave signal onto the d9 pin which controlls the led brightness. The arduino then reads a photoresistor's resistance which is exposed to the led. The result however, is very noisy. To remedy this we apply a smoothing function found here Link. This has two amazing functions, firstly it gets rid of the spikes formed from when a reported value is inbetween two intergers, and secondly it gets rid of most noise that might be effecting the value.

Link to the code: Link

4 Data

For the first experiment I used various averaging widths to compare them. The first graph is the function when the average width is 50. The second graph is when I ran the code 3 seperate times each with different

widths. I imported all data through github and made the graphs in python.

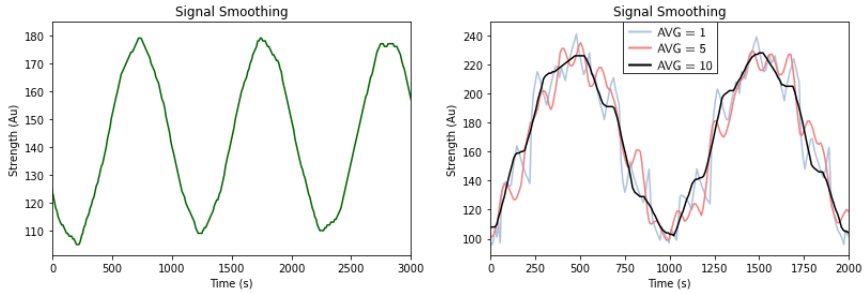

Figure 3: Comparing Average Widths

Of course, none of my graphs are as smooth as they should be. I'm guessing this has something to do with my LED. The yellow LED I have is extremely flickery and jumps between brightnesses rather than having a smooth gradient. You can see this especially in my 10 width average, there are very obvious steps where my line becomes horizontal. I think with a better LED a lot of this noise could be removed.

Link to comparison data and python code: Link

For the second experiment I changed the amplitude of the brightness signal from 50 to 2. I then tried to find a signal by using the smoothing function. To do this I ended up needing a really large avg width being 50. Below is the resulting graph.

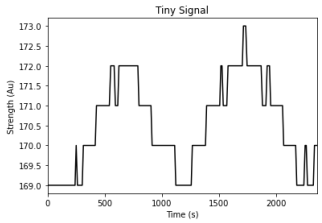

Figure 4: Signal of Small Amplitude

Its interesting to note just how powerful this smoothing function is. The brightness difference between the peaks and troughs is minascule, yet a computer running a simple bit of code can realize that there is a signal here immediately. Link to data: Link

Link to code: Link