Group members：Bhautik (Brian) Amin, Yimai Zhou, Jayden Chen
ECE T580 Project 3 Final Report

Objective:
To find the noise and measurement uncertainty characteristics of an ultrasonic transducer sensor, and design a Kalman Filter to estimate the position. Taking comparisons between estimated data, measured data, and ground truth data and drawing conclusions regarding the statistics.

Part 1: Obtaining general statistics of three ultrasonic transducers

We tested three sensors along a small testing area, where we used a ruler to measure out marks for 6”, 12”, 18” and 24” distances away from a stationary ping sensor. The ping sensor was placed on top of a block of wood a few inches away from the ground to avoid any disruptions (See Figure 1 for depiction of testing environment). A folder and straightedge was used as a “wall” for the ultrasonic sensor. For the manual measurements, a function generator was used to output trigger pulses to the sensor, and an oscilloscope was used to read and measure the echo pulses being put out of the sensor. For each distance, three measurements were taken from the scope in different time positions along the scope (Figure 2), per each sensor. In Figure 2 for example, the oscilloscope image is frozen and three pulses from this set will be measured using the oscilloscope cursors and the time between the pulses will be recorded for further analysis. In Figure 1, note that the Arduino shown is only to supply power to the sensor, and it is not currently being used for any data collection.

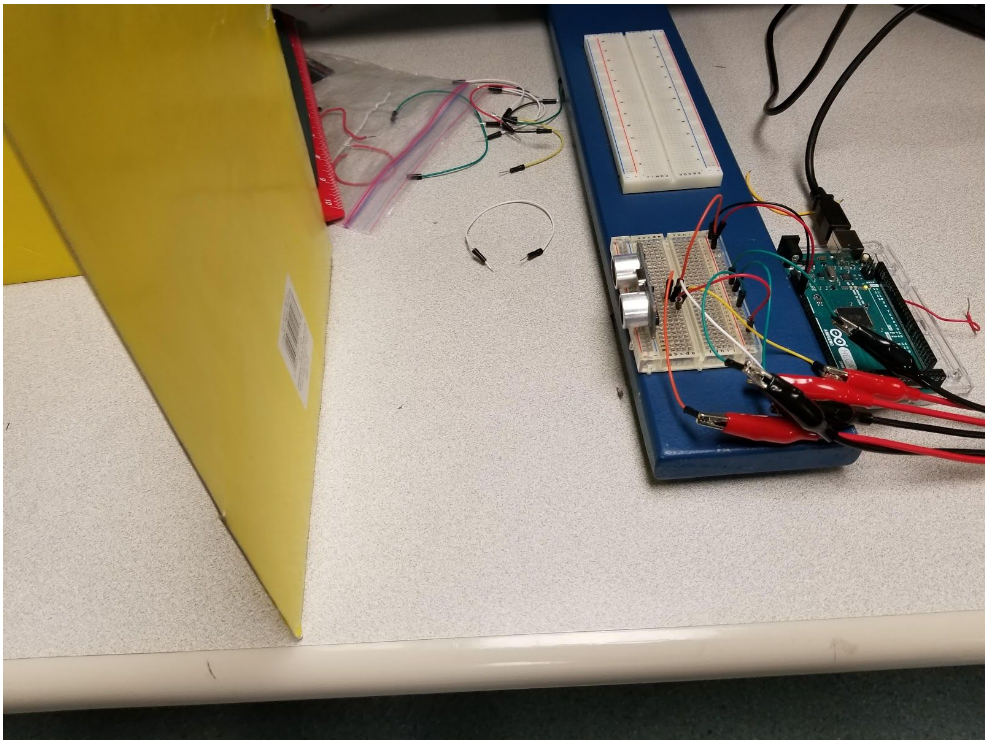


Figure 1. Testing environment for sensors

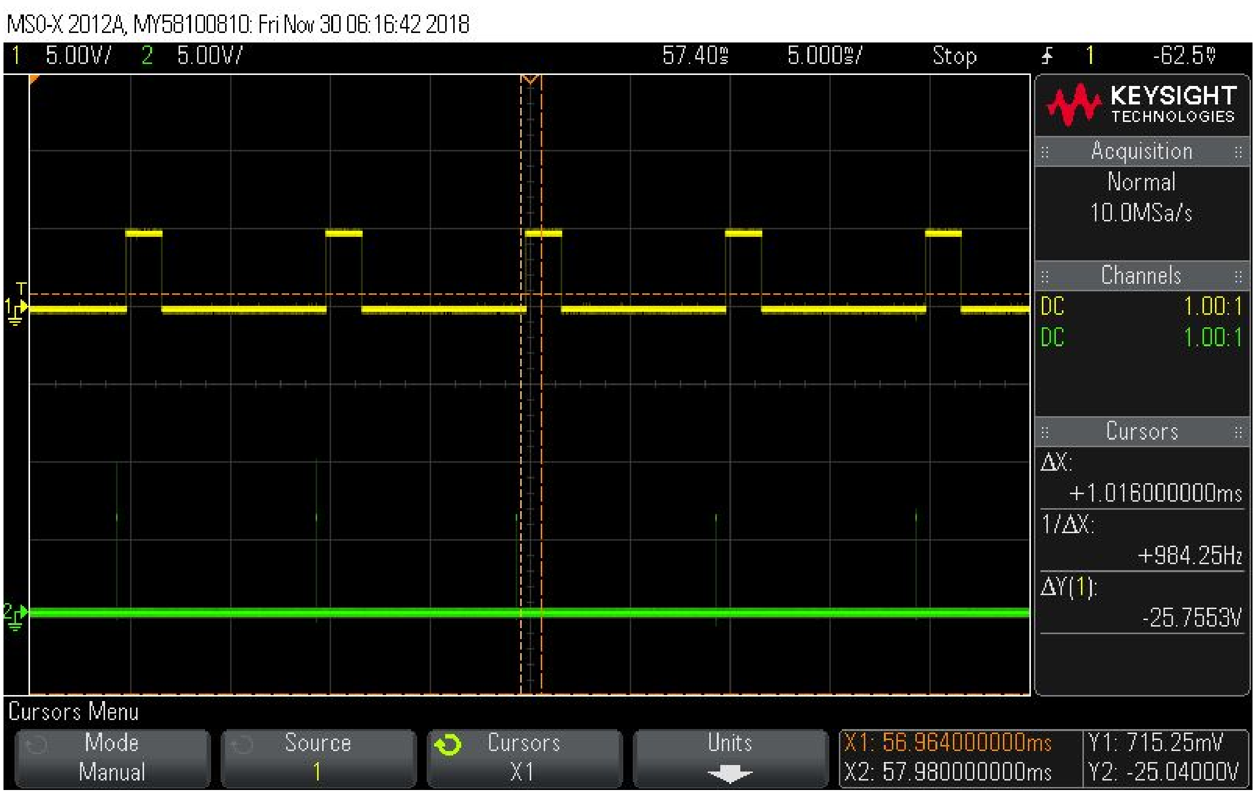


Figure 2. Sensor Sample Output

Actual Measurement (inches)	Echo Pulse (us or ms)	Sensor Measurement Calculated (inches)	Time Position on Oscilloscope (ms)	Normal Distribution
Sensor 1				
6	868	5.859	1	0.457470078
6	900	6.075	9	0.839375751
6	900	6.075	19	0.839375751
Sensor2				
6	1	6.75	36.98	0.58232444
6	1.02	6.885	46.98	0.358728937
6	1.016	6.858	56.96	0.399585642
Sensor3				
6"	920	6.21	58.94	1.026341026
6"	900	6.075	68.94	0.839375751
6"	930	6.2775	78.94	1.07802612

Table 1. 6” Manual Tracked Data for Three Sensors

Measurement (inches)	Mean between three sensors (Inches)	Variance between three sensors (Standard Deviation)
6	**6.3405**	**0.364583324**

Table 2. Mean and Standard Deviation between three sensors

For post analysis, we used Excel to convert the measured pulses into a distance in inches - as well as calculate the mean, standard deviation, and the normal distribution of our collected values as well as plot them. In Table 1, we see our data for 6” measurements. In Table 2 we see that the mean of our data is 6.3405 inches, and the standard deviation is 0.364583324. We can calculate the normal distribution using these numbers for each collected point, shown in Table 1. Normal distribution (Or Gaussian distribution), in Excel the function used is called Norm.DIST and it will return a pmf (Probability mass function) for a discrete set of values inputted into it. We can plot such values here:

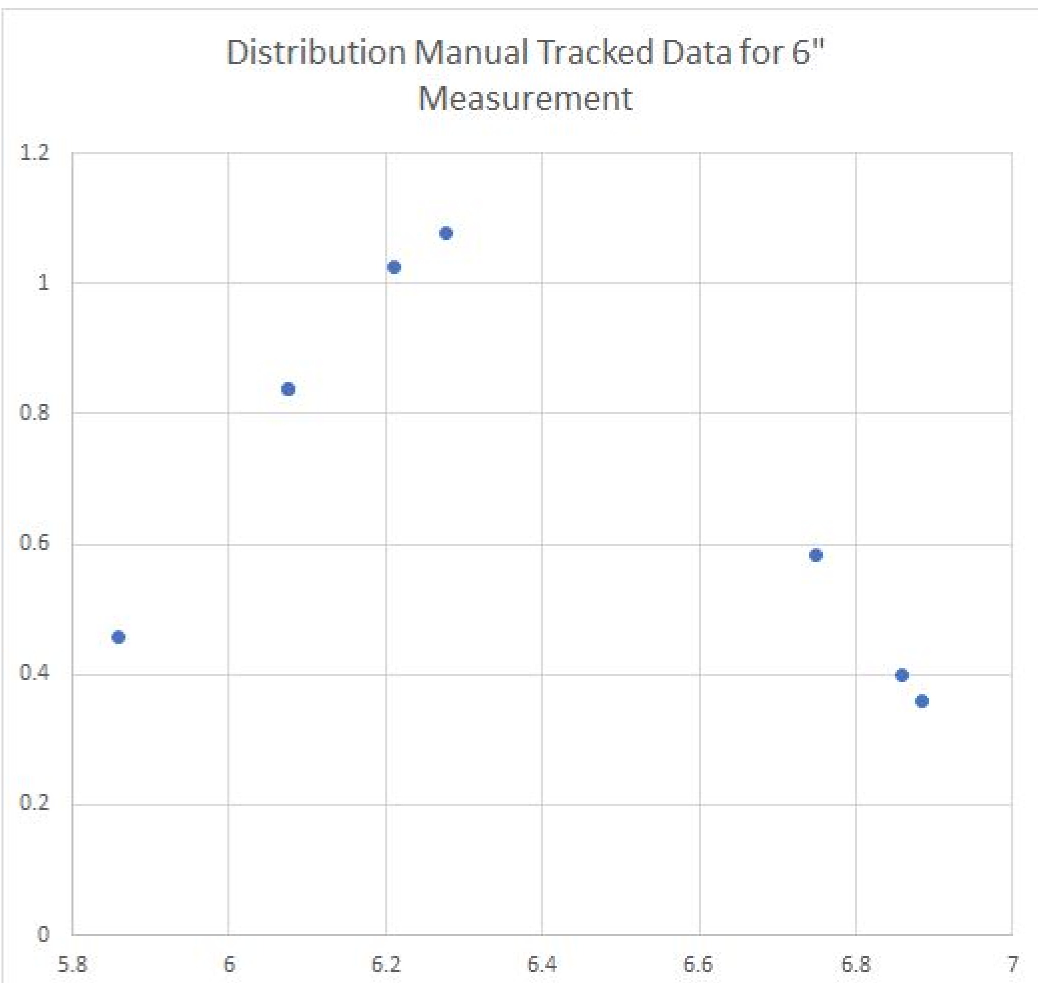


Figure 3. Distribution for 6” Measurements

As seen in Figure 3, the collected data does indeed look gaussian. More manual data points would have allowed for a clearer distribution. The rest of the data is shown as followed:

Actual Measurement (inches)	Echo Pulse (us or ms)	Sensor Measurement Calculated (inches)	Time Position on Oscilloscope (ms)	Distribution
Sensor1				
6	868	5.859	1	0.457470078
6	900	6.075	9	0.839375751
6	900	6.075	19	0.839375751
12	1.74	11.745	8.24	0.909007187
12	1.76	11.88	18.24	2.067862635
12	1.76	11.88	28.24	2.067862635
18	2.64	17.82	10.74	0.269911436
18	2.68	18.09	20.74	1.155420565
18	2.66	17.955	30.74	0.646270746
24	3.52	23.76	26.51	0.70167275
24	3.52	23.76	36.5	0.70167275
24	3.52	23.76	46.48	0.70167275
Sensor2				
6	1	6.75	36.98	0.58232444
6	1.02	6.885	46.98	0.358728937
6	1.016	6.858	56.96	0.399585642
12	1.81	12.2175	37.24	0.8063275
12	1.8	12.15	47.24	1.371022245

12	1.81	12.2175	57.25	0.8063275
18	2.75	18.5625	36.37	0.884702362
18	2.73	18.4275	46.46	1.375650954
18	2.728	18.414	56.36	1.414821469
24	3.56	24.03	39.15	1.536771786
24	3.62	24.435	49.15	0.599845848
24	3.61	24.3675	59.15	0.837030789
Sensor3				
6	920	6.21	58.94	1.026341026
6	900	6.075	68.94	0.839375751
6	930	6.2775	78.94	1.07802612
12	1.76	11.88	69.04	2.067862635
12	1.762	11.8935	79.04	2.1620051
12	1.764	11.907	89.04	2.245004018
18	2.74	18.495	25.34	1.144224315
18	2.73	18.4275	35.24	1.375650954
18	2.73	18.4275	45.24	1.375650954
24	3.58	24.165	82.57	1.489328114
24	3.58	24.165	92.57	1.489328114
24	3.6	24.3	102.57	1.088429943

Table 3. All Manual Tracked Data for 4 difference distances, with three difference sensors

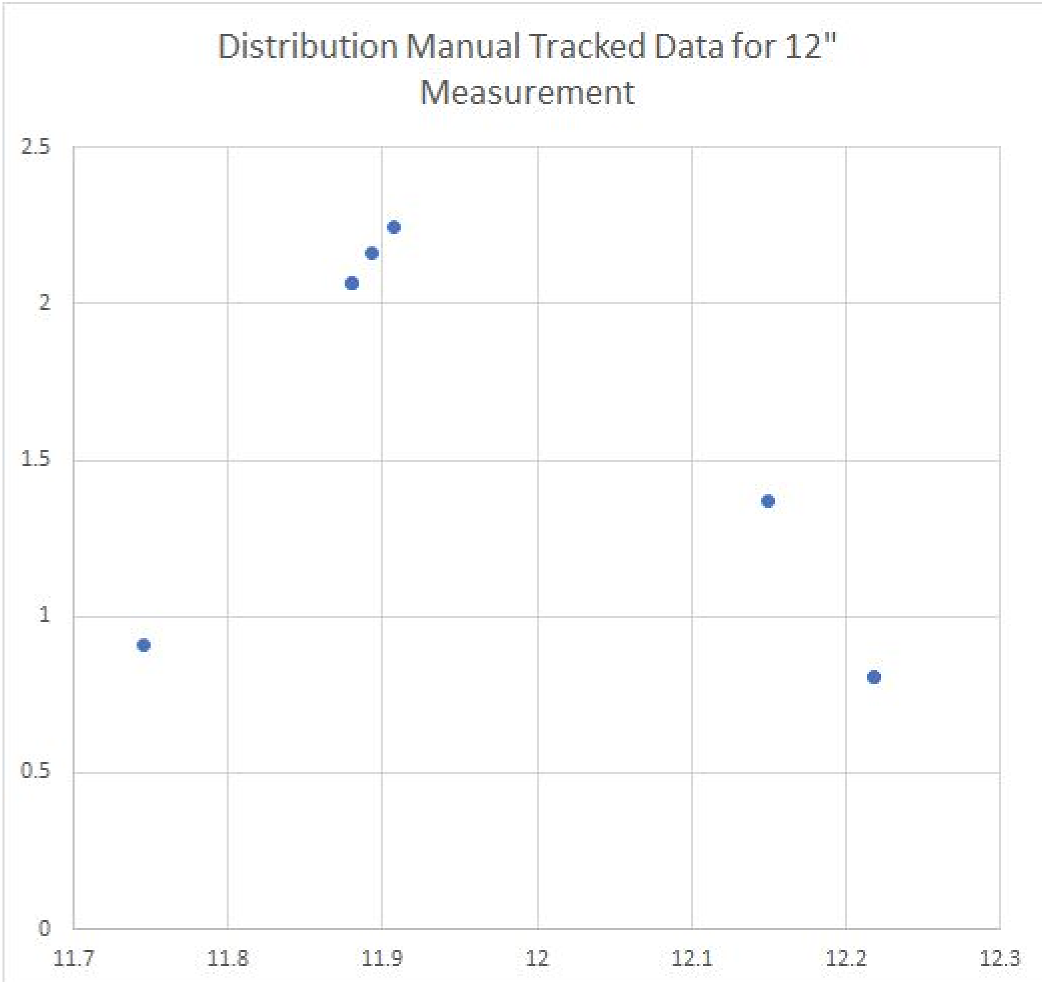


Figure 4. Distribution Graph of 12” Collected Data

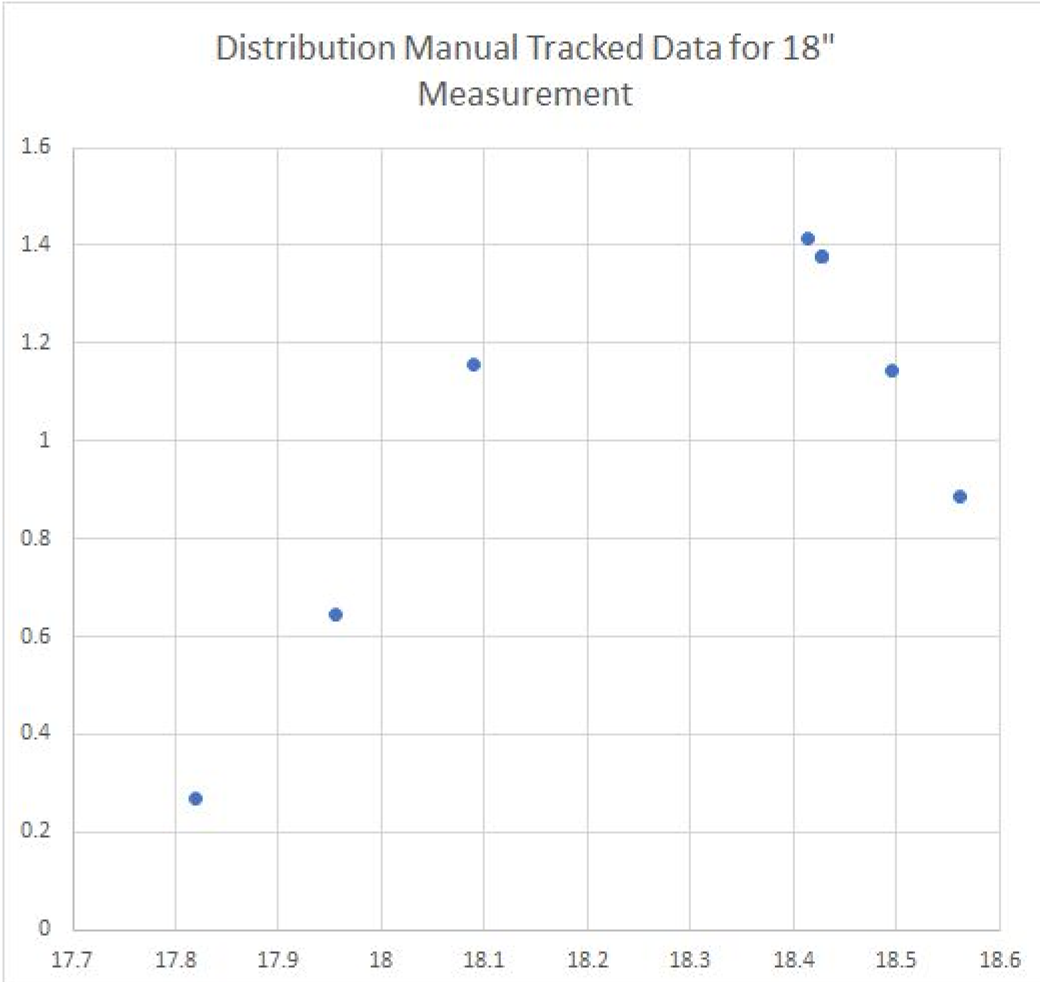


Figure 5. Distribution Graph of 18” Collected Data

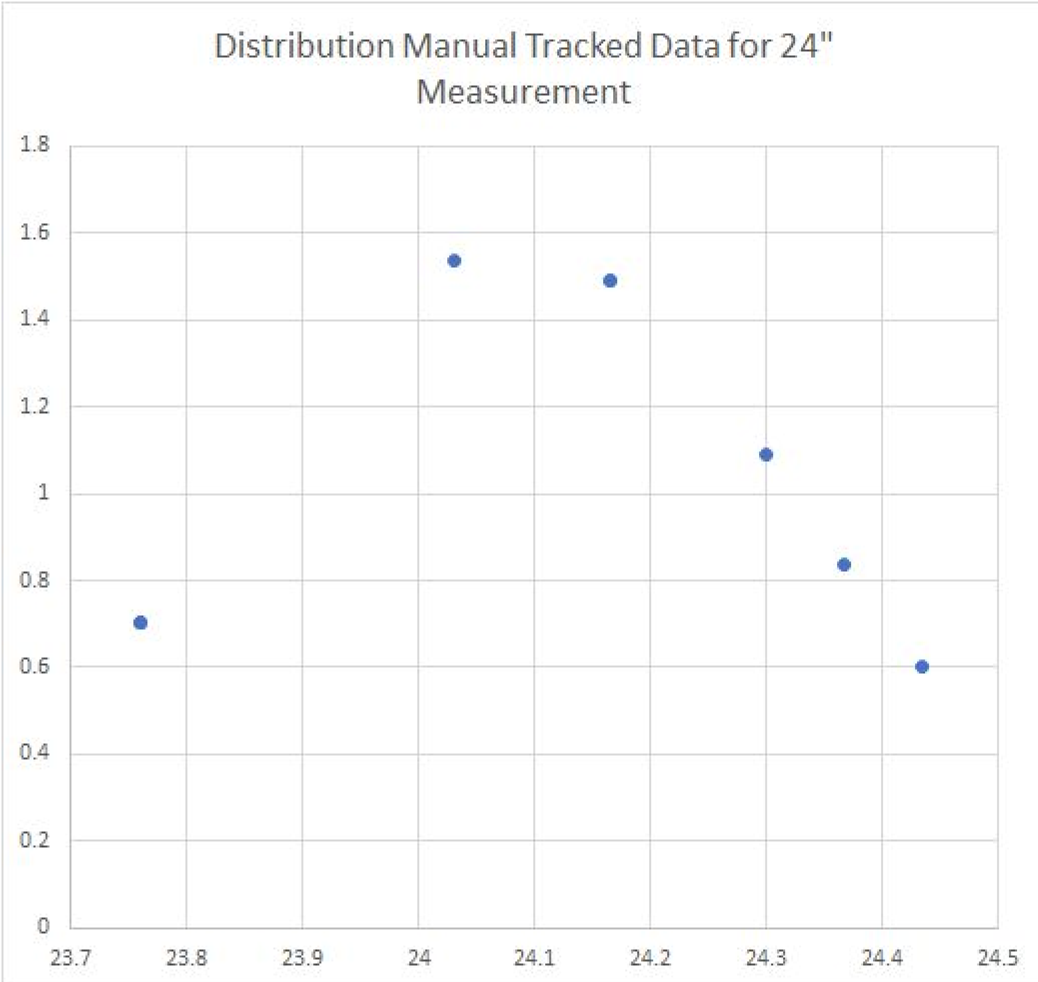


Figure 6. Distribution Graph of 24” Collected Data

To characterize which sensor is underperforming,what we did was calculate the averages within each sensor for each distance, and subtract it from the actual real life distance:

	6" Mean	Difference	12" Mean	Diff	18" Mean	Diff	24" Mean	Diff
Sensor 1	6.003	0.003	11.835	0.165	17.955	0.045	23.76	0.24
Sensor 2	6.831	0.831	12.195	0.195	18.468	0.468	24.2775	0.2775
Sensor 3	6.1875	0.1875	11.8935	0.1065	18.45	0.45	24.21	0.21

Table 4. Performance Analysis of the three sensors

From our analysis, we can see that sensor 2 seems to be underperforming, and it would serve as a good test case for our implementation of the kalman filter. We'll move onto to talk about testing sensor 2 with the Arduino to reproduce this manual collected data.

Part 2: Connecting the Arduino and reviewing results

We programmed an Arduino MEGA to output a 10 us HIGH pulse to trigger the ultrasonic sensor. We used the pulseIn method to track the time of flight of the sensor, and used Serial to output the data to the monitor. We had the Arduino do this up to 20 samples, in which it will alert us that it has completed the run, and we then copy and paste the data into an Excel file. We compare the collected data to the manual tracked data's mean and standard deviation:

	Manual Tracked Data	
	Mean	**Variance**
6"	**6.3405**	**0.364583324**
12"	**11.9745**	**0.163121121**
18"	**18.291**	**0.2497759**
24"	**24.0825**	**0.254116115**

Echo Pulse (us)	**Sensor Measurement (Inches)**
864	**5.832**
912	**6.156**
896	**6.048**
920	**6.21**
876	**5.913**
872	**5.886**
876	**5.913**
900	**6.075**
868	**5.859**
924	**6.237**
896	**6.048**

920	**6.21**
892	**6.021**
900	**6.075**
892	**6.021**
896	**6.048**
892	**6.021**
920	**6.21**
900	**6.075**
892	**6.021**
Mean	**Standard Deviation**
6.04395	**0.117465516**

Table 5. Arduino 6” Sensor Measurements

We can see that for our 6” measurements, the standard deviation is lower from our manual measurements. This could be attributed to possible manual measurement error (We didn’t measure accurately enough with our ruler during the manual trials). Looking at the further distances:

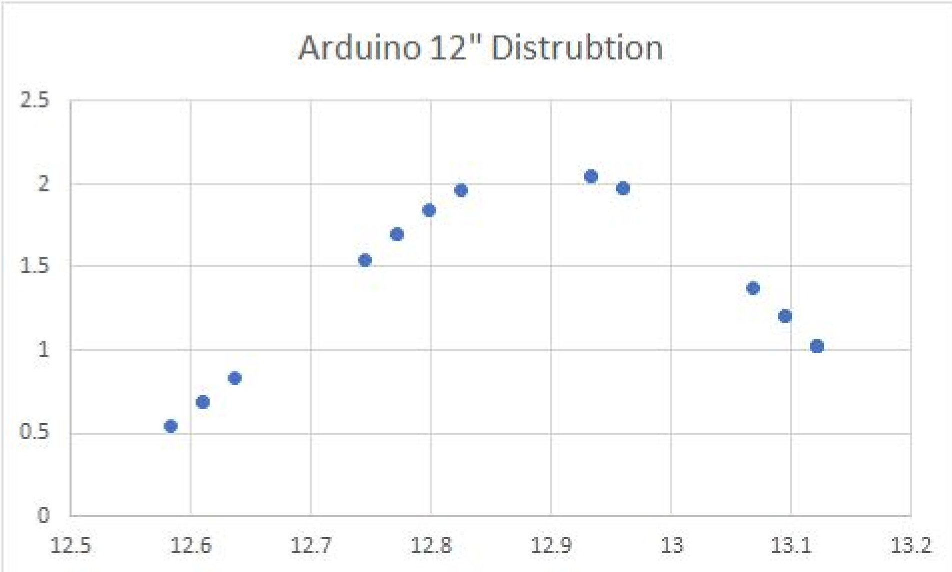
Echo Pulse (us)	Sensor Measurement (Inches)
1896	**12.798**
1944	**13.122**
1892	**12.771**
1900	**12.825**
1916	**12.933**
1868	**12.609**
1944	**13.122**
1944	**13.122**

1892	**12.771**
1920	**12.96**
1916	**12.933**
1920	**12.96**
1936	**13.068**
1944	**13.122**
1872	**12.636**
1940	**13.095**
1888	**12.744**
1940	**13.095**
1868	**12.609**
1864	**12.582**
Mean	**Standard Deviation**
12.89385	**0.190244652**

Table 6. Arduino 12" Measurements

For 12" we can see our sensor has a slightly higher standard deviation than the global standard deviation, but it is closely attenuated to its individual hand tracked data as shown in Table 2.

Picking the 12" data set, we now plot the normal distribution:



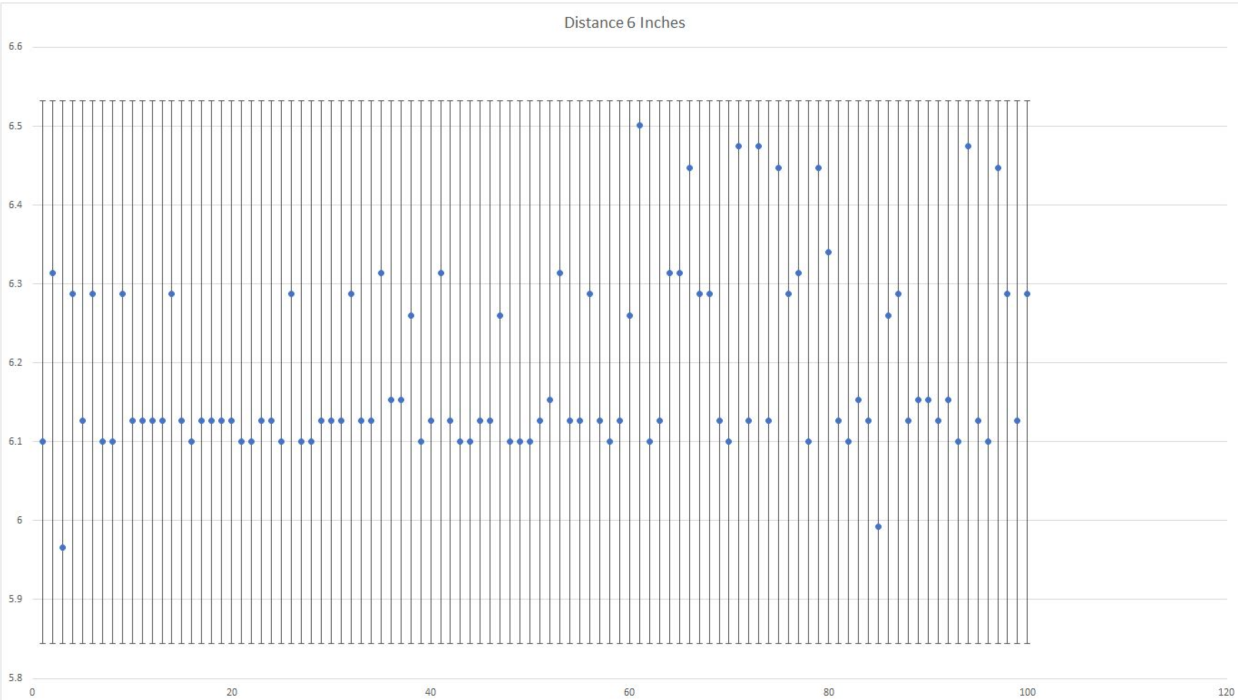
Where our mean is centered around 12.89385, and we can see that the plot is similar to gaussian distribution.

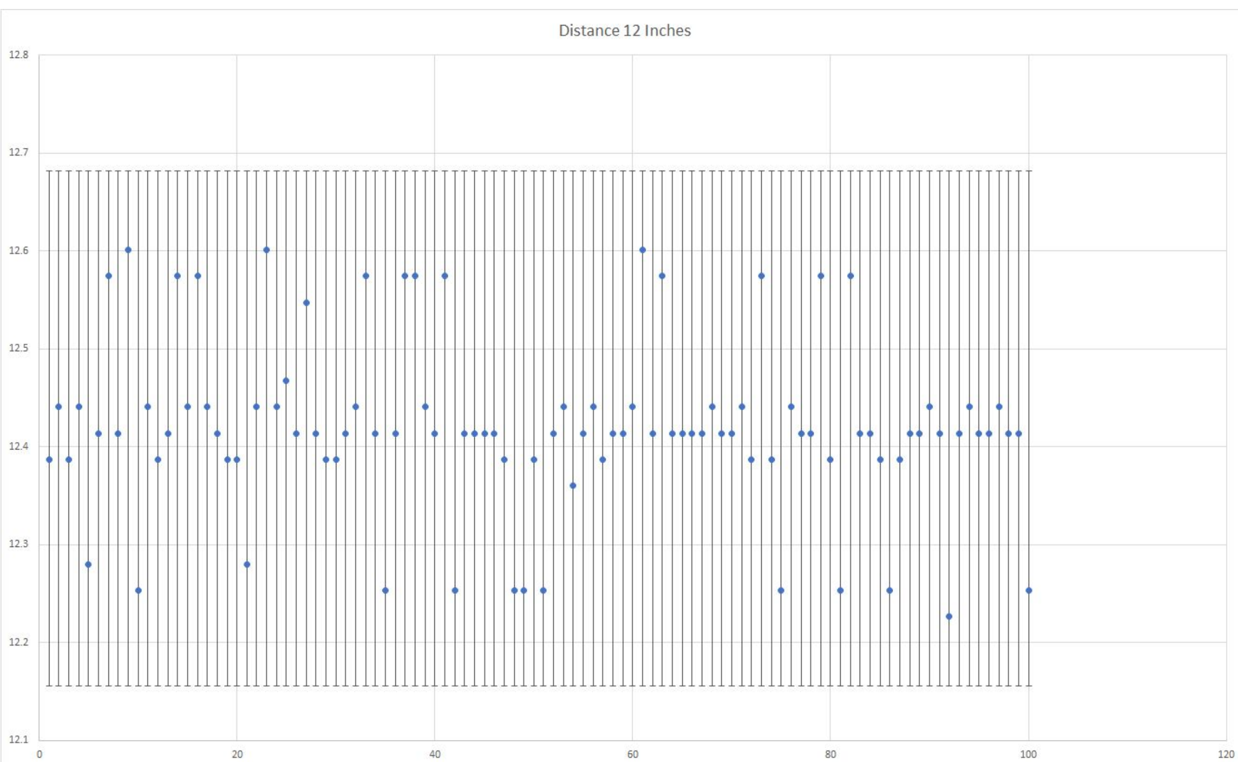
Part 3: Sensor Calibration

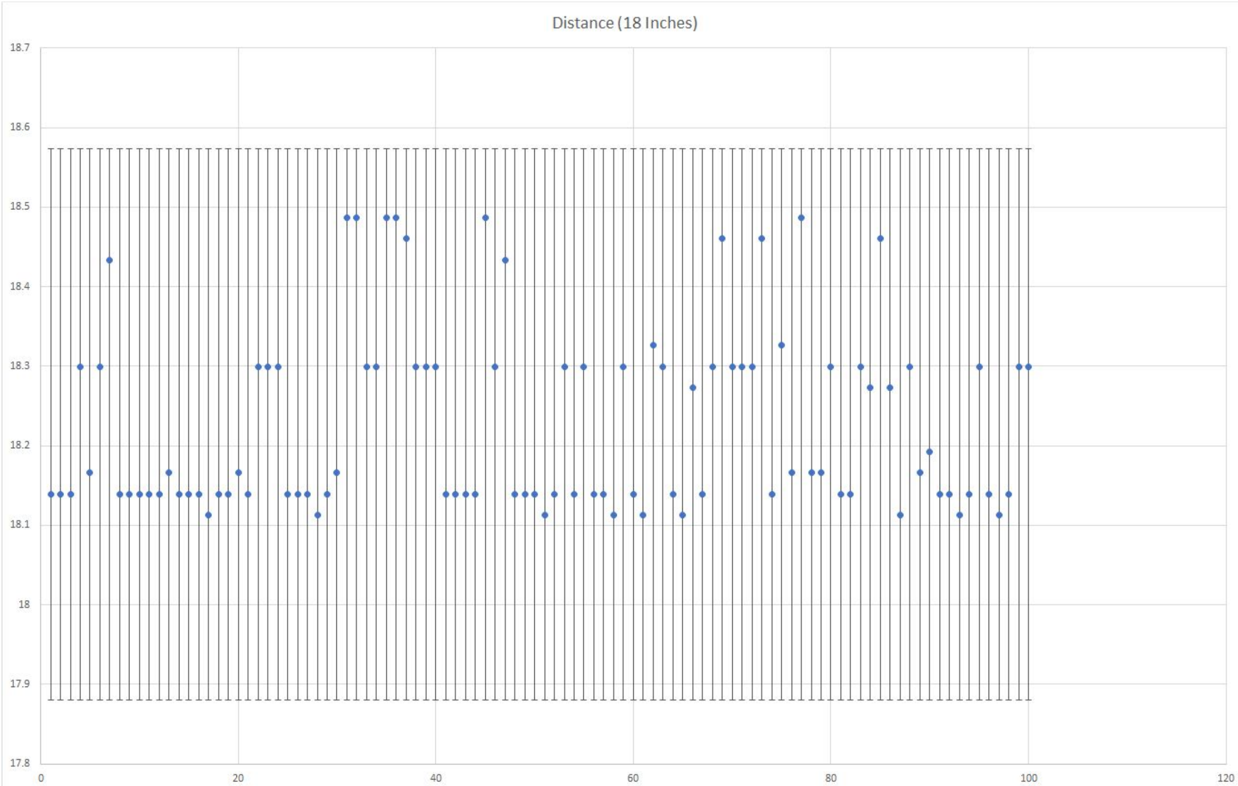
Moving forward from where we left off in the progress report, we tested the 6", 12", 18", and 24" ranges with a new testing environment. In the new testing environment (Figure 7) we used a general ruler to guide the initial ground truth measurement, and this time we followed corroborated the initial ground truth measurement by using a measuring tape and a smaller ruler to make sure the physical measurement is accurate as possible.

Figure 7. Sensor Calibration testing environment

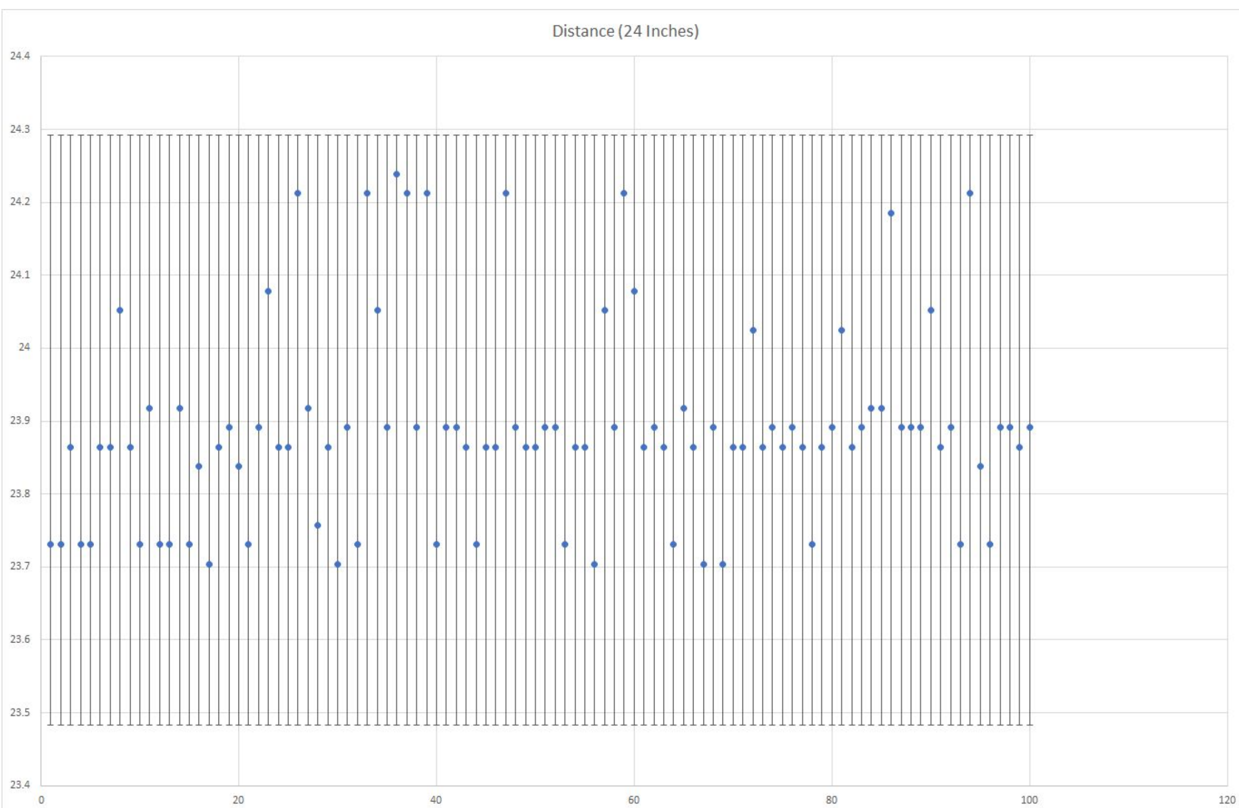
For each distance, we created a logging program that collected 100 data points that will be used for the calibration of the sensor and design of the kalman filter. Plotting with +- 3 standard deviations we observe the data (Due to the scale of these graphs, it may be easier to refer to the Standard-Deviation-Graph.xlsx file included in the online submission)







Distance (18 Inches)
18.7
18.6
18.5
18.4
18.3
18.2
18.1
18
17.9
17.8
0
20
40
60
80
100
120



Distance (24 Inches)
24.4
24.3
24.2
24.1
24
23.9
23.8
23.7
23.6
23.5
23.4
0
20
40
60
80
100
120

From here we are able to calculate the measurement uncertainty required for the kalman filter gain of the filter. Where the measurement noise is defined as the variance between the measured data and ground truth. We move to Matlab to develop and run this data through the kalman filter algorithm.

Part 4: The Kalman Filter

In this test scenario, we started the sensor as close to the board as possible, and moved back for about 9 points (Figure 8).

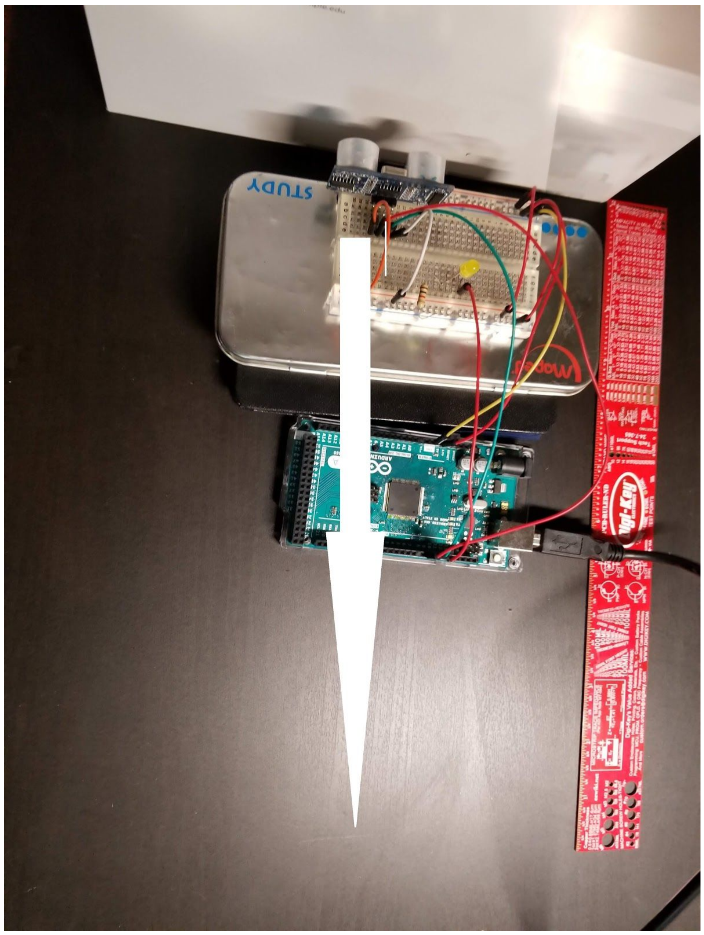
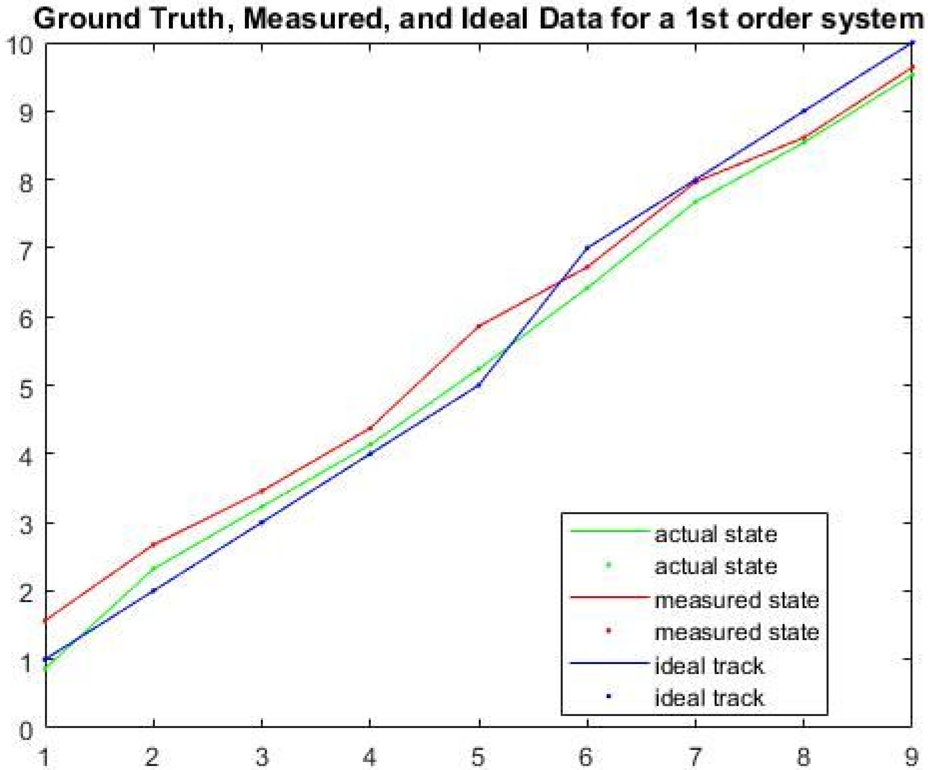


Figure 8. Position Movement Simulation

We moved the marker and induced error by making our ground truth measurements within some standard deviation from what the intended (or Normal tracked) measurement was supposed to be. We summate the following observations in the table:

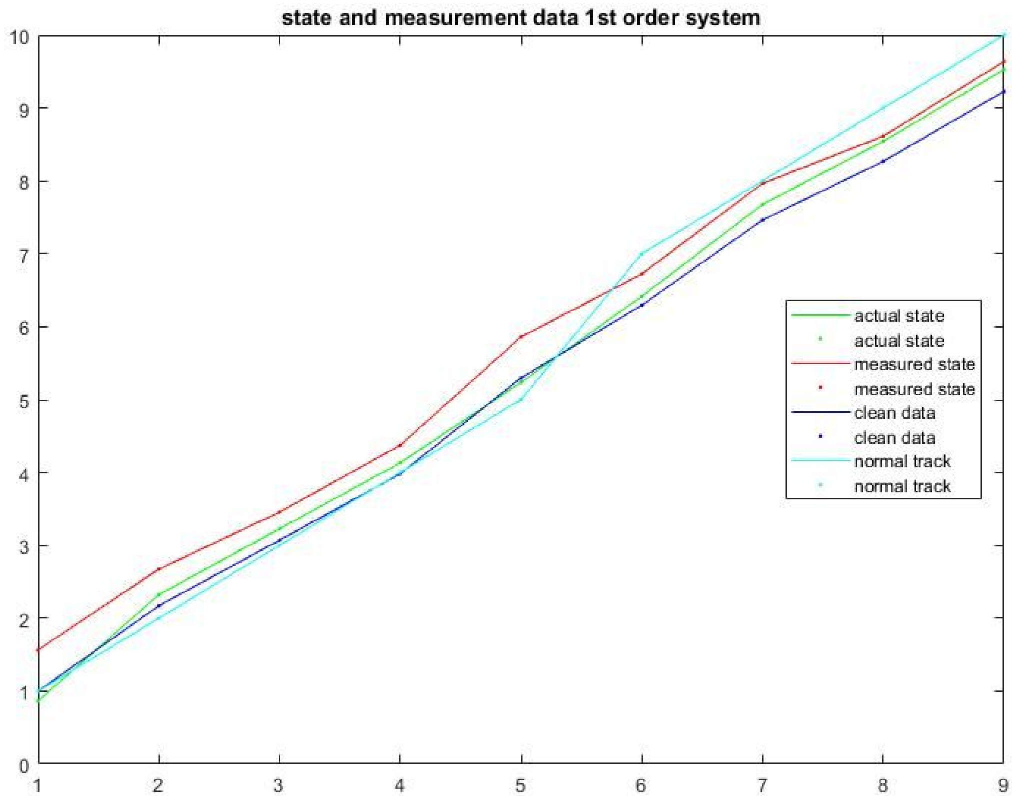
Intended/Normal Track	Ground Truth (measured) inches	Duration (us)	Sensor Measurement (Inches)
1	0.866	232	1.566
2	2.322	396	2.673
3	3.228	512	3.456
4	4.133	648	4.374
5	5.236	868	5.859
7	6.417	996	6.723
8	7.677	1180	7.965
9	8.543	1276	8.613
10	9.527	1428	9.639

Moving into Matlab, we plot the intended, ground truth, and measured data to get an initial idea of how the variance between the data sets are:



From MATLAB we calculate the variances between each of these data sets as follows:
Variance between Ground Truth and Measured = 0.0448
Variance between Ideal Track and Measured = 0.2287
Variance between Ground Truth and Ideal Track = 0.1251

We can see that the variance between the Ground truth and measured is lower than ideal track and measured. With the induced error this will give a better scenario to see if we tuned the Kalman filter to estimate this sort of offset.



From the figure above, after running the measured data through the kalman filter, we plot the clean data and take a look at the variances between the measured data and the ideal data: Variance between clean data and ideal (normal) data = 0.1861

Which is lower than the original variance. This shows that the Kalman filter estimated data is better than the original measured data. The KF gain in this scenario could be recalculated to provide a much more accurate estimate with better sensor calibrations and more sample points.