Project 1 Report: Sensing Signal Processing

1. Signal Processing for One Ultrasonic Sensor

1.1 Problem Statement

In problem 1.1 we are tasked with utilizing a signal ultrasonic sensor to sample, filter and report distances from the sensor front to a fixed object. The primary objective of the sensing system is to report as accurate a distance from an object as possible, with a secondary objective of having the minimum time cost possible in which to read the measurement. There are also a few additional system constraints including: a requirement to flash a light and fire a buzzer at the end of a test run, continuously printing the measured distance until the system has completed its task, printing the time cost once the system has "completed" its measurement.

1.2 Technical Approach

In order to address this task, we model our system in a state space model. The ultrasonic sensor directly measures the time that it takes a sound pulse to travel from the sensor's emitter, be bounced off the object, and trigger the sensor's detector. We need to establish a relationship between the time that it takes for this process to take place, and the distance away from the sensor an object is. Since our system has no control input, only states, our system is described by equations 1, where k is the discrete time state, y is the system state, z is the system measurement, and v is measurement noise.

$$y_k = Ay_{k-1} \qquad \textit{Eqn. 1}$$
$$z_k = Hy_k + v_k \qquad \textit{Eqn. 2}$$

In our case, since the state we are interested in is the objects position, and we are measuring this position directly, our *A* and *B* matrices are each equal to 1. This model is then used to construct a Kalman filter in order to filter the values from the sensor as it attempts to take a measurement. The Kalman filter is structured iteratively with a prediction first being made on the value of the state and the state error covariance. The Kalman gain is then calculated, the measurement is updated, and the error covariance is updated. In our process, the relatively simple system state model simplifies the Kalman filter equations to the following, where *p* represents the error covariance, *k* the Kalman gain, *m* is the measured value and *r* the standard deviation of the measurement noise distribution.

$$\widehat{y_k}^- = y_{k-1} \qquad \textit{Eqn. 3}$$
$$p_k^- = p_{k-1} \qquad \textit{Eqn. 4}$$
$$K = \frac{p_k^-}{p_k^- + r} \qquad \textit{Eqn. 5}$$
$$y = \widehat{y_k}^- + K * (m - \widehat{y_k}^-) \qquad \textit{Eqn. 6}$$
$$p_k = (1 - K) * p_k^- \qquad \textit{Eqn. 7}$$

This filter is used to directly filter the time measurement from the sensor in order to stabilize it, meaning that during the calibration process, it is important to remember that *r* is the standard deviation of the duration distribution rather than that of the measurement distribution. Once implemented, this filter will stabilize the measurement values until the error covariance falls below a target value.

After the filter returns a duration it must be passed through a calibration curve in order to converted into a distance. In order to calibrate the systems, a series of measurements must be taken at a variety of distances around the desired operating range. These values are graphed such that a relationship can be identified between them and a curve assigned. This curve has the pulse time duration as and input and returns the distance.

1.3 Hardware and Software Implementation

In order to complete the required task, we first construct a circuit. The simple circuit consisted of an Arduino uno device, a bread board and an ultrasonic sensor. A visual circuit representation is pictured in figure 1.

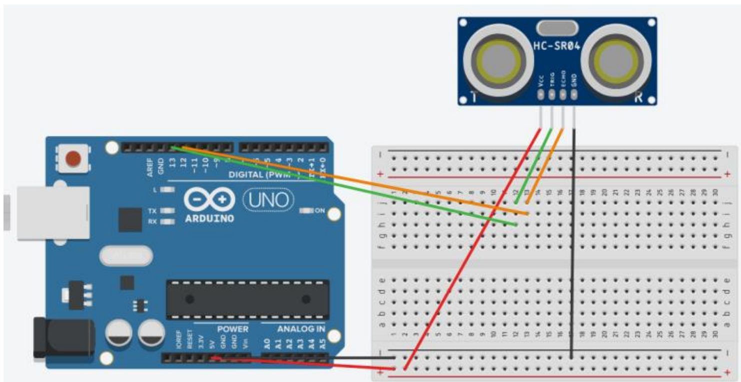

Figure 1. Single sensor circuit configuration.

In order to calibrate the sensor measurements are taken at 16 different distances, from 10 cm to 160 cm away from the object. The actual distance is evaluated by a standard measuring tape and around 500 duration measurements are taken at each point. These measurements at each distance are averaged and the corresponding distance, average duration points are plotted in order to visually identify a curve. In our case the curve appeared linear, which was supported by a high linear R^2 value of 1.0000 and our intuition that since the speed of sound should be relatively constant, duration should increase linearly as distance increases.

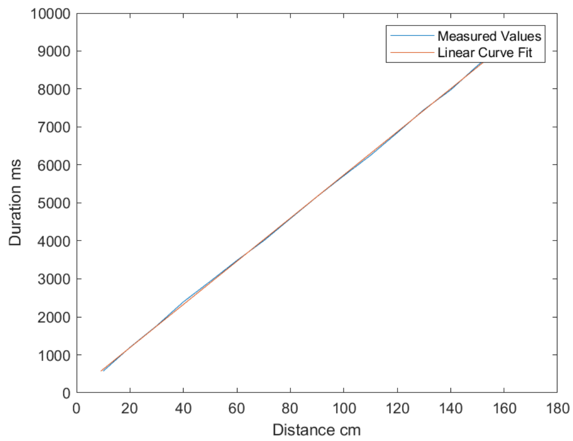

Figure 2. Duration vs Distance for a variety of distances from a test object.

In order to determine an *r* value for our Kalman filter, we also plotted the variance at each point in order to see if a curve existed.

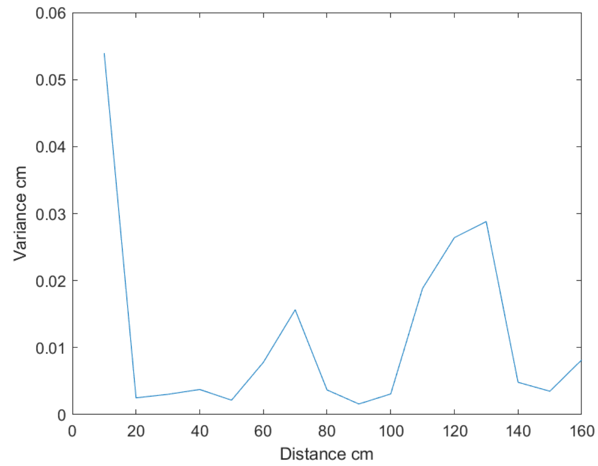

Figure 3. Variance vs Distance for a variety of distance measurements.

Our variance plot showed no obvious pattern so we will use the average variance across all distances for our *r* value. While investigating the setting of our R value we also found some samples with very stepped input patterns. We suspect that this pattern is caused a limitation of the input of from the pulse-in command. This only happened when the sensor appeared to be at a distance that put it between two measurable values, and the input would jitter between them without settling in the middle.

1.4 Experimental Results

The single sensor configuration was tested both in the test day and independently at 3 separate distances in order to gain insight into its success in the following two critical areas: time cost and error. For three random measurements (chosen such that they do not lie directly at calibrated points) the following data was achieved:

Final Value (cm)	**27.8**	**45.6**	**70.3**
Actual Value (cm)	**28.3**	**45.8**	**70.8**
Error (cm)	**-0.5**	**-0.2**	**-0.5**
Time Cost (ms)	**4526**	**4526**	**4512**

Table 1. Data from independent test.

These 3 trials demonstrate that the system has very consistent time cost, even at varying distances. Additionally, due to all three lengths being shorter than the actual length, it is likely that the calibration was slightly offset and could be optimized further to better improve accuracy.

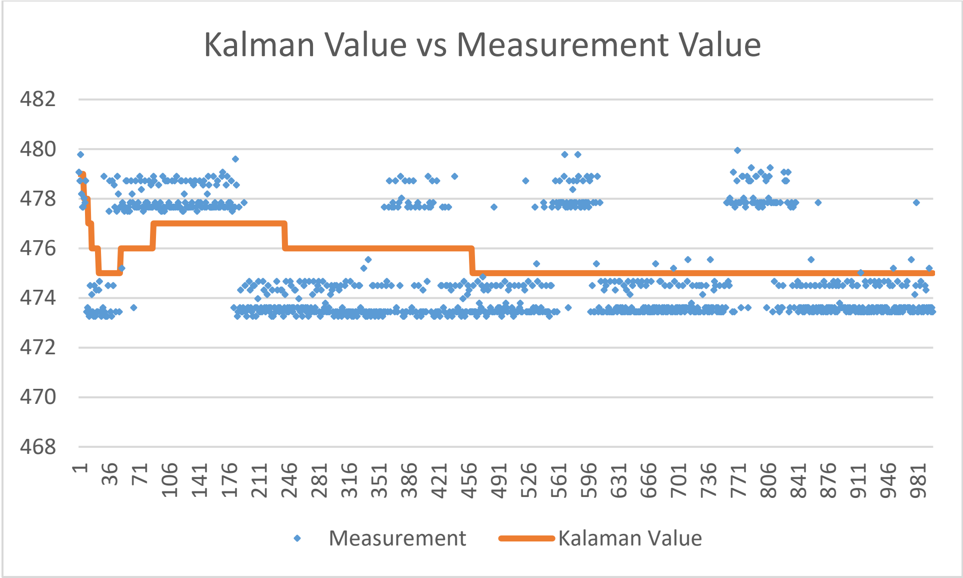

Figure 4. Sample dataset.

The above dataset is the output of the Kalman filter demonstrates the ability of the Kalman filter to quickly settle the measurement into a precise value. This dataset also demonstrates a characteristic seen in many samples, a non-continuous input distribution.

2. Sensor Fusion of Multiple Ultrasonic Sensors

2.1 Problem Statement

In problem 1.1 we are tasked with utilizing a combination of multiple ultrasonic sensor to sample, filter and report distances from the sensor front to a fixed object. The primary objective of the sensing system is to report as accurate a distance from an object as possible, with a secondary objective of having the minimum time cost possible in which to read the measurement. There are also a few additional system constraints including: a requirement to flash a light and fire a buzzer at the end of a test run, continuously printing the measured distance until the system has completed its task, printing the time cost once the system has "completed" its measurement.

2.2 Technical Approach

A large portion of our approach to this task is the same as that in in task 1. There are a few minor changes that must be conducted such that the data from multiple sensors can be combined. Our system state model remains the same as in task 1, however, since each of the sensors may have a slightly different calibration curve, in order to combine them, we must instead use a Kalman filter to filter the distance measurements instead of the duration measurements. This way the output of each sensor can be directly compared, unlike the duration output. In order to combine the measurements of multiple sensors, for each measurement taken, an individual correction step is conducted. For an initial measurement to be input into the model, a single measurement is taken from one of the sensors to serve as the initial state value. Due to nature of the filter, as long as the calibration has been conducted carefully it should not make a difference which sensor the original measurement is taken from because it will eventually settle between them (assuming some difference in calibrated sensor output).

2.3 Hardware and Software Implementation

It is important that the position of the two sensors relative to each other remains constant such that the calibration curves of the sensors are as accurate as possible. We securely tape our sensors to the breakout board and both the breakout board and Arduino to a piece of cardboard such that they are as stable as possible during each test run. We decided to use two sensors because of the potential difficulty of maintaining consistent mountings since our breadboard did not fit more than two sensors in parallel. This additionally had the added benefit of reducing the cycle time since each additional measurement requires not only processing time but also time for the sound wave to actually travel from the sensor to the object and back. In the case of an object at near the end of the range of the sensors, this can take nearly 10000 μs from the time the command is sent to the sensor to fire the pulse to the time it has returned the duration value. This leads to a greater time cost.

2.4 Experimental Results

Again, the setups performance was both evaluated in the final test in the class room and in a set of three random distances. The performance for these random distances is as follows:

Final Value(mm)	**36.87**	**58.54**	**72.31**
Actual Value(mm)	**37.1**	**58.5**	**72.1**
Error(mm)	**-0.23**	**0.04**	**0.21**
Time Cost	**2896**	**2902**	**2905**

Similar to the one sensor configuration, this sensor had very good accuracy, however, unlike the one sensor configuration the errors were not uniformly below actual value, indicating that the likely better calibration of the second sensor (or inversely bad calibration) yielded a better result. This configuration's time cost did increase with its distance from the object, indicating that there were likely more measurement inconstancies at greater distances.

3. Conclusions and Discussions

3.1 Conclusions (a summary the results of different approaches)

In summary, the results of both sensor configurations were very consistent, both with each other when compared side by side and at a variety of different measurements. Both systems were useful for understanding the effect of a Kalman filter and how it can be used to process data from sensors in order to turn it into a useful measurement.

These systems both performed the task as stated. The two-sensor system was useful for decreasing the overall time cost of the system but did not seem to significantly increase accuracy.

3.2 Discussions (a comparison of different approaches and potential future work to further improve each approach)

Each of these two approaches was highly accurate, with no obvious differences between their accuracy, however, the addition of a secondary sensor increased the speed at which the measurement converged significantly. Additionally, in both the one and two sensor configurations, the Kalman filter was able to interpret a quick hand wave in front of the sensor array as measurement noise and recover.

In future work it would be interesting to design a loop that allows the system to function if each sensor is receiving wildly different values. Like a car backup system, a multi-sensor array could be used to more precisely and quickly measure distance when multiple sensors operate in parallel or switch to a series operation and report one or multiple distances read if needed.

Additional work could also be conducted to maximize the relationship between number of sensors and convergence time. Each sensor added adds additional readout time so it is likely there is a point at which a theoretical system is balanced such that adding or removing an a sensor increases time cost and decreases accuracy (an optimum).

Another interesting task could be to force the system into an emergency backup mode if one of the sensors appears to fail, and to notify the user through a unique tone through the speakers.

Notice: The data in an individual report should be recorded individually. Students in one group can use the same hardware and software but cannot use the same data in the report.