Multiple Model Adaptive Estimation Kalman Filtering for Position Estimation

Jakob Frabosilio, Sam Hudson, Joseph Lyons

1. Introduction

Various Kalman Filter (KF) types and their applications were covered in [1]. This project will develop a practical application of a Multiple Model Adaptive Estimation Kalman Filter (MMAEKF). The MMAEKF will be applied to estimate the position of a linear carriage. Measurements of the carriage are taken using an ultrasonic sensor, accelerometer, and encoder to provide distance, acceleration, and actual (or close to actual) position measurements, respectively. The data is processed by an ESP32 microcontroller, which applies a MMAEKF to estimate the carriage position. The estimated positions from the ultrasonic sensor, accelerometer, and encoder are then compared. A real life application could be tracking an object which cannot easily be measured. With the limited measurements (ultrasound and accelerometer), the position can then be estimated, which may then be used in a controller.

This report is organized as follows: Section 2 provides the model of the system. Section 3 describes the theoretical background of the filter and the implementation of the system. Section 4 presents the results of the system evaluation, including an analysis of the accuracy and reliability of the different position estimation techniques. Section 5 compares the measurement technique to classical methods. Finally, Section 6 concludes the report and provides recommendations for future work.

2. System Model

2.1. State-space representation

The primary goal of this project is to improve the performance of two linear position tracking methods: accelerometer data integration and acoustic transmitter-receiver triangulation. The performance is verified using the position measured from an encoder on the motor. The two measurements are filtered using individual Kalman Filters, and also a combined state-space particle model as described below. Lastly, the MMAEKF combines all three filters (the two individual filters and the combined filter) using a selection algorithm.

This project uses two forms of KFs for signal processing. Firstly, one-dimensional KFs are used to individually filter the accelerometer and acoustic sensor data. Secondly, a combined measurement model (using both the accelerometer and acoustic sensor data) is used. To do this, a state-space system model is used to describe the system. The system uses a state-space representation of the equations of motion to model the platform as a particle:

$$\begin{bmatrix} \mathrm{x} \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \mathrm{dt} & \frac{1}{2}dt^2 \\ 0 & 1 & \mathrm{dt} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \sim N(0, Q)$$

$$\begin{bmatrix} x \\ \ddot{x} \end{bmatrix}_{k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sim N(0, \sigma_{acoustic}^2) & 0 \\ 0 & \sim N(0, \sigma_{IMU}^2) \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot 0.1$$

Where x is the position measured via acoustic triangulation and $\ddot{x}$ is the acceleration measured via the IMU.

2.2. MATLAB Simulation

The above system was modeled in MATLAB/Simulink with simulated random Gaussian noise inputs for position and acceleration. Figure 1 shows the simulation output, plotting a desired input position, simulated position data from acoustic sensors, and the estimated position from our Kalman Filter. Note that the KF data is able to converge close to the true position as time increases.

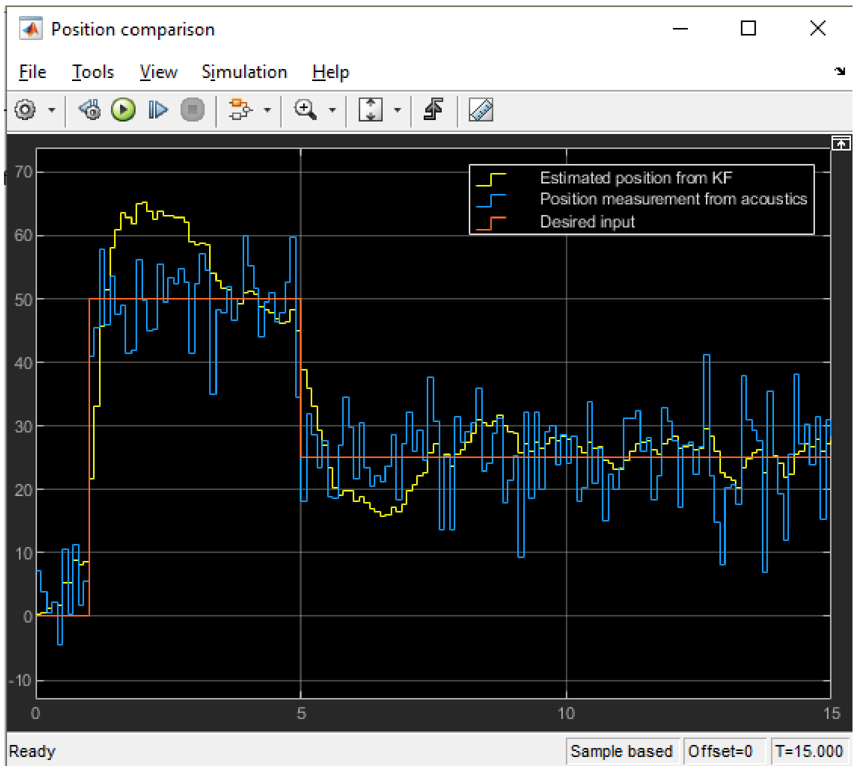

Figure 1: MATLAB Simulation of Kalman Filter

Additionally, simulated accelerometer data can be seen in Figure 2, but that data does not significantly impact this system as there is no true acceleration in the simulation (step responses are instantaneous). Also, the accelerometer data has an offset of 100 mm/s^2, mimicking that of our real system. In our real system, the accelerometer/IMU used was not properly calibrated; we only read the acceleration values from one of the axes and ignored all other data. If we had more time, we would have made a proper KF for the IMU itself to provide more accurate acceleration/position data. Since we did not do this, we had an offset of -400 to 400 mm/s^2 of acceleration at any given point due to the IMU not being completely level (and vibrating/changing orientation when moving).

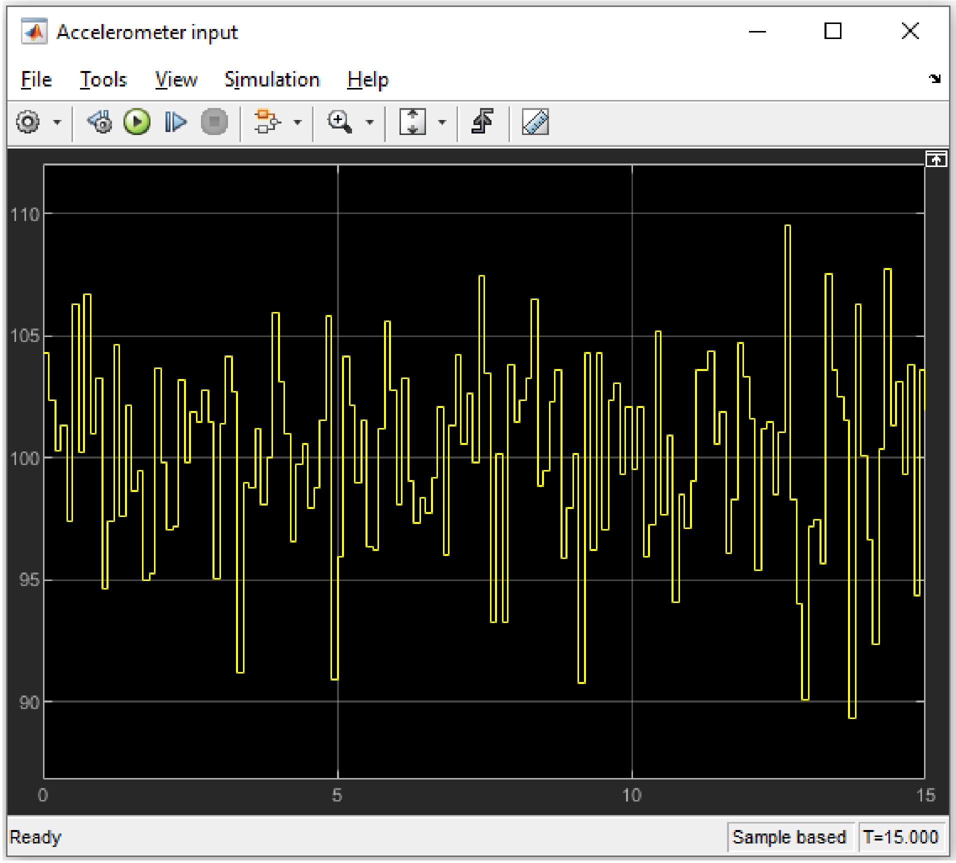

Figure 2: Raw Accelerometer Measurements

Figure 3 shows the Simulink block diagram used to simulate our KF. Note that a Kalman Filter subsystem does the majority of the processing; this Simulink file can be found in the GitHub repository at the end of this report.

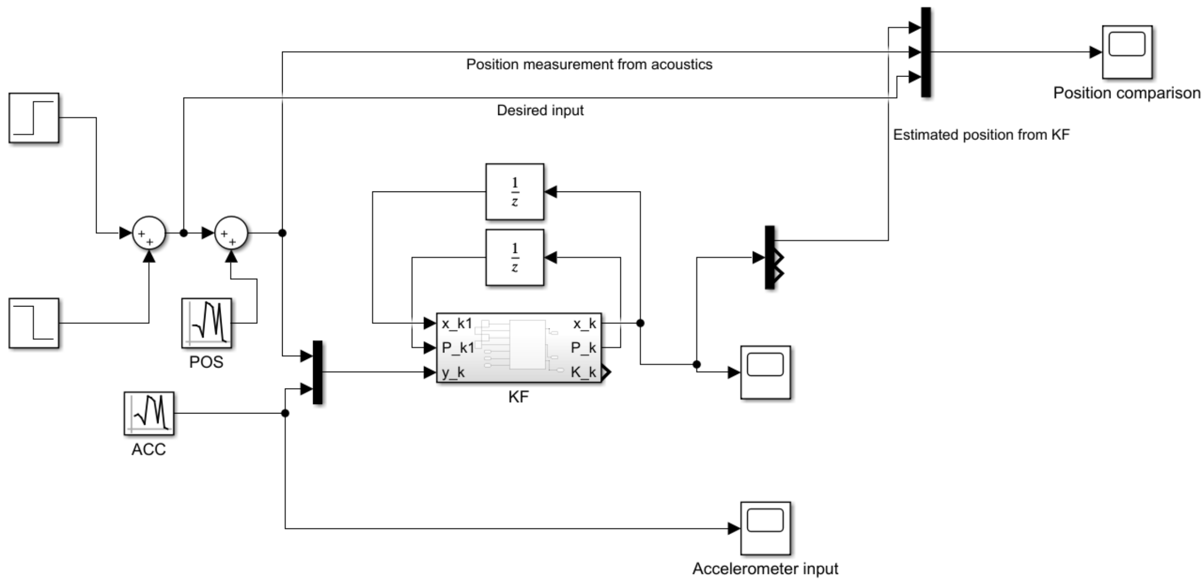

Figure 3: Kalman Filter Simulation Block Diagram

3. Multiple Model Adaptive Estimation

3.1. Dynamic theory and background

Multiple Model Adaptive Estimation Kalman Filter (MMAEKF) is a technique used to estimate the state of a dynamic system by combining estimations from multiple Kalman filter models. Each KF model represents a possible dynamic model of the system, and the MMAE algorithm selects the model that produces the most accurate estimate of the system's state based on the available sensor measurements.

MMAEKF is a powerful technique that can improve the accuracy of state estimation in systems with changing dynamics or unknown model parameters. By using multiple KF models in parallel, MMAEKF can adapt to changes in system dynamics and improve estimation accuracy by selecting the model that best represents the current state of the system.

Many MMAEKF selection algorithms exist. To show the implementation of the MMAEKF, the simplest form of selection algorithm was chosen for this project, which is a weighted selection algorithm. This MMAEKF algorithm works by weighting estimations based on the residual of each model (the absolute difference between the current measurement and the previous estimation of the measurement, at each time step). The MMAEKF selection algorithm used in the system is defined in equations (1), (2), and (3):

$$r_{KFn} = \left| y - \hat{y}^{-} \right| \qquad (1)$$

Where r_{KFn} is the residual for the n-th KF previous estimation, y is the measured state, and $\hat{y}^{-}$ is the previous estimation for the current measured state.

$$c_{KFn} = \frac{1}{m-1}\left(1 - \frac{r_{KFn}}{r_{KF1} + r_{KF2} + ... + r_{KFm}}\right) \qquad (2)$$

Where c_{KFn} is the weight coefficient for the n-th KF estimation.

$$\hat{x}_{MMAE} = (c_{KF1}\hat{x}_{KF1})(c_{KF2}\hat{x}_{KF2}) \dots (c_{KFm}\hat{x}_{KFm}) \qquad (3)$$

Where $\hat{x}_{MMAE}$ is the MMAEKF estimation output, and m is the total number of KFs used in the system.

It is important to note that in a practical application, x is not accessible, and so this selection algorithm is not practically useful. However, for the sake of demonstration, the system uses the encoder measurement as the 'actual' state reference point, to be able to more simply show the impact of having multiple KFs working in conjunction. A selection algorithm that does not use a direct state measurement is presented in [3].

3.2. Stability, Controllability, and Observability

The KFs used in this project are for processing of measurements to reduce noise, and therefore are not used in a state space controller. The stability, controllability, and observability of the plant cannot be investigated. However, in addition to the KFs used for MMAE, a separate full-state controller was developed to control the DC motor in positioning the linear carriage, based on a user's given input. The user input is controlled through a sliding potentiometer, and is mapped to replicate the position of the carriage. The derivation of the DC motor model for the system can be found in [2].

3.3. Application

In the research paper referenced for this project, several types of KFs and their applications are discussed. The MMAEKF control method is applied to a linear carriage system, which is displayed in Figure 4. The intention with the system is to attempt to accurately estimate the position of the system using an ultrasonic sensor (for the position) and an accelerometer (for the acceleration) with multiple KF models. Using MMAE, the filtered measurements from the KF models were then weighted according to their perceived accuracy. The position gathered from the encoder is presented to compare the MMAEKF estimation to the actual position. From empirical measurements, the acoustic sensor measurements have lower variance than the accelerometer measurements, and are hence weighted more heavily by the MMAE algorithm.

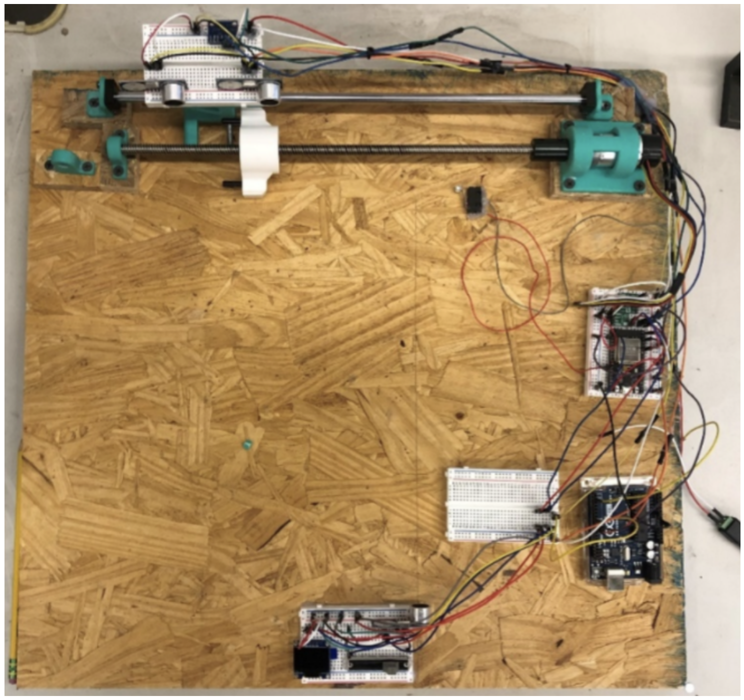

Figure 4: Linear Carriage System

3.4. Controller Design

Using the MMAEKF technique to obtain accurate position estimations, a full-state feedback LQR/LQG controller with set point control (as shown in Figure 5) could be used.

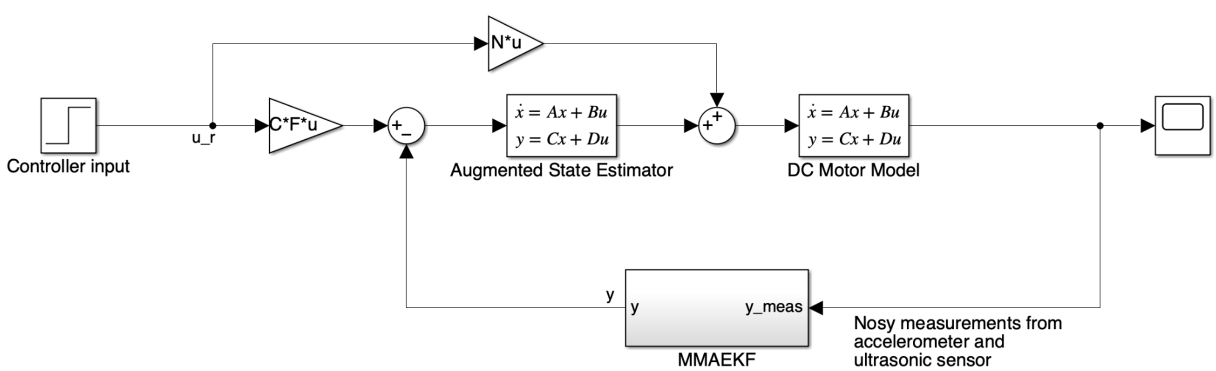

Figure 5: Full-State Feedback LQR/LQG Controller

Where the DC Motor is modeled using the system found in [2], and the signal y is the filtered measured position of the carriage (which is gathered from the MMAEKF technique using acoustic sensor and accelerometer data).

4. System Evaluation

Displaying the raw measurements and estimated positions alongside each other provides a clear visual of the performance of each. An example of the output is shown below in Figure 6. The six values at the bottom of the output are compared against each other, with the top performing model displayed above the nine measurements/estimations. At the top of the output, the best performing model over time is displayed, with how often that model performs the best. MMAE KF Proportions, at the bottom of the output, represents the weighting of the three outputs into the MMAE algorithm, with percentages for Accelerometer KF, Acoustic KF, and Combined KF, represtively. The Accelerometer KF is typically weighted higher, although the Acoustic KF is actually more accurate, as shown in Figure 7. This is because the IMU outputs less noisy data than the acoustics, even though the IMU is less accurate. The IMU is less accurate because only one axis of accelerometer data was measured. To improve the IMU’s accuracy, all three axes and gyrations could be measured and used to more fully understand the acceleration in the desired direction.

```
-----------------------------
           Average best model: Acoustic time-of-flight (with KF)
                Model is best: 53.85% of the time

  Best model for this dataset: Acoustic time-of-flight
        Model percent error: 1.79%

        Desired position (mm): 100
         Actual position (mm): 98.82, DC motor controller percent error: 1.18%
         Accel. position (mm): 155.89, percent error: 58.07%
       Acoustic position (mm): 100.59, percent error: 1.79%
      KF accel. position (mm): 156.02, percent error: 57.93%
    KF acoustic position (mm): 94.37, percent error: 5.05%
    KF combined position (mm): 172.67, percent error: 78.74%
           MMAE position (mm): 128.14, percent error: 28.34%
         MMAE KF proportions: 49.91%, 46.26%, 3.84%
-----------------------------
```

Figure 6: Sample Data Output

Figure 7 displays the performance of each raw measurement and estimation type listed in Figure 3, and compares these to the actual position represented by the dashed line. Evidently, the acoustic positioner performs best. Interestingly, the raw acoustic measurement matches closer to the actual position while the carriage is in motion, while the acoustic KF estimation performs best when the carriage is at rest.

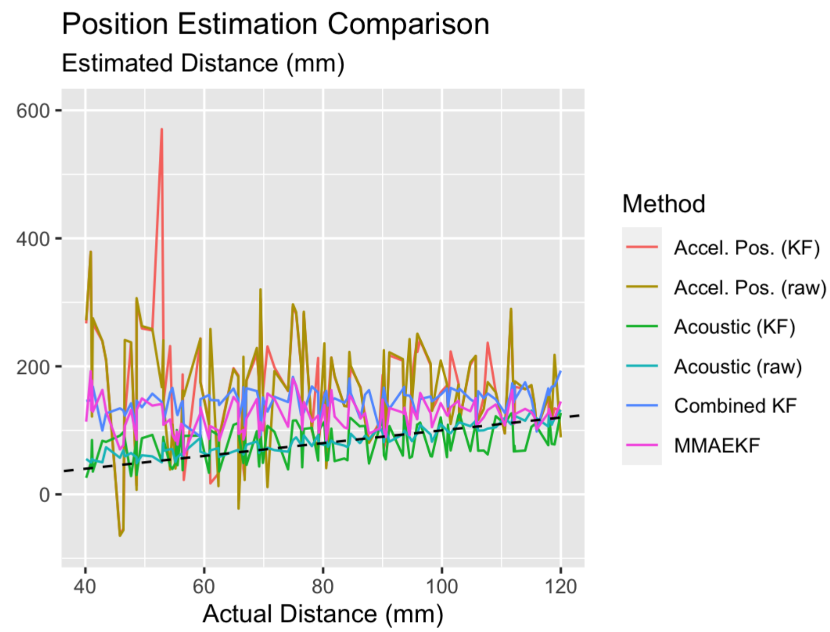

Figure 7: Position Estimation Comparison

Figure 8 displays the proportion of the three filtered estimates used in the MMAE algorithm. As discussed earlier, the accelerometer proportion is most commonly the greatest, averaging ~43% of the MMAE. Fluctuations exist in the proportion of the acoustic and combined KFs, varying greatly from ~48% to >10%.

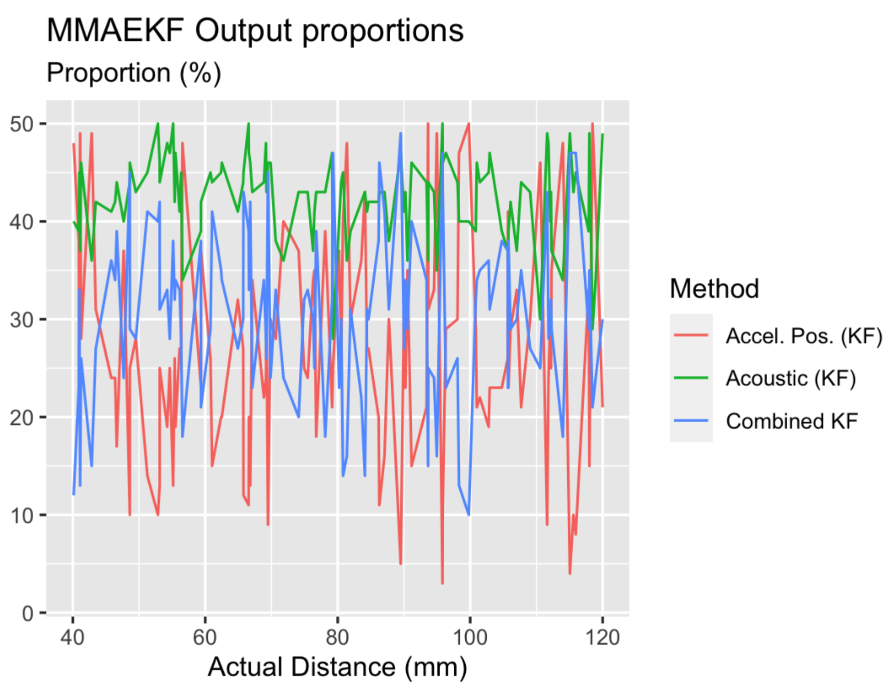

Figure 8: MMAE KF Output Proportions

Figure 9 shows both the raw acoustic data and the KF-filtered acoustic data versus the true position (dashed line). Note that the KF data appears to be less accurate - this is true for this test, which is essentially a ramp input, where the position is changing at a constant rate.

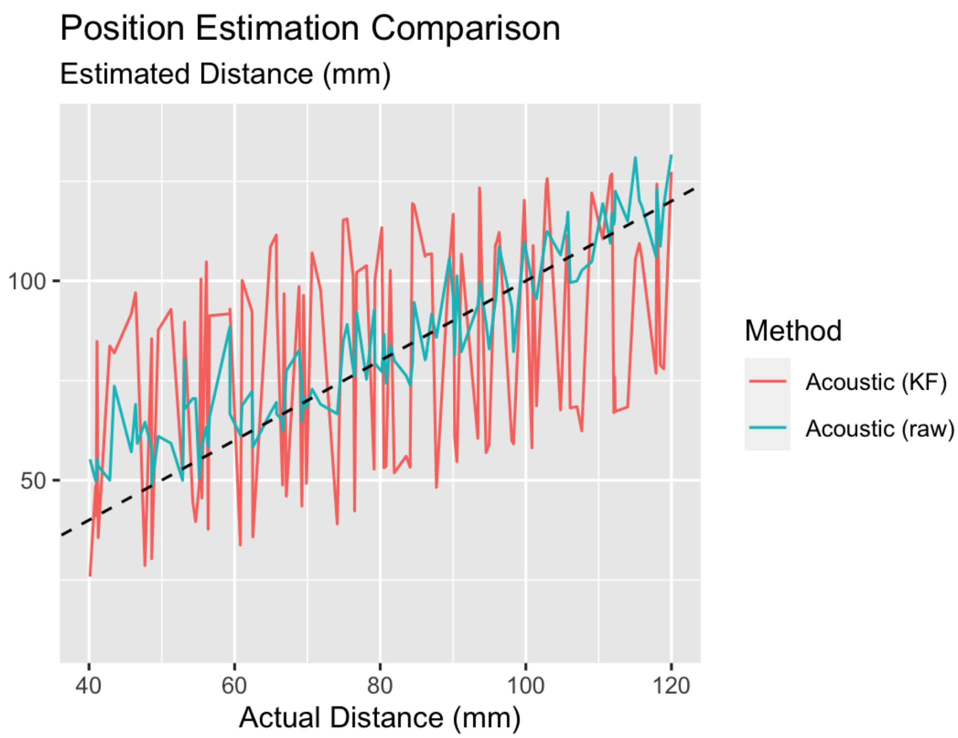

Figure 9: Comparison of Acoustic Sensor Position Measurements

For stationary/step inputs where the positioner stays at a constant location, however, the KF data is much more accurate. This can be seen in Figure 10, where the true position is held constant and the KF data is much more representative of the true position at any given time.

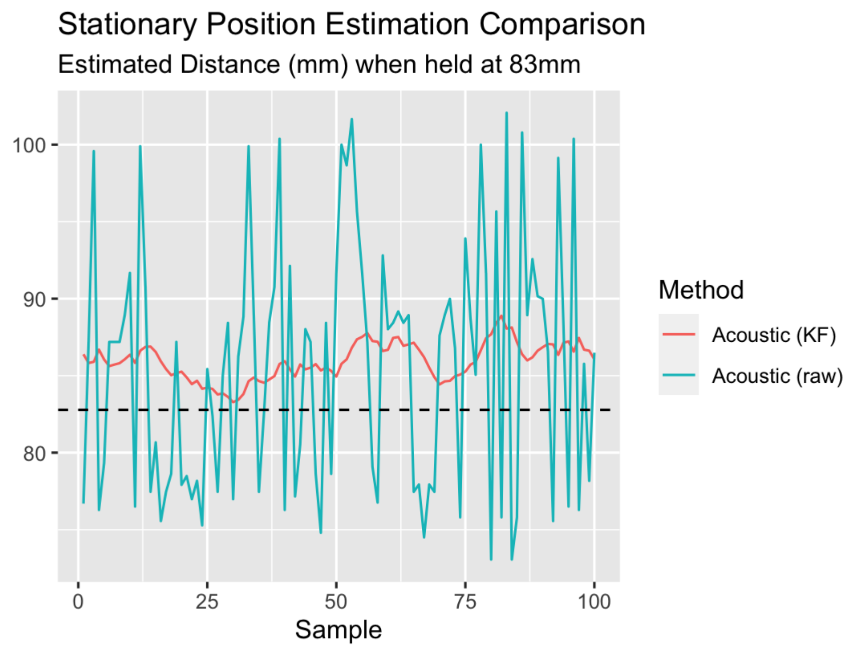

Figure 10: Comparison of Acoustic Sensor Position Measurements, Stationary Input

5. Comparison to Classical Methods

5.1. Comparison

The use of MMAEKF to process output data in this project transforms noisy, essentially useless accelerometer measurements into a quality description of the carriage's position. Alternative, "classical" filtering methods include a low- or high- pass filter to eliminate noise either above or below the cutoff frequency.

MMAEKF is superior to low- or high- pass filtering because it allows for adaptation to the system, providing a more accurate reading with each time step, rather than simply eliminating outlying data measurements. Furthermore, it allows for multiple 'noise state' models which may be particularly useful for systems with varying noise. An example of this could be an accelerometer on an aircraft wing which undergoes varying noise vibration modes during different operation times. The MMAEKF will use the selection algorithm to select the KF that best represents the current noise. Classical signal processing techniques do not have this advantage.

5.2. Wider Applications

MMAE has broad applications in position measurements filtering, including in Global Positioning Systems (GPS) and navigation systems. The main applications of MMAEKF can be found in guidance, navigation, and control [3]. MMAE techniques may be used on missile guidance systems to select the optimal path towards a target, which may have an unpredictable flight path which requires multiple estimation models. Conversely, MMAE may be used on airplanes to select a path to avoid incoming missiles, as discussed in [4].

6. Conclusion

In conclusion, MMAEKF is an effective signal processing technique for combining multiple measurement inputs, especially when applied to systems with changing uncertainties and conditions. By using multiple models to represent the system dynamics, and adaptively selecting the most suitable model based on the best performing estimates, the algorithm can provide accurate and reliable state measurements, which can then be used to control the system effectively. This technique is particularly useful in applications such as navigation, guidance, and control, where multiple sensors are utilized, and where accurate estimation of the system state is crucial for achieving the desired performance. Furthermore, the MMAEKF approach can be implemented in real-time on embedded systems with limited computational resources, as explored in this project. Overall, the MMAEKF technique represents an important advance in signal processing and estimation theory, and has significant potential for improving the performance of complex systems in dynamic real-world environments.

7. References

[1] Khodarahmi, M., & Maihami, V. (2022). A review on Kalman filter models. Archives of Computational Methods in Engineering, 1-21.

[2] Frabosilio, J., & Lyons, J. (2023, March 20). Linear Position Controller. EE 513 - Winter 2023. Retrieved from https://cpslo-my.sharepoint.com/:w:/g/personal/jlyons06_calpoly_edu/EVHZh_uWLDNAoM7PYSkW0x4BvMAR2L0xoFjLaxz6DwgfRA?e=YE741o

[3] Weicun Zhang, Sufang Wang, Yuzhen Zhang, "Multiple-Model Adaptive Estimation with A New Weighting Algorithm", Complexity, vol. 2018, Article ID 4789142, 11 pages, 2018. https://doi.org/10.1155/2018/4789142

[4] Y. WANG, S. FAN, J. WANG, and G. WU, “Quick identification of guidance law for an incoming missile using multiple-model mechanism,” Chinese Journal of Aeronautics, vol. 35, no. 9, pp. 282–292, Sep. 2022, doi: https://doi.org/10.1016/j.cja.2021.10.032.

8. Appendix

8.1. C++ code and MATLAB files:

Github Repository Available at: https://github.com/sam-hud/AERO553_MMAKF