ECE 8540: Lab 4

KALMAN FILTER

October 8, 2021

John Lawler
Clemson University
Department of Electrical and Computer Engineering
lawler6@clemson.edu

1 Introduction

Kalman filters are used widely across many different applications of data anaylsis and manipulation. The benefits to using a Kalman filter revolve around optimizing for best lines of fit to a set of data. The Kalman filter is very similar to a moving average in a sense; it looks to find the "middle ground" between a set of data over a given period of time. The filtering algorithm is used widely in predicting correct sensor data, machine learning algorithms, and more.

2 Background

Two different sets of data are used for the modeling process. First, a set of 1-dimensional velocity data is filtered against. The implementation of the filter for the 1D data is explained under section 2.1. For the second data set, two different financial statistics are chosen using market data on the Ethereum cyrptocoin from May 25, 2020 to present day. The two data sets under the coin chosen are the volume traded on a given day and the opening price of the coin for a given day. Due to the nature of the prediction-update cycle, the Kalman filter adjusts very well to non-linear sets of data.

Financial predictive modeling implements moving averages and weighted summations to form various generalizations about a market in a given time span. Using the Kalman filter to analyze financial data can pose useful for removing high outlying values and finding/predicting general market trends. Depending on how the measurement and dynamic noise is initialized, the kalman filter can be used to mimic a simple or exponential moving average over sets of financial data.

2.1 Model Fitting Velocity via Kalman Filter

The starting matricies for dynamic noise covariance (Q), the state estimate covariance (S), the measurement noise covariance (R), initial state (X) are as follows:

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

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2}$$

$$R = \begin{bmatrix} 1 \end{bmatrix} \tag{3}$$

$$X = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4}$$

This same configuration is used for the initial values in part two also; only Q and R are altered to the desired fit.

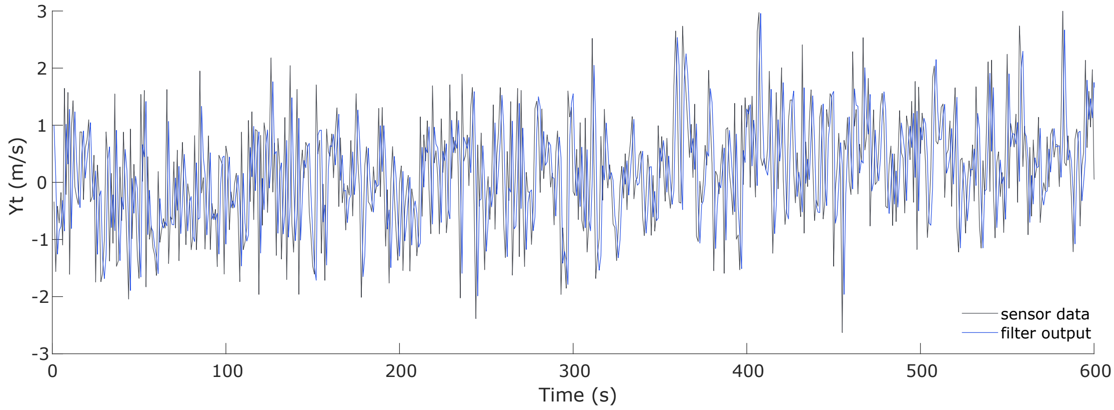

Figure 1: Initial Output

3 Discussion and Results

3.1 Filtering 1D Velocity Data

Figure 1 shows the results of the Kalman filter output when using the initialized setup shown in section 2.1. As shown, the filtered output in blue directly aligns with the provided one-dimensional velocity data. It does a very poor job at averaging itself throughout the data.

Several match up variations of dynamic noise covariance (Q) and measurement noise covariance (R) are shown in the following figures. First an extreme value of Q is chosen and shown against a constant value of R. Two different extreme values of R are used to reflect the difference a factor of 10 may show in the filtered outputs. Finally, a value of Q and R is chosen such that a accurate and smoothed average is found amongst the velocity data.

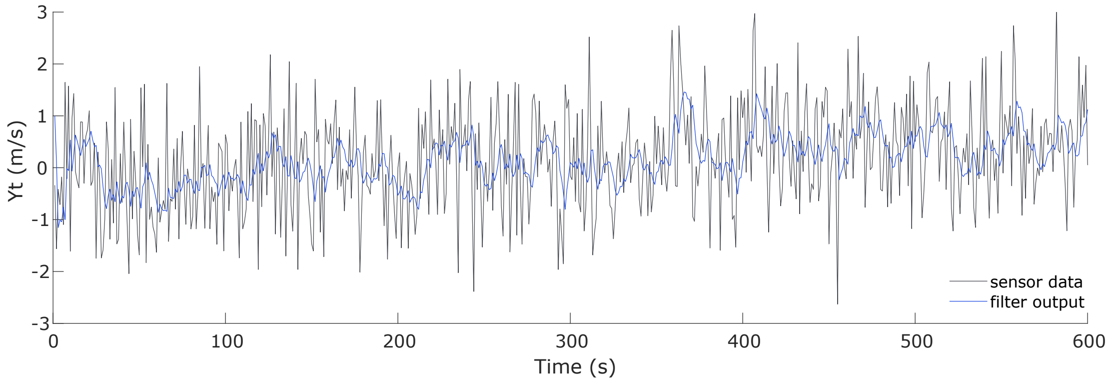

Figure 2: Output with extreme Q, R = 1

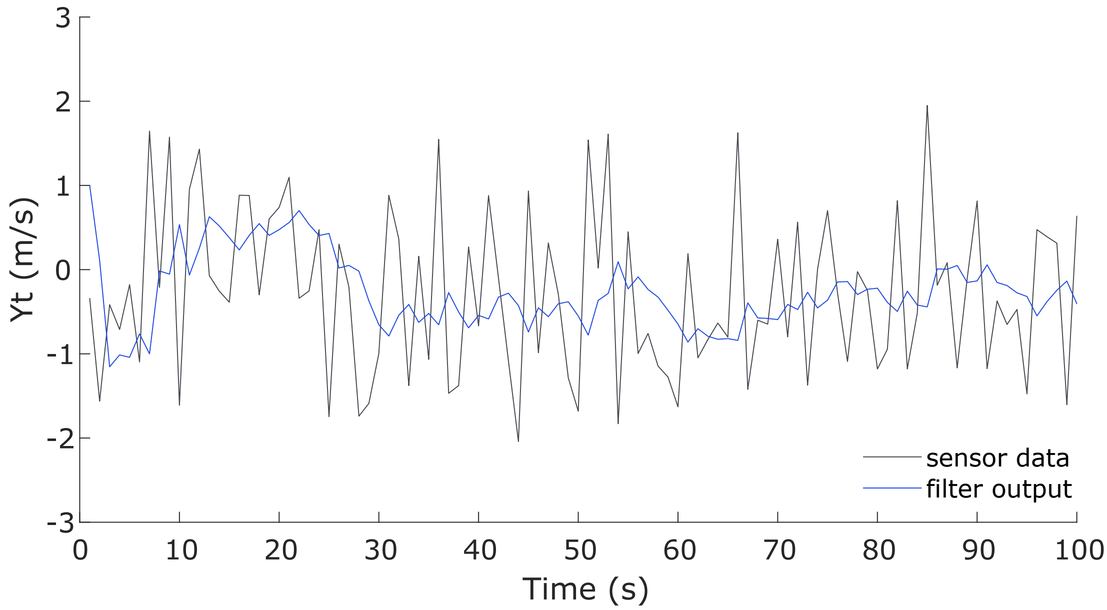

Figure 3: Figure 2 close-up

Figure 2 shows the result of setting the filter with an extreme weight on Q

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.0005 \end{bmatrix} \tag{5}$$

while holding R at R = 1. Figure 3 shows a close up of the results of figure 2, however, on a time scale of 0 - 100 seconds as opposed to the entire data set provided. Relative to figure 1, the results shown by adjusting Q to a very small fraction produces a smoothed filter effect.

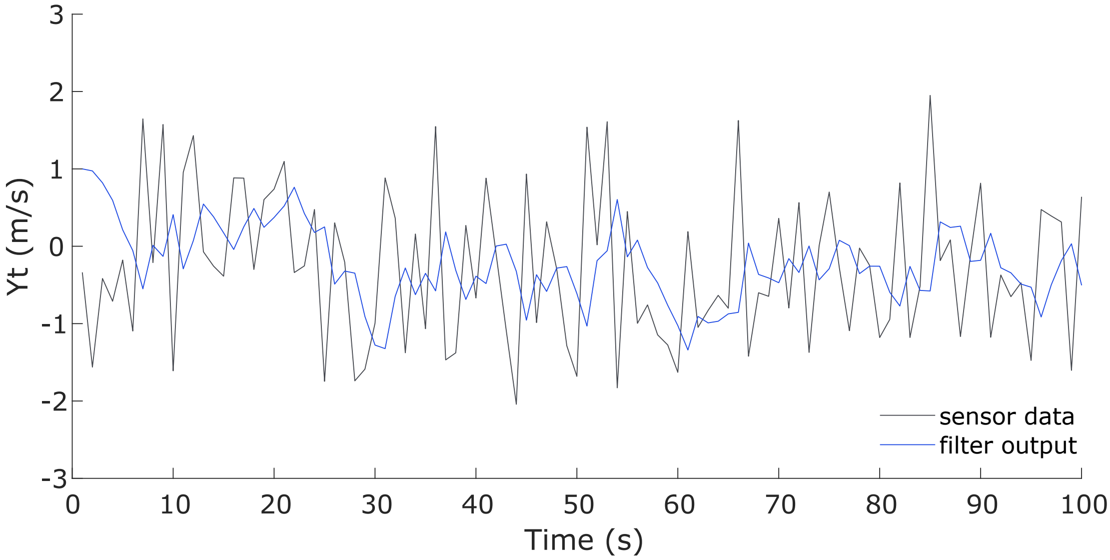

Figure 4: Q held constant, R = 100

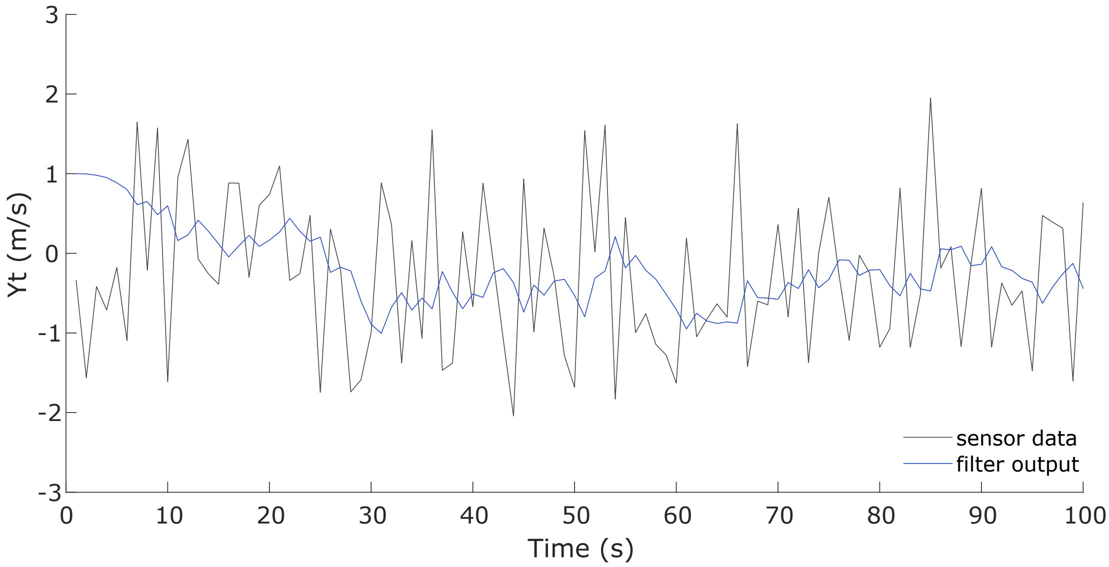

Figure 5: R set to 1000

Figure 4 and 5 sets the dynamic noise covariance matrix to

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{6}$$

while adjusting R.Between figures 4 and 5, R is adjusted by a factor of 10. The relationship shown by increasing measurement noise factor, the filter weights less heavily towards the peaks and dips. We can also see that the difference between adjusting R and adjust Q create the same result when comparing figures 3 and 5.

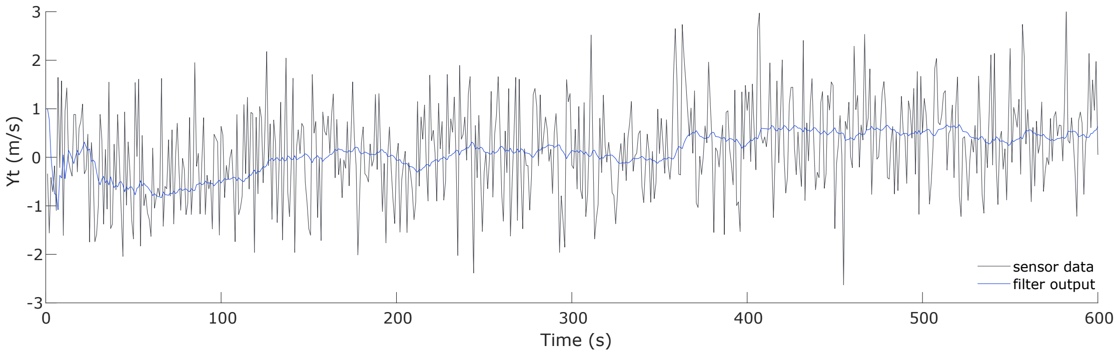

Figure 6: Smoothed filter output

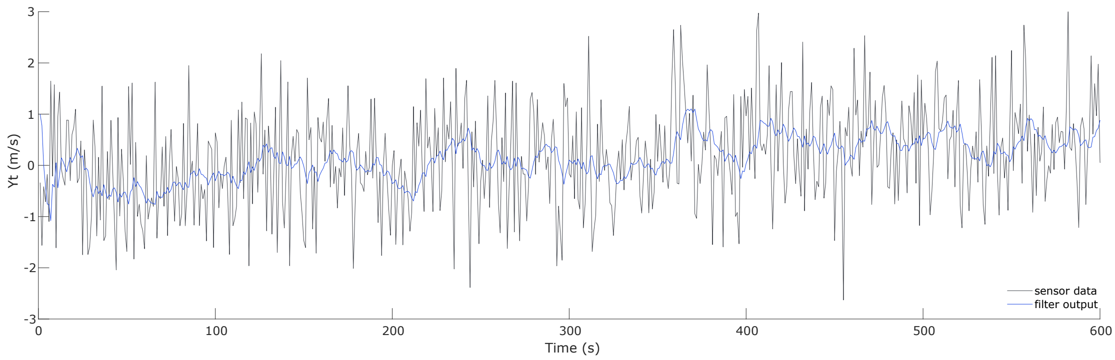

Figure 7: Filter output with heavier weighting towards variance

Figures 6 and 7 show the best fit results found, with figure 6 weighted with a smaller Q value than figure 7. Both figure 6 and figure 7 are weighted with R = 10. The results shown here show a dinstict difference in how the dynamic noise measurement affects the smoothness of the Kalman filter. A Q matrix weighted with a smaller fraction such as

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.00001 \end{bmatrix} \tag{7}$$

which is the Q used in figure 6, will produce a smoother continuous curve versus a covariance matrix consisting of

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.0005 \end{bmatrix} \tag{8}$$

which is used in figure 7.

3.2 Filtering Ethereum Coin Data

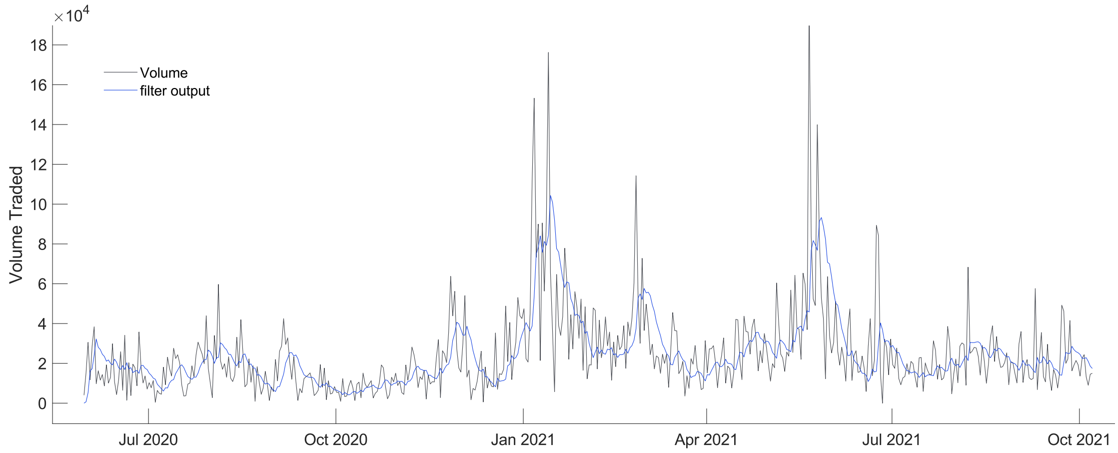

Figure 8: Ethereum volume traded against filtered output

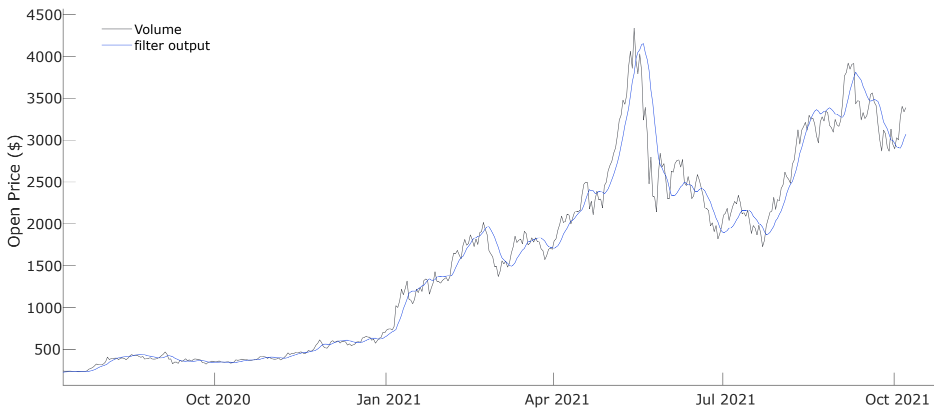

Figure 9: Ethereum opening price against filtered output

The data used for part two was retrieved from Gemini currency exchange; a trading platform which provides historical data on Ethereum price. Figure 8 and 9 shows the filter output of the Kalman filter using the optimized solution for Q and R found in part 1:

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.0005 \end{bmatrix}, R = 10 \tag{9}$$

The output reflected in figures 8 and 9 is more tightly adjusted to the actual data set. This resembles a moving average across several days or a week, as the ratio of Q and R is weighted less heavily towards ignoring noise. Shorter moving averages that might resemble this filter are useful for analyzing short term (months at a time) trends in market data. An observation of figure 9 may be that, at a given peak price point with an initial dip, if the price crosses the filter in a decreasing direction, there is more often than not a relatively large drop in price.

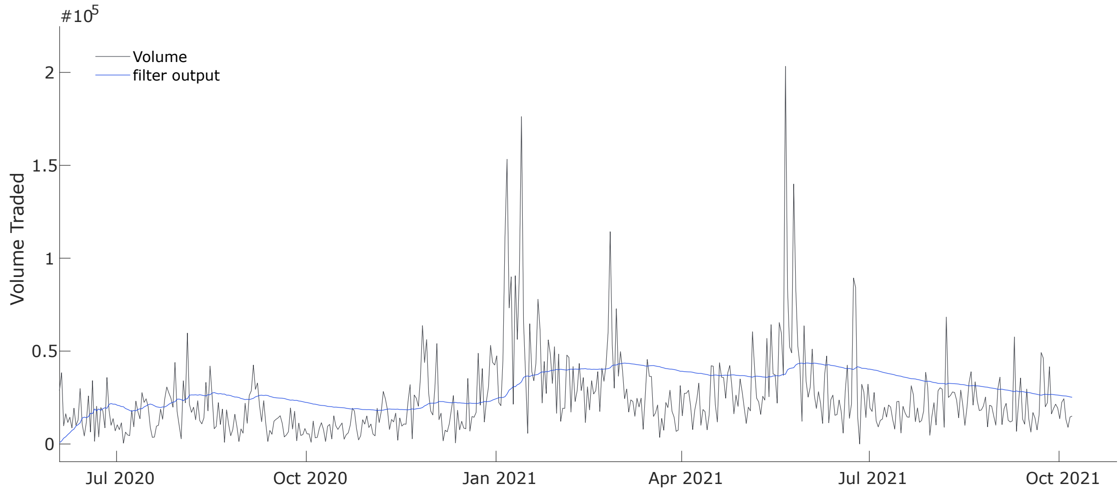

Figure 10: Ethereum volume against Kalman filter

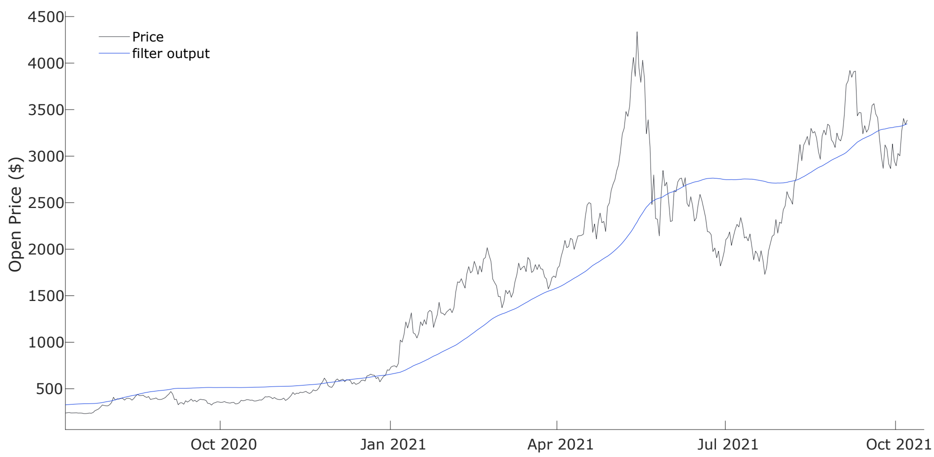

Figure 11: Ethereum Opening Price filtered

Figure 11 shows the filter output using a solution found to best fit the opening price data for Ethereum. These values were then applied to the volume traded across the time span shown in Figure 10.

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.00001 \end{bmatrix}, R = 1000 \tag{10}$$

Here, we see a very smooth filtered curve over our figures in 10 and 11. This curve is similar to a moving average across several weeks or a month versus a moving average across a shorter time span such as shown in figures 8 and 9. For figure 11, it can be especially useful to see potential buy/sell times, evaluate trends of when the price cross the Kalman filter, and see a generalized market evaluation of the average price of Ethereum. For instance, when the price crosses the filter in an increasing direction, it may be a good indicator to buy - as it seems the price tends to increase further after crossing the line initially.

4 Conclusion

Overall, it can be shown how useful the Kalman filter may be at determining actual real-world sensor measurements to predicting long-term profit/loss scenarios in finance, with a range of possibilities in between. Accurate predictive representations of non-linear data is much easier to calculate using a predict-update algorithm such as the Kalman filter versus fundamental linear regression. Accurate tuning of the filter to adjust for dynamic and measurement noise will allow a user to optimize their predictive model to their liking.