Kalman Filters for Noise Reduction

Keshav Ganapathy [1] and Joshua McClellan (AMDS)

[1]Centennial High School

Abstract

In everyday tasks, we often depend on sensors and other tools to provide us accurate measurements for a variety of tasks ranging from driving down the street to launching the next rocket to space. Naturally, with these sensors comes slight inaccuracies that will be referred to throughout as noise. Throughout this study, we provide an empirical comparison between two common filtering techniques to remove this noise: a kalman filter and moving average filter.

Kalman Filter

The Kalman Filter is a Linear Quadratic Estimation algorithm that uses various noisy measurements to produce estimates that are often more accurate. The Kalman Filter consists of two steps: a prediction step and update step.

Prediction

The prediction step implemented throughout this project can be mathematically represented with the equations below. Variable definitions can also be found below.

$$\hat{\mathbf{x}}_k = \mathbf{F}_k \hat{\mathbf{x}}_{k-1}$$
$$\mathbf{P}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k * a$$

Variable	Definition
$\hat{\mathbf{x}}_k$	The state vector that is currently being predicted. In this experiment, the state vector contained the state's estimates of the position, velocity, and acceleration.
$\mathbf{F}_k$	The state transition model. This is multiplied to the previous state vector to produce the new state vector.
$\hat{\mathbf{x}}_{k-1}$	The previous state vector.
$\mathbf{P}_k$	The covariance matrix used in the Kalman Filter. Covariance matrices are used to indicate the relationship between variables in the Kalman Filter.
$\mathbf{Q}_k$	The $\mathbf{Q}_k$ matrix acts as process noise added to the Kalman Filter in the prediction step.
a	a is a preset parameter used to add noise to the $\mathbf{Q}_k$ matrix within the Kalman Filter. This parameter should be tuned to optimize the Kalman Filter's performance.

Visually, with a state vector containing only position and velocity, the prediction step can shown below. The Kalman Filter used in this experiment has the expanded functionality of acceleration in the state vector as well.

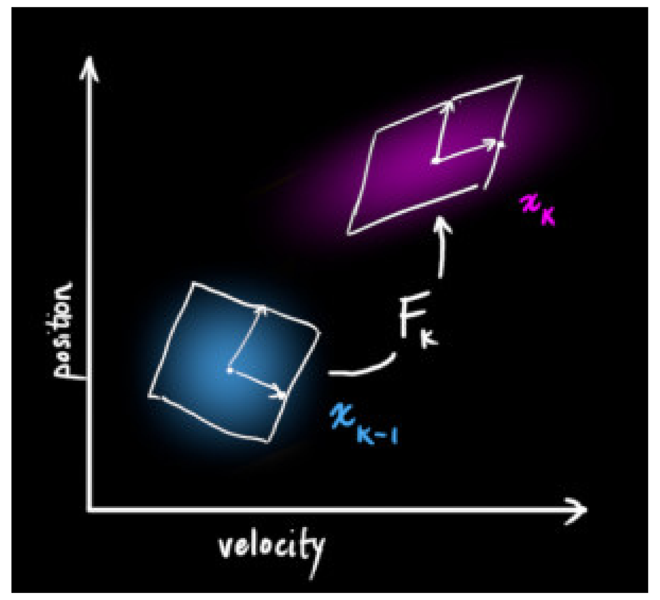

Figure 1. Figure from [1]

Update

The update step implemented throughout this project can be mathematically represented with the equations below. Variable definitions can also be found below.

$$\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$
$$\hat{\mathbf{x}}'_k = \hat{\mathbf{x}}_{\mathbf{k}} + \mathbf{K}(\vec{\mathbf{z}_k} \mathbf{H}_k \hat{\mathbf{x}}_k)$$
$$\mathbf{P}'_k = \mathbf{P}_k \mathbf{K} \mathbf{H}_k \mathbf{P}_k$$

Variable	Definition
$\mathbf{H}_k$	The observational model which is used to represent the state space into observed space.
$\mathbf{K}$	The Kalman Gain is used to give relative weight given to the measurements and current state estimate. This can be tuned to optimize performance.
$\mathbf{R}_k$	The variance of the observed measurement. Quantifies the uncertainty of the measured value.
$\vec{\mathbf{z}_k}$	The noisy observed measurement value.

The formulas used in both steps and their derivations can be found visually in [1] and mathematically in [2].

Moving Average Filter

We primarily implemented the moving average filter to compare its effectiveness to the Kalman Filter. The filter takes in a subset of size N (the window size) and a list of noisy measurements , L. At some index i of list L, the filter takes the value of the list at index i, as well as the next N-1 values in the list. These N values are then averaged, and stored in index i of list y. This can be mathematically represented with the equation below:

$$y[i] = \frac{1}{N} \sum_{j=0}^{N-1} x[i+j]$$

Let's do an example to see this in practice. Take the list: [45, 24, 49, 29, 16, 39, 45, 34, 30, 2] with N=2 (the window size). The moving average filter takes the list aforementioned. Using the logic mentioned before, take the first N integers, 2, and average them. In this case, the first element of the returned list y will be represented by $\frac{(45+24)}{2} = 34.5$, the second by $\frac{(24+49)}{2} = 36.5$, etc. This is then added to another list. We continue this process with the entire list and return a filtered list with values [34.5, 36.5, 39 , 22.5, 27.5, 42, 39.5, 32, 16]. The moving average filter will cause the size of the list to reduce.

Results

In this experiment, we use both a Kalman Filter and a Moving Average filter to filter artificial noise added to the position, velocity, and acceleration data. We used the following functions to generate data points for the true position, velocity, and acceleration with respect to time, t. Velocity and Acceleration are the the first and second derivative respectively of the position function.

$$x(t) = 100sin(t) \qquad x'(t) = v(t) = 100cos(t) \qquad x''(t) = a(t) = -100sin(t)$$

Measurements were taken in time increments of .1, and measurements received a random amount of noise ranging from -10 to 10.

Results of the Kalman Filter vs Moving Average are below:

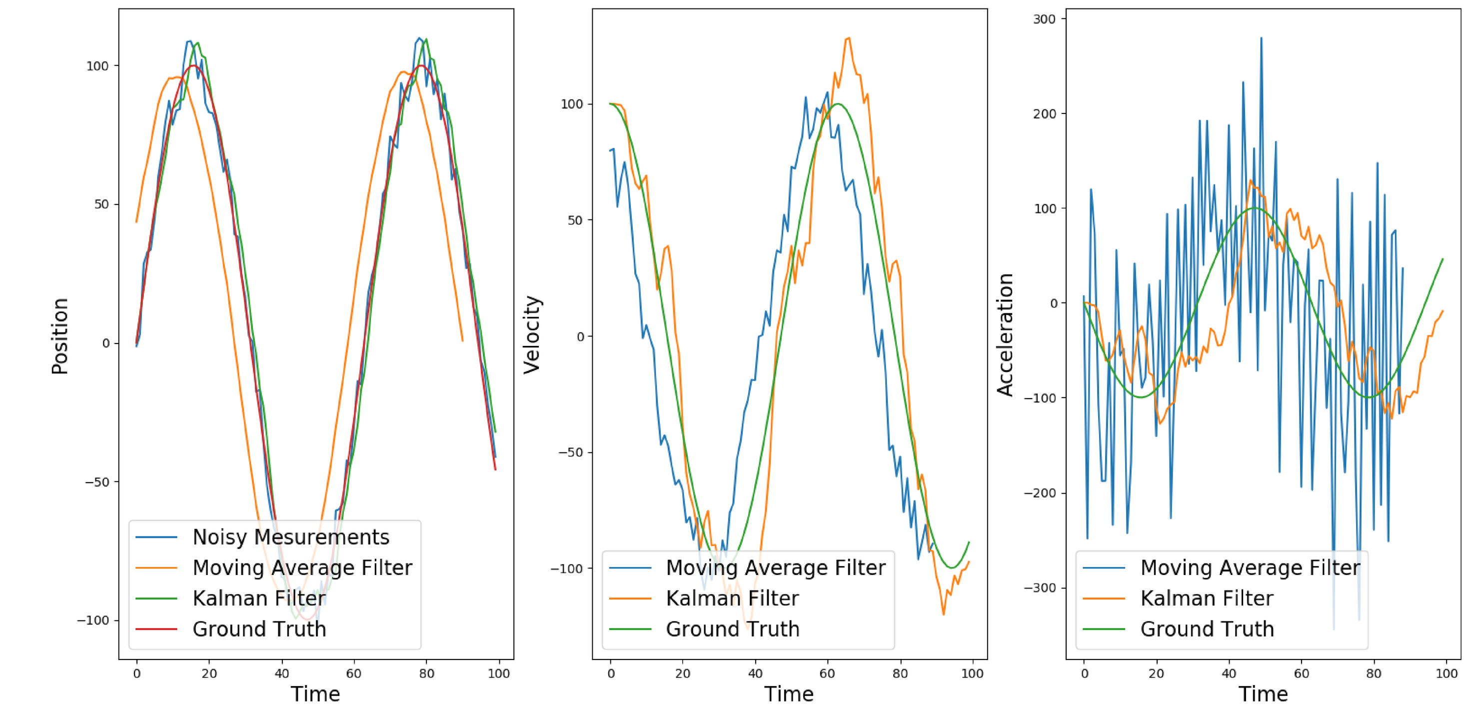

Figure 2.

To quantify the performance of the various filtering algorithms, we use the absolute value of the mean residual, M_r, for the Kalman filter and the moving average filter.

$$M_r = avg(|predicted - true|)$$

Below, are the results of both filters over three trials:

Filter and Variable	Trial 1	Trial 2	Trial 3	Overall Average
Kalman Position	5.318	5.760	5.837	5.638
Moving Average Position	27.660	27.378	27.284	27.441
Kalman Velocity	15.845	15.515	14.307	15.222
Moving Average Velocity	32.495	32.041	32.405	32.314
Kalman Acceleration	42.464	41.346	40.800	41.537
Moving Average Acceleration	91.385	95.461	81.952	89.601

Conclusion

This work introduces the mathematics behind a Kalman Filter and applies an implementation of the filter to reduce artificially added noise. We demonstrate that noise reduction via a Kalman Filter significantly outperforms more naive methods such as a moving average filter. More specifically, outperform is defined by having a lower mean residuals.

Acknowledgements

I would like to thank my mentor Joshua McClellan (JHUAPL) for giving me the opportunity to work with him as a part of the ASPIRE Program for close to a year. Thank you for the continuous support, the feedback and comments on this poster, and much more.

References

[1] How a kalman filter works, in pictures. `https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/`. Accessed: 03-10-2021.

[2] Kalman filter. `https://en.wikipedia.org/wiki/Kalman_filter`. Accessed: 03-15-2021.

Keshav.Ganapathy@jhuapl.edu