ECE 8540

# Lab 5 - Extended Kalman Filter

Tim Driscoll

October 26, 2021

## 1 Introduction

This lab report concerns the problem of applying an extended Kalman filter to a set of measurements. The measurements are numerical data that has been recorded over time in order to provide samples to filter. The provided data is to be filtered using a sinusoidal model. The measurements provide recorded data on the current height of the sinusoid. Similar to the Kalman filter, the extended Kalman filter would be used actively to filter the measurements in real time. The extended Kalman filter is used to more accurately determine the desired state variables in non-linear models. This expands upon the regular Kalman filter to help handle more complex problems with more complex models. In practice there will be some level of noise that we want to remove. The extended Kalman filter allows us to weight the accuracy of our predictions and measurements to produce the best results. Previously, methods such as the regular Kalman filter allowed us to filter linear models, but this technique fails to produce accurate results when observing non-linear problems. The extended Kalman filter uses methods to linearize non-linear problems and produce accurate results.

## 2 Methods

The extended Kalman filter also follows a cycle of prediction and update. In this cycle the next state and state covariance is predicted. Then measurements are obtained from the sensor that is being actively used. After the measurements are obtained the Kalman gain is calculated. The Kalman gain represents the weights that are placed on both the predictions and measurements. The state and state covariance is then updated. This process is then looped over the next time step. These steps can be modeled in the follow series of equations:

1. Predict next state
$$X_{t,t-1} = f(X_{t-1,t-1}, 0) \tag{1}$$
Equation 1 represents the matrix of nonlinear equations used to predict the next state. The inputs to $f$ include the state variables and noise. The noise is set to 0 when predicting the next state in this step.

2. Predict next state covariance
$$S_{t,t-1} = \left(\frac{\partial f}{\partial x}\right) S_{t-1,t-1} \left(\frac{\partial f}{\partial x}\right)^T + \left(\frac{\partial f}{\partial a}\right) Q \left(\frac{\partial f}{\partial a}\right)^T \tag{2}$$
In equation 2 the partial derivative terms are the Jacobians of the state transition equations, $f$.

3. Obtain measurement(s) $Y_t$

4. Calculate the Kalman gain (weights)

$$K_t = S_{t,t-1}\left(\frac{\partial g}{\partial x}\right)^T\left[\left(\frac{\partial g}{\partial x}\right)S_{t,t-1}\left(\frac{\partial g}{\partial x}\right)^T + \left(\frac{\partial g}{\partial n}\right)R\left(\frac{\partial g}{\partial n}\right)^T\right]^{-1} \tag{3}$$

   In equation 3 the partial derivative terms are the Jacobians of the measurement equations, $g$.

5. Update state

$$X_{t,t-1} + K_t[Y_t - g(\tilde{x}_t, 0)] \tag{4}$$

   In equation 4, $g(\tilde{x}_t, 0)$ represents a noiseless measurement of the predicted state.

6. Update state covariance

$$S_{t,t} = \left[I - K_t\left(\frac{\partial g}{\partial x}\right)\right]S_{t,t-1} \tag{5}$$

7. Loop ($t$ is incremented by $\Delta t$)

It is important to note that in this series of equations initial values need to be defined. These values include the state at time 0, the state covariance at time 0, the dynamic noise covariance and the measurement noise covariance. It is the job of the designer to determine how these values should be defined based on available information. The designer must also define the nonlinear equations that compose both $f$ and $g$. After these equations are defined the initial Jacobians must also be calculated for and defined. The Jacobians are calculated by taking partial derivatives of all the transition and observation equations with respect to the noises and state variables. Lastly, it is important to note that if the Jacobians are not constant they must be updated each iteration.

## 2.1 EKF with Sinusoidal Model

For the measurements that were being filtered for this lab a sinusoidal model was used. There were three state variables, where the most important variable was $h_t$. The state variable $h_t$ represented the height of the sinusoid at the given time $t$. All the state variables were represented in the following matrix:

$$X_t = \begin{bmatrix} x_t \\ \dot{x}_t \\ h_t \end{bmatrix} \tag{6}$$

For this model at time 0 it was chosen that that the first two state variables were set to zero and the height of the sinusoid was set to be the value obtained by the first measurement. The state transition equations for the given model were defined as follows:

$$f(x_t, a_t) = \begin{bmatrix} x_{t+1} = x_t + \dot{x}_t T \\ \dot{x}_{t+1} = \dot{x}_t + a_t \\ h_{t+1} = \sin\frac{x_t}{10} \end{bmatrix} \tag{7}$$

These equations allowed for the prediction of future states. The value, $a_t$ represents the dynamic noise. The dynamic noise is a random value obtained from a normal Gaussian distribution. For measurements it was assumed that the sensor used detected the current height of the sinusoid. The height was represented by $d_t$ in the following matrix:

$$Y_t = \begin{bmatrix} d_t \end{bmatrix} \tag{8}$$

The observation equation for the measurements were represented by the following matrix.

$$g(x_t, n_t) = \begin{bmatrix} d_t = h_t + n_t \end{bmatrix} \tag{9}$$

In this matrix there is only the need for one equation, because only one state variable was being observed. The inputs to the equation includes both the state variable ($h_t$) and the measurement noise ($n_t$). The measurement noise represents a random value obtained from a normal Gaussian distribution. After the transition equations, observation equations and accompanying variables were defined four Jacobians were defined. The first two Jacobians include the derivative of the state transition equations with respect to the state variables and then the dynamic noise. The second two Jacobians are the derivative of the observation equations with respect to the state variables and then the measurement noise. All four Jacobians are defined as follows:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ \frac{1}{10}\cos\frac{x}{10} & 0 & 0 \end{bmatrix} \tag{10}$$

$$\frac{\partial f}{\partial a} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{11}$$

$$\frac{\partial g}{\partial x} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$\frac{\partial g}{\partial n} = \begin{bmatrix} 1 \end{bmatrix} \tag{13}$$

It is important to note that for this example the value $T$ was set to equal one. $T$ was set equal to one because the time step for each iteration of new measurements was defined to be one second. It can also be noted that all but one of the Jacobians are constant. These Jacobians are constant because many of the state transition and observation equations were linear. Finally it is important to note that during the beginning of each time iteration Jacobian 10 must be updated.

It was then important to define the initial state estimate covariance, $S_t$. The state estimate covariance represents the level of uncertainty that there is in the state variables shown in matrix 6. Specifically, the diagonal of matrix 14 represents the uncertainty of the individual state variable. The elements off of the diagonal represent the covariances of the three state variables. The state covariance matrix was initialized to the following:

$$S_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

For this model at time 0 the state covariance matrix was defined to be the identity matrix. This placed an initial uncertainty on the state variables, but not their covariances. The final two variables that needed to be initialized were the dynamic noise covariance and the the measurement noise covariance, represented by $Q$ and $R$ respectively. These matrices would directly impact how the filtering would behave. Depending on the ratio between the two noises the filter would either more closely follow the measurements or the state transition equations. In this lab the noises were tuned to adjust the behavior of the filter and observe the results. The generic form of both the Q and R matrix used for this sinusoidal model are as follows:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_a^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{15}$$

$$R = [\sigma_n^2] \tag{16}$$

In matrix 16 the singular value represents the variance of the measurement noise of sinusoid height. Matrix 15 is only composed of the variance for the $\dot{x}_t$ portion of the dynamic noise. The nonzero values in both these matrices were used to tune the filter and achieve desired results.

### 2.2 MATLAB Implementation

Once it was determined how the matrices were defined for the model equations 1-6 were implemented in software. The 2020a version of MATLAB on a Windows operating system was used to implement the code. All the initial matrices were constructed in the code and a for loop was used to iterate over the data for each time step. It was assumed that a new measurement would be read every second, thus $\Delta T$ was defined to be 1. The loop was ran over the entire length of the measurement data. The estimated state was calculated and stored in a separate matrix at the end of each loop. This saved data was then used to plot the results of the filtering.

## 3 Results

The extended Kalman filter with the sinusoidal model was tuned and observed at three different ratios of dynamic noise and measurement noise. The tuning of this filter was done by holding the measurement noise as a constant value and adjusting the dynamic noise to alter the ratio between the two. The three different ratios represent the three different general cases that can be observed in the estimated data outputted by the Kalman filter. A summary of the variances used for each case can be observed in table 1. Note that both the measurement variance and the dynamic variance were adjusted in case two so the desired results were achieved. It is also important to observe the raw measurement data in figure 1. This plot represents the measurements obtained from the sensor and used to produce the filtered estimate. As shown in the figure a very rough sinusoid can be made out, but there is a large amount of noise. The first ratio case is when the measurement noise is smaller then the dynamic noise. This ratio will add more weight to the measurements and the estimated output will more closely resemble the measurements. If the difference between the

Table 1: Summary of three noise ratios observed.

| Ratio Case | $Q$ ($\sigma_a^2$) | $R$ ($\sigma_n^2$) |
|---|---|---|
| 1 | 100 | 1 |
| 2 | $10^{-12}$ | $10^7$ |
| 3 | 0.001 | 1 |

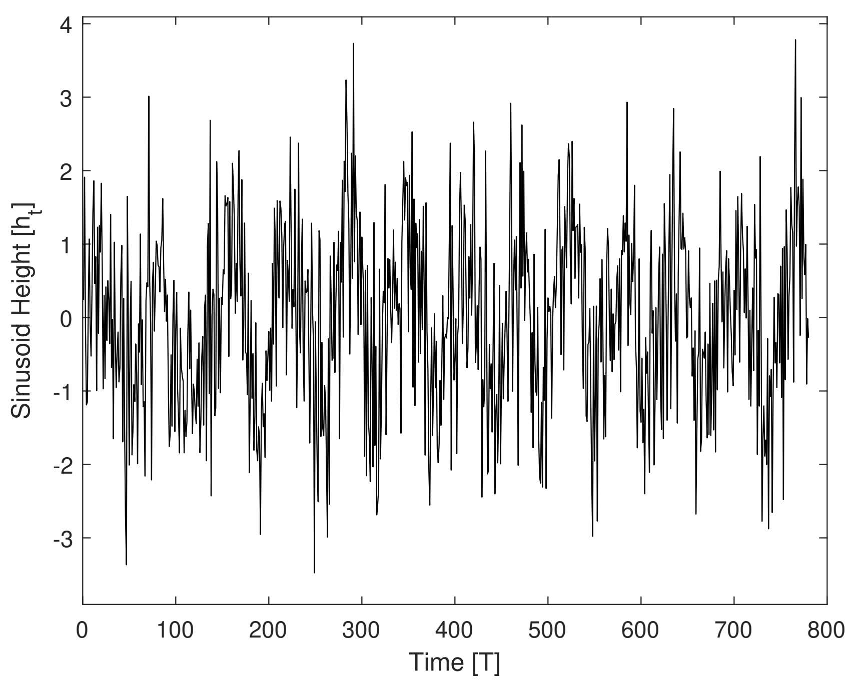


Figure 1: Raw measurement data.

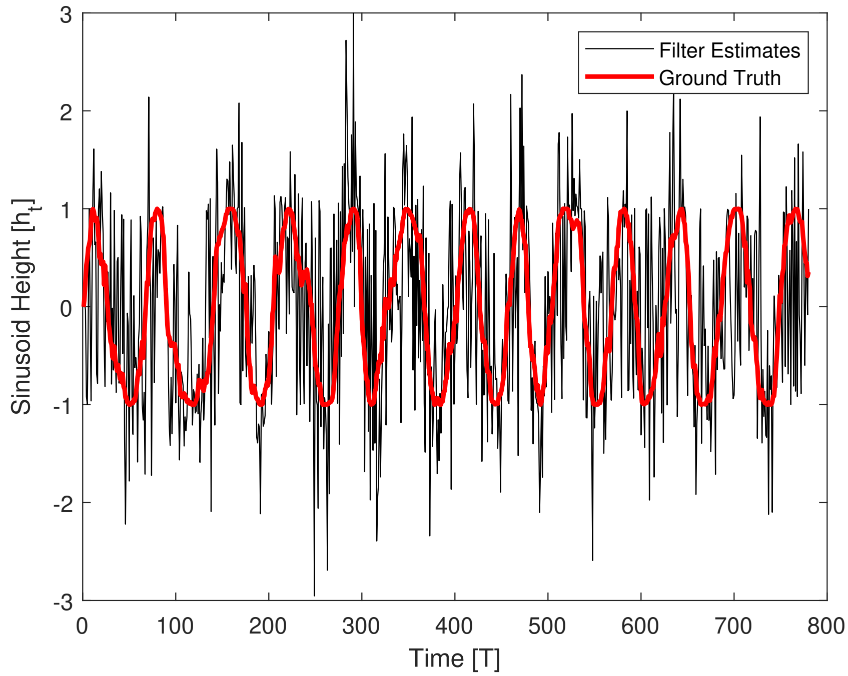


Figure 2: Ground truth data and filtered estimate when $(\sigma_a^2) = 100$ and $(\sigma_n^2) = 1$.

two noises in this case is large enough the estimated output will almost completely match the measurement data. As shown by figure 2 the filtered estimate almost completely follow the measurement data from figure 1. The ground truth is displayed in red and follows the general trend of the filtered estimate. It is clear though that the estimate more closely resembles the raw measurements. The second ratio case was when the dynamic noise was much smaller then the measurement noise. With this ratio the behavior of the filtered estimate should more closely match the values produced by the third equation in matrix 7 with respect to the other state variables. When viewing the model this output should reflect *sin* of a very small value. This will result in the filtered estimate resembling a straight line. When observing figure 3 the filtered estimate almost represents a straight line, but with some very slight oscillation. It can also be observed that this ratio case does a poor job of following the plotted ground truth. The final case was more of a balanced ratio between the dynamic noise and measurement noise variance. The balanced case allowed for a more equal weighting to be placed on the measurements and the state equation prediction. The ratio that was chose places slightly more weight on the prediction to produce an estimate more closely resembling the provided ground truth. This final case can be observed in figure 4. In this case it can be observed that the filtered estimate does a good job of closely resembling the provided ground truth.

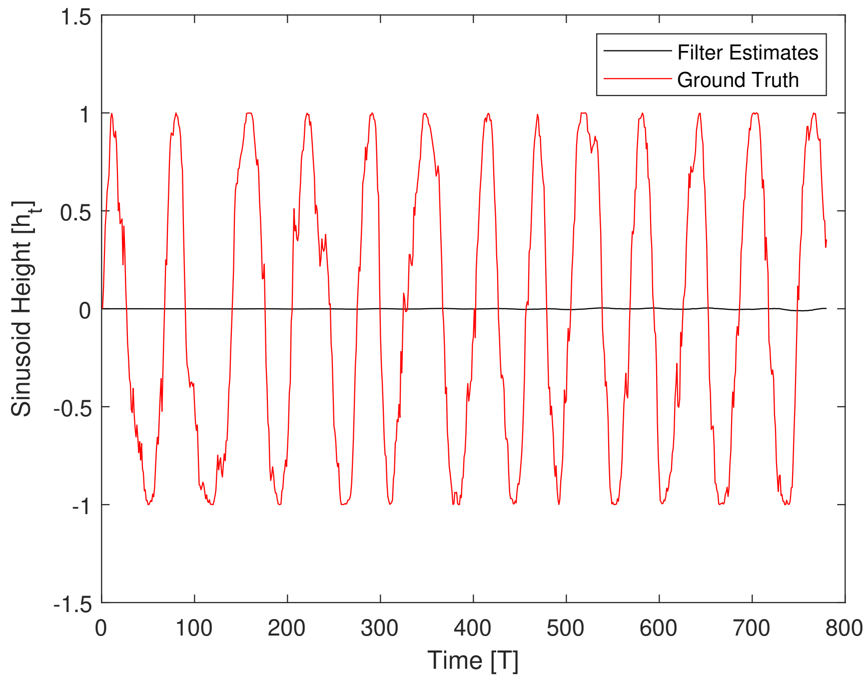


Figure 3: Ground truth data and filtered estimate when $(\sigma_a^2) = 10^{-12}$ and $(\sigma_n^2) = 10^7$.

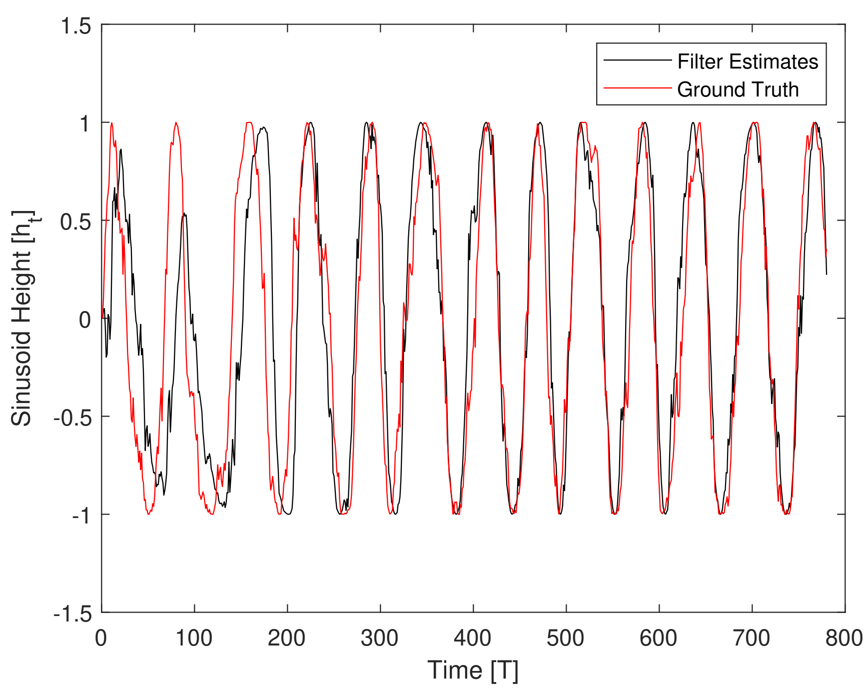


Figure 4: Ground truth data and filtered estimate when $(\sigma_a^2) = 0.001$ and $(\sigma_n^2) = 1$.

# 4 Conclusion

This lab clearly showed how the extended Kalman filter uses measurements in conjunction with state predictions to produce estimates. The extended Kalman filter provides an advanced filtering technique to more complex problems. By implementing the use of Jacobians the filtering method is able to linearize non-linear problems and accurately filter the measurements. The filter allows for the designer to place varying weights on both the measurements and predictions to gain the most accurate state estimates and handle noise. These varying weights can be achieved by adjusting the ratio between dynamic and measurement variance. If the measurement noise is held constant and the dynamic noise is increased an increasing amount of weight will be placed on the measurements. The exact opposite is true if the measurement noise is held constant and the dynamic noise is decreased. In the lab the results behaved according to the relationship between dynamic and measurement noise. A typical limitation in filtering is the lack of a known ground truth making it hard to most accurately tune the filter. Fortunately, in this implementation a ground truth was provided and allowed for the filter to be tuned to the best of its abilities. Even though a ground truth was provided and used for tuning it was impossible to replicate the ground truth with the extended Kalman filter estimate. This shows that the main limitation in filters such as the extended Kalman filter is how accurately the chosen model represents the measurements. Given there is noise in the measurements the accuracy of the model will also limit the accuracy of the estimation.