Concept Review

Kalman Filters

How to Estimate Unknown States of a System?

With the improvement in computation and sensor information a roboticist has access to different tools for estimating system states. Using intuition to derive the motion of a system in time and the noise which the system has, statistical tools have been designed to predict the states of a system. Primarily designed by Rudolf Kalman, the Kalman filter can be utilized to filter a signal and when expanded to an entire robot it can be used to estimate system states.

Background

A Kalman Filter (KF), also sometimes known as a Linear Quadratic Estimator (LQE), is a controls tool used for estimating properties of a system. Prior to looking at Kalman Filters it's highly recommended to be familiar with:

- Probability Theory
- Bayes Filter
- System Motion Models
- Sensor Models

Bayes Filter is an ideal filter which formulates the probability likelihood of system states using measurements up to and including the current timestep. By making assumptions about the shape of the probability likelihood such as:

- The probability distribution closely fitting a gaussian distribution
- Markov assumption of a system's state being only dependent on the present state and not any past states

The Bayes Filter is transformed into a gaussian filter. This is where assumptions about the motion model and measurement model can be made.

What are some assumptions to implement the Kalman Filter:

- Motion model is linear, and the measurement model is also linear, this leads to a Linear Kalman Filter.
- If the system's motion model and/or measurement model are now non-linear this is a Extended-Kalman Filter implementation.

A practical approach is used to demonstrate the benefit of the Kalman Filter. First a linear Kalman Filter is implemented within the context of measuring an objects height. Secondly a non-linear Kalman Filter is implemented to measure the angle of a pendulum as it swings back and forth. While not covered in this document, Kalman Filters have also been utilized in the context of Simultaneous Localization and Mapping (SLAM).

Kalman Filter

Motivation

Consider a falling object at constant gravity with some initial height. How would someone go about estimating the height of the object during freefall? An initial approach could be to use a constant gravity motion model to estimate the height of the object as it falls. This approach makes assumptions about the system motion model to estimate position of the object as it falls. A secondary step would be to attach a height sensor to the object and measure the height above ground.

Given the nature of sensors, the data being read could be either noisy or slow. Kalman Filters allow users to combine a motion model with sensor measurement information to generate a fast state estimate while using sensor data to correct for uncertainty over time.

In a generic sense, prior to implementing the Kalman Filter, some considerations to make:

- The accuracy of the motion model, how many assumptions are being made about the system for the motion model to be valid?
- The accuracy of the measurement model, does the measurement model encapsulate all the components which make up the sensor measurement?
- Noise in the motion model, does it follow a gaussian distribution or not?
- Noise in the measurement model, does it follow a gaussian distribution or not?

Motion Model

Continuing with the falling object example, the assumption was made that over time the object would traverse from an initial height and eventually hit the ground. What can be written are the following equations of motion

$$\ddot{x}(t) = -g \tag{1}$$

$$\dot{x}(t) = \dot{x}(t_0) - g(t - t_0) \tag{2}$$

$$x(t) = x(t_0) + \dot{x}(t - t_0) - \frac{g}{2}(t - t_0)^2 \tag{3}$$

The complicated part becomes these equations are formulated in continuous time. By changing to a discrete time system, the assumption now is the time step in the system is now 1 sample in length. That is:

$$t - t_0 = 1$$

We assume now that instead of an initial time estimate t_0 it is the k^{th} step in our estimate we can rewrite time as:

$$t = 1 + k$$

The changes to our model equations become:

$$\ddot{x}(k+1) = -g \tag{1}$$

$$\dot{x}(k+1) = \dot{x}(k) - g \tag{2}$$

$$x(k+1) = x(k) + \dot{x}(k) - \frac{g}{2} \tag{3}$$

The height estimate only depends on the height at the previous step and the velocity at the previous step, gravity is a constant input to the system. The states of the system can be defined as:

$$\boldsymbol{X}[k+1] = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

$$\boldsymbol{X}[k+1] = \boldsymbol{F}\boldsymbol{X}[k] + \boldsymbol{B}\boldsymbol{u}[k] \tag{4}$$

For the example provided:

$$\boldsymbol{F} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} -0.5 \\ -1 \end{bmatrix} g$$

The sensor used in our system can only measure height, not the speed at which the object falls therefore the measurement model becomes:

$$\boldsymbol{Z}[k+1] = \boldsymbol{H}\boldsymbol{X}[k] \tag{5}$$

Where:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Within any system there will be uncertainty. This can manifest as sensor noise or uncertainty in the motion model of the system. For a Kalman Filter to work it is assumed the uncertainty is Gaussian and zero mean. The process of combining the motion and measurement model is done in the following sequence of steps:

Prediction:

This step takes into account the prior information about the system and based on the motion model generates a prediction of what the states should be given a command input. The prediction steps also predict how the covariance matrix evolves for the next time step based on the system uncertainty and the motion model.

State prediction

$$\hat{X}_{k|k-1} = F_k X_{k-1|k-1} + B_k u_k \tag{6}$$

Covariance prediction

$$\hat{P}_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{7}$$

Where term Q_k represents the uncertainty in the motion model.

Once a new sensor measurement is received the Kalman Filter can now enter the **Update** portion of the algorithm.

Update:

The objective of this step is to calculate what the state estimate would be at the current timestep given the predicted covariance matrix, observation matrix, and sensor measurements.

Innovation

$$S_k = H_k \hat{P}_{k|k-1} H_k^T + R_k \tag{8}$$

Kalman Gain

$$K_k = \hat{P}_{k|k-1} H_k^T S_k^{-1} \tag{9}$$

Measurement Pre-Correction

$$z_k = y_k - H_k \hat{X}_{k|k-1} \tag{10}$$

Updated State Estimate

$$X_{k|k} = \hat{X}_{k|k-1} + K_k z_k \tag{11}$$

Updates Covariance Estimate

$$P_{k-1|k-1} = (I - K_k H_k) \hat{P}_{k|k-1} \tag{12}$$

Nomenclature:

Symbol	Description
H	Observation Matrix
Q	System noise
R	Observation noise
P	Covariance Matrix
K	Kalman Gain

Table 1: Kalman Filter Nomenclature definition.

Example result for falling mass:

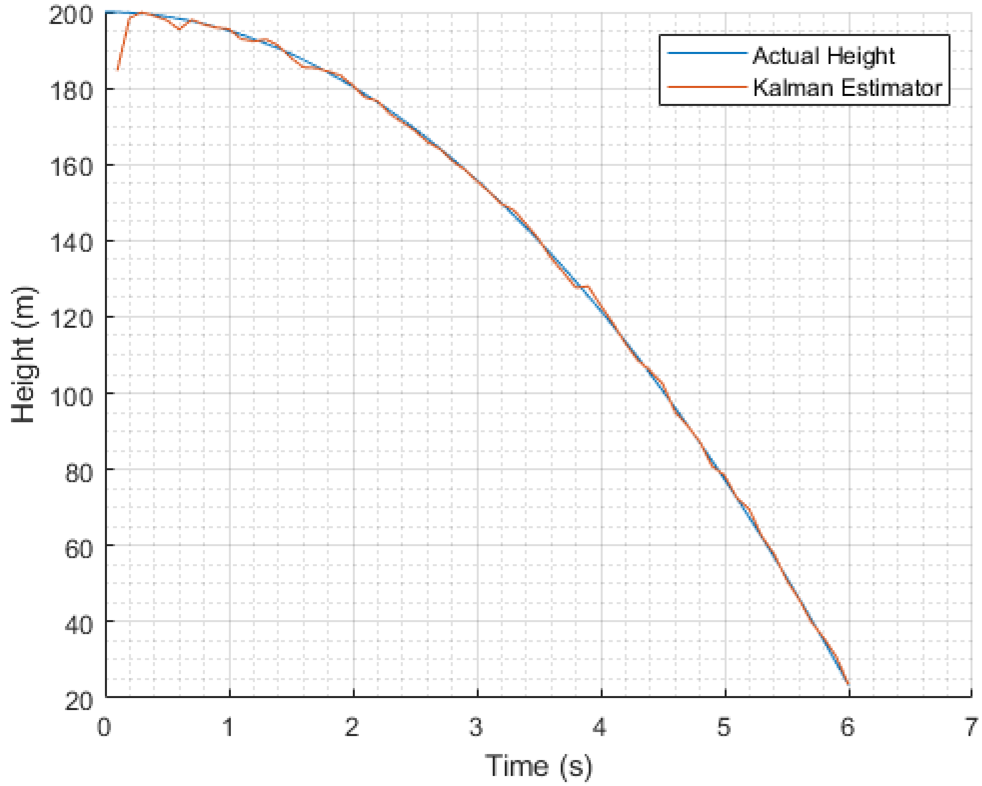

Figure 1. Height for falling mass under 1g of gravity

Assuming a $[x, \dot{x}] = [0,0]$ starting condition for the predictor, the Kalman Filter quickly converges to closely track the height of the ball above the ground. In this example the observation and system noise were estimated to be small. Depending on the application and sensor data, the observation noise R can be calculated by reading a sequence of samples and generating a Gaussian distribution. Concept review sensor noise describes a method for describing a sensor using a gaussian distribution.

Extended Kalman Filters

Linear Kalman Filters are a useful tool for systems that behave linearly in their motion model (or system equations) and/or in the measurement model. The next step is to ask, what happens when there are non-linear equations?

In this example let's consider the case of estimating the angle of a pendulum as it swings.

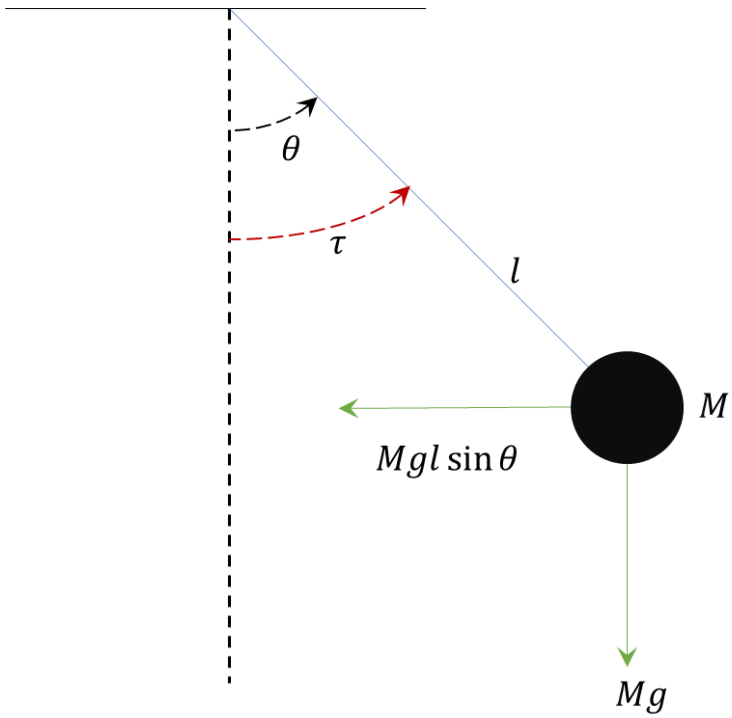

Figure 2. Pendulum Free Body Diagram

Start by writing the equations of motion. The state of interest is the angle θ. For this example, friction is ignored, it's assumed to be an ideal system.

The sum of torques around the origin yields:

$$I\ddot{\theta}(t) + mglsin(\theta(t)) = \tau\,, \qquad I = ml^2$$

Rearrange to have a torque input on one side and the system states on the opposite side:

$$\ddot{\theta}(t) = \frac{\tau}{ml^2} - \frac{g}{l} sin(\theta(t)) \tag{13}$$

The equations of motion give a non-linear dependency on θ. θ can also be estimated to be the direct integration of $\dot{\theta}$:

$$\theta(t) = \theta(t-1) + \dot{\theta}(t)dt \tag{14}$$

For sensors used in this example let's make sure of an inclinometer. Inclinometers are used to measure the angular orientation of an object. One of their downsides is the update rate which can be as slow as 10Hz. An Extended Kalman Filter can be used in this case to measure the angle of the mass as it swings back and forth based on some initial condition and input torque.

Compared to the Kalman Filter, the Extended Kalman Filter does not assume the motion model/measurement models are linear therefore the following equations are written:

Motion Model

$$\boldsymbol{X_k} = f(\boldsymbol{x_{k-1}}, \boldsymbol{u_k}, \boldsymbol{w_k})$$

Observation Model

$$\boldsymbol{Z_k} = h(\boldsymbol{X_k}, \boldsymbol{v_k})$$

When the motion and observation models are linearized about a local point the equations become:

$$f(\boldsymbol{x_{k-1}}, \boldsymbol{u_k}, \boldsymbol{w_k}) \approx \boldsymbol{\hat{X}_{k|k-1}} + \boldsymbol{F}_{X,k}\left(\boldsymbol{X}_{k-1} - \boldsymbol{\hat{X}}_{k-1}\right) + \boldsymbol{F_{w,k}}\boldsymbol{w_k}$$

And

$$h(\boldsymbol{x_k}, \boldsymbol{v_k}) \approx \boldsymbol{\hat{Z}_k} + \boldsymbol{H}_{X,k}\left(\boldsymbol{X}_k - \boldsymbol{\hat{X}_{k|k-1}}\right) + \boldsymbol{H_{v,k}}\boldsymbol{v_k}$$

Each sequence in the linearized forms becomes:

Nomenclature:

Symbol	Equation	Description
$\boldsymbol{\hat{X}_{k\|k-1}}$	$f(\boldsymbol{\hat{X}}_{k-1}, \boldsymbol{u}_k, 0)$	State prediction, zero noise
$\boldsymbol{F}_{X,k}$	$\left.\frac{\partial f(\boldsymbol{x_{k-1}}, \boldsymbol{u_k}, \boldsymbol{w_k})}{\partial \boldsymbol{x_{k-1}}}\right\|_{\hat{X}_{k-1}, u_k, 0}$	Jacobian of the motion model linearized about the prior state
$\boldsymbol{F}_{w,k}$	$\left.\frac{\partial f(\boldsymbol{x_{k-1}}, \boldsymbol{u_k}, \boldsymbol{w_k})}{\partial \boldsymbol{w_{k-1}}}\right\|_{\hat{X}_{k-1}, u_k, 0}$	Jacobian of the noise model linearized about the prior state
$\boldsymbol{\hat{Y}_k}$	$h(\boldsymbol{\hat{X}_{k\|k-1}}, 0)$	Measurement model, zero noise
$\boldsymbol{H}_{X,k}$	$\left.\frac{\partial h(\boldsymbol{x_k}, \boldsymbol{v_k})}{\partial \boldsymbol{x_k}}\right\|_{\hat{X}_{k\|k-1}, 0}$	Jacobian of measurement model linearized about the current state prediction
$\boldsymbol{H}_{v,k}$	$\left.\frac{\partial h(\boldsymbol{x_k}, \boldsymbol{v_k})}{\partial \boldsymbol{v_k}}\right\|_{\hat{X}_{k\|k-1}, 0}$	Jacobian of the measurement noise model linearized about the current state prediction

From the linearized equations the Extended Kalman Filter shows the local linearization done at each timestep to estimate information about the system. For the Extended Kalman Filter the prediction and correction steps are:

Prediction:

State prediction

$$\boldsymbol{\hat{X}_{k|k-1}} = f(\boldsymbol{x_{k-1}}, \boldsymbol{u_k}, 0) \tag{15}$$

Covariance Prediction

$$\widehat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}} = \boldsymbol{F}_{X,k}\,\widehat{\boldsymbol{P}}_{\boldsymbol{k}-\mathbf{1}}\boldsymbol{F}_{X,k}{}^{\boldsymbol{T}} + \boldsymbol{F}_{\boldsymbol{w},\boldsymbol{k}}\boldsymbol{Q}_{\boldsymbol{k}}\boldsymbol{F}_{\boldsymbol{w},\boldsymbol{k}}{}^{\boldsymbol{T}} \tag{16}$$

Correction:

Kalman Gain:

$$\boldsymbol{K}_{\boldsymbol{k}|\boldsymbol{k}} = \widehat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}}\boldsymbol{H}_{\boldsymbol{X},\boldsymbol{k}}^{T}\left(\boldsymbol{H}_{\boldsymbol{X},\boldsymbol{k}}\widehat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}}\boldsymbol{H}_{\boldsymbol{X},\boldsymbol{k}}^{T} + \boldsymbol{H}_{\boldsymbol{v},\boldsymbol{k}}\boldsymbol{R}_{\boldsymbol{k}}\boldsymbol{H}_{\boldsymbol{v},\boldsymbol{k}}^{T}\right)^{-\mathbf{1}} \tag{17}$$

Updated State Estimate

$$\widehat{\boldsymbol{X}}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}} = \widehat{\boldsymbol{X}}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}} + \boldsymbol{K}_{\boldsymbol{k}|\boldsymbol{k}}(\boldsymbol{Z}_{\boldsymbol{k}} - h(\boldsymbol{x}_{\boldsymbol{k}}, \mathbf{0})) \tag{18}$$

Covariance Update:

$$\widehat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}} = (\boldsymbol{I} - \boldsymbol{K}_{k|k}\boldsymbol{H}_{\boldsymbol{X},\boldsymbol{k}}^{T})\widehat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}} \tag{19}$$

When applied to the swinging pendulum example the result for estimating θ using an Extended Kalman Filter:

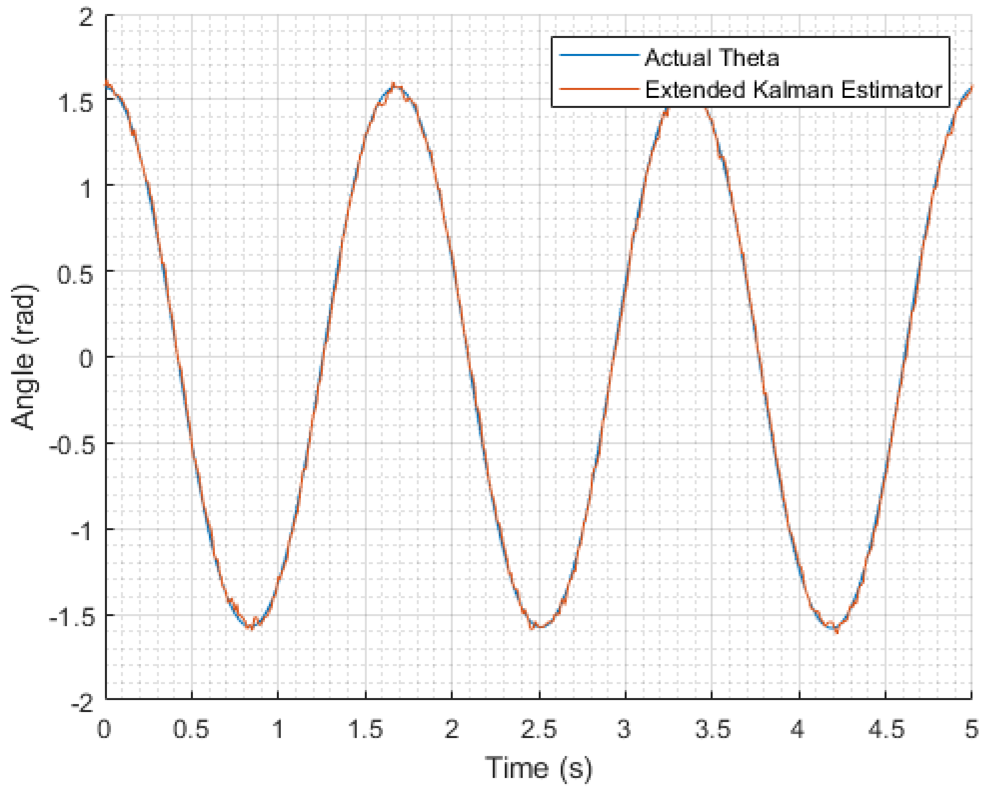

Figure 2. Pendulum angle estimate starting at $\pi/2$.

Assuming a $[\theta, \dot{\theta}] = [\frac{\pi}{2}, 0]$ starting condition for the predictor, the Extended Kalman Filter does a good job at estimating a large starting angle which would not be possible using a linear Kalman estimator.

Conclusion

Consider the system equations when implementing a Kalman Estimator. There are instances where a linearized model is useful since the Jacobian calculations are not needed. When using an Extended Kalman Filter consider if the analytical Jacobian can be calculated. In the case where there is no analytical solution a numerical approximation can be calculated however numerical solutions can be computationally extensive.

Solutions for teaching and research. Made in Canada.