Linear Kalman Filter Tutorial

Leo Wilson

August 27, 2020

1 What is a Kalman Filter

A Kalman Filter is a model-based estimation algorithm. Why model-based? Because it uses a model of the system and using that model takes in noisy observations from which it estimates the current "state" of the system. The linear system the Kalman Filter is based on:

The state, $\bar{x}$, evolution over time:

$$\bar{x}_k = F\bar{x}_{k-1} + w, \text{ and } w \sim N(0, Q) \tag{1}$$

The observations, $\bar{z}$, as a function of the state:

$$\bar{z}_k = H\bar{x}_k + \nu, \text{ and } \nu \sim N(0, R) \tag{2}$$

An estimate of the uncertainty, P, of the state:

$$P_k = cov(\bar{x}_k) \tag{3}$$

Where:

F is known as the state transition matrix. F "takes" the state $\bar{x}$ from one time step to the next.
H transforms the state space, $\bar{x}_k$ into the observation (measurement) space of $\bar{z}_k$ to.
Q is a covariance matrix that models the errors (or uncertainty) in the state dynamics model.
R is a covariance matrix that models the errors in the observations.

Each of these models, i.e. each matrix, will be described in more detail in section 2.

1.1 The two steps in the Kalman Filter

1.1.1 Time Update (Prediction)

Prediction is done whenever the state must be propagated forward in time.

Using the previous state, $\bar{x}_{k-1}$ and the state transition model, F, predict the state, $\hat{x}$, at time k.

$$\hat{x}_k = F\bar{x}_{k-1} \tag{4}$$

Using the previous uncertainty, P_{k-1}, and the state transition model, predict the uncertainty, $\hat{P}$ at time k.

$$\hat{P}_k = FP_{k-1}F^T + Q \tag{5}$$

Some key points about the prediction step are:

- The system noise covariance, Q from equation (1) is accounted for when propagating the covariance, P.
- Prediction provides the best estimate of the state at the current time (or some future time).
- Prior to updating the state with a new measurement, prediction is used to "align" the state with the measurement in time.

1.1.2 Observation Update

An update of the Kalman Filter is done whenever a new measurement is available. After propagating the state to the current measurement time, using the prediction step above, the uncertainties of the system and the measurement are used to compute the Kalman gain, K.

$$K = \hat{P}_k H^T (H \hat{P}_k H^T + R)^{-1} \tag{6}$$

Then the state is updated based on the difference between the measurement and what the Kalman Filter predicted the measurement should be (i.e. the expected measurement).

$$\bar{x}_k = \hat{\bar{x}}_k + K(\bar{z}_k - H\hat{\bar{x}}_k) \tag{7}$$

We also update the Kalman Filter's estimate of uncertainty.

$$P_k = (I - KH)\hat{P}_k \tag{8}$$

Note that the covariance update can be written as: $P_k = \hat{P}_k - KH\hat{P}_k$. From this one can see that the covariance is reduced as a result of incorporating a new measurement.

1.2 Important Terms from the Observation Update

The residual (or innovation) is the difference between the measurement and what the Kalman filter "expected" the measurement to be, just prior to incorporating the new observation.

$$\bar{z}_k - H\hat{\bar{x}}_k \tag{9}$$

The innovation covariance is the total uncertainty of the state and covariance. It is defined to be in the observation space.

$$H\hat{P}_k H^T + R \tag{10}$$

Here innovation is a statistical term that refers to the residuals in a time series that includes all prior information in its estimate up to but not including the current time (see `https://en.wikipedia.org/wiki/Innovation(signal_processing)`).

These terms are very useful when analyzing the performance of the filter in real-time or as a tool for tuning. Tuning will be discussed later in the document.

2 Example of a Simple Linear Kalman Filter

Now that we have an overview of the basics of the Kalman Filter algorithm, we will construct a simple linear Kalman Filter. The filter model is based on constant acceleration dynamics and it will take in a single observation of position. The following discussion will include defining the state, $\bar{x}$, the dynamics model, F and Q, as well as the observation model, $\bar{z}$ and R and H. Much of what follows will be applicable to the asteroids project.

2.1 State Space: $\bar{x}$

We start by defining the state space. The object being observed by the Kalman Filter accelerates in a single dimension, x, and its position is measured periodically. So a good choice for the state might be:

$$\bar{x} = [x, \dot{x}, \ddot{x}]^T \tag{11}$$

In this model, the entire state is represented by the vector $\bar{x}$ while the position and its derivatives are represented by the scalars x, $\dot{x}$ (velocity), and $\ddot{x}$ (acceleration). So given 3 state variables, the state has dimension 3. We will refer to the state dimension in general as n.

To start the filter off requires an initial value for $\bar{x}$. This can be set a number of ways, perhaps most commonly using 1 or more of the first observations. For the simple Kalman Filter observing position only, the position element could be a average of the first one or two observations. But what about the velocity and acceleration terms? These can be initialized using differences of initial measurements or simply set to 0, letting the filter figure them out over time. Naturally the more closely the initial state is set to the truth, the more quickly the filter can converge on a quality estimate. The bottom line though is that the initialization must be based on the actual physics of the problem. For instance, if the total space the object is operating in is +-1, as it is in the asteroids project, the initial position should be within those bounds.

2.2 State Dynamics Model: F

Given the state space chosen for our linear Kalman Filter the state dynamics model encoded in F must be derived. Recall that a constant acceleration model (think of the example of a falling object from physics) is:

$$x_t = x_{t-1} + \dot{x}_{t-1}\Delta t + \ddot{x_{t-1}}\frac{\Delta t^2}{2} \tag{12}$$

The state transition model F has dimension $nxn = 3x3$ and (for reasons we won't go into here) encodes a consistent set system of differential equations. These can be found from the constant acceleration model as follows. Starting from equation (12), the first derivative with respect to time (velocity) is:

$$\dot{x}_t = \dot{x}_{t-1} + \ddot{x}_{t-1}\Delta t \tag{13}$$

and the second derivative with respect to time (acceleration) is:

$$\ddot{x}_t = \ddot{x}_{t-1} \tag{14}$$

Let's stack these equations and rearrange and group the Δt terms slightly for clarity. We'll also multiply the state elements that are not included in a derivative by 0. Doing this we can almost see F directly.

$$\begin{array}{rclcrcl} x_t & = & (1)x_{t-1} & + & (1\Delta t)\dot{x}_{t-1} & + & (1\frac{\Delta t^2}{2})\ddot{x}_{t-1} \\ \dot{x}_t & = & (0)x_{t-1} & + & (1)\dot{x}_{t-1} & + & (1\Delta t)\ddot{x}_{t-1} \\ \ddot{x}_t & = & (0)x_{t-1} & + & (0)\dot{x}_{t-1} & + & (1)\ddot{x}_{t-1} \end{array} \tag{15}$$

The coefficients for the state variables in each row of (15), i.e. those values in parentheses, are used to fill in each row of F. And the entire state transition model from equations (1) and (4) is:

$$\begin{bmatrix} x_t \\ \dot{x}_t \\ \ddot{x}_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \\ \ddot{x}_{t-1} \end{bmatrix} \tag{16}$$

Note that in the asteroids project, the following hold:

- F is the same for all asteroids for all time.
- Δt is always 1.0 units of simulation time.
- This dynamics model will need to be extended to account for two spatial dimensions, x and y.

2.3 State Uncertainty Model: P

The Kalman filter keeps an estimate of its uncertainty in $\bar{x}$ with a covariance matrix, P, defined in equation (3) and implemented in equations (5) and (8), the time update and observation update respectively. P will be of dimension nxn where n is the state dimension and must be initialized to reflect the initial uncertainty in the state estimate.

The diagonal terms of P are the variances (also referred to as covariances) of the uncertainties for each state element. Remember that variances are standard deviations squared (i.e. σ^2) and the quantities one generally thinks in are standard deviations, $\sigma's$. So when analyzing or setting the initial diagonal values in P, be sure to remember to square or take a square root as necessary. Note, because the correlation terms of the initial

uncertainty are seldom known, P is generally initialized as a diagonal matrix and for our simple example will look like:

$$P_0 = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & \sigma_{\ddot{x}}^2 \end{bmatrix}$$

When initializing the diagonal elements of P, almost never will an initial σ be set to 0 because that implies perfect information about that state element at the beginning of the estimation process. Having perfect knowledge of a state is unlikely because one would most likely not include some perfectly known quantity into the state. Further, especially when getting into nonlinear filtering, if the state information is not perfect, but the uncertainty is set to 0 (or just too small a value) it might cause the filter to be unstable. Generally, one has some idea of the initial uncertainty of the state estimate based on the physics of the measurement process and the world the sensor is operating in. For instance for the asteroids problem, the world is +1.0 and so the initial uncertainty in position (as well as velocity and acceleration) should be consistent with that. It is generally a good idea to set the initial values a little bit higher than expected; the Kalman filter will fairly quickly correct these confidences, as long as they are not orders of magnitude too large.

One final thought relates to the way the state, $\bar{x}$ is initialized. The values for the initial $\sigma's$ should be consistent with the method chosen for initializing $\bar{x}$. For instance, if several measurements of position were averaged, perhaps the initial σ_x could be smaller than if a single value is used.

2.4 State Dynamics Uncertainty Model: Q

Uncertainty or error in the state dynamics is modeled using another covariance matrix, Q, an additive noise process shown in equation (1) and implemented in equation (5). An example of where system dynamics error could arise would be modeling a constant acceleration process with a constant velocity model, for instance in our example using 2 states $\bar{x} = [x, \dot{x}]^T$, vs 3 states $\bar{x} = [x, \dot{x}, \ddot{x}]^T$. Another example could be using reduced precision floating point that introduces numerical noise into the process. Often, and certainly in this class, Q is assumed to be diagonal because the off-diagonal correlations may be insignificant.

$$Q = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & \sigma_{\ddot{x}}^2 \end{bmatrix} \tag{17}$$

Given some insight into the errors due to choice of dynamics model, one can estimate what the values of Q should be. In our case Q could be a constant matrix where the values are chosen through a tuning process (discussed later). Because Q is a covariance matrix, the covariances (or variances) are contained on the diagonal, just as described for P.

The intuition in using Q is that if its values are large, we have low confidence in the dynamics model and will tend to put more weight on the measurements, so the estimate will follow the measurements, including their noise, more closely. Conversely, if the values of Q are small, we believe the state dynamics model and tend to put less weight on the measurements and more on the filter's previous estimate of the state, resulting in "smoother" estimates but slower convergence.

For the asteroids problem, because the simulation does not introduce significant noise, if the state dynamics are modeled with a constant acceleration model, Q may not be necessary at all. However, if found necessary, Q can be set once and used for all asteroids.

2.5 Observation Model: $\bar{z}$ and H

The second important "model" required for the Kalman Filter is the observation model. Recall from equation (2) that it is defined as $\bar{z}_k = H\bar{x}_k$. Here again noise is not added to this step; the noise R is accounted

for when computing the Kalman Gain in equation (7).

The example filter will be provided a measurement of the position, while the derivative terms, $\dot{x}$ and $\ddot{x}$ will be estimated over time by the filter. To map the observation of position into the state is a simple linear operation. We are observing the 1st state element directly, so $H = [1, 0, 0]$. It is easy to see that this is correct by writing the matrix equations out.

$$\begin{bmatrix} z_x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{18}$$

Extension to a system where both x and y are part of the state and are observed would look like:

$$\begin{bmatrix} z_x \\ z_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{19}$$

2.6 Observation Uncertainty: R

Recall from equation (2) that the measurement relates to the state by: $\bar{z}_k = H\bar{x}_k + \nu$, where $\nu \sim N(0, R)$, and R is used in the innovation covariance when computing the Kalman Gain in equation (6). So R is also a covariance matrix and it is used to model the measurement uncertainty. If the measurements are independent, and in this class they generally will be, R will be a diagonal matrix in which the diagonals will be the observation uncertainties (squared). For the example filter observing only position this is:

$$R = \begin{bmatrix} \sigma_{z,x}^2 \end{bmatrix} \tag{20}$$

For an uncorrelated two dimension observation of x and y:

$$R = \begin{bmatrix} \sigma_{z,x}^2 & 0 \\ 0 & \sigma_{z,y}^2 \end{bmatrix} \tag{21}$$

If the measurements are not independent, either the off-diagonal terms must be accounted for or the filter redesigned using some linear algebraic techniques to diagonalize R.

When choosing the values for R one generally starts with the sensor specifications to set the initial values. These values, must reflect the sensor capability as well as the world the sensor is operating in. Remember that for the asteroids problem, the world is +- 1, so the σ^2 in R should be consistent with that.

There are various ways to get initial values for the asteroids project. One could be to examine the magnitudes of noise added in the test cases and use values around the largest. Another might be to get several measurements and estimate the magnitude of the variance from them. One could also use the residuals in the Kalman Filter to iteratively refine the estimates of R over time. Often, the simplest approach works out well.

The intuition behind R is that smaller values in R mean there is more confidence in the measurement and so the filter will tend to follow the measurements more closely, including any noise in them. Conversely, larger values of R will cause the filter estimate to be smoother but may converge more slowly.

3 Tuning the Kalman Filter

Now we're at the point where the filter has been designed and implemented and is now being tested in simulation or with real world data? In this section we discuss one method of analyzing performance that works whether or not truth data is available.

Recall, the innovation (or residual) in equation (9) is defined as:

$$\Delta\bar{z} = \bar{z}_k - H\hat{\bar{x}}_k \tag{22}$$

That is it is the difference between what the measurement is expected to be and what the Kalman Filter "thinks" the measurement should be. This quantity is in the measurement space after a time update and prior to a measurement udpate.

Also, recall from equation (10) the innovation covariance is defined as:

$$S = H\hat{P}_k H^T + R \tag{23}$$

and that it reflects the total system uncertainty of the system and measurement in the measurement space after a time update and prior to a measurement udpate.

Using these two terms allows one to evaluate how well the Kalman Filter is doing at predicting both future state values and its uncertainty regarding those predictions.

3.1 χ^2 Statistic

A useful metric, which can also be used as a data editing method to reject measurement outliers in real-time, is to evaluate the χ^2 statistic associated with the current measurement. This is computed as:

$$\chi^2 = \Delta\bar{z}^T S^{-1} \Delta\bar{z} \tag{24}$$

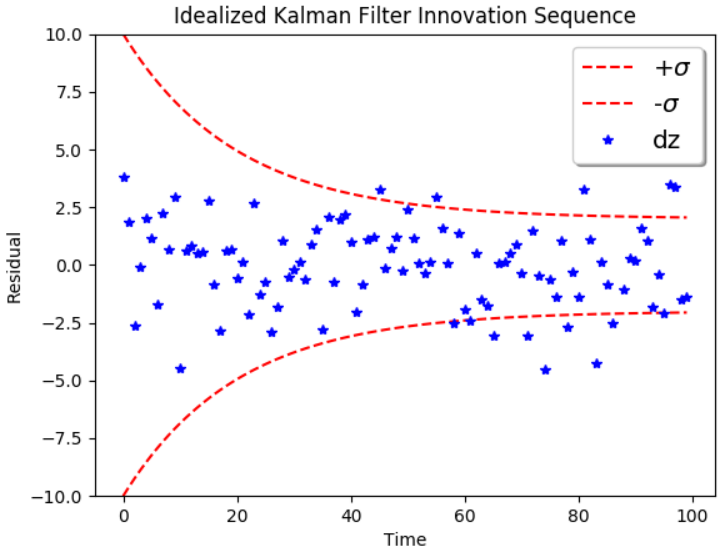
For a single dimensional observation this simplifies to:

$$\chi^2 = \Delta z^2 / S(1,1) \tag{25}$$

For real time data editing, logic can be added so that if this χ^2 value exceeds some threshold, the measurement is rejected.

3.2 Innovation Sequence

To use the innovation terms to support tuning, a chart can be generated to compare the residual vs the square root of the innovation covariance for each observation dimension. For a properly performing filter the residuals should be a white noise process with about two thirds of the values falling within an envelope of +- the square root of the innovation covariance (a 1 σ value). An idealized sequence is shown in the figure below.



Notice that the residuals are approximately randomly and evenly distributed about 0 with a $\sigma = 2$ throughout. This reflects a fixed measurement uncertainty. Also notice that the innovation covariance "contains" approximately 2/3 of the residuals even after it has converged. This indicates that the filter is maintaining a realistic uncertainty since the square root of the diagonals in S are the filter's estimate of what the 1 σ uncertainty.

Following are suggestions on how to use the chart in the tuning process.

What to look for:

- The square root of the innovation covariance should start out large and reduce to a steady state (if the measurement uncertainty is constant)
- The square root of the innovation covariance is/should be a 1 σ bounds on the residuals
- The residuals should be a white noise sequence, i.e. evenly and randomly distributed about 0.

Rules of thumb:

What if the residual sequence is not nicely distributed as a white noise sequence?

- Make sure that R is large enough based on sensor specs etc.

What are the steps to take if the $+ - \sigma$ is too small?

- Make sure that R is large enough based on sensor specs etc.
- Incrementally add noise to Q, starting with higher derivative terms. item.
- Make sure the initial covariance P is large enough.

What about if $+ - \sigma$ is far too large?

- Make sure that R is not too conservative.
- Incrementally reduce noise from Q starting with the lower order derivatives.
- Be sure the initial covariance is large enough but not ridiculously large.

Intuitions to work with:

- The intuition behind R is that smaller values in R mean that there is more confidence in the measurement and so the filter will tend to follow the measurements more closely, including any noise in them. Conversely, larger values of R will cause the filter estimate to be smoother but its error may not reduce as quickly.
- The intuition in using Q is that if its values are large, we have low confidence in the dynamics model and will tend to put more weight on the measurements, so the estimate will follow the measurements, including their noise, more closely. Conversely, if the values of Q are small, we believe the state dynamics model and tend to put less weight on the measurements and more on the filter's previous estimate of the state, resulting in "smoother" estimates but slower convergence of the filter.
- When setting the initial (or constant) values in the various uncertainty matrices and the initial state, be sure they reflect the physics of the environment and sensor.