Lecture Notes: Kalman filter

Last lecture we developed matrix notation for filtering. We also looked at the weights for the state update equation, and showed the Kalman formulation for an arbitrary number of variables. Now we will put it all together and show the Kalman filter equations implemented in practice.

Recall that the Kalman filter is a continuous cycle of predict-update. The following equations form the main loop of the filter:

1. predict next state
$$X_{t,t-1} = \Phi X_{t-1,t-1} \tag{1}$$
2. predict next state covariance
$$S_{t,t-1} = \Phi S_{t-1,t-1} \Phi^T + Q \tag{2}$$
3. obtain measurement(s) Y_t
4. calculate the Kalman gain (weights)
$$K_t = S_{t,t-1} M^T [M S_{t,t-1} M^T + R]^{-1} \tag{3}$$
5. update state
$$X_{t,t} = X_{t,t-1} + K_t (Y_t - M X_{t,t-1}) \tag{4}$$
6. update state covariance
$$S_{t,t} = [I - K_t M] S_{t,t-1} \tag{5}$$
7. loop (now t becomes $t+1$)

The output from the filter is the result of the state update and state covariance update equations. These provide the combined estimate from the model (prediction equations) and latest observation (measurements). The state matrix provides the mean value of the distribution for each state variable, and the covariance matrix provides the variances.

In calculating the gain and update, it can be helpful to break the calculations in the following meaningful manner. First, calculate the "innovation", which is the difference between the new data and the previous prediction:

$$J = Y_t - M X_{t,t-1} \tag{6}$$

Then calculate the covariance of the innovation:

$$\mathrm{COV}(J) = MS_{t,t-1}M^T + R \tag{7}$$

The gain matrix can then be calculated as

$$K_t = S_{t,t-1}M^T\mathrm{COV}(J)^{-1} \tag{8}$$

and the updated state can be calculated as

$$X_{t,t} = X_{t,t-1} + K_t J \tag{9}$$

To get into the loop, several matrices must be initialized. The initial values of the state variables $X_{0,0}$ can be set to reasonable values according to the first measurement (or first few measurements). They can also be set to zero or some other starting value, but in this case the filter will take some time (a number of iterations) to converge until tracking is stable. The covariance of the state can be initialized similarly, using for example the dynamic noise covariance matrix Q or the identity matrix. It will also take some time to become stable.

The measurement noise covariance R can be calculated via calibration. The sensing system can be operated to take a large number of readings of a known ground truth state, from which the variance(s) can be calculated. For example, the measurement noise for a radar could be calculated by taking a number of readings of a plane at a fixed, known distance. The variance of the measurements provides the value of σ_n^2 in R.

The dynamic noise covariance Q can be harder to determine. Some state transition equations may have a literal interpretation that can be used to place bounds on the expected dynamic noise. For example, using a constant velocity model with dynamic noise interpreted as acceleration, the known maximum acceleration of the target could be used to calculate 3 sigma in σ_a^2 in Q.

In practice, what matters is the relative ratio of the measurement noise to the dynamic noise. This is what determines how the Kalman gain (weights) are calculated. Therefore it is common to fix one of the noise covariance matrices and adjust the other until the desired performance is achieved. For example, one can fix the measurement noise and turn the dynamic noise up/down to achieve more/less smoothing. This process will be demonstrated in class.

A filter is not intended to run for just a few iterations. It takes at least a few iterations for the state and state covariances to stabilize. The performance of a filter should not be evaluated over its first few iterations, and sometimes not until after tens or hundreds of iterations.

Note that we have defined Q and R as constant while the filter is running. If the measurement and dynamic noise are in fact constant, then this is okay. However, in many problems, these noises change over time, and the matrices are updated as the filter runs.

If Q and R are constant, then K and S will also converge to a constant. This is because the gain matrix and state covariance matrix are not affected by the measurements. They assume that the relative noise distributions are given in Q and R. Their job is simply to combine those matrices for use in the estimate of state.

The Kalman filter can be applied to problems even when the basic assumptions are not true (when the system is not exactly linear, or when noise models are not exactly Gaussian).

It is possible for the Kalman filter to still give reasonable results, but there are other filters that are better for non-linear and non-Gaussian systems. These will be discussed in future lectures.