Kalman Filter Algorithm

Note: This section is currently under revision.

This section covers the Kalman Filter Algorithm. First we'll cover the State Space format of modeling and measuring a discrete-time dynamic system of estimated states, noisy inputs, and noisy measurements. Second, we'll explore all the different pieces of information about our system necessary to inform the algorithm. Third, the specific Kalman Filter Algorithm constructed based off of those parameters. Finally, we'll use some example state spaces and measurements to see how well we track.

Note: all images below have been created with simple Matlab Scripts. If seeing the code helps clarify what's going on, the .m files can all be found under internal location cs:localization:kalman.

Section 1 - State Space Format

The State Space Format is a universally standardized format for dynamic systems in the signals and controls community. In the continuous-time domain, the derivative of the state $\dot{x}(t)$ is a linear function of $x(t)$. In the discrete-time domain, where we'll be operating, the next state $x[k+1]$ is a linear function of the current state $x[k]$.

$$ X_{k+1} = AX_k + B(u_k + \sim\mathcal{N}(0,Q^2)) \\ Y_k = CX_k + \sim\mathcal{N}(0,\sigma_m^2) $$

Vector X_k is the State Vector which contains all states of the system at time-step k. These include things like position, velocity, orientation, voltage, etc. Matrix A is the Transmission Matrix, which contains the dynamics of the system, and calculates the next state given the current state. B is the input matrix, which describes the dynamics of inputs u.

Vector Q is the Process Noise of the system, which is the combination of the variance of the inputs σ_u^2 and an estimated degree of possible external forces σ_{ext}^2. $Q^2 = \sigma_u^2 + \sigma_{ext}^2$ Random forces from bumping into walls, random currents in the water, diver interaction, and other unpredictable perturbations. Put another way, Q describes the uncertainty involved when predicting into to future, even given perfect information about the present State. Were there no external forces, perfect actuators, and a perfect initial state x_0, the system state could be predicted perfectly into the future. This is obviously not the case in the real world, hence the need to specify uncertainty in the model, and thus in predicting the future states.

Vector Y_k is the measurement vector. It contains the values taken from the sensors. Matrix C is the Emission Matrix, which describes the linear function that relates the system state to the measurement values. σ_m is the noise vector which describes how noisy each individual sensor measurement is.

Section 2 - Kalman Filter Algorithm

The Kalman Filter is a two-stage process of prediction and measurement. First, based on the previous state estimate $\hat{X}_{k-1}$ and inputs u_{k-1}, an initial current state estimate $\hat{X}_{k'}$ is predicted. The confidence of the Previous Estimate is contained in the Covariance Matrix P From this estimate, a further estimate $\hat{Y}_k$ of what the sensors *should* be reading given $\hat{X}_{k'}$ is calculated using the Emission Matrix C.

Second, the true, noisy measurements Y_k are received and compared with the expected sensor values $\hat{Y}_k$. A compromise between the noisy measurements and the expected measurement is arrived at based upon the noise of the sensors σ_m and the uncertainty of the measurement predictions calculated from P. This compromise is encapsulated in a value $\hat{K}$ terms the 'Kalman Gain'. The official $\hat{X}_{k}$ is then calculated from the Emission Matrix C and the compromise of sensor values. The estimate of our state for this time step is made, and the process repeats to estimate the state X_{k+1} at the next time step.

$$ \hat{X}_{k'} = A\hat{X}_{k-1} + Bu_{k-1}, \qquad P_{k'} = AP_{k-1}A' + Q^2 \\ \hat{K} = P_{k'}H'/(HPH'+diag(\sigma_m^2)) \\ \hat{X}_k = \hat{X}_{k'} + \hat{K}(Y_k - C\hat{X}_{k'}),\qquad P_k = (I_{n} - \hat{K}C)P_{k'} $$