Machine Learning.

Kalman Filtering (KF):

Covariance matrix

$$x_{k+1} = \Phi x_k + w_k \quad , \quad w_k \sim N(\underline{0}, Q_k).$$

$$z_k = H x_k + v_k \quad , \quad v_k \sim N(\underline{0}, R_k)$$

Covariance matrix

$\hat{x}_k$: Estimation on x_k once we get z_k

$\hat{x}'_k$: " " " before getting z_k.

Kalman Filter Recursive Algorithm:

Kalman Gain: $K_K = P_K' \mathcal{H}^T (\mathcal{H} P_K' \mathcal{H}^T + R)^{-1}$

Update Estimate: $\hat{x}_K = \hat{x}_K' + K_K (I - K_K \mathcal{H}) P_K'$

Update Covariance: $P_K = (I - K_K \mathcal{H}) P_K'$

Project into K+1 : $\begin{cases} \hat{x}_{K+1}' = \Phi \hat{x}_K \\ P_{K+1} = \Phi P_K \Phi^T + Q \end{cases}$

EX.

$$\begin{cases} \underline{x}[n+1] = \underset{3\times 3}{A}\, x[n] + B\, \overset{\text{input}}{u[n]} + G\, \overset{\text{noise}}{w[n]}. & : \text{State Eq.} \\ \underset{1\times 1}{y[n]} = \underset{1\times 3}{C}\, \underset{3\times 1}{x[n]} + \underbrace{D u[n]}_{\to 0} + v[n]. & : \text{Measurement Eq.} \end{cases}$$

$$w[n] \sim N(\underline{0}, Q). \qquad v[n] \sim N(\underline{0}, R). \qquad \begin{cases} Q: 2.3 \\ R: 1 \end{cases}$$

Generate a random number from a normal (Gaussian distribution

randn $\qquad N(0,1)$

$N(a,b) \rightarrow a + \sqrt{b}$ randn

3 + randn $\qquad N(3,1)$.

3 + randn(5,1)

$3 + \sqrt{10}$ randn $\qquad N(3,10)$.