

Kalman Filter

선형 시스템 모델

$$\mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k$$

■ 0. 초기값 선정

$$\hat{\mathbf{x}}_0, \mathbf{P}_0$$

■ 1. 추정값 예측, 오차 공분산 예측

$$\hat{\mathbf{x}}_k^{\dagger} = \mathbf{A} \hat{\mathbf{x}}_{k-1}$$

$$\mathbf{P}_k^{\dagger} = \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q}$$

■ 2. 칼만 이득 계산

$$\mathbf{K}_k = \mathbf{P}_k^{\dagger} \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^{\dagger} \mathbf{H}^T + \mathbf{R})^{-1}$$

■ 3. 추정값 계산 (측정값 $\mathbf{z}_k \rightarrow$ 추정값 $\hat{\mathbf{x}}_k$)

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{\dagger} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^{\dagger})$$

■ 4. 오차 공분산 계산

$$\mathbf{P}_k = \mathbf{P}_k^{\dagger} - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^{\dagger}$$

Kalman Filter Algorithm

■ 0. 초기값 선정

$$\hat{\mathbf{x}}_0, \mathbf{P}_0$$

■ 예측과정

■ 1. 추정값 예측, 오차 공분산 예측

$$\hat{\mathbf{x}}_k^{\dagger} = \mathbf{A} \hat{\mathbf{x}}_{k-1}$$

$$\mathbf{P}_k^{\dagger} = \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q}$$

■ 추정과정

■ 2. 칼만 이득 계산

$$\mathbf{K}_k = \mathbf{P}_k^\dagger \mathbf{H}^\top (\mathbf{H} \mathbf{P}_k^\dagger \mathbf{H}^\top + \mathbf{R})^{-1}$$

■ 3. 추정값 계산 (측정값 $z_k \rightarrow$ 추정값 $\hat{x}_k$)

$$\hat{x}_k = \mathbf{x}_k^\dagger + \mathbf{K}_k (z_k - \mathbf{H} \mathbf{x}_k^\dagger)$$

■ 4. 오차 공분산 계산

$$\mathbf{P}_k = \mathbf{P}_k^\dagger - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^\dagger$$

■ 용도별 변수

■ 외부입력

z_k (측정값)

■ 최종 출력

$\hat{x}_k$ (추정값)

■ 시스템 모델

$\mathbf{A}, \mathbf{H}, \mathbf{Q}, \mathbf{R}$

■ 내부 계산용

$\mathbf{x}_k^\dagger, \mathbf{P}_k^\dagger, \mathbf{P}_k, \mathbf{K}_k$

■ 추정값과 오차 공분산

$\mathbf{x}_k$ = 상태변수 추정값

$\mathbf{x}_k^\dagger$ = 상태변수 예측값

$\mathbf{P}_k$ = 오차 공분산 추정값

$\mathbf{P}_k^\dagger$ = 오차 공분산 예측값

■ 추정값 계산

$$\hat{x}_k = \mathbf{x}_k^\dagger + \mathbf{K}_k (z_k - \mathbf{H} \mathbf{x}_k^\dagger)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{\dagger} + \mathbf{K}_k \mathbf{z}_k - \mathbf{K}_k \mathbf{H} \hat{\mathbf{x}}_k^{\dagger}$$

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \hat{\mathbf{x}}_k^{\dagger} + \mathbf{K}_k \mathbf{z}_k$$

$$\hat{\mathbf{x}}_k = \alpha \hat{\mathbf{x}}_{k-1} + (1 - \alpha) \mathbf{x}_k$$

$$\alpha = 1 - K$$

$$\hat{\mathbf{x}}_k = (1 - K) \hat{\mathbf{x}}_{k-1} + K \mathbf{x}_k$$

$\mathbf{x}_k \sim \mathcal{N}[\hat{\mathbf{x}}_k, \mathbf{P}_k]$ (normal distribution)

■ 오차 공분산

$$\mathbf{P}_k = \mathbf{E} \{ (\mathbf{x}_k - \hat{\mathbf{x}}_k) (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \}$$

추정과정

■ 2. 칼만 이득 계산

$$\mathbf{K}_k = (\mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q}) \mathbf{H}^T (\mathbf{H} (\mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q}) \mathbf{H}^T + \mathbf{R})^{-1}$$

■ 3. 추정값 계산 (측정값 $\mathbf{z}_k \rightarrow$ 추정값 $\hat{\mathbf{x}}_k$)

$$\hat{\mathbf{x}}_k = \mathbf{A} \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \mathbf{A} \hat{\mathbf{x}}_{k-1})$$

$(\mathbf{z}_k - \mathbf{H} \mathbf{A} \hat{\mathbf{x}}_{k-1})$ 측정값의 예측 오차

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{A} \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k \mathbf{z}_k$$

■ 4. 오차 공분산 계산

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) (\mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q})$$

시스템 모델 - 실제 시스템 모델과 비슷하게 설계

$$\mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k$$

$\mathbf{x}_k$ = 상태변수 (state variable), $(n \times 1)$ Column vector

$\mathbf{z}_k$ = 측정값 (sensor), $(m \times 1)$ Column vector

$\mathbf{A}$ = 상태전이행렬 (System 운동방정식), $(n \times n)$ Matrix

$H = (m \times n)$ Matrix

$w_k =$ 잡음 (system white noise), $(n \times 1)$ Column vector

$v_k =$ 잡음 (sensor noise), $(m \times 1)$ Column vector

$Q = w_k$ 의 공분산 행렬, $(n \times n)$ diagonal matrix (대각행렬)

$R = v_k$ 의 공분산 행렬, $(m \times m)$ diagonal matrix

Extended Kalman Filter, EKF

비선형 시스템 $\rightarrow$ 선형화

선형화 칼만 필터 (Linearized Kalman Filter)

$$x_{k+1} = f[x_k] + w_k$$

$$z_k = h[x_k] + v_k$$

■ 0. 초기값 선정

$$\hat{x}_0, P_0$$

■ 1. 추정값 예측, 오차 공분산 예측

$$\hat{x}_k^+ = f[\hat{x}_{k-1}]$$

$$P_k^+ = A P_{k-1} A^T + Q$$

■ 2. 칼만 이득 계산

$$K_k = P_k^+ H^T (H P_k^+ H^T + R)^{-1}$$

■ 3. 추정값 계산 (측정값 $z_k \rightarrow$ 추정값 $\hat{x}_k$)

$$\hat{x}_k = \hat{x}_k^+ + K_k (z_k - h[\hat{x}_k^+])$$

■ 4. 오차 공분산 계산

$$P_k = P_k^+ - K_k H P_k^+$$

■ Jacobian Matrix

$$A \hat{x}_{k-1} \Leftrightarrow f[\hat{x}_{k-1}]$$

$$H \hat{x}_k^+ \Leftrightarrow h[\hat{x}_k^+]$$

$$A \equiv \frac{\partial f}{\partial x} [\hat{x}_k]$$

$$\mathbf{H} \equiv \frac{\partial \mathbf{h}}{\partial \mathbf{x}} [\mathbf{x}_k^\dagger]$$

Unscented Kalman Filter

선형화 (jacobian) 과정 생략

■ 0. 초기값 선정

$$\mathbf{x}_0, \mathbf{P}_0$$

■ 1. 시그마 포인트와 가중치 계산

$$(\chi_i, w_i) \leftarrow (\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}, \kappa)$$

■ 2. 추정값 예측, 오차 공분산 예측

$$(\mathbf{x}_k^\dagger, \mathbf{P}_k^\dagger) = \mathbf{UT}[\mathbf{f}[\chi_i], w_i, \mathbf{Q}]$$

■ 3. 추정값 예측, 오차 공분산 예측

$$(\hat{\mathbf{z}}_k, \mathbf{P}_z) = \mathbf{UT}[\mathbf{h}[\chi_i], w_i, \mathbf{R}]$$

■ 4. 칼만 이득 계산

$$\mathbf{P}_{xz} = \sum_{i=1}^{2n+1} w_i \{ \mathbf{f}[\chi_i] - \mathbf{x}_k^\dagger \} \{ \mathbf{h}[\chi_i] - \hat{\mathbf{z}}_k \}^\top$$

$$\mathbf{K}_k = \mathbf{P}_{xz} \mathbf{P}_z^{-1}$$

■ 5. 추정값 계산 (측정값 $z_k \rightarrow$ 추정값 $\hat{x}_k$)

$$\mathbf{x}_k = \mathbf{x}_k^\dagger + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

■ 6. 오차 공분산 계산

$$\mathbf{P}_k = \mathbf{P}_k^\dagger - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^\dagger$$

$$\mathbf{x} \sim \mathbf{N}[\mathbf{x}_m, \mathbf{P}_k]$$

Monte Carlo Simulation

$$\chi_1 = \mathbf{x}_m$$

$$\chi_{i+1} = \mathbf{x}_m + \mathbf{u}_i, \quad i = 1, 2, \dots, n$$

$$\chi_{i+n+1} = \mathbf{x}_m - \mathbf{u}_i, \quad i = 1, 2, \dots, n$$

$$W_1 = \frac{\kappa}{n + \kappa}$$

$$W_{i+1} = \frac{1}{2 (n + \kappa)}, \quad i = 1, 2, \dots, n$$

$$W_{i+n+1} = \frac{1}{2 (n + \kappa)}, \quad i = 1, 2, \dots, n$$

$$\chi = \text{kai}$$

$$\kappa = \text{kappa}$$

$$u_i = \text{U row vector}$$

Cholesky decomposition (분해)

$$U^T U = (n + \kappa) P_x$$

■ $y=f[x]$ 의 평균과 공분산 $\Rightarrow$ 시그마 포인트와 가중치

$$y_m = \sum_{i=1}^{2n+1} W_i f[\chi_i]$$

$$P_y = \sum_{i=1}^{2n+1} W_i \{f[\chi_i] - y_m\} \{f[\chi_i] - y_m\}^T$$

weighted mean

$$x_m = \sum_{i=1}^{2n+1} W_i \chi_i$$

$$P_y = \sum_{i=1}^{2n+1} W_i \{\chi_i - x_m\} \{\chi_i - x_m\}^T$$

$$P_{xz} = \sum_{i=1}^{2n+1} W_i \{f[\chi_i] - \hat{x}_k^T\} \{h[\chi_i] - \hat{z}_k\}^T$$

■ SigmaPoints

$$(\chi_i, W_i) \leftarrow (\hat{x}_{k-1}, P_{k-1}, \kappa)$$

$$n = \text{numel}[\hat{x}_{k-1}]$$

$$\chi_i = \text{Zeros}[n, 2n+1]$$

$$W = \text{Zeros}[2n+1, 1]$$

$$\chi_i(:, 1) = \hat{x}_{k-1}$$

$$W(1) = \frac{\kappa}{n + \kappa}$$

$$U = \text{Chol}[(n + \kappa) P]$$

$$\mathbf{U}^T \mathbf{U} = (\mathbf{n} + \kappa) \mathbf{P}$$

for $k = 1 : n$

$$\chi_{k+1} = \hat{\mathbf{x}}_{k-1} + \mathbf{U}_k^T$$

$$\mathbf{u}_i = \mathbf{U} \text{ row vector}$$

$$\chi_i (: , k + 1) = \hat{\mathbf{x}}_{k-1} + \mathbf{U} (k, :)^T$$

$$\mathbf{W} (k + 1) = \frac{1}{2 (\mathbf{n} + \kappa)}$$

for $k = 1 : n$

$$\chi_{n+k+1} = \hat{\mathbf{x}}_{k-1} - \mathbf{U}_k^T$$

$$\chi_i (: , n + k + 1) = \hat{\mathbf{x}}_{k-1} - \mathbf{U} (k, :)^T$$

$$\mathbf{W} (n + k + 1) = \frac{1}{2 (\mathbf{n} + \kappa)}$$

■ UT

$$(\hat{\mathbf{x}}_k^\dagger, \mathbf{P}_k^\dagger) = \mathbf{UT}[\mathbf{f}[\chi_i], \mathbf{W}_i, \mathbf{Q}]$$

$$(\hat{\mathbf{z}}_k, \mathbf{P}_z) = \mathbf{UT}[\mathbf{h}[\chi_i], \mathbf{W}_i, \mathbf{R}]$$

$$\chi_i = \chi (: , i)$$

$$\mathbf{W}_i = \mathbf{W} (i)$$

$$\mathbf{x}_m = \sum_{i=1}^{2n+1} \mathbf{W}_i \chi_i$$

($n \times n$) matrix

$$\mathbf{P}_y = \sum_{i=1}^{2n+1} \mathbf{W}_i \{ \chi_i - \mathbf{x}_m \} \{ \chi_i - \mathbf{x}_m \}^T$$