Optimal Estimation

Estimators

- LQR (linear quadratic regulator)

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t),$$

$$\mathbf{z}(t) = \mathbf{C}(t)\mathbf{x}(t) \qquad\qquad (\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t))$$

$$J = \frac{1}{2}\mathbf{x}^T(t_f)\mathbf{H}\mathbf{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left[\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)\right] dt$$

$\Rightarrow$

$$\boxed{\begin{aligned} &\mathbf{u}^*(t) = -\mathbf{R}^{-1}(t)\mathbf{B}(t)\mathbf{P}(t)\mathbf{x}(t) \equiv \mathbf{F}(t)\,\mathbf{x}(t) \qquad \textit{: state feedback controller} \\ &\dot{\mathbf{P}}(t) = -\mathbf{P}(t)\mathbf{A}(t) - \mathbf{A}^{\mathbf{T}}(t)\mathbf{P}(t) - \mathbf{Q}(t) + \mathbf{P}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{P}(t) \qquad \mathbf{P}(t_f) = \mathbf{H} \end{aligned}}$$

- Problem
  - Most often all information of state $\mathbf{x}(t)$ is not available (measurable) $(\mathbf{H}(t) \neq \mathbf{I})$
  - There is usually error in measurement of $\mathbf{x}(t)$

Estimators

- Approach 1
  - State feedback → output feedback : $\mathbf{u}(t) = \mathbf{F}(t)\mathbf{x}\,(t) \;\rightarrow\; \mathbf{u}(t) = \hat{\mathbf{F}}(t)\,\mathbf{z}(t)$
  - Output feedback controller is hard to design

- Approach 2
  - State feedback control
  - But "estimate" of the system states based on the measured output

- New plan : Separation principle

  S1. Develop estimate of $\mathbf{x}(t)$, called $\hat{\mathbf{x}}(t)$

  S2. Then switch from $\mathbf{u}(t) = \mathbf{F}(t)\mathbf{x}(t)$ to $\mathbf{u}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t)$

- How do we find $\hat{\mathbf{x}}(t)$? → state estimation problem

Estimation Scheme

- System model

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}\,, \quad \mathbf{x}(0) \text{ unknown}$$
$$\mathbf{z} = H\mathbf{x}$$

where

- $\mathbf{x} \in \mathcal{R}^n, \mathbf{u} \in \mathcal{R}^p, \mathbf{z} \in \mathcal{R}^m$
- $A = A(t),\ B = B(t),$ and $H = H(t)$ are known, possibly time-varying
- Control inputs $\mathbf{u} = \mathbf{u}(t)$ are known
- Outputs $\mathbf{z} = \mathbf{z}(t)$ are measurable and $H \neq I$

- Goal

Develop a dynamic system whose state

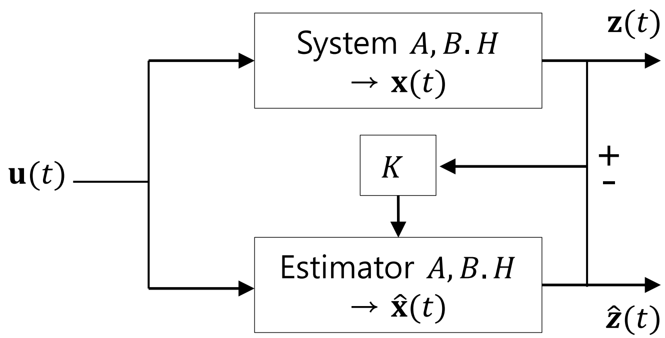
$$\hat{\mathbf{x}}(t) = \mathbf{x}(t) \quad \forall t \geq 0$$

Estimation Scheme

- Closed-loop estimator (observer)
  - Comparing the measured output ($\mathbf{z}$) with the estimated output ($\hat{\mathbf{z}}$)

$$\tilde{\mathbf{z}} = \mathbf{z} - \hat{\mathbf{z}} = H(\mathbf{x} - \hat{\mathbf{x}}) = H\tilde{\mathbf{x}}$$

  - Feedback $\tilde{\mathbf{z}}$ to estimator to improve the state estimation



  - Basic form of the estimator

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B\mathbf{u}(t) + K\tilde{\mathbf{z}}(t)$$

$$\hat{\mathbf{z}}(t) = H\hat{\mathbf{x}}(t)$$

where $K \in \mathcal{R}^{n \times m}$ is a user selectable gain matrix.

(analysis) $$\begin{aligned} \dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} &= [A\mathbf{x} + B\mathbf{u}] - [A\hat{\mathbf{x}} + B\mathbf{u} - K(\mathbf{z} - \hat{\mathbf{z}})] \\ &= A(\mathbf{x} - \hat{\mathbf{x}}) - KH\,(\mathbf{x} - \hat{\mathbf{x}}) \\ &= A\tilde{\mathbf{x}} - KH\tilde{\mathbf{x}} = (A - KH)\tilde{\mathbf{x}} \end{aligned}$$

$$\Longrightarrow \quad \boxed{\tilde{\mathbf{x}}(t) = e^{(A-KH)t}\,\tilde{\mathbf{x}}(0)}$$

Estimation Scheme

(Ex) $A = \begin{bmatrix} -1 & 1.5 \\ 1 & -2 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $H = [1 \quad 0]$ $x(0) = \begin{bmatrix} -0.5 \\ -1 \end{bmatrix}$

Assume that the initial conditions are not well known

⇨ Closed-loop estimator

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u} + K\tilde{\mathbf{y}} = A\hat{\mathbf{x}} + B\mathbf{u} + K(\mathbf{z} - \hat{\mathbf{z}})$$

$$= (A - KH)\hat{\mathbf{x}} + B\mathbf{u} + KH\mathbf{x}$$

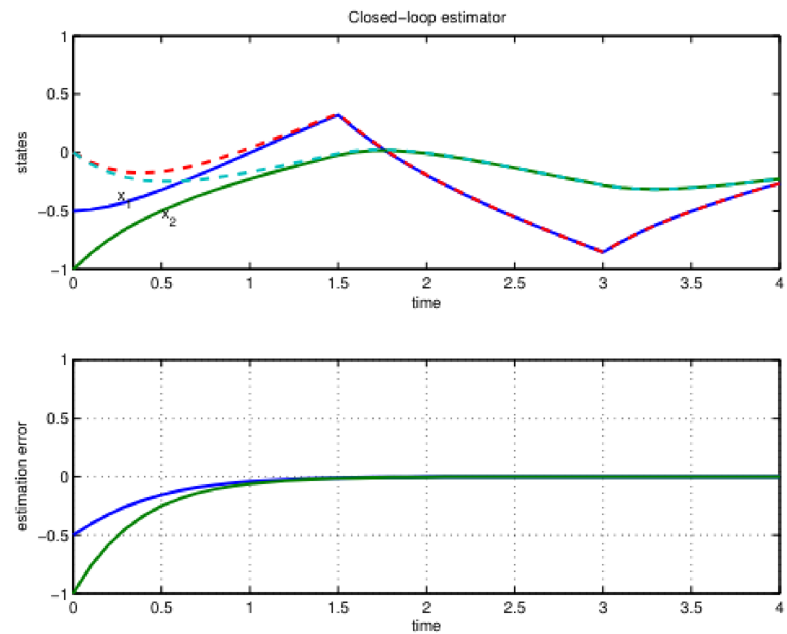
Overall system

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\dot{\hat{\mathbf{x}}} = (A - KH)\hat{\mathbf{x}} + B\mathbf{u} + KH\mathbf{x}$$

$$\rightarrow \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\hat{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ KH & A - KH \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u$$

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



How to determine K matrix ?

Optimal Estimator

- Random noise in the system

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + G\mathbf{w}$$
$$\mathbf{z} = H\mathbf{x} + \mathbf{v}$$

  - w: "process noise" – models uncertainty in the system model. $\mathbf{w} \in \mathcal{R}^{n\times 1}$
  - v: "sensor noise" – models uncertainty in the measurements. $\mathbf{v} \in \mathcal{R}^{m\times 1}$

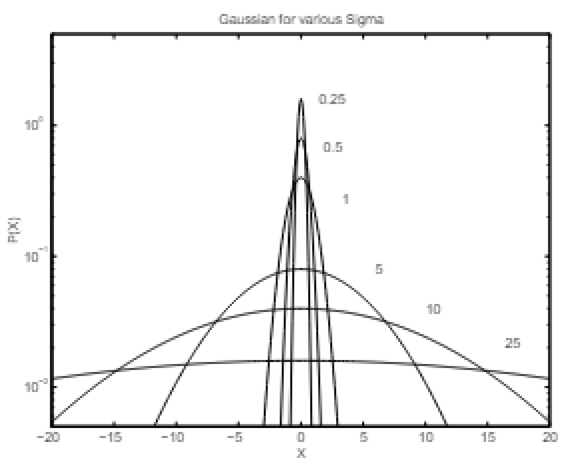
- Uncorrelated Gaussian white noise

$$E[\mathbf{w}(t)] = E[\mathbf{v}(t)] = 0$$

$$E[\mathbf{w}(t_1)\mathbf{w}(t_2)^T] = Q(t_1)\delta(t_1 - t_2) \;\rightarrow \mathbf{w}(t) \sim \mathcal{N}(0, Q)$$

$$E[\mathbf{v}(t_1)\mathbf{v}(t_2)^T] = R(t_1)\delta(t_1 - t_2) \;\rightarrow \mathbf{v}(t) \sim \mathcal{N}(0, R)$$

$$E[\mathbf{w}(t_1)\mathbf{v}(t_2)^T] = 0$$



- Closed-loop estimator

$$\boxed{\begin{aligned} \dot{\hat{\mathbf{x}}} &= A\hat{\mathbf{x}} + B\mathbf{u} + K(\mathbf{z} - \hat{\mathbf{z}}) \\ \hat{\mathbf{z}} &= H\hat{\mathbf{x}} \end{aligned}}$$

How to determine K matrix ?

Optimal Estimator

- Covariance of estimation error

$$\boxed{P(t) = E[\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}(t)^T]}$$

where
$$\begin{aligned}\dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} &= [A\mathbf{x} + B\mathbf{u} + G\mathbf{w}] - [A\hat{\mathbf{x}} + B\mathbf{u} - K(\mathbf{z} - \hat{\mathbf{z}})] \\ &= A(\mathbf{x} - \hat{\mathbf{x}}) - K(H\mathbf{x} - H\hat{\mathbf{x}}) + G\mathbf{w} - K\mathbf{v} \\ &= A\tilde{\mathbf{x}} - KH\tilde{\mathbf{x}} + G\mathbf{w} - K\mathbf{v} \\ &= (A - KH)\tilde{\mathbf{x}} + G\mathbf{w} - K\mathbf{v}\end{aligned}$$

$$\Longrightarrow \quad \dot{P}(t) = (A - KH)P(t) + P(t)(A - KH)^T + GQG^T + KRK^T \quad (\bigstar)$$

differential Lyapunov equation

- Finding K matrix
  - Minimization of trace($P(t)$)
  - $\dfrac{\partial \, \text{trace}(P(t))}{\partial K} = 0$

  $\Rightarrow K = P(t)H^TR^{-1}$

$$(\bigstar) \quad \boxed{\dot{P}(t) = AP(t) + P(t)A^T + GQG^T - P(t)H^TR^{-1}HP(t)}$$

differential Ricatti equation

Continuous-Time Kalman Filter

- Given

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + G\mathbf{w} \qquad \mathbf{w}(t) \sim \mathcal{N}(0, Q)$$
$$\mathbf{z} = H\mathbf{x} + \mathbf{v} \qquad \mathbf{v}(t) \sim \mathcal{N}(0, R)$$

linear system with zero-mean white noise

- Goal

Minimize $J = \text{trace}\ (P)$

where $P = E[(\mathbf{x} - \hat{\mathbf{x}})((\mathbf{x} - \hat{\mathbf{x}})^T]$

Covariance of estimation error

- Solution

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u} + K(\mathbf{z} - H\hat{\mathbf{x}})$$

Closed-loop estimator

where $K = P\, H^T R^{-1}$

Estimator gain (Kalman gain)

$$\dot{P} = AP + PA^T + GQG^T - PH^T R^{-1} HP$$

Matrix Ricatti eq.

- Remarks
  - $\hat{x}(0)$ and $P(0)$ are known. $P(t)$ is solved forward in time.
  - Increase in $P(t)$ corresponds to increased uncertainty in the state estimate
  - The estimator gain $K(t)$ is weight on the innovation $\mathbf{z} - \hat{\mathbf{z}}$
    - If the uncertainty about the state is high $P \uparrow$, then the innovation is weighted heavily ($K \uparrow$)
    - If the measurements are very accurate $R \downarrow$, then the innovation is weighted heavily ($K \uparrow$)

Discrete-Time Kalman Filter

- Linear discrete system

$$\boxed{\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Gw_k \\ z_k &= Hx_k + v_k \end{aligned}}$$

$$x_0 \sim (\bar{x}_0, P_{x_0}), \qquad w_k \sim (0, Q_k), \qquad v_k \sim (0, R_k)$$

, where $x_k \in \mathcal{R}^n, z_k \in \mathcal{R}^m,\ u_k \in \mathcal{R}^p,$

- Kalman filter algorithm

Initialization

$$P_0 = P_{x_0}, \qquad \hat{x}_0 = \bar{x}_0$$

Time update (effect of system dynamics)

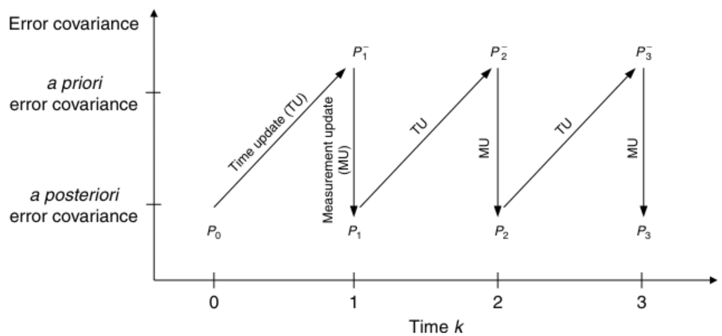
Error covariance $\quad P_{k+1}^- = A_k P_k A_k^T + G_k Q_k G_k^T$

Estimate $\quad \hat{x}_{k+1}^- = A_k \hat{x}_k + B_k u_k$

Measurement update (effect of measurement z_k)

Error covariance $\quad P_{k+1} = \left[(P_{k+1}^-)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \right]^{-1}$

Estimate $\quad \hat{x}_{k+1} = \hat{x}_{k+1}^- + P_{k+1} H_{k+1}^T R_{k+1}^{-1} (z_{k+1} - H_{k+1} \hat{x}_{k+1}^-)$



$$\begin{aligned} K_{k+1} &= P_{k+1}^- H_{k+1}^T \left(H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1} \right)^{-1} \\ P_{k+1} &= (I - K_{k+1} H_{k+1}) P_{k+1}^- \\ \hat{x}_{k+1} &= \hat{x}_{k+1}^- + K_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1}^-) \end{aligned}$$

Discrete-Time Kalman Filter

(Ex) Ship navigation fixes

Suppose a ship is moving east at 10 mph. This velocity is assumed to be constant except for the effects of wind gusts and wave action. An estimate of the easterly position d and velocity $s = \dot{d}$ is required every hour. The navigator guesses at time zero that $d_0 = 0, s_0 = 10$, and his "guesses" can be modeled as having independent Gaussian distributions with variances $\sigma_{d_0}^2 = 2$, $\sigma_{s_0}^2 = 3$. Hence the initial estimates are $\hat{d}_0 = 0, \hat{s}_0 = 10$. The north-south position is of no interest. If d_k, s_k indicate position and velocity at hour k, we are required to find estimates $\hat{d}_k, \hat{s}_k$.

(sol)

- State equations

$$d_{k+1} = d_k + s_k \qquad s_{k+1} = s_k + w_k$$

$$\Longrightarrow \quad x_k \triangleq \begin{bmatrix} d_k \\ s_k \end{bmatrix}$$

$$\Longrightarrow \quad x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

$$w_k \sim N(0, Q) = N(0, 1)$$

$$x_0 \sim N(\bar{x}_0, P_0) = N\left(\begin{bmatrix} 0 \\ 10 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \right)$$

- Measurement equation

$$z_k = [1 \quad 0] x_k + v_k, \qquad v_k \sim N(0, 2)$$

- Measurements

$$z_1 = 9, z_2 = 19.5, z_3 = 29$$

Discrete-Time Kalman Filter

- Estimates

$k = 0$

$$\hat{x}_0 = \bar{x}_0 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

$k = 1 \quad z_1 = 9$

Time update: propagate estimate to $k = 1$ using system dynamics

$$\hat{x}_1^- = A\hat{x}_0 + Bu_0 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \qquad a\ prior \text{ estimate at } k = 1$$

$$P_1^- = AP_0A^T + GQG^T = \begin{bmatrix} 5 & 3 \\ 3 & 4 \end{bmatrix}$$

Measurement update: include the effects of z_1

$$P_1 = \left[(P_1^-)^{-1} + H^TR^{-1}H\right]^{-1} = \begin{bmatrix} 1.429 & 0.857 \\ 0.857 & 2.714 \end{bmatrix}$$

$$\hat{x}_1 = \hat{x}_1^- + P_1H^TR^{-1}(z_1 - H\hat{x}_1^-)$$

$$= \begin{bmatrix} 9.286 \\ 9.571 \end{bmatrix}, \qquad a\ posteriori \text{ estimate at } k = 1$$

$k = 2 \quad z_2 = 19.5$

Time update: effects of system dynamics

$$\hat{x}_2^- = A\hat{x}_1 + Bu_1 = \begin{bmatrix} 18.857 \\ 9.571 \end{bmatrix}, \qquad a\ priori \text{ estimate at time } k = 2$$

$$P_2^- = AP_1A^T + GQG^T = \begin{bmatrix} 5.857 & 3.571 \\ 3.571 & 3.714 \end{bmatrix}$$

Measurement update: effects of z_2

$$P_2 = \left[(P_2^-)^{-1} + H^TR^{-1}H\right]^{-1} = \begin{bmatrix} 1.491 & 0.909 \\ 0.909 & 2.091 \end{bmatrix}$$

$$\hat{x}_2 = \hat{x}_2^- + P_2H^TR^{-1}(z_2 - H\hat{x}_2^-)$$

$$= \begin{bmatrix} 19.336 \\ 9.864 \end{bmatrix}, \qquad a\ posteriori \text{ estimate at } k = 2$$

$k = 3 \quad z_3 = 29$

Time update: effects of system dynamics

$$\hat{x}_3^- = A\hat{x}_2 + Bu_2 = \begin{bmatrix} 29.2 \\ 9.864 \end{bmatrix}, \qquad a\ priori \text{ estimate}$$

$$P_3^- = AP_2A^T + GQG^T = \begin{bmatrix} 5.4 & 3 \\ 3 & 3.091 \end{bmatrix}$$

Measurement update: effects of z_3

$$P_3 = \left[(P_3^-)^{-1} + H^TR^{-1}H\right]^{-1} = \begin{bmatrix} 1.46 & 0.811 \\ 0.811 & 1.875 \end{bmatrix}$$

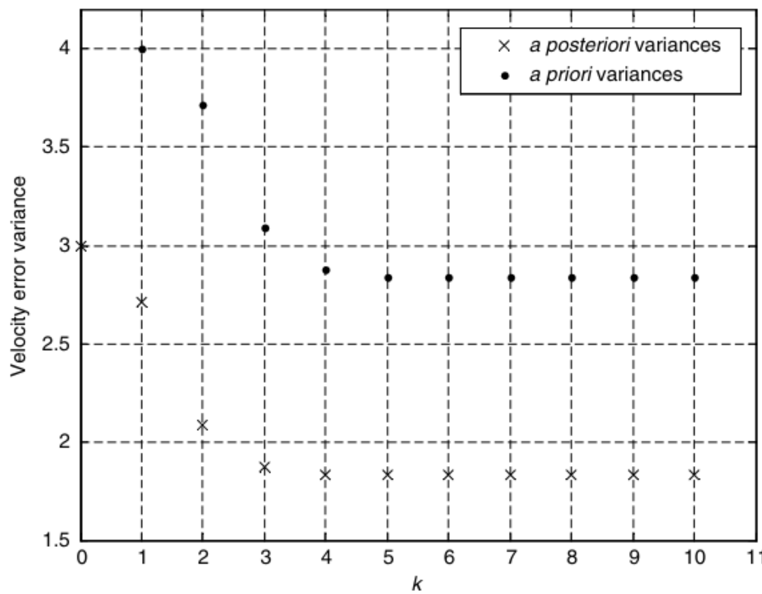
$$\hat{x}_3 = \hat{x}_3^- + P_3H^TR^{-1}(z_3 - H\hat{x}_3^-)$$

$$= \begin{bmatrix} 29.054 \\ 9.783 \end{bmatrix}, \qquad a\ posteriori \text{ estimate}$$

Discrete-Time Kalman Filter

- Velocity error variances

$$\overline{(s_k - \hat{s}_k)(s_k - \hat{s}_k)^T}$$




```
% Example 2.3
% Program to plot the apriori and aposteriori error covariance
x=[0;10]; p=[2 0 ; 0 3]; % Initialization
A=[1 1 ; 0 1]; G=[0;1]; Q=1; R=2; H=[1 0];
p22(1,1)= p(2,2);
for i=1:10
    pm= A*p*A'+ G*Q*G';
    pm22(i+1,:)= pm(2,2); % apriori error covariance p(2,2) for velocity s
    p= inv(inv(pm)+H'* inv(R)*H);
    p22(i+1,:)= p(2,2); % aposteriori error covariance p(2,2) for velocity s
end
plot([0:10],p22,'k.',[1:10],pm22([2:11]),'k*'); % Plotting
grid on; legend('aposteriori error covariance','apriori error covariance');
xlabel('k'); ylabel('Velocity error variances');
```

Discrete-Time Kalman Filter

(Ex) Falling body example

$y(t)$: height of the object

$$\ddot{y}(t) = -g$$
$$\Rightarrow \dot{y}(t) = \dot{y}(t_0) - g(t - t_0)$$
$$\Rightarrow y(t) = y(t_0) + \dot{y}(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2$$

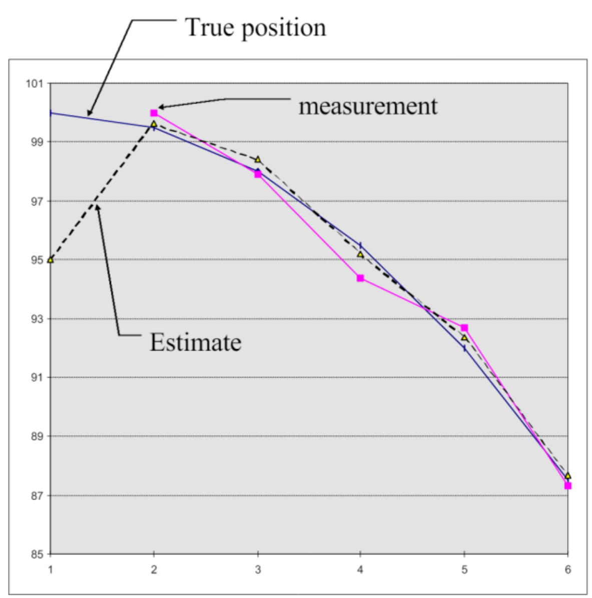
$$y(k+1) = y(k) + \dot{y}(k) - \frac{g}{2}$$

$$\mathbf{x}(\mathrm{k}) \equiv [\mathrm{y(k)} \quad \dot{\mathrm{y}}(\mathrm{k})]$$

$$\begin{aligned}\mathbf{x}(\mathrm{k}+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(\mathrm{k}) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}(-g) \\ &= \mathbf{F}\,\mathbf{x}(k) + \mathbf{G}\,\mathbf{u}\end{aligned}$$

measurement: height of the ball directly

$$\begin{aligned}\mathbf{z}(\mathrm{k}) &= [1 \quad 0]\,\mathbf{x}(\mathrm{k}) + \mathrm{w(k)} \\ &= \mathbf{H}\,\mathbf{x}(k) + w(k)\end{aligned}$$



23

Extended Kalman Filter

- Non-linear system (continuous time)

$$\dot{x} = a(x,u,t) + G(t)w$$
$$z = h(x,t) + v$$
$$x(0) \sim (\bar{x}_0, P_0), w(t) \sim (0,Q), v(t) \sim (0,R)$$

- Linearization

$$A(x,t) = \frac{\partial a(x,u,t)}{\partial x}$$
$$H(x,t) = \frac{\partial h(x,t)}{\partial x}$$

Jacobian matrix

- EKF algorithm

Initialization

$$P(0) = P_0, \hat{x}(0) = \bar{x}_0$$

Estimate update

$$\dot{\hat{x}} = a(\hat{x},u,t) + K[z - h(\hat{x})]$$

Error covariance update

$$\dot{P} = A(\hat{x},t)P + PA^T(\hat{x},t) + GQG^T - PH^T(\hat{x},t)R^{-1}H(\hat{x},t)P$$

Kalman gain

$$K = PH^T(\hat{x},t)R^{-1}$$

Extended Kalman Filter

- Non-linear system (continuous & discrete time)

$$\dot{x} = a(x,u,t) + G(t)w$$
$$z_k = h[x(t_k),k] + v_k$$
$$x(0) \sim (\bar{x}_0, P_0), w(t) \sim (0,Q), v_k \sim (0,R)$$

- Linearization

$$A(x,t) = \frac{\partial a(x,u,t)}{\partial x}$$
$$H(x,t) = \frac{\partial h(x,t)}{\partial x}$$

Jacobian matrix

- EKF algorithm

Initialization

$$P(0) = P_0, \hat{x}(0) = \bar{x}_0$$

Time update

Estimate $\dot{\hat{x}} = a(\hat{x},u,t)$

Error covariance $\dot{P} = A(\hat{x},t)P + PA^T(\hat{x},t) + GQG^T$

Measurement update

Kalman gain

$$K_k = P^-(t_k)H^T(\hat{x}_k^-)[H(\hat{x}_k^-)P^-(t_k)H^T(\hat{x}_k^-) + R]^{-1}$$

Error covariance $P(t_k) = [I - K_k H(\hat{x}_k^-)P^-(t_k)]$

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

Extended Kalman Filter

- Non-linear system (continuous & discrete time)

$$\dot{x} = a(x, u, t) + G(t)w$$
$$z_k = h[x(t_k), k] + v_k$$
$$x(0) \sim (\bar{x}_0, P_0), w(t) \sim (0, Q), v_k \sim (0, R)$$

- Linearization

$$A(x, t) = \frac{\partial a(x, u, t)}{\partial x}$$
$$H(x, t) = \frac{\partial h(x, t)}{\partial x}$$

Jacobian matrix

- EKF algorithm

Initialization

$$P(0) = P_0, \hat{x}(0) = \bar{x}_0$$

Time update

Estimate $\dot{\hat{x}} = a(\hat{x}, u, t)$

Error covariance $\dot{P} = A(\hat{x}, t)P + PA^T(\hat{x}, t) + GQG^T$

Measurement update

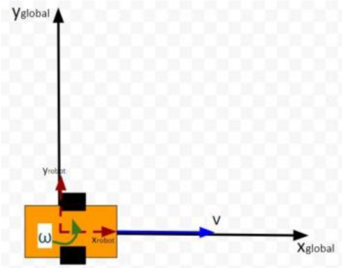
Kalman gain

$$K_k = P^-(t_k)H^T(\hat{x}_k^-)[H(\hat{x}_k^-)P^-(t_k)H^T(\hat{x}_k^-) + R]^{-1}$$

Error covariance $P(t_k) = [I - K_k H(\hat{x}_k^-)P^-(t_k)]$

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

Extended Kalman Filter



(Ex) Mobile robot

- Motion

$$\boldsymbol{x}_k = [x_k \quad y_k \quad \theta_k]^T \qquad \boldsymbol{u}_k = [v_k \quad \omega_k]^T$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{a}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k \quad \leftrightarrow \quad \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k - \frac{v_k}{\omega_k}\sin\theta_k + \frac{v_k}{\omega_k}\sin(\theta_k + \omega_k \Delta t) \\ y_k + \frac{v_k}{\omega_k}\cos\theta_k - \frac{v_k}{\omega_k}\cos(\theta_k + \omega_k \Delta t) \\ \theta_k + \omega_k \Delta t \end{bmatrix} + \boldsymbol{w}_k$$

$$A_k = \frac{\partial \boldsymbol{a}(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{x}_k} = \begin{pmatrix} 1 & 0 & -\frac{v_k}{\omega_k}\cos\theta_k + \frac{v_k}{\omega_k}\cos(\theta_k + \omega_k \Delta t) \\ 0 & 1 & -\frac{v_k}{\omega_k}\sin\theta_k + \frac{v_k}{\omega_k}\sin(\theta_k + \omega_k \Delta t) \\ 0 & 0 & 1 \end{pmatrix}$$

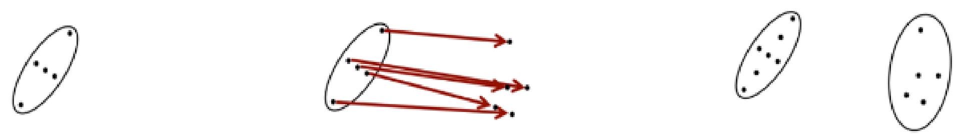
- Measurement

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \quad \leftrightarrow \quad \boldsymbol{z}_k = \begin{bmatrix} \sqrt{(m_x - x_k)^2 + (m_y - y_k)^2} \\ m_x - x_k \\ m_y - y_k \end{bmatrix} + \boldsymbol{v}_k$$

$$H_k = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} = \begin{pmatrix} \frac{-m_x + x_k}{\sqrt{(m_x - x_k)^2 + (m_y - y_k)^2}} & \frac{-m_y + y_k}{\sqrt{(m_x - x_k)^2 + (m_y - y_k)^2}} & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

Unscented Kalman Filter

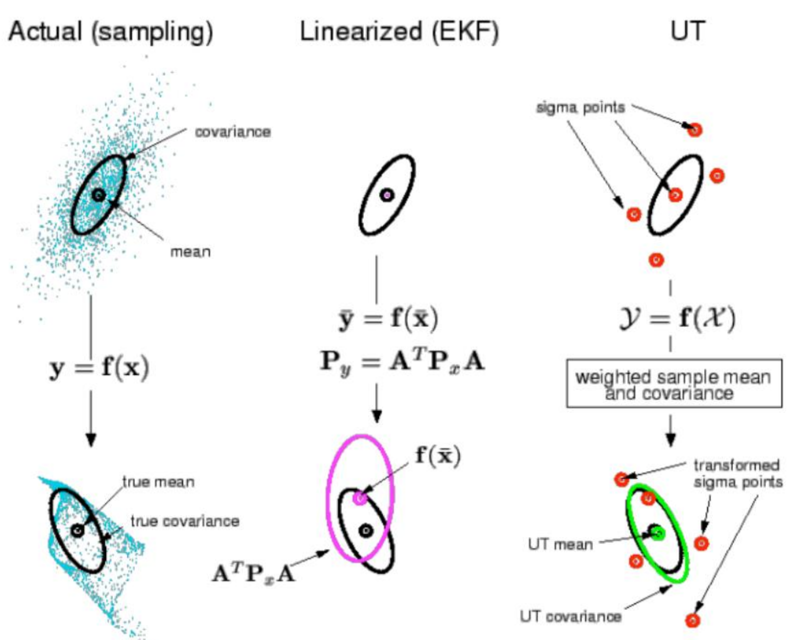
- Unscented Kalman Filter (UKF)
  - EKF 의 선형화 과정 제외 : Jacobian 계산 없음
  - Sigma points 를 사용하여 보다 정확한 Gaussian 분포 구함

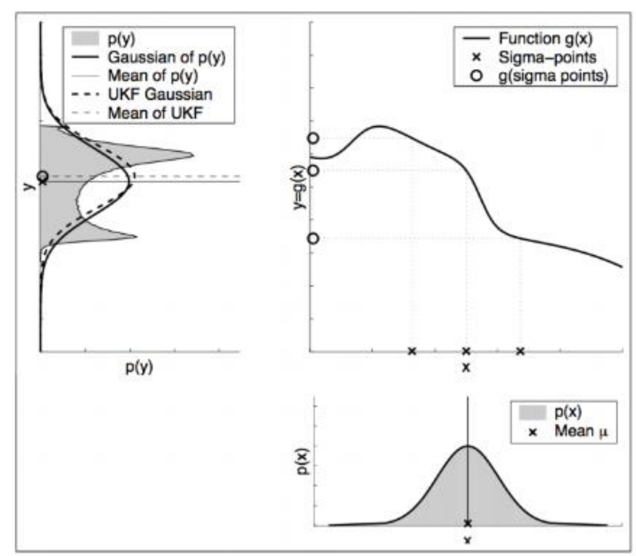


Compute a set of (so-called) sigma points

Transform each sigma point through the non-linear function

Compute Gaussian from the transformed and weighted points





Stochastic Optimal Control

- LQG

$$\dot{\mathbf{x}}(\boldsymbol{t}) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + G(t)\mathbf{w}(t) \qquad \mathbf{w}(t) \sim \mathcal{N}(0, Q)$$

$$\mathbf{z}(t) = H(t)\mathbf{x}(t) + \mathbf{v}(t) \qquad \mathbf{v}(t) \sim \mathcal{N}(0, R)$$

$$J_s = E\left\{\frac{1}{2}\mathbf{x}^T(t_f)P_{t_f}\mathbf{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(\mathbf{x}^T(t)R_{\mathrm{xx}}(t)\mathbf{x}(t) + \mathbf{u}^T(t)R_{\mathrm{uu}}(t)\mathbf{u}(t))dt\right\}$$

(optimizer)

$$\boxed{\mathbf{u}^*(t) = -R_{uu}^{-1}\, B(t)\, P(t)\, \hat{\mathrm{x}}(t)}$$

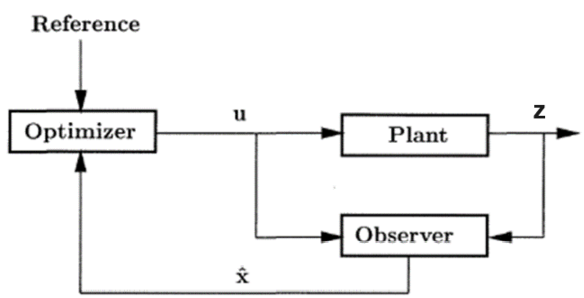
$$\dot{P}(t) = -P(t)A(t) - A^T(t)P(t) - R_{xx}(t) + P(t)B(t)\, R_{uu}^{-1}(t)B^T(t)P(t) \qquad P(t_f) = P_{t_f}$$

(estimator/observer)

$$\boxed{\dot{\hat{\mathbf{x}}}(t) = A(t)\hat{\mathbf{x}}(t) + B(t)\mathbf{u}(t) + K(t)(\mathbf{z}(\mathbf{t}) - H(t)\hat{\mathbf{x}}(t))}$$

$$K(t) = P(t)\, H(t)^T R^{-1}$$

$$\dot{P}(t) = A(t)P(t) + P(t)A(t)^T + G(t)QG(t)^T - P(t)H(t)^T R^{-1} H(t)P(t)$$



In Memoriam

(photo: NAE)

Rudolf E. Kalman

May 19, 1930 - July 2, 2016

이름 모를 노학자, GPS·자율차·로봇 은인이었다

송고시간 | 2016-09-07 08:30

| '오류 제거 알고리즘' 만든 美 칼만…"IoT 시대엔 잡스와 동급될 수도"

(서울=연합뉴스) 김태균 기자 = GPS(인공위성 위치정보), 자율주행차, 우주 탐사 로봇, 사물인터넷…

이런 유망 ICT(정보통신기술) 업종이 올해 세상을 떠난 노학자 1명에게 큰 빚을 지고 있다는 것을 아는 일반인은 얼마나 될까?

7일 미국의 과학 전문지 MIT 테크놀로지리뷰에 따르면 데이비드 민델 MIT 교수(공학사·工學史)와 프랭크 모스 전 MIT 미디어랩 디렉터는 '당신이 전혀 들어보지 못했을 한 발명가가 어떻게 현대 사회를 바꿨을까'란 글을 최근 이 잡지에 실었다.

글의 주인공은 헝가리 태생의 미국 수학자 겸 엔지니어였던 루돌프 칼만(1930~2016)이다.

그의 성과는 1960년대에 내놓은 '칼만 필터'(Kalman Filter). 어감과 달리 정수기나 방독면에 쓰이는 거름장치가 아니라 알고리즘(데이터를 처리하는 논리체계)의 이름이다.

칼만 필터는 복잡한 현실 데이터에서 잡음(노이즈)을 걸러내고 가장 정확한 수치를 통계 기법으로 추정한다.

기계를 오작동시키는 각종 오류와 부정확 데이터를 효율적으로 없애주는 효능 덕에 칼만 필터는 이후 아폴로 11호 달착륙선의 컴퓨터, GPS, 심해 로봇 탐사선, 항공기 제어 장치 등 수많은 기기에 쓰였다.

민델 교수 등은 기고문에서 "칼만 필터는 매우 빠르게 움직이는 우주선이 정확하게 실시간으로 위치·속력·방향을 파악하게 해줬다"며 "우리가 쓰는 스마트폰 GPS가 사용자의 복잡한 움직임 속에서도 위치정보를 잘 잡는 것 역시 현대화된 칼만 필터의 덕이다"고 설명했다.

생전 칼만은 과학기술계에서 선구자로 존경을 받았지만, 대중 사이에서는 인지도가 없었다. 칼만 필터의 전문적 성격 때문이다. 그는 2009년 버락 오바마 미국 대통령으로부터 '국가 과학 메달'을 받았다.

민델 교수 등은 "무인자동차나 로봇이 문제없이 작동하고 가상현실(VR) 디스플레이가 사용자에게 멀미를 일으키지 않게 하려면 실시간으로 정밀한 위치정보를 계산해야 한다"며 "이는 결국 우리 삶에 셀 수 없이 많은 칼만 필터가 쓰여야 한다는 뜻이 된다"고 강조했다.

필자들은 이어 "사물인터넷(IoT) 시대에 들어서면서 일상 속 수많은 IoT 기기를 정확하게 제어하려면 역시 칼만 필터가 필요하다"며 "이쯤 되면 칼만이 스티브 잡스나 마크 저커버그 같은 혁신가처럼 유명해질 날도 올 것"이라고 내다봤다.