Last Time:

- Finish ILC
- Stochastic Optimal Control
- LQG

Today:

- Optimal Estimation
- Finish LQG
- Robust Control

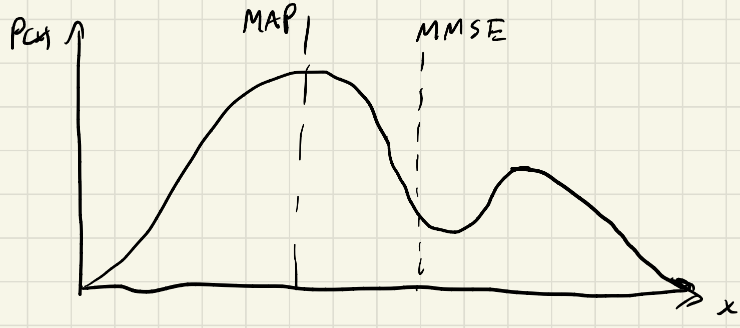
---

* From Last Time:

  - "Certainty Equivalence"
  - "Separation Principle"
  - Frequently Applied to nonlinear systems in practice

* Optimal State Estimation

  - What should I try to optimize?



- Minimum mean-squared error (MMSE):

$$\underset{\hat{x}}{\operatorname{argmin}} \; \underbrace{E[(x-\hat{x})^T(x-\hat{x})]}$$ "least squares" "minimum variance"

$$\| \quad E[tr((x-\hat{x})^T(x-\hat{x}))] = E[tr(\overbrace{(x-\hat{x})(x-\hat{x})^T}^{\text{covariance }(P)})]$$

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

- Maximum a-posteriori (MAP):

$$\underset{x}{\operatorname{argmax}} \; \underbrace{p(x|y)}$$

probability of state conditioned on measurement

- These are the same for a Gaussian.

---

* Kalman Filter:

  - Recursive linear MMSE estimator

  - Assume an estimate of the state that includes all measurements up to the current time:

$$\hat{x}_{k|k} = E[x_n | y_{1:n}]$$

- Assume we also know the error covariance:

$$\Sigma_{n|n} = E\left[(x_n - \hat{x}_{n|n})(x_n - \hat{x}_{n|n})^T\right]$$

- We want to update $\hat{x}$ and Σ to include a new measurement at t_{n+1}
- The KF can be broken into 2 steps:

<u>Prediction:</u>

$$\hat{x}_{n+1|n} = E\left[Ax_n + Bu_n + w_n \mid y_{1:n}\right]$$

$$= A\hat{x}_{n|n} + Bu_n$$

$$\Sigma_{n+1|n} = E\left[(x_{n+1} - \hat{x}_{n+1|n})(x_{n+1} - \hat{x}_{n+1|n})^T\right]$$

$$= E\left[(Ax_n + \cancel{Bu_n} + w_n - A\hat{x}_{n|n} - \cancel{Bu_n})(\cdots)^T\right]$$

$$= A\underbrace{E\left[(x_n - \hat{x}_{n|n})(x_n - \hat{x}_{n|n})^T\right]}_{\Sigma_{n|n}}A^T + \underbrace{E\left[w_n w_n^T\right]}_{W}$$

$$= A\Sigma_{n|n}A^T + W$$

Measurement Update:

- Define "innovation"

$$z_{n+1} = y_{n+1} - C\hat{x}_{n+1|n}$$

$$= Cx_{n+1} + v_{n+1} - C\hat{x}_{n+1|n}$$

- Innovation Covariance

$$S_{n+1} = E[z_{n+1} z_{n+1}^T]$$

$$= E[(Cx_{n+1} + v_{n+1} - C\hat{x}_{n+1|n})(\cdots)^T]$$

* v_{n+1} and x_{n+1} are uncorrelated

$$\Rightarrow S_{n+1} = C\underbrace{E[(x_{n+1} - \hat{x}_{n+1|n})(\cdots)^T]}_{\Sigma_{n+1|n}}C^T + \underbrace{E[v_{n+1} v_{n+1}^T]}_{V}$$

$$= C\Sigma_{n+1|n}C^T + V$$

- Innovation is the thing we feed back into the estimator

- State Update:

$$\hat{x}_{n+1|n+1} = \hat{x}_{n+1|n} + \underbrace{L_{n+1}}_{L} z_{n+1}$$

"Kalman Gain"

- Covariance Update:

$$\Sigma_{n+1|n+1} = E[(x_{n+1} - \hat{x}_{n+1|n+1})(\cdots)^T]$$

$$= E[(x_{n+1} - \hat{x}_{n+1|n} - L_{n+1}(Cx_{n+1} + V_{n+1} - C\hat{x}_{n+1|n}))(\cdots)^T]$$

* V_{n+1} and x_{n+1} are uncorrelated

$$= \underbrace{(I - L_{n+1}C)\Sigma_{n+1|n}(I - L_{n+1}C)^T + L_{n+1}VL_{n+1}^T}$$

"Joseph Form"

- Kalman Gain

MMSE $\Rightarrow$ Minimize $E[(x_n - \hat{x}_{n|n})^T(x_n - \hat{x}_{n|n})]$

$= tr(\Sigma_{n|n})$

$\Rightarrow$ set $\dfrac{\partial tr(\Sigma_{n|n})}{\partial L_{n+1}} = 0$ and solve for L_{n+1}

$$\Rightarrow \boxed{L_{n+1} = \Sigma_{n+1|n} C^T S_{n+1}^{-1}}$$

★ KF Algorithm Summary:

1) Start with $\hat{x}_{0|0}$, $\Sigma_{0|0}$, W, V

2) Predict:

$$\hat{x}_{n+1|n} = A\hat{x}_{n|n} + Bu$$

$$\Sigma_{n+1|n} = A\Sigma_{n|n}A^T + W$$

3) Calculate Innovation + Covariance:

$$z_{n+1} = y_{n+1} - C\hat{x}_{n+1|n}$$

$$S_{n+1} = C\Sigma_{n+1|n}C^T + V$$

4) Calculate Kalman Gain

$$L_{n+1} = \Sigma_{n+1|n}C^T S_{n+1}^{-1}$$

5) Update

$$\hat{x}_{n+1|n+1} = \hat{x}_{n+1|n} + L_{n+1} z_{n+1}$$

$$\Sigma_{n+1|n+1} = (I - L_{n+1}C)\Sigma_n(I - L_{n+1}C)^T + L_{n+1}VL_{n+1}^T$$

6) Goto 2

* How do we apply this to nonlinear systems?

  - Extended KF (EKF): Linearize about $\hat{x}$ and proceed as in standard KF

  - Many other generalizations.

* Duality + Trajectory Optimization

  - MMSE estimation problem is equivalent to the following traj opt problem:

$$\min_{\substack{x_{1:N} \\ w_{1:N}}} \underbrace{\sum_{n=1}^{N} \frac{1}{2}(y_n - g(x_n))^T V^{-1}(y_n - \overbrace{g(x_n)}^{\text{measurement model}}))}_{\text{state cost}} + \underbrace{\frac{1}{2} w_n^T W^{-1} w_n}_{\text{control cost}}$$

$$\text{s.t.} \quad x_{n+1} = f(x_n) + \underbrace{w_n}_{\text{"controls"}}$$

  - If $f(x) = Ax$ and $g(x) = Cx$, this is an LQR problem.

  - Solving this with dynamic programming will result in the Kalman filter.