CSE4421/COSC5323

Kalman Filtering

Quick review from last time

- Seek optimal estimate
  - Minimize variance
  - Independent noise
  - Least squares sense

$$\hat{x} = \omega_1 x_1 + \omega_2 x_2.$$

$$\hat{x} = (\sigma_2^2 x_1 + \sigma_1^2 x_2)/(\sigma_1^2 + \sigma_2^2) \quad \sigma^2 = \sigma_1^2 \sigma_2^2/(\sigma_1^2 + \sigma_2^2).$$

Recursive Least Squares

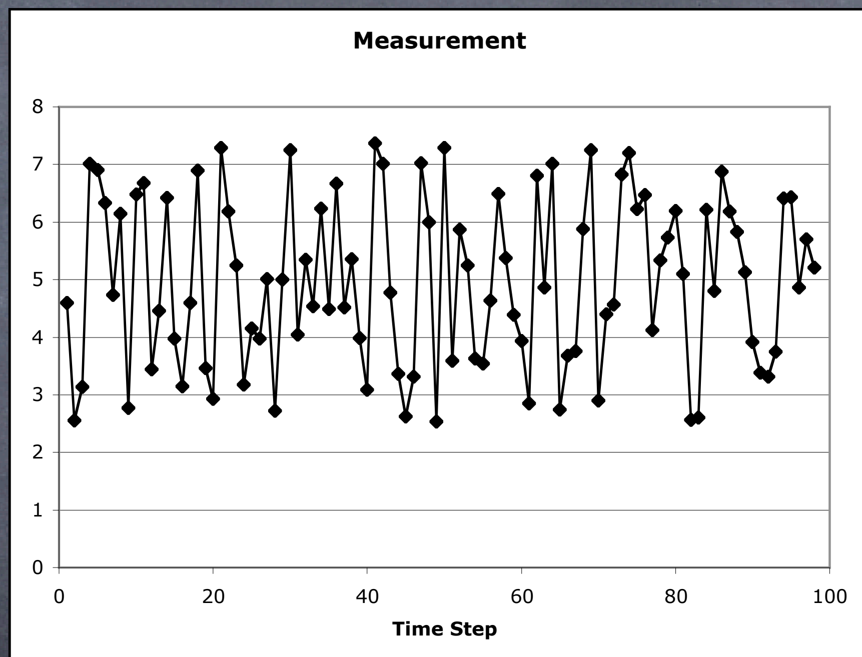
Initialization

$$\hat{x}(0) = x(0)$$
$$\sigma^2(0) = \sigma^2$$

At each time step k do

$$\omega = \sigma^2(k)/(\sigma^2(k) + v^2)$$
$$\hat{x}(k+1) = \hat{x}(k) + \omega(x(k) - \hat{x}(k))$$
$$\sigma^2(k+1) = \sigma^2(k)(1 - \omega)$$

How does this work in practice?

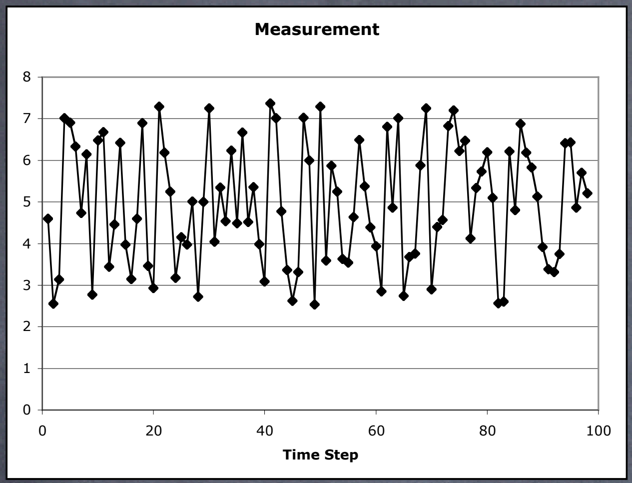


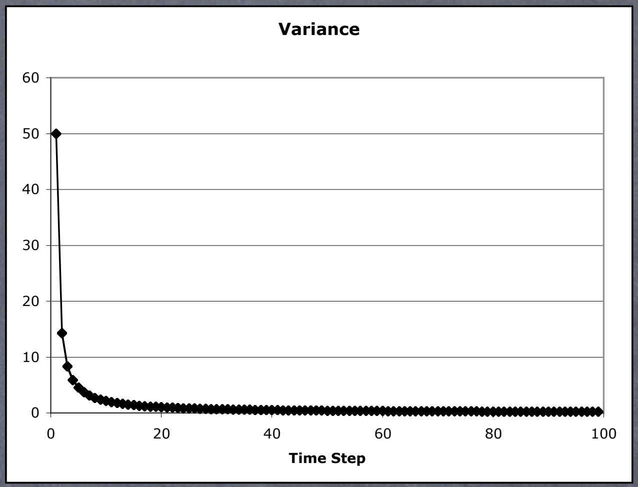
Want to estimate mean at all time
(Answer is 5)

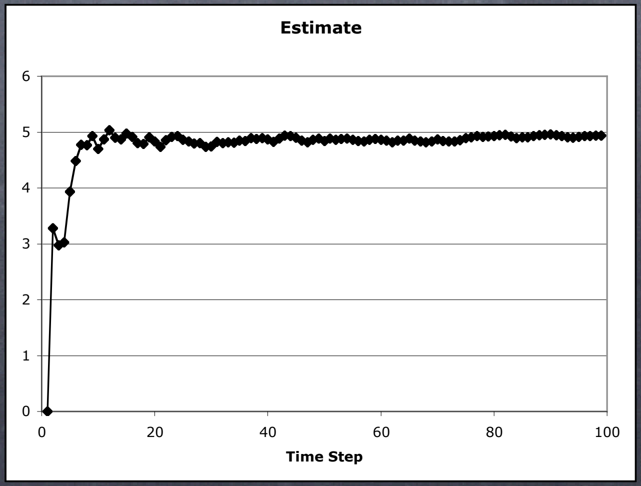
Initiate the Filter

- Initial guess - 0
  - (really wrong)
- Initial estimate of variance - 50
  - (we have no idea)

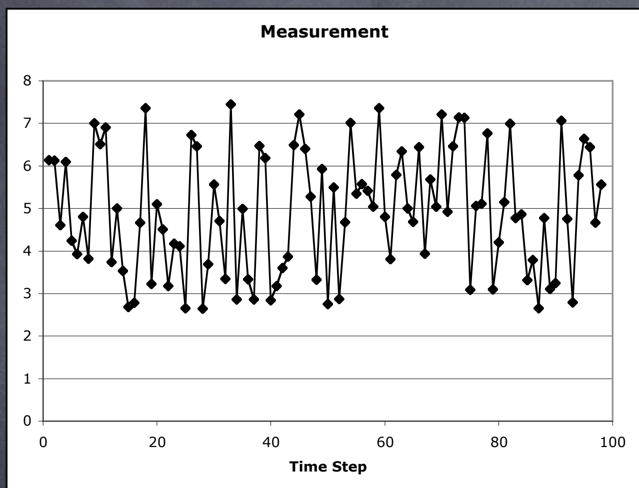
Variance Estimate 20

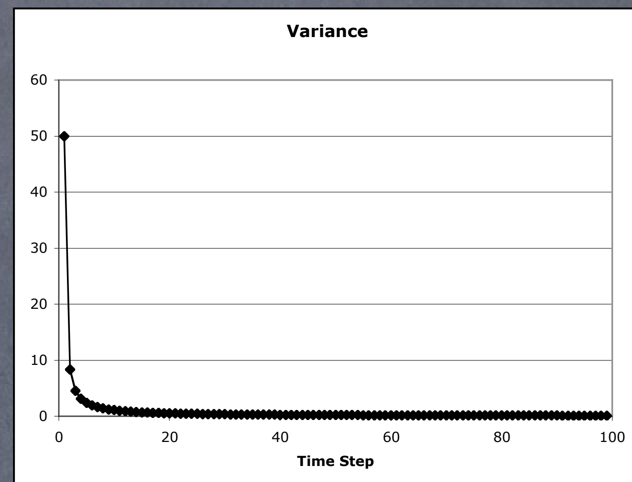


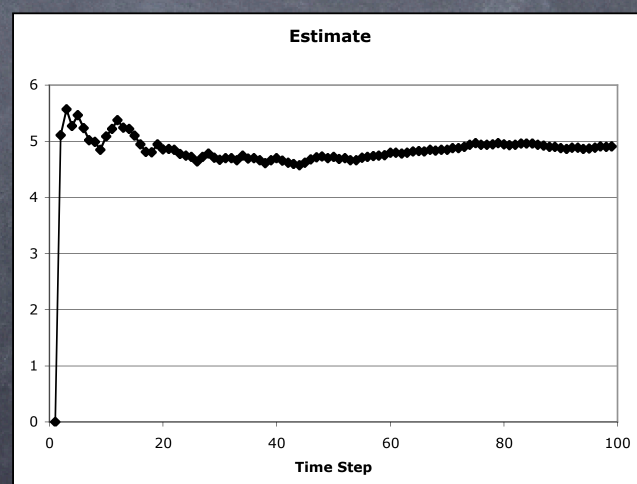




Variance Estimate 10







Vector Version

- x->column vector
- sigma->covariance matrix

$$\begin{bmatrix} E[(x_1-E[x_1])^2] & E[(x_1-E[x_1])(x_2-E[x_2])] & .... & E[(x_1-E[x_1])(x_n-E[x_n])] \\ E[(x_2-E[x_2])(x_1-E[x_1])] & E[(x_2-E[x_2])^2] & .... & E[(x_2-E[x_2])(x_n-E[x_n])] \\ & & ... & \\ E[(x_n-E[x_n])(x_1-E[x_1])] & E[(x_n-E[x_n])(x_1-E[x_1])] & .... & E[(x_n-E[x_n])^2] \end{bmatrix}$$

Recursive Vector Estimator

$$\omega = \sigma^2(k)/(\sigma^2(k) + v^2(k))$$
$$\hat{x}(k+1) = \hat{x}(k) + \omega(x(k) - \hat{x}(k))$$
$$\sigma^2(k+1) = \sigma^2(k)(1-\omega).$$

$$\boldsymbol{K} = \boldsymbol{P}(k)(\boldsymbol{P}(k) + \boldsymbol{C}_v)^{-1}$$
$$\hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{x}}(k) + \boldsymbol{K}(\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k))$$
$$\boldsymbol{P}(k+1)) = (\boldsymbol{I} - \boldsymbol{K})\boldsymbol{P}(k).$$

Scalar Version

Vector Version

Recursive Vector Estimator

$$\omega = \sigma^2(k)/(\sigma^2(k) + v^2(k))$$
$$\hat{x}(k+1) = \hat{x}(k) + \omega(x(k) - \hat{x}(k))$$
$$\sigma^2(k+1) = \sigma^2(k)(1 - \omega).$$

Scalar Version

$$\boldsymbol{K} = \boldsymbol{P}(k)(\boldsymbol{P}(k) + \boldsymbol{C}_v)^{-1}$$
$$\hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{x}}(k) + \boldsymbol{K}(\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k))$$
$$\boldsymbol{P}(k+1)) = (\boldsymbol{I} - \boldsymbol{K})\boldsymbol{P}(k).$$

Vector Version

Variance

Covariance Matrix

Recursive Vector Estimator

$$\omega = \sigma^2(k)/(\sigma^2(k) + v^2(k))$$
$$\hat{x}(k+1) = \hat{x}(k) + \omega(x(k) - \hat{x}(k))$$
$$\sigma^2(k+1) = \sigma^2(k)(1-\omega).$$

$$\boldsymbol{K} = \boldsymbol{P}(k)(\boldsymbol{P}(k) + \boldsymbol{C}_v)^{-1}$$
$$\hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{x}}(k) + \boldsymbol{K}(\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k))$$
$$\boldsymbol{P}(k+1)) = (\boldsymbol{I} - \boldsymbol{K})\boldsymbol{P}(k).$$

Scalar Version

Vector Version

Gain

Gain

Extending the linear estimator (Kalman)

- Plant model

$$\boldsymbol{x}(k+1) = \Phi(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{v}(k)$$

State of the system (robot)

Extending the linear estimator (Kalman)

- Plant model

$$x(k+1) = \Phi(x(k), u(k)) + v(k)$$

Commanded input
(motion command)

Extending the linear estimator (Kalman)

- Plant model

$$x(k+1) = \Phi(x(k), u(k)) + v(k)$$

Plant model
(kinematic model)

Extending the linear estimator (Kalman)

- Plant model

$$\boldsymbol{x}(k+1) = \Phi(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{v}(k)$$

Plant noise

How uncertain we are about the plant

Assume Covariance matrix $\boldsymbol{C}_v(k)$

Extending the linear estimator (Kalman)

- Plant model

$$\boldsymbol{x}(k+1) = \Phi(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{v}(k)$$

$$\downarrow$$

$$\boldsymbol{x}(k+1) = \boldsymbol{\Phi}\boldsymbol{x}(k) + \boldsymbol{\Gamma}\boldsymbol{u}(k) + \boldsymbol{v}(k).$$

Linear Plant

Extending the linear estimator (Kalman)

- Measurement Model

$$\boldsymbol{z}_i(k) = \boldsymbol{h}(\boldsymbol{x}(k), \mathcal{E}) + \boldsymbol{w}_i(k)$$

Extending the linear estimator (Kalman)

- Measurement Model

$$z_i(k) = h(x(k), \mathcal{E}) + w_i(k)$$

Robot State

Extending the linear estimator (Kalman)

- Measurement Model

$$\boldsymbol{z}_i(k) = \boldsymbol{h}(\boldsymbol{x}(k), \mathcal{E}) + \boldsymbol{w}_i(k)$$

World Model

Extending the linear estimator (Kalman)

- Measurement Model

$$\boldsymbol{z}_i(k) = \boldsymbol{h}(\boldsymbol{x}(k), \mathcal{E}) + \boldsymbol{w}_i(k)$$

Measurement Function
(what the robot should see from x(k)
NB: Not necessarily a fully observable model.

Extending the linear estimator (Kalman)

- Measurement Model

$$z_i(k) = h(x(k), \mathcal{E}) + w_i(k)$$

Measurement Noise
with covariance $C_w(k)$.

Extending the linear estimator (Kalman)

- Measurement Model

$$z_i(k) = h(x(k), \mathcal{E}) + w_i(k)$$

↓

$$z_i(k) = \Lambda_E x(k) + w_i(k)$$

Linear Measurement Model

More Notation

- Q(k|k)=Q(k) - value of Q at time k given all measurements up to and including time k
- Q(k|k-1) - value of Q at time k given all measurement up to and including time k-1
  - NOT MEASUREMENTS AT TIME k

Kalman Filter

$$\hat{\boldsymbol{x}}(k+1|k) = \Phi\hat{\boldsymbol{x}}(k) + \Gamma\boldsymbol{u}(k).$$

$$\boldsymbol{P}(k+1|k) = \Phi\boldsymbol{P}(k)\Phi^T + \boldsymbol{C}_v(k).$$

$$\boldsymbol{K}(k+1) = \boldsymbol{P}(k+1|k)\Lambda_E^T(\Lambda_E\boldsymbol{P}(k+1|k)\Lambda_E^T + \boldsymbol{C}_w(k+1))^{-1}.$$

$$\boldsymbol{r}(k+1) = \boldsymbol{z}(k+1) - \Lambda_E\hat{\boldsymbol{x}}(k+1|k)$$

$$\hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{x}}(k+1|k) + \boldsymbol{K}(k+1)\boldsymbol{r}(k+1)$$

$$\boldsymbol{P}(k+1) = (\boldsymbol{I} - \boldsymbol{K}(k+1)\Lambda_E)\boldsymbol{P}(k+1|k)$$

$$\omega = \sigma^2(k)/(\sigma^2(k) + v^2(k))$$
$$\hat{x}(k+1) = \hat{x}(k) + \omega(x(k) - \hat{x}(k))$$
$$\sigma^2(k+1) = \sigma^2(k)(1-\omega).$$

Remember this?

Still Following?

- Critical observations
  - Can estimate a (partially) hidden system from measurements in a linearly optimal way
  - Its all just linear least squares
    - Even if the math is pretty dense at first view.

But Wait, it gets uglier

- Almost no robotic system is linear
- So we take the nonlinear systems and linearize them about our current estimate of the system.
- And use this in the Kalman equations
    - Known as the Extended Kalman filter

Enjoy Reading Week