Estimation: parts of Chapters 12-13

Wiener and Kalman Filtering

Natasha Devroye
devroye@ece.uic.edu
http://www.ece.uic.edu/~devroye

Spring 2011

Summary

We look at two examples of sequential estimation:

- Wiener filtering: filtering, smoothing and prediction *(wide-sense stationary signals)* in sequential LMMSE framework
- Kalman filtering: generalization of Wiener filtering to *(non-stationary signals)*, i.e. sequential MMSE estimator of a signal in noise, where signal characterized by a dynamical model (i.e. tracking)

Signal model: $x[n] = s[n] + w[n], \;\; n = 0, 1, 2, \cdots N-1$ where noise $w[n]$ is WSS, zero-mean with $\mathbf{C}_{ww} = \mathbf{R}_{ww}$

Problem: Process $x[n]$ using a *linear* filter to obtain a "de-noised" version of the signal that has *minimum mean square error* relative to the desired signal $s[n]$.

Wiener filtering

Signal model: $x[n] = s[n] + w[n], \;\; n = 0, 1, 2, \cdots N-1$ where:

- noise $w[n]$ is WSS, zero-mean with $\mathbf{C}_{ww} = \mathbf{R}_{ww}$
- desired signal $s[n]$ is WSS, zero mean with $\mathbf{C}_{ss} = \mathbf{R}_{ss}$
- observed noisy signal $x[n]$ is WSS, zero mean with $\mathbf{C}_{xx} = R_{xx}$

Filtering

Given: $x[0], x[1], \ldots, x[n]$

Find: $\hat{s}[n]$

Smoothing

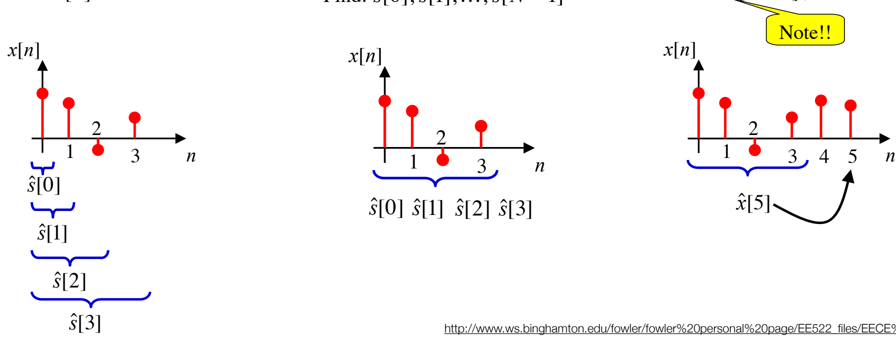
Given: $x[0], x[1], \ldots, x[N\text{-}1]$

Find: $\hat{s}[0], \hat{s}[1], \ldots, \hat{s}[N-1]$

Prediction

Given: $x[0], x[1], \ldots, x[N\text{-}1]$

Find: $\hat{x}[N+l], \;\; l > 0$

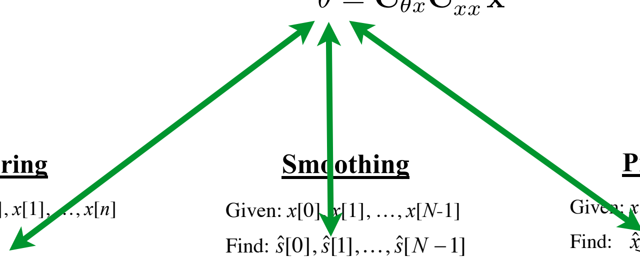


http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_28%20Ch_12B.pdf

Wiener filtering

Solve all three using general LMMSE estimation

$$\hat{\theta} = \mathbf{C}_{\theta x} \mathbf{C}_{xx}^{-1} \mathbf{x}$$



Filtering

Given: $x[0], x[1], \ldots, x[n]$

Find: $\hat{s}[n]$

Smoothing

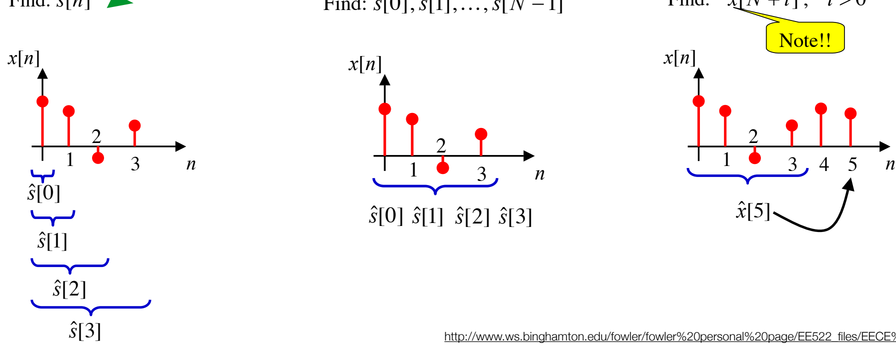
Given: $x[0], x[1], \ldots, x[N\text{-}1]$

Find: $\hat{s}[0], \hat{s}[1], \ldots, \hat{s}[N-1]$

Prediction

Given: $x[0], x[1], \ldots, x[N\text{-}1]$

Find: $\hat{x}[N+l], \;\; l > 0$



http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_28%20Ch_12B.pdf

Wiener filtering

$$\hat{\boldsymbol{\theta}} = \mathbf{C}_{\boldsymbol{\theta}\mathbf{x}}\mathbf{C}_{\mathbf{xx}}^{-1}\mathbf{x}$$

Filtering

$\theta = s[n]$ (scalar)

$$\mathbf{C}_{\theta\mathbf{x}} = E\left\{s[n]\mathbf{x}^T\right\} = E\left\{s[n]\mathbf{s}^T\right\} = \left[r_{ss}[n]\cdots r_{ss}[0]\right] = \tilde{\mathbf{r}}_{ss}^{\mathrm{T}} \text{ (vector!)}$$

$$\mathbf{C}_{xx} = E\left\{(\mathbf{s}+\mathbf{w})(\mathbf{s}+\mathbf{w})^T\right\} = E\left\{\mathbf{ss}^T + \mathbf{ww}^T\right\} = \mathbf{R}_{ss} + \mathbf{R}_{ww}$$

$$\hat{s}[n] = \tilde{\mathbf{r}}_{ss}^T(\mathbf{R}_{ss} + \mathbf{R}_{ww})^{-1}\mathbf{x}$$

$$[1\times(n+1)][(n+1)\times(n+1)][(n+1)\times 1]$$

Smoothing

$\boldsymbol{\theta} = \mathbf{s}$ (vector)

$$\mathbf{C}_{\boldsymbol{\theta}\mathbf{x}} = E\left\{\mathbf{sx}^T\right\} = E\left\{\mathbf{s}(\mathbf{s}+\mathbf{w})^T\right\} = E\left\{\mathbf{ss}^T + \mathbf{sw}^T\right\} = \mathbf{R}_{ss} \text{ (Matrix!)}$$

$$\mathbf{C}_{xx} = E\left\{(\mathbf{s}+\mathbf{w})(\mathbf{s}+\mathbf{w})^T\right\} = E\left\{\mathbf{ss}^T + \mathbf{ww}^T\right\} = \mathbf{R}_{ss} + \mathbf{R}_{ww}$$

$$\hat{\mathbf{s}} = \mathbf{R}_{ss}(\mathbf{R}_{ss} + \mathbf{R}_{ww})^{-1}\mathbf{x}$$

$$[N\times N][N\times N][N\times 1]$$

Prediction

$\theta = x[N-1+l]$ (scalar)

x not s!

$$\mathbf{C}_{\theta\mathbf{x}} = E\left\{x[N-1+l]\mathbf{x}^T\right\} = \left[r_{xx}[N-1+l]\cdots r_{xx}[l]\right] = \tilde{\mathbf{r}}_{xx}^{\mathrm{T}} \text{ (vector!)}$$

$$\mathbf{C}_{xx} = E\left\{\mathbf{xx}^T\right\} = \mathbf{R}_{xx}$$

$$\hat{x}[N-1+l] = \tilde{\mathbf{r}}_{xx}^T\mathbf{R}_{xx}^{-1}\mathbf{x}$$

$$[1\times N][N\times N][N\times 1]$$

12

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_28%20Ch_12B.pdf

Wiener filtering

$$\hat{s}[n] = \underbrace{\tilde{\mathbf{r}}_{ss}^T(\mathbf{R}_{ss} + \mathbf{R}_{ww})^{-1}}_{\mathbf{a}^T}\mathbf{x} = \mathbf{a}^T\mathbf{x}$$

$$\mathbf{h} = \left[h^{(n)}[0] \quad h^{(n)}[1] \quad \ldots \quad h^{(n)}[n]\right]^T = \left[a_n \quad a_{n-1} \quad \ldots \quad a_0\right]^T$$

$$\hat{s}[n] = \sum_{k=0}^{n} h^{(n)}[k]x[n-k]$$

Wiener Filter as Time-Varying FIR Filter
- Causal!
- Length Grows!

Wiener-Hopf Filtering Equations

$$\underbrace{(\mathbf{R}_{ss} + \mathbf{R}_{ww})}_{\mathbf{R}_{xx}}\mathbf{h} = \mathbf{r}_{ss}$$

$$\mathbf{r}_{ss} = \left[r_{ss}[0] \quad r_{ss}[1] \quad \ldots \quad r_{ss}[n]\right]^T$$

$$\underbrace{\begin{bmatrix} r_{xx}[0] & r_{xx}[1] & \cdots & r_{xx}[n] \\ r_{xx}[1] & r_{xx}[0] & \cdots & r_{xx}[n-1] \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}[n] & r_{xx}[n-1] & \cdots & r_{xx}[0] \end{bmatrix}}_{\text{Symmetric \& Toeplitz}} \begin{bmatrix} h^{(n)}[0] \\ h^{(n)}[1] \\ \vdots \\ h^{(n)}[n] \end{bmatrix} = \begin{bmatrix} r_{ss}[0] \\ r_{ss}[1] \\ \vdots \\ r_{ss}[n] \end{bmatrix}$$

<u>In Principle</u>: Solve WHF Eqs for filter **h** at <u>each</u> n

<u>In Practice</u>: Use Levinson Recursion to Recursively Solve

13

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_28%20Ch_12B.pdf

Kalman filtering

Borrowed heavily from the excellent notes

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_29%20Ch_13A.pdf

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_29%20Ch_13B.pdf

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_29%20Ch_13C.pdf

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_29%20Ch_13D.pdf

Kalman filtering

- Rudolf Kalman developed in 1960s
- discrete-time and continuous time versions
- used in Control systems, Navigation systems, Tracking systems

Background

- **Wiener filter:** LMMSE of changing signal (varying parameter)
- **Sequential LMMSE:** sequentially estimate fixed parameter
- **State-space models:** dynamical models for varying parameters

- **Kalman filter:** sequential LMMSE estimation for a time-varying parameter vector that follows a ``state-space” dynamical model (i.e. not arbitrary dynamics)

State-space / dynamical models

- **System state:** variables needed to predict system at future times in absence of inputs (i.e. what you need to keep track of)
- **Example:** constant velocity aircraft in 2-D

$$s(t) = [r_x(t) r_y(t) v_x(t) v_y(t)]^T, \text{ where } v_x(t) = V_x, v_y(t) = V_y \text{ for constant velocity}$$

In discrete-time, take samples every Δ seconds, then if the state at time n is $\mathbf{s}[n]$:

$$\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n] + \mathbf{B}\mathbf{u}[n]$$

$$\begin{bmatrix} r_x[n] \\ r_y[n] \\ v_x[n] \\ v_y[n] \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_x[n-1] \\ r_y[n-1] \\ v_x[n-1] \\ v_y[n-1] \end{bmatrix}$$

Vector Gauss-Markov Model

Linear state model: $\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n], \;\; n \geq 0$, where

$$\begin{aligned}
\mathbf{s}[n] &: \text{“state vector” is a vector Gauss-Markov process} \\
\mathbf{A} &: \text{“state transition matrix”, with } |\lambda_i| < 1 \text{ for stability} \\
\mathbf{B} &: \text{“input matrix”} \\
\mathbf{u}[n] &: \text{“driving noise” is vector WGN with zero mean} \\
&\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \;\; E[\mathbf{u}[n]\mathbf{u}^T[m]] = 0, n \neq m \\
\mathbf{s}[-1] &: \text{“initial state”} \sim \mathcal{N}(\mu_2, \mathbf{C}_s) \text{ and independent of } \mathbf{u}[n]
\end{aligned}$$

Thm. 13.1 (Vector Gauss-Markov Model)

For the Gauss-Markov model on the previous page, the signal process $\mathbf{s}[n]$ is Gaussian with mean

$$E[\mathbf{s}[n]] = \mathbf{A}^{n+1}\mu_{\mathbf{s}},$$

and

$$\begin{aligned}
\mathbf{C}_s[m,n] &= E\left[[\mathbf{s}[m] - E[\mathbf{s}[m]]\,[\mathbf{s}[n] - E[\mathbf{s}[n]]^T\right] \\
&= \mathbf{A}^{m+1}\mathbf{C}_s\left(\mathbf{A}^{n+1}\right)^T + \sum_{k=m-n}^{m} \mathbf{A}^k\mathbf{B}\mathbf{Q}\mathbf{B}^T(\mathbf{A}^{n-m+k})^T \\
&\text{for } m \geq n,
\end{aligned}$$

$$\mathbf{C}_s[m,n] = \mathbf{C}_s^T[n,m] \qquad\qquad \text{for } m < n,$$

and

$$\begin{aligned}
\mathbf{C}_[n] &:= \mathbf{C}_s[n,n] \\
&= \mathbf{A}^{n+1}\mathbf{C}_s(\mathbf{A}^{n+1})^T + \sum_{k=0}^{n} \mathbf{A}^k\mathbf{B}\mathbf{Q}\mathbf{B}^T(\mathbf{A}^{\mathbf{k}})^T
\end{aligned}$$

The mean and covariance propagation equations are

$$\begin{aligned}
E[\mathbf{s}[n]] &= \mathbf{A}E[\mathbf{s}[n-1]] \\
\mathbf{C}[n] &= \mathbf{A}\mathbf{C}[n-1]\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T
\end{aligned}$$

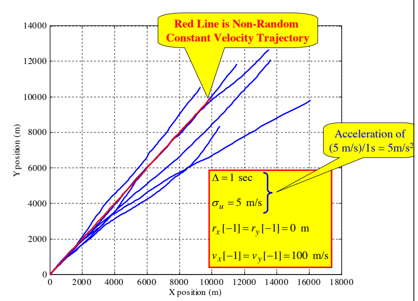
Example - constant velocity 2-D aircraft

$$\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n] + \mathbf{B}\mathbf{u}[n]$$

$$\mathbf{s}[n] = \begin{bmatrix} r_x[n] \\ r_y[n] \\ v_x[n] \\ v_y[n] \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_x[n-1] \\ r_y[n-1] \\ v_x[n-1] \\ v_y[n-1] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_x[n] \\ u_y[n] \end{bmatrix}$$

where we have a random perturbation of the constant velocities with

$$\mathbf{C}_u = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_u^2 & 0 \\ 0 & 0 & 0 & \sigma_u^2 \end{bmatrix}$$



Have state model, now observation model

- Have a state-space or dynamical system model for the desired signal
- Need a model for the noisy measurements and how it relates to the states (depends on how we acquire the data)

Linear observation model: $\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n]$, where

- $\mathbf{x}[n]$ is the measured observation vector at each time
- $\mathbf{H}[n]$ is the Observation Matrix and can change with time
- $\mathbf{s}[n]$ is the state vector process being observed
- $\mathbf{w}[n]$ is a vector noise process

Estimation problem

Given the linear state-space and observation models, giving us a sequence $\mathbf{x}[0], \mathbf{x}[1], \cdots \mathbf{x}[n]$, we wish to compute an estimate of the state vector $\mathbf{s}[n]$, that is we wish to obtain

$$\hat{\mathbf{s}}[n|n], \text{ where } \hat{\mathbf{s}}[n|m] = \text{ estimate of } \mathbf{s}[n] \text{ using } \mathbf{x}[0], \mathbf{x}[1], \cdots \mathbf{x}[m]$$

Furthermore, we want a *recursive* solution, i.e.

$$\begin{aligned} \text{Given} &: \hat{\mathbf{s}}[n|n] \text{ and a new observation vector } \mathbf{x}[n+1] \\ \text{Find} &: \hat{\mathbf{s}}[n+1|n+1] \end{aligned}$$

Three cases of interest:

- scalar state, scalar observation
- vector state, scalar observation
- vector state, vector observation

Scalar state, scalar observation Kalman

State model: $s[n] = as[n-1] + u[n], \ \ u[n] \sim \mathcal{N}(0, \sigma_u^2)$, WGN, WSS

Observation model: $x[n] = s[n] + w[n], \ \ w[n] \sim \mathcal{N}(0, \sigma_n^2)$, WGN, can depend on n

Minimum MSE: let $M[n|m]$ be the minimum MSE incurred when $s[n]$ is estimated based on $x[0] \cdots x[m]$

Additional assumptions:

- initial state if $s[-1] \sim \mathcal{N}(\mu_s, \sigma_s^2)$
- $u[n], w[n], s[-1]$ are all independent
- for now, assume $\mu_s = 0$

Goal: to recursively compute $\hat{s}[n|n] = E[s[n]|x[0], x[1], \cdots x[n]]$
To exploit:

- MMSE estimator for uncorrelated data x_1, x_2 is additive, i.e. $\hat{\theta} = E[\theta|x_1, x_2] = E[\theta|x_1] + E[\theta|x_2]$
- If $\theta = \theta_1 + \theta_2$ then $\hat{\theta} = E[\theta|x] = E[\theta_1|x] + E[\theta_2|x]$

Scalar state, scalar observation Kalman

State model: $s[n] = as[n-1] + u[n], \;\; u[n] \sim \mathcal{N}(0, \sigma_u^2)$, WGN, WSS

Observation model: $x[n] = s[n] + w[n], \;\; w[n] \sim \mathcal{N}(0, \sigma_n^2)$, WGN, can depend on n

Initialization: $\hat{s}[-1|-1] = E[s[-1]] = \mu_s$, $M[-1|-1] = \sigma_s^2$

Prediction: $\hat{s}[n|n-1] = a\hat{s}[n-1|n-1]$

Prediction MSE: $M[n|n-1] = a^2 M[n-1|n-1| + \sigma_u^2$

Kalman gain: $K[n] = \frac{M[n|n-1]}{\sigma_n^2 + M[n|n-1]}$

Update: $\hat{s}[n|n] = \hat{s}[n|n-1] + K[n](x[n] - \hat{s}[n|n-1])$

Estimation MSE: $M[n|n] = (1 - K[n])M[n|n-1]$

Let's derive this!!!

Some observations

- dynamical model provides update from estimate to prediction
- in the Kalman filter, prediction acts like the prior information about the state at time n before we observe the data at time n
- must know noise and initial state distributions and dynamical and observation model

Kalman Filter: Scalar State & Scalar Observation

State Model: $s[n] = as[n-1] + u[n]$ $u[n]$ WGN; WSS; $\sim N(0,\sigma_u^2)$

Observation Model: $x[n] = s[n] + w[n]$ $w[n]$ WGN; $\sim N(0,\sigma_n^2)$ Varies with n

Must Know: $\mu_s, \sigma^2_s, a, \sigma^2_u, \sigma^2_n$

Initialization:
$$\hat{s}[-1|-1] = E\{s[-1]\} = \mu_s$$
$$M[-1|-1] = E\{(s[-1]\} - \hat{s}[-1|-1])^2\} = \sigma_s^2$$

Prediction:
$$\hat{s}[n|n-1] = a\hat{s}[n-1|n-1]$$

Pred. MSE:
$$M[n|n-1] = a^2 M[n-1|n-1] + \sigma_u^2$$

Kalman Gain:
$$K[n] = \frac{M[n|n-1]}{\sigma_n^2 + M[n|n-1]}$$

Update:
$$\hat{s}[n|n] = \hat{s}[n|n-1] + K[n]\big(x[n] - \hat{s}[n|n-1]\big)$$

Est. MSE:
$$M[n|n] = \big(1 - K[n]\big)M[n|n-1]$$

Kalman Filter: Vector State & Scalar Observation

State Model: $\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n]$ $\mathbf{s}\ p\times 1; \mathbf{A}\ p\times p; \mathbf{B}\ p\times r; \mathbf{u} \sim N(\mathbf{0},\mathbf{Q})\ r\times 1$

Observation Model: $x[n] = \mathbf{h}^T[n]\mathbf{s}[n] + w[n];\quad \mathbf{h}^T[n]\ p\times 1$ $w[n]$ WGN; $\sim N(0,\sigma_n^2)$

Must Know: $\mu_s, \mathbf{C}_s, \mathbf{A}, \mathbf{B}, \mathbf{h}, \mathbf{Q}, \sigma^2_n$

Initialization:
$$\hat{\mathbf{s}}[-1|-1] = E\{\mathbf{s}[-1]\} = \boldsymbol{\mu}_s$$
$$\mathbf{M}[-1|-1] = E\Big\{(\mathbf{s}[-1]\} - \hat{\mathbf{s}}[-1|-1])(\mathbf{s}[-1]\} - \hat{\mathbf{s}}[-1|-1])^T\Big\} = \mathbf{C}_s$$

Prediction:
$$\hat{\mathbf{s}}[n|n-1] = \mathbf{A}\hat{\mathbf{s}}[n-1|n-1]$$

Pred. MSE ($p\times p$):
$$\mathbf{M}[n|n-1] = \mathbf{A}\mathbf{M}[n-1|n-1]\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T$$

Kalman Gain ($p\times 1$):
$$\mathbf{K}[n] = \frac{\mathbf{M}[n|n-1]\mathbf{h}[n]}{\sigma_n^2 + \underbrace{\mathbf{h}^T[n]\mathbf{M}[n|n-1]\mathbf{h}[n]}_{1\times 1}}$$

Update:
$$\hat{\mathbf{s}}[n|n] = \hat{\mathbf{s}}[n|n-1] + \mathbf{K}[n]\underbrace{(x[n] - \underbrace{\mathbf{h}^T[n]\hat{\mathbf{s}}[n|n-1]}_{\hat{x}[n|n-1]})}_{\tilde{x}[n]:\ innovations}$$

Est. MSE ($p\times p$):
$$\mathbf{M}[n|n] = \big(\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]\big)\mathbf{M}[n|n-1]$$

Kalman Filter: Vector State & Vector Observation

State Model: $\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n] \quad \mathbf{s}\ p\times 1; \mathbf{A}\ p\times p; \mathbf{B}\ p\times r; \mathbf{u} \sim \mathrm{N}(\mathbf{0},\mathbf{Q})\ r\times 1$

Observation: $\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n]; \quad \mathbf{x}\ M\times 1; \mathbf{H}[n]\ M\times p; \mathbf{w}[n] \sim \mathrm{N}(\mathbf{0},\mathbf{C}[n])\ M\times 1$

Must Know: $\boldsymbol{\mu}_s, \mathbf{C}_s, \mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{Q}, \mathbf{C}[n]$

Initialization:

$$\hat{\mathbf{s}}[-1\,|\,-1] = E\{\mathbf{s}[-1]\} = \boldsymbol{\mu}_s$$
$$\mathbf{M}[-1\,|\,-1] = E\left\{(\mathbf{s}[-1]\} - \hat{\mathbf{s}}[-1\,|\,-1])(\mathbf{s}[-1]\} - \hat{\mathbf{s}}[-1\,|\,-1])^T\right\} = \mathbf{C}_s$$

Prediction:

$$\hat{\mathbf{s}}[n\,|\,n-1] = \mathbf{A}\hat{\mathbf{s}}[n-1\,|\,n-1]$$

Pred. MSE (*p*×*p*):

$$\mathbf{M}[n\,|\,n-1] = \mathbf{A}\mathbf{M}[n-1\,|\,n-1]\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T$$

Kalman Gain (*p*×*M*):

$$\mathbf{K}[n] = \mathbf{M}[n\,|\,n-1]\mathbf{H}^T[n]\left(\mathbf{C}[n] + \underbrace{\mathbf{H}[n]\mathbf{M}[n\,|\,n-1]\mathbf{H}^T[n]}_{M\times M}\right)^{-1}$$

Update:

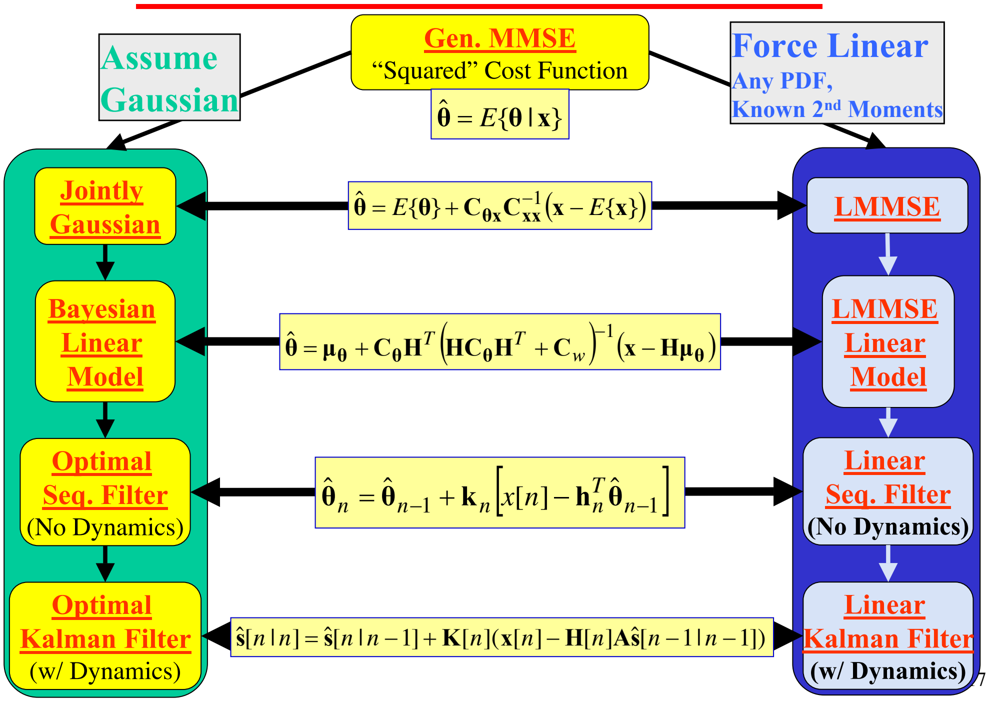
$$\hat{\mathbf{s}}[n\,|\,n] = \hat{\mathbf{s}}[n\,|\,n-1] + \mathbf{K}[n]\underbrace{(\mathbf{x}[n] - \underbrace{\mathbf{H}[n]\hat{\mathbf{s}}[n\,|\,n-1]}_{\hat{\mathbf{x}}[n|n-1]})}_{\tilde{\mathbf{x}}[n]:\ \text{innovations}}$$

Est. MSE (*p*×*p*):

$$\mathbf{M}[n\,|\,n] = \left(\mathbf{I} - \mathbf{K}[n]\mathbf{H}[n]\right)\mathbf{M}[n\,|\,n-1]$$

http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_29%20Ch_13B.pdf

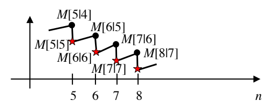
Overview of MMSE Estimation



http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE522_files/EECE%20522%20Notes_29%20Ch_13B.pdf

Properties of the Kalman filter

- extension of sequential MMSE (fixed parameter) to time-varying parameter with a known dynamical model
- Kalman filter is a time-varying filter (Kalman gain changes with n)
- Kalman filter computes and uses its performance measure M[n|n]
- prediction increases error, update decreases error



- as n→∞ Kalman filter reaches ``steady-state” and becomes a linear time-invariant filter (i.e. k[n] constant, M[n|n] constant)
- Kalman filter created uncorrelated sequence of ``innovations”
- Kalman filter is optimal for Gaussian, if not Gaussian, optimal Linear MMSE
- M[n|n-1], M[n|n], K[n] can be computed off-line (ahead of time)

More observations

- Kalman vs Wiener?
- What about if don’t have linear observation and dynamical models?
- Much more to know about Kalman filter!!!