Optimal Filters: Wiener and Kalman Filters

Aalto University
School of Engineering

Optimal Filtering

- Estimation of signals: dynamic, time varying parameter
- In state or signal estimation, 3 situations are possible depending upon the relationship of current time index k and and the sample size N. Let $y(N)$ be the last available measurement and k is the present time point.
  - If $N < k$ we are estimating a future value. A predicted estimate.
  - If $N = k$ we are using all past measurements and the most recent one to estimate the state. A filtered estimate.
  - When $N > k$ we are estimating an earlier value. A smoothed estimate.
- Predicting and filtering can be done in real time whereas smoothing can never be done in real time.
- Any estimate based on finite number of observations is expected to contain some error (noise, distortions).

Optimal Filtering

- Index of performance: the mean square error. The error $e(k) = s(k) - y(k)$ is the difference between the filter output $y(n)$ and desired filter output $s(k)$ (signal) and the expected value of the squared error is minimized.
- Such error criterion is mathematically tractable, single unique minimum exists for the error surface. Mean Square calculus.
- Principle of orthogonality $E[\epsilon_{opt}(k)x(k-m)] = 0$
- Geometrical interpretation for it, error is orthogonal to the observed signal
- The desired output is determined by the signal component we are interested in, i.e., the assumed signal model.
- Other performance indeces: mean absolute error.
- Applications include equalization, noise cancellation, spatial filtering, channel estimation, deconvolution

Commonly used Optimal Filters

- **Wiener Filter: optimal filter for scalar signals in wide sense stationary (WSS) scenarios.**
  - Minimizes the Mean Square Error (expected value of squared error)
  - FIR Wiener Filter, Causal IIR Wiener Filters, non-causal IIR Wiener Filter
- **Kalman Filter (KF) extends optimal filters to multichannel signals, multidimensional states and nonstationary scenarios**
  - Linear model and Gaussian probability distributions are assumed
  - Optimal Bayesian filter in minimum mean square error (MMSE) sense
- **Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Particle Filter extend KF for nonlinear and non-Gaussian (PF) filtering problems.**
- **Optimality may be lost because of linearization/approximation in EKF and UKF. They work very well in many applications anyway.**

Optimal Filtering

- Causality in an important topic if we are doing real time processing. However, in case of we have spatial parameter instead of time parameter e.g., in array and image processing, causality is not an important issue.
- Often (in particular in communications) signals are presented as follows

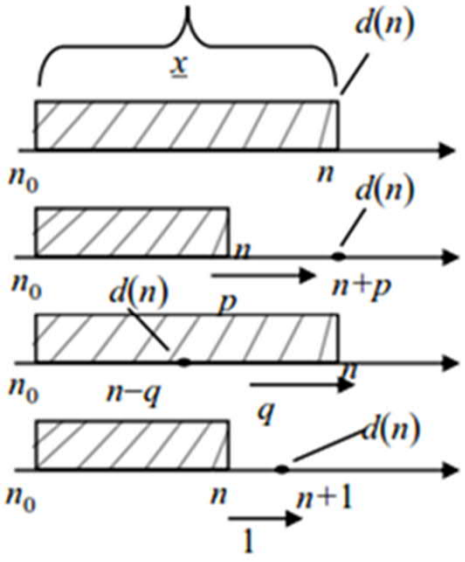
$$s(k) = s_I(k) + j s_Q(k)$$

  where $s_I(k)$ is in-phase (real) component and $j s_Q(k)$ is quadrature (imaginary) component.
- Magnitude-phase angle form $s(k) = |s(k)| e^{j\omega(k)}$ is used as well.
- Filters for real signals are special cases of filters for complex signals.
- In real form the operation of complex conjugation is removed and conjugate transposition is replaced by ordinary transposition.

Optimal Filtering

- We will study Wiener filtering (FIR, IIR, causal/noncausal) and Kalman Filtering here

Optimal Filtering: filtering, predicting, smoothing



Problem	Form of Observations	Desired Signal
Filtering of signal in noise	$x(n) = s(n)+w(n)$	$d(n) = s(n)$
Prediction of signal in noise	$x(n) = s(n)+w(n)$	$d(n) = s(n+p)$; $p > 0$
Smoothing of signal in noise	$x(n) = s(n)+w(n)$	$d(n) = s(n-q)$; $q > 0$
Linear prediction	$x(n) = s(n-1)$	$d(n) = s(n)$

Wiener Filters

- Estimation of a signal from another
- Applies to wide sense stationary (WSS) processes
- Discrete time formulation is given here
- Model for noisy observations $x(k) = s(k) + v(k)$
- $x(k)$ and $s(k)$ are jointly WSS and their autocorrelations
  $r_{xx}(l) = E[x(k)x^*(k-l)]$ and
  $r_{ss}(l) = E[s(k)s^*(k-l)]$ as well as cross-correlation
  $r_{sx}(l) = E[s(k)x^*(k-l)]$ are known.
- Autocorrelation matrix of x is R_{xx}.
- If the data are available to infinite past, the optimum filter has infinite impulse response (IIR)
- If only finite number of observations are available, the optimum filter is FIR.
- Input-output relationship using convolution $\hat{s} = y = h * x$.

Wiener Filters

- The observed data is input sequence $\{x(k)\}$ which passes through a Linear Time-Invariant (LTI) system producing an output sequence $\{y(k)\}$. $\{h(k)\}$ is the impulse response of the LTI-system.
- We will design a filter h that filters $x(k)$ and produces an estimate $\hat{s}(k)$ of desired signal $s(k)$.
- The error is defined $\epsilon(k) = s(k) - \hat{s}(k)$
- The performance index in Wiener filtering is the mean square error

$$J = E[|\epsilon(k)|^2] = E[(\hat{s}(k) - s(k))^2]$$

- By plotting $J = E[|\epsilon(k)|^2]$ as a function of filter coefficients $\{h\}$ a parabolic surface with single unique minimum is obtained.

Wiener Filters

- In order to find optimal coefficients h we need to differentiate J with respect to filter coefficients and set the derivatives to zero

$$\frac{\partial J}{\partial h^*(m)} = \frac{E[\epsilon(k)\epsilon^*(k)]}{\partial h^*(m)} = 0$$

  for all $m = 0, ..., p-1$ in the case of FIR filter and for infinite number of coefficients in the case of IIR filter.

- A necessary and sufficient condition for the cost function to attain its minimum is that the estimation error is orthogonal to each input observation in the estimation at time k.

Wiener Filters

- In case of noncausal filtering we need to have all the data available and the processing cannot be real time. In case of spatial processing (array and image processing) causality is less important issue.
- The goal is to find the impuse response $h(k)$ of the IIR filter with transfer function

$$H(z) = \sum_{m=-\infty}^{\infty} h(m) z^{-m}$$

such than mean square error

$$J = E[|\epsilon(k)|^2]$$

is minimized.

Wiener Filters

- Error $\epsilon(k)$ is defined here as

$$\epsilon(k) = s(k) - \hat{s}(k) = s(k) - \sum_{m=-\infty}^{\infty} h(m)x(k-m)$$

- From orthogonality principle, best h satisfies

$$\sum_{m=-\infty}^{\infty} h(m)r_{xx}(k-m) = r_{sx}(k),$$

where $r_{xx}()$ is autocorrelation of x and $r_{sx}()$ is cross correlation of s and x. The expression on the LHS is in the familiar form of convolution sum and it can be written as $h(k) * r_{xx}(k)$

- In terms of z-transforms of time sequences ($k = -\infty, ..., \infty$) and we have

$$H(z)\Phi_{xx}(z) = \Phi_{sx}(z)$$

Wiener Filters

- The transfer function of the filter is

$$H(z) = \frac{\Phi_{sx}(z)}{\Phi_{xx}(z)} \tag{1}$$

- The frequency response is obtained by substituting $z = e^{j\omega}$

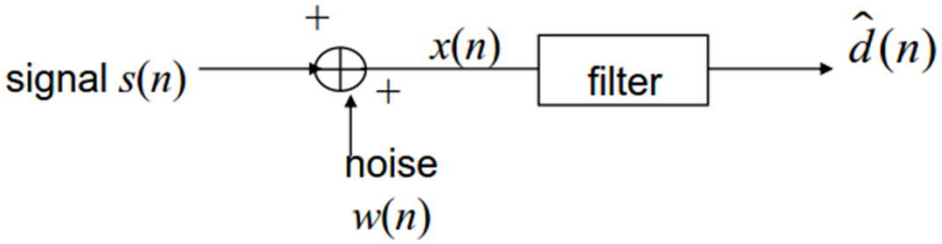
$$H(e^{j\omega}) = \frac{\Phi_{sx}(e^{j\omega})}{\Phi_{xx}(e^{j\omega})}.$$

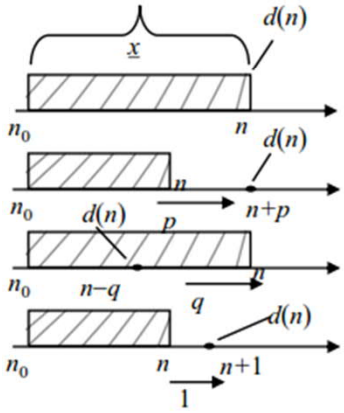
- In many applications the desired output is forward predicted α time units. The transfer function is then

$$H(z) = \frac{\Phi_{sx}(z)z^{\alpha}}{\Phi_{xx}(z)}$$

which differs from (??) by z^{α} caused by the forward prediction.

Wiener Filters





Typical Wiener Filtering Problems

Problem	Form of Observations	Desired Signal
Filtering of signal in noise	$x(n) = s(n)+w(n)$	$d(n) = s(n)$
Prediction of signal in noise	$x(n) = s(n)+w(n)$	$d(n) = s(n+p)$; $p > 0$
Smoothing of signal in noise	$x(n) = s(n)+w(n)$	$d(n) = s(n-q)$; $q > 0$
Linear prediction	$x(n) = s(n-1)$	$d(n) = s(n)$

Wiener Filters

- Filter coefficients h can the be obtained from the frequency response $H(e^{j\omega})$ by inverse Fourier transform.
- Recall that autocorrelation and power spectra are a Fourier transform pair, so $\Phi_{xx}(e^{j\omega})$ is the power spectrum of x and $\Phi_{sx}(e^{j\omega})$ is the cross power spectrum between x and s.
- The minimum mean square error

$$J_{min} = r_{ss}(0) - \sum_{m=-\infty}^{\infty} h(m) r_{sx}^{*}(m)$$

EX: Smoothing Wiener Filter

- Observation model is $x(k) = s(k) + v(k)$.
- Signal and noise are assumed to be uncorrelated and zero mean. The autocorrelation of x can then be expressed as $r_{xx}(m) = r_{ss}(m) + r_{vv}(m)$.
- Power spectra were obtained by applying Fourier transform to autocorrelations. As a result

$$\Phi_{xx}(e^{j\omega}) = \Phi_{ss}(e^{j\omega}) + \Phi_{vv}(e^{j\omega})$$

- The cross correlation is

$$r_{sx}(m) = E[s(m)x^*(k-m)] = E[s(m)s^*(k-m)] + E[s(m)v^*(k-m)]$$

  where the latter term on the RHS is zero based on orthogonality and we get $r_{sx}(m) = r_{ss}(m)$

- The corresponding power spectra are:

$$\Phi_{sx}(e^{j\omega}) = \Phi_{ss}(e^{j\omega})$$

EX: Smoothing Wiener Filter

- The frequency response of a Wiener filter for smoothing is

$$H(e^{j\omega}) = \frac{\Phi_{ss}(e^{j\omega})}{\Phi_{ss}(e^{j\omega}) + \Phi_{vv}(e^{j\omega})}$$

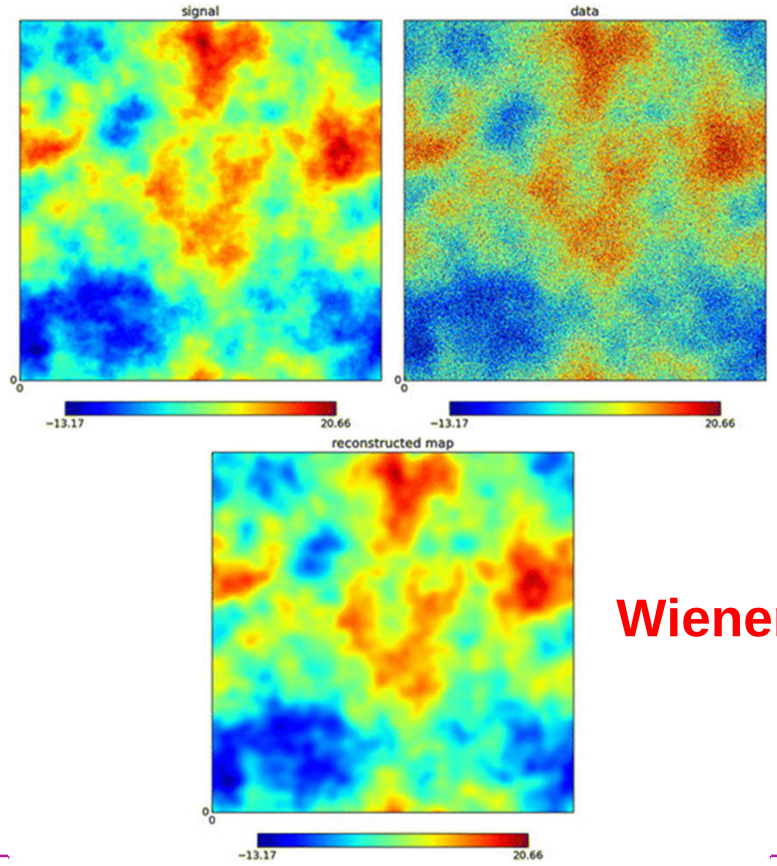
- The quotient above and consequently the filter gain $|H(e^{j\omega})| \approx 1$ for frequencies where the noise power is small compared to the signal power whereas gain are close to zero and the noise is attenuated in those frequencies where the noise power is large compared to the signal power.
- The mean square error of smoothing filter is

$$J_{min} = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\Phi_{ss}(e^{j\omega}) - H(e^{j\omega})\Phi_{sx}(e^{j\omega})] d\omega$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{ss}(e^{j\omega})[1 - H(e^{j\omega})] d\omega$$

Ex: noisy image smoothing using Wiener Filter

Ideal noise-free image



Observed noisy image

Wiener filter output

Causal FIR Wiener Filter

- In this case the error is

$$\epsilon(k) = s(k) - \hat{s}(k) = s(k) - \sum_{i=0}^{p-1} h(i)x(k-i)$$

- discrete time version of the Wiener-Hopf equation

$$\sum_{l=0}^{p-1} h_{opt}(l) r_{xx}(k-l) = r_{sx}(k-l)$$

***p* linear equations and *p* unknowns *h(l)* in Matrix form: $R_{xx}\, h_{opt} = r_{sx}$**

where r_{xx} are autocorrelations, r_{sx} cross-correlations and h_{opt} are the coefficients of the optimum filter.

- Matrix form solution in case of finite number of observations.

$$h_{opt} = R_{xx}^{-1} r_{sx},$$

where $r_{sx} = [r_{sx}(0)\ r_{sx}(1)\ \ldots\ r_{sx}(p-1)]^T$

Causal FIR Wiener Filter

Autocorrelation matrix has Toeplitz structure for WSS signals. Such matrix is easier to invert. Autocorrelation sequence is conjugate symmetric, i.e. $r_x(k) = r_x^*(k)$ for complex-valued data

- Ex: (p+1) x (p+1) dimensional Rx for complex data x

$$\begin{bmatrix} r_x(0) & r_x^*(1) & \cdots & r_x^*(p-1) \\ r_x(1) & r_x(0) & \cdots & r_x^*(p-2) \\ r_x(2) & r_x(1) & \cdots & r_x^*(p-3) \\ \vdots & \vdots & & \vdots \\ r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{bmatrix}$$

Large Toeplitz matrices can be well approximated with circulant matrices that are diagonalized using DFT

Causal FIR Wiener Filter

- Minimum mean square error

$$J = E[|\epsilon(k)|^2] = E[\epsilon(k)\{s(k) - \sum_{i=0}^{p-1} h(i)x(k-i)\}^*]$$

and rerranging by changing the order of summation and expectation and using the orthogonality property $E[\epsilon_{opt}(k)x(k-i)] = 0$ for Wiener solution we get

$$J_{min} = E[\epsilon_{opt}(k)s^*(k)] = E[\{s(k) - \sum_{i=0}^{p-1} h_{opt}(i)x(k-i)\}s^*(k)]$$

$$= r_{ss}(0) - r_{sx}^H R_{xx}^{-1} r_{sx}$$

- Causal FIR Wiener filtering, using data upto and including current observation

EX: Causal FIR Wiener Filter

- Signal $s(k)$ is assumed to be corrupted by additive noise and we observe

$$x(k) = s(k) + v(k)$$

the cross correlation is now

$$r_{sx}(l) = E[s(k)x^*(k-l)] = E[s(k)s^*(k-l)] + E[s(k)v^*(k-l)] = r_{ss}(l)$$

where $E[s(k)v^*(k-l)] = 0$ based on orthogonality. The autocorrelation is

$$r_{xx}(l) = E[x(k)x^*(k-l)] = E[\{s(k) + v^*(k)\}\{s(k-l) + v^*(k-l)\}^*]$$

- If signal and noise are uncorrelated

$$r_{xx}(l) = r_{ss}(l) + r_{vv}(l)$$

and the Wiener-Hopf equations may be presented by

$$h_{opt} = [R_{ss} + R_{vv}]^{-1} r_{ss}$$

EX: Causal FIR Wiener Filter

- Linear predition of future values using current and past observations.

$$\hat{x}(k+\alpha) = \sum_{i=0}^{k-1} h(i)x(k-i)$$

- We will address the future value of the signal as the desired signal, i.e., $s(k) = x(k+\alpha)$, where α is the number of time units forward predicted. The cross correlation between $s(k)$ and $x(k)$:

$$r_{sx}(l) = E[s(k)x^*(k-l)] = E[x(k+\alpha)x^*(k-l)] = r_{xx}(\alpha+l)$$

- Wiener-Hopf equations are now in matrix form

$$\mathbf{h}_{opt} = R_{xx}^{-1}\mathbf{r}_{x\alpha}$$

where $\mathbf{r}_{x\alpha} = [r_{xx}(\alpha)\ r_{xx}(\alpha+1)\ \ldots\ r_{xx}(\alpha+p-1)]^T$

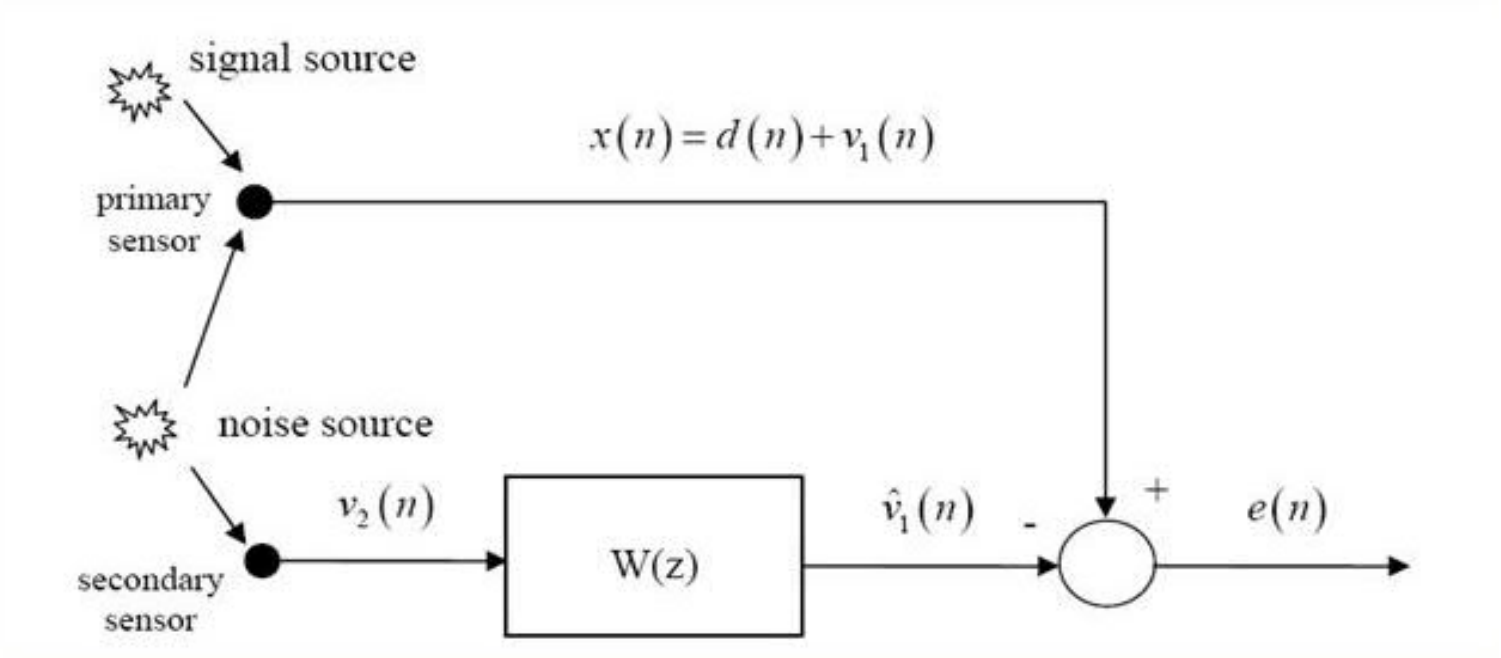
EX: Causal FIR Wiener Filter

- The minimum mean square error is then

$$J_{min} = E[s(k) - \sum_{i=0}^{p-1} h_{opt}(i)x(k-i)s^*(k)]$$

$$= r_{ss}(0) - r_{x\alpha}^H h_{opt}$$

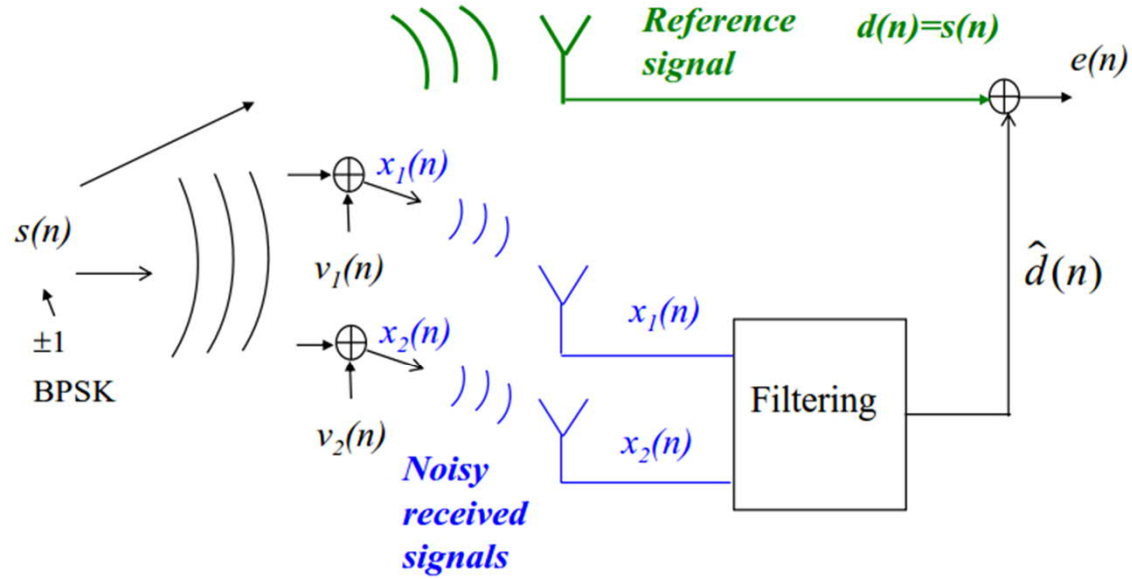
EX: Causal FIR Wiener Filter as noise canceller



W(z) is FIR Wiener Filter

EX: Spatial filtering

❖ **Application to Spatial Filtering**



❖ Information Available: *snapshot in time of received signal retrieved at two antennas & reference signal*

❖ Assumption: $v_1(n)$, $v_2(n)$ *zero mean wss white noise RPs independent of each other and of s(n).*

❖ Goal: Denoise received signal

EX2: Causal FIR Wiener Filter

- Noise Cancellation using FIR Wiener filter
- Let us assume that our desired signal is sinusoidal

$$d(n) = cos(n\omega_0 + \phi)$$

where frequency $\omega_0 = 0.05\pi$

- The noise sequences $v_1(n)$ and $v_2(n)$ (signal from reference sensor) are AR(1) processes obtained by the following difference equations

$$v_1(n) = 0.8v_1(n-1) + g(n)$$

$$v_2(n) = -0.6v_2(n-1) + g(n)$$

where $g(n)$ are zero mean white noise sequences uncorrelated with the desired signal $d(n)$

EX2: Causal FIR Wiener Filter

- The observed signal is

$$y(n) = d(n) + v_1(n)$$

and reference signal $v_2(n)$ is used to estimate $v_1(n)$ and cancel its effect.

- Autocorrelation for $v_2(n)$

$$\hat{r}_{v_2 v_2}(k) = \frac{1}{N} \sum_{n=0}^{N-1} v_2(n) v_2(n-k)$$

and cross-correlation

$$\hat{r}_{y v_2}(k) = \frac{1}{N} \sum_{n=0}^{N-1} y(n) v_2(n-k)$$

EX2: Causal FIR Wiener Filter

- The Wiener filter h is obtained by solving Wiener-Hopf equation

$$\hat{h} = R_{v_2}^{-1} r_{yv_2}$$

- Obtained filter is used to estimate $v_1(n)$ which is subtracted from observations $y(n)$ in order to obtain the desired sinusoidal $d(n)$.

Sketch of Causal IIR Wiener Filter

- Now the goal is to find a stable and causal linear filter witha a stable and causal inverse filter that will convert the input sequence $\{x(k)\}$ into white sequence $\{\tilde{x}(k)\}$.

- The combination of the causal whitening filter and the best causal estimate on filter acting on $\{\tilde{x}(k)\}$ will form the best causal filter acting on $\{x(k)\}$.
- Let us assume that $\frac{1}{G(z)}$ is the transfer function of such a causal, stable, linear invertible whitening filter.

$$\{x(k)\} \rightarrow \frac{1}{G(z)} \rightarrow \{\tilde{x}(k)\}.$$

The autocorrelation for the white sequence is $r_{\tilde{x}\tilde{x}}(n) = \delta(n)$ and its z-transform

$$\Phi_{\tilde{x}\tilde{x}}(z) = 1 = \frac{1}{G(z)G^*(1/z^*)}\Phi_{xx}(z)$$

Sketch of Causal IIR Wiener Filter

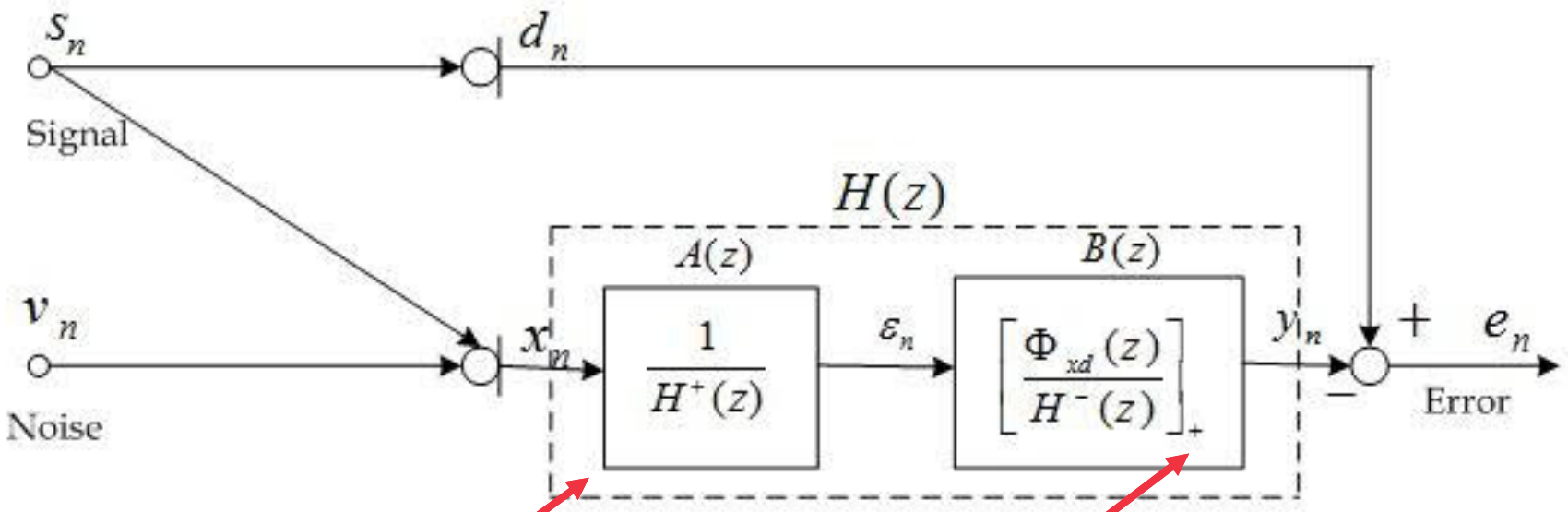
- The overall optimum causal filter is the combination of the whitening filter and the optimum filter for $\{\tilde{x}(k)\}$, i.e.,

$$H(z) = \frac{1}{G(z)} \left[\frac{\Phi_{sx}(z)}{G^*(1/z^*)} \right]_+$$

Causal part of filter

which means that the impulse response truncated at $m = 0$ and $m < 0$ values are set to zero.

EX: Causal IIR Wiener Filter noise canceller



Whitening filter

Causal Wiener filter acting on $\{\tilde{x}(k)\}$ (ouput of whitening filter)

Steepest Descent and Adaptive Filtering

- Steepest descent is an old, deterministic optimization method.
- Stochastic gradient-based methods are based on it.
- It is an approach that finds the minimum of the error performance surface such as Mean Square Error

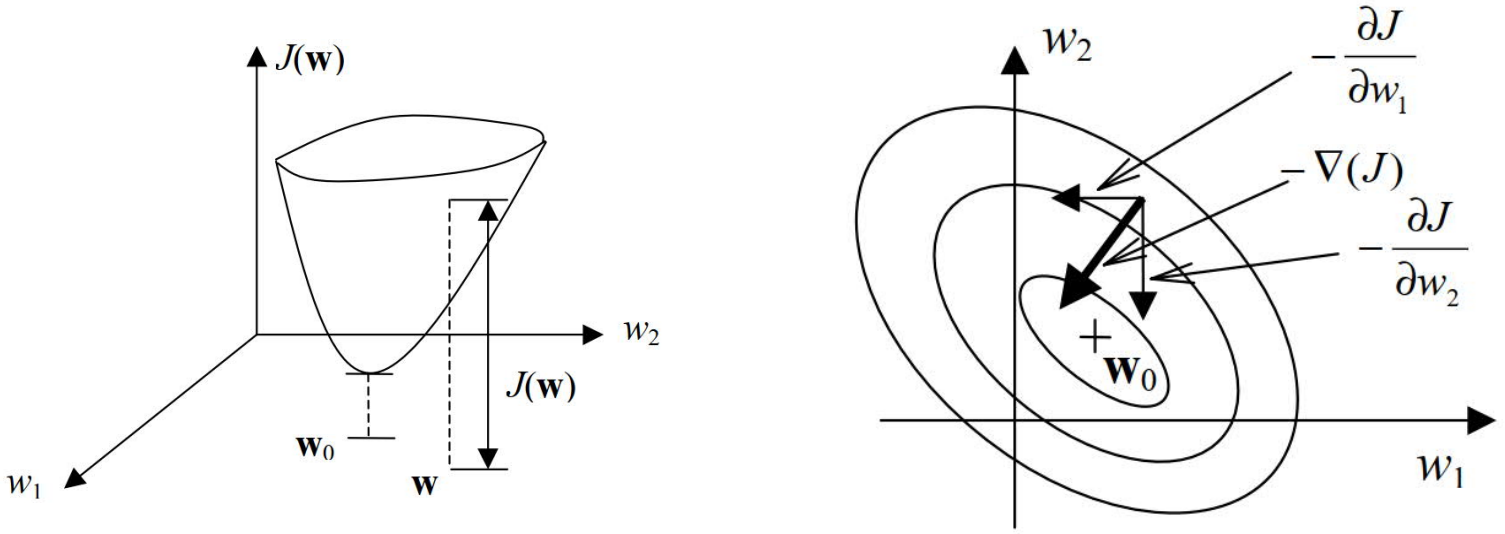
$$J(n) = E\{|e(n)|^2\}$$

- Error surface must be known
- Adaptive approach that finds the optimal filter

$$w_o = R^{-1}_{xx}\, r_{dx}$$

without inverting matrix R. Here d denotes the desired signal and x observed signal

EX: Error surface for 2 coefficients w_1, w_2



- If we would drop a ball to this bowl shaped error surface, it would reach the minimum following the path of steepest descent.
- We will move in the direction of negative gradient until the optimum is reached

Steepest Descent and Adaptive Filtering

- At each time instance *n*, we are taking a step in the direction of negative gradient until the optimum coefficients *w* are found

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)]$$

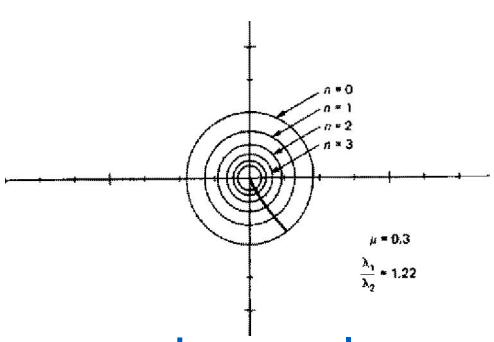
- The stepsize is denoted by μ
- Since the method uses feedback, the step size must be chosen appropriately to guarantee stability
- The eigenvalues of autocorrelation matrix *R* can be used to ensure stability

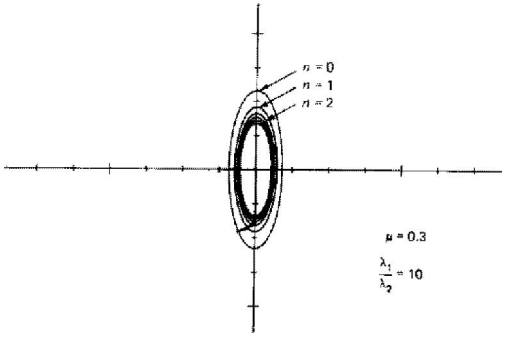
Steepest Descent and Adaptive Filtering

- Safe choice (Widrow) based on maximum eigenvalue of *R*:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

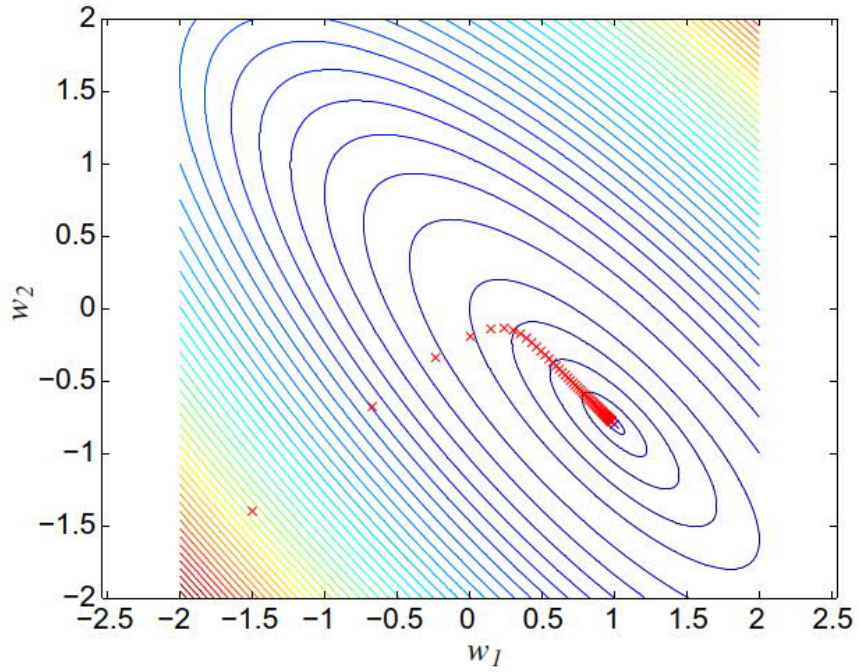
- Eigenvalue spread impacts the convergence: $\frac{\lambda_{\max}}{\lambda_{\min}}$





- Large eigenvalue spread leads to slower convergence

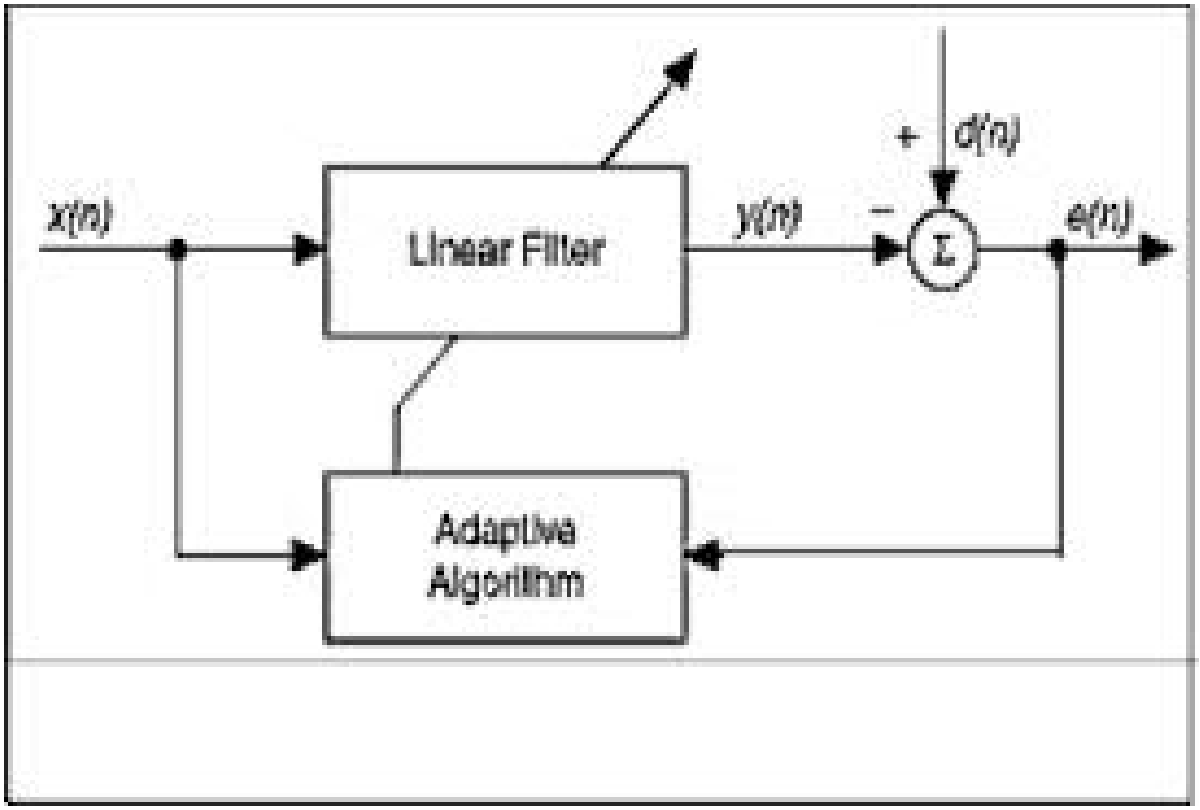
SD convergence when large ratio of eigenvalues (eigenvalue spread)



Least Mean Square (LMS) adaptive filter

- In practise we are often dealing with time varying systems and the signal properties may change as a function of time. Hence, the assumption on WS Stationarity used in Wiener filtering does not hold.
- In order to perform filtering somehow in optimal manner the filter transfer function $H(z)$ and consequently filter coefficients must vary based on changing signal characteristics.
- Consequently, the coefficients are data dependent and the superposition principle does not hold. Hence, we can say that adaptive filter is a nonlinear device (although the filter itself may perform linear combination of input values)
- Adaptive filter consists of two elements: a discrete time filter and an adaptive algorithm for changing the coefficients of $H(z)$.

LMS adaptive filter



LMS adaptive filter

- Similarly to Wiener filtering adaptive algorithms use the difference between the true response $y(k) = \hat{s}(k)$ and the desired signal $s(k)$ of the system to modify the coefficients h_k. The error is defined as $\epsilon(n) = s(k) - y(k) = s(k) - \hat{s}(k)$.
- FIR LMS filter with p coefficients
- Least Mean Square algorithm is widely used and simplest adapive filtering algorithm. Let the input-output relationship be

$$y(k) = \sum_{m=0}^{p-1} h_k(m)x(k-m),$$

i.e., the system is a time variant FIR-filter where $h_k(i)$'s are p varying filter coefficients, k is a time index, x and y are the filter input and output, respectively.

LMS adaptive filter

- The error is defined as follows

$$\epsilon = s(k) - \hat{s}(k) = s(k) - \sum_{m=0}^{p-1} h_k(m)x(k-m)$$

and the filter coefficients are changed for each sample such that mean square error (MSE) is minimized:

$$J = E[|\epsilon(k)|^2].$$

- Similarly to Wiener filtering: In order to find optimal coefficients, the error function J is differentiated with respect to the filter coefficients h_k. As a result we get the surface gradient with respect to the coefficients h_k.

$$\nabla_k J = \frac{\partial J}{\partial \mathbf{h}_k^*} = \frac{E[\epsilon(k)\epsilon^*(k)]}{\partial \mathbf{h}_k^*} \quad \text{where} \quad \mathbf{h}_\mathrm{k}^* = [\mathrm{h}_\mathrm{k}^*(0)\ \mathrm{h}_\mathrm{k}^*(1)\ \cdots\ \mathrm{h}_\mathrm{k}^*(\mathrm{p}{-}1)]^\mathrm{T}$$

LMS adaptive filter

- The gradient may be evaluated
$$\nabla_k J(k) = -E[\epsilon(k)x^*(k)]$$
- LMS is a steepest descent method which iteratively looks for the minimum. Steepest descent direction is the direction of negative gradient.
- No matrix inversion is required.
- The coefficients are updated at the arrival of new observation as follows
$$\mathbf{h}_{k+1} = \mathbf{h}_k - \mu\nabla_k J(k) = \mathbf{h}_k + \mu E[\epsilon(k)x^*(k)]$$
where μ is a parameter that controls the rate of convergence.
- The update of the coefficients is proportional to the negative gradient of the error surface. MSE gets smaller on each step and the gradient equals zero at the optimum. This solution would correspond to the Wiener filtering solution.

LMS adaptive filter

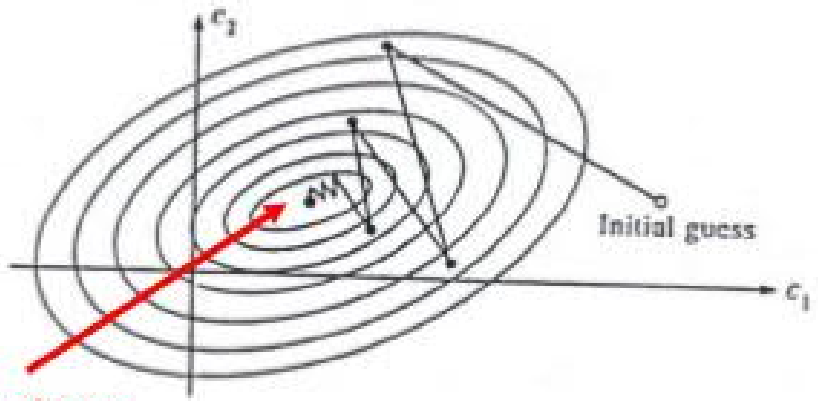
- In the Wiener filtering derivation we were minimizing the expected value of the squared error. However, this expecteted value and consequently the gradient are usually unknown and have to be estimated.
- The coefficient update is rewritten using the instanteneous estimate of the error instead of expected error

$$h_{k+1} = h_k - \mu \hat{\nabla}_k J(k) = h_k + \mu \epsilon(k) x^*(k)$$

  i.e., $-E[\epsilon(k)x^*(k)]$ is replaced by $-\epsilon(k)x^*(k)$. The error $\epsilon(k)$ is given in vector form $y(k) - h_k^T x(k)$
- If we have the input sequence $\{x(k)\}$ and the desired signal $\{s(k)\}$, only the parameter μ controlling the convergence have to be chosen.
- If μ is very small the filter adapts very slowly.

EX: LMS adaptive channel estimation, steepest descent

: Example for the Unknown Channel of 2nd order



Desired Combination of taps

LMS adaptive filter

- On the other hand, if μ is large the update may contain a significant error and the adaptive filter does not converge to the optimum.
- Typically the optimal (Wiener) solution is not obtained and the actual solution fluctuates in the vicinity of the optimal solution.
- A general condition for the convergence is given by

$$0 < \mu < \frac{2}{\lambda_{max}},$$

where λ_{max} is the largest eigenvalue of the autocorrelation matrix R_{xx} of data x.

- A commonly used condition for μ may be given as follows

$$0 < \mu < \frac{2}{Tr(R_{xx})} = \frac{2}{\text{total signal power}},$$

where $Tr(R_{xx})$ is the trace of the autocorrelation matrix.

Least Mean Square (LMS) adaptive filter

- Such step-size guarantees convergence in the Mean Square sense.
- Haykin: The mean square error J converges to a steady-state which exceeds the minimum mean square error J_{min} because we are estimating the gradient based on one observation (high variance in estimates). The excess error is denoted by J_{ex} and the mean square error is given as follows:

$$J_{MS} = J_{min} + J_{ex}(\infty) = J_{min} \frac{1}{1 - \mu \sum_{k=0}^{p} \frac{\lambda_k}{2-\lambda_k}}$$

- The misadjustment is the ratio of the steady state excess MS error to the minimum mean square error

$$\mathcal{M} = \frac{J_{ex}(\infty)}{J_{min}}$$

Least Mean Square (LMS) adaptive filter

- It can be approximated by

$$\mathcal{M} \approx \frac{1}{2}\mu\, tr(R_{xx}).$$

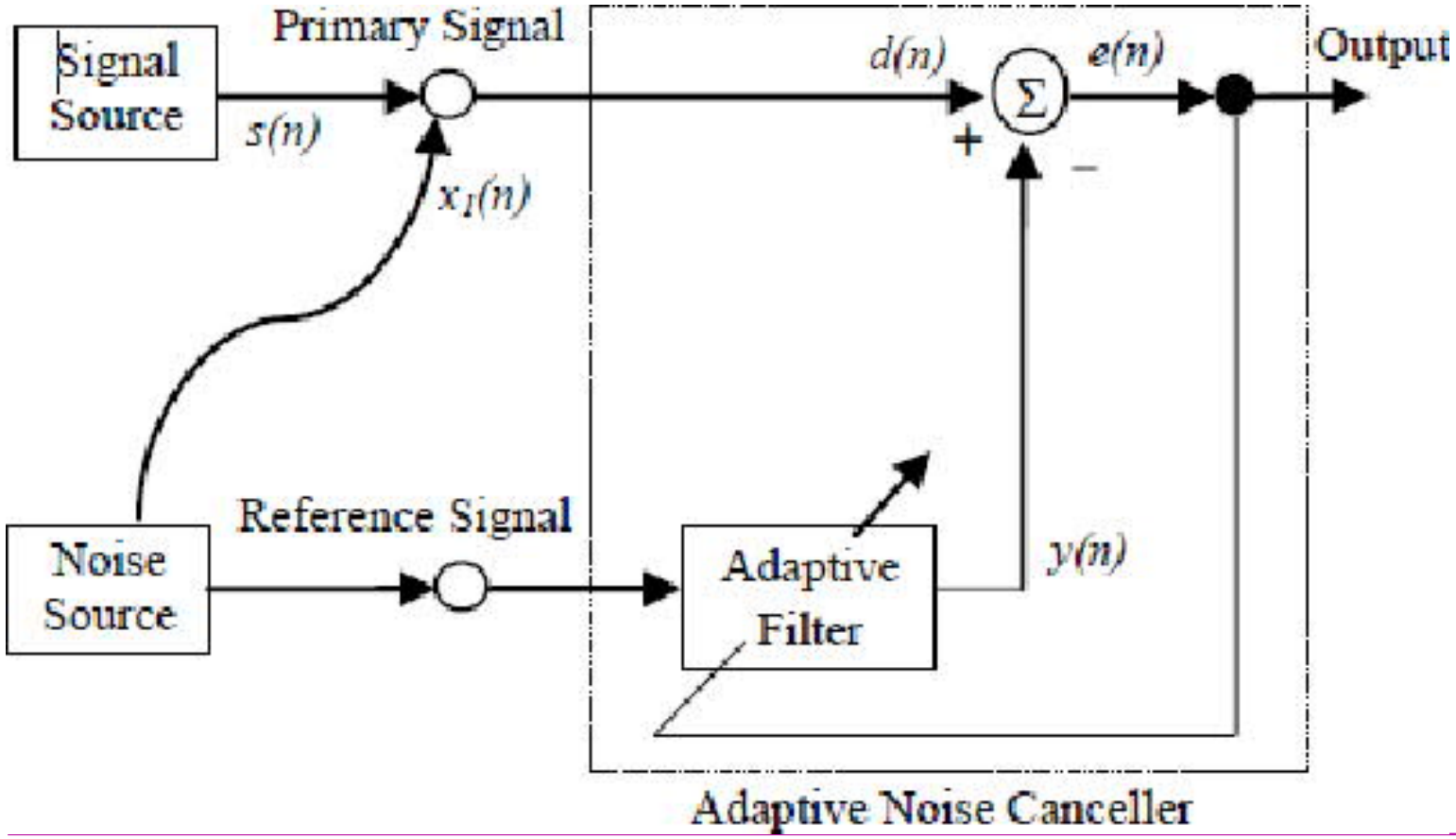
- By using the estimate $\hat{\sigma}_k^2$ for the total signal power, the LMS coefficient update may be rewritten

$$h_{k+1} = h_k + \frac{2\mu\epsilon(k)x^*(k)}{p\hat{\sigma}_k^2} \quad \text{and} \quad \hat{\sigma}_{\mathrm{k}}^2 = \alpha \mathrm{x}^2(\mathrm{k}) + (1-\alpha)\hat{\sigma}_{\mathrm{k-1}}^2,$$

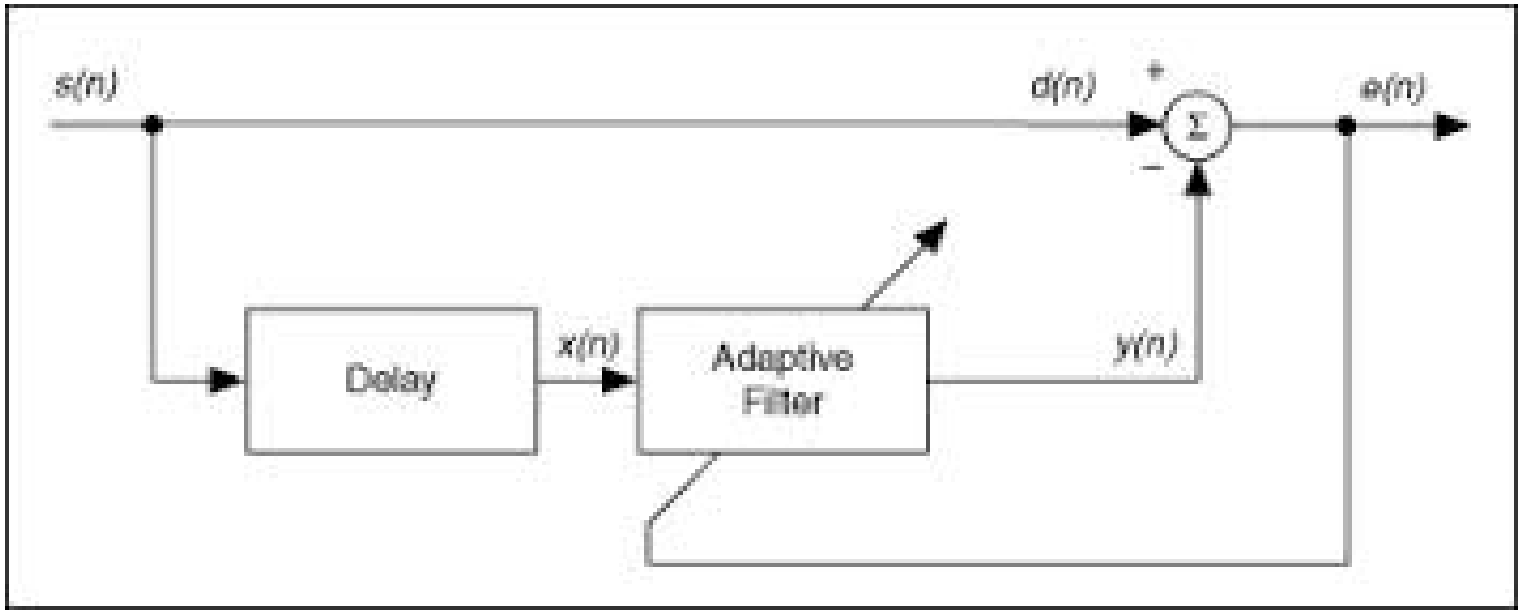
  where $0 < \alpha \ll 1$ is a forgetting factor which decreases the the influence of the past observations to the current estimate. This is necesary since we are dealing with time varying systems.

- There are quite a few modifications of the basic LMS algorithm including Normalized LMS, Leaky LMS, etc and the convergence parameter may vary, etc.

EX: LMS adaptive noise canceller



EX: LMS linear prediction



EX: LMS linear prediction

- Second order AR-model is defined by the difference equation

$$y(n) = 1.2728y(n-1) - 0.81y(n-2) + v(n)$$

  where $v(n)$ is white noise with unit variance.
- The optimum coefficients $h(1) = 1.2728, h(2) = -0.81$ can be directly estimated if autocorrelation sequence of $y(n)$ is known.
- The input to the adaptive filter is a delayed version of the observed signal.

EX: LMS linear prediction

- The adaptive linear predictor for estimation of the coefficients $h_n(1), h_n(2)$ is

$$\hat{y}(n) = h_n(1)y(n-1) + h_n(2)y(n-2)$$

- The LMS-update for the coefficients is

$$h_{n+1}(k) = h_n(k) + \mu e(n)x^*(n-k)$$

- In order to converge towards Wiener solution, the value of μ have to be chosen sufficiently small such as $\mu = 0.004$.

EX: LMS adaptive line enhancement

- The task at hand is adaptive noise cancellation from

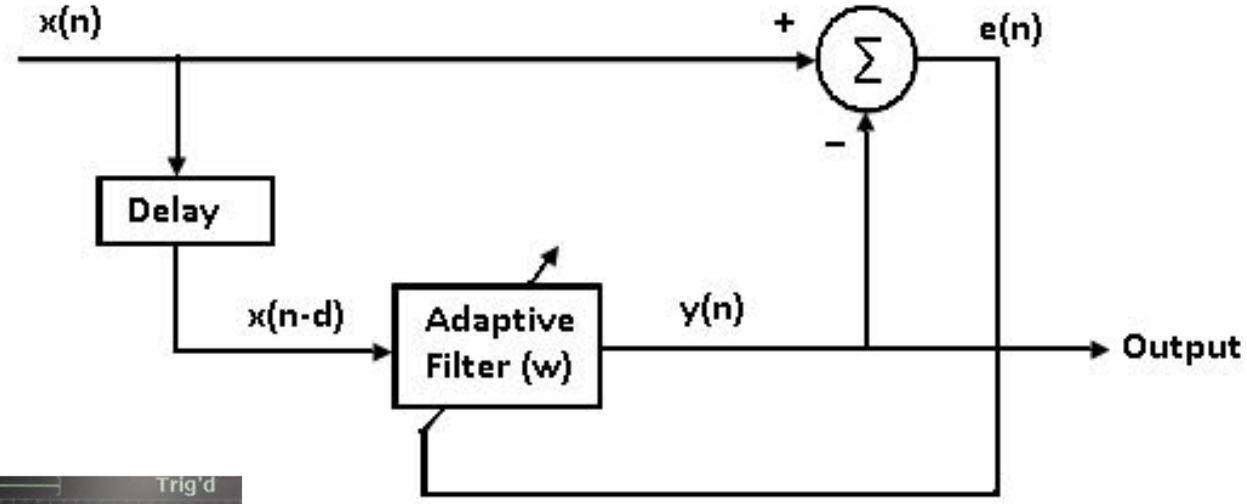
$$y(n) = d(n) + v(n)$$

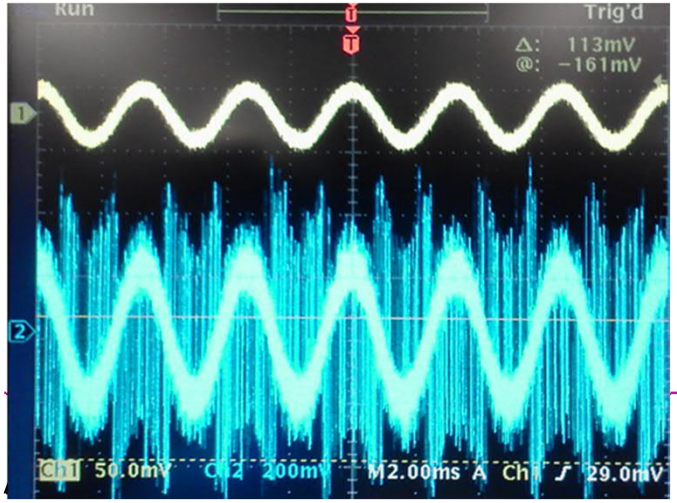
  where $v(n)$ is the noise to be cancelled, $d(n)$ is desired signal uncorrelated with the noise and $y(n)$ is the measured signal.
- Unfortunately, no secondary reference signal that could be used to cancel $v(n)$ is available
- Therefore, a delayed version of the input signal is used a reference signal.
- Noise are assumed to be uncorrelated over time and desired signal is correlated over time.

EX: LMS adaptive line enhancement

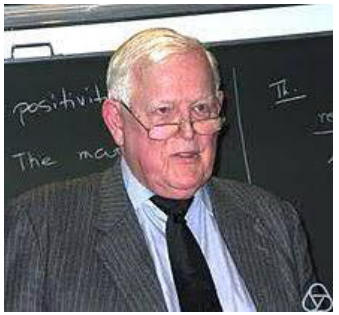
- Time delay makes reference signal correlated with the desired signal component $d(n)$ but the noise components are uncorrelated.
- The output of the adaptive filter is an estimate of the desired signal and the difference between the measured signal and adaptive filter output is an estimate of the noise to be cancelled.
- The method can be used for spectrum estimation as well. The amplitude response of the adaptive filter is proportional to the frequency content of input data.

EX: LMS adaptive line enhancement





Kalman Filters



- R.E. Kalman 1960, Swerling/Rand corporation memo has also claimed the results.
- Gaussian assumption is crucial for optimality (Gaussian thermal noise in sensors), random inputs to the system (e.g., due to turbulence)
- Kalman filters extend the Wiener filter to nonconstant coefficient, multivariate systems and nonstationary signal/noise, nonlinear models and gives a sequential solution
- Kalman filter is derived here as an optimal mean square error filter using the Bayesian approach. (no linearity needs to be assumed but it follows from the Bayesian approach that the optimal filter will be linear)

Kalman Filters

- Recall conditional mean $E[x|y]$: Suppose that x and y are jointly Gaussian with means μ_x, μ_y (and covariances...) then $E[x|y]$ is Gaussian with mean

$$E[x|y] = \mu_x + C_{xy}C_{yy}^{-1}(y - \mu_y)$$

and covariance

$$Cov(x|y) = E[(x - E[x|y])(x - E[x|y])^T] = C_{xx} - C_{xy}C_{yy}^{-1}C_{yx}$$

where C is a covariance matrix.

- The computation of conditional mean at arrival of each new observation is time consuming.
- The amount of memory required would grow.
- In order to do real time estimation it would be convenient to have recursive formula for $E[x|y] = E[x(k)|y(k), ..., y(0)]$

Kalman Filters

- Kalman filter is a collection of recursive formulas for computing the conditional mean for linear system with Gaussian noises and initial state.
- It avoids growing memory problem.

Kalman Filters: state variable model

- A necessary model in state estimation problems
- Allows for describing nonstationary time varying systems. The model is given as follows:

$$x(k) = F(k|k-1)x(k-1) + B(k|k-1)u(k-1)$$
$$+G(k|k-1)w(k-1)$$
$$y(k) = H(k)x(k) + v(k),$$

  where w and v are mutually uncorrelated, jointly Gaussian white noise sequences and u is a known input vector (e.g. control input, sensor platform motion), y is the measurement.
- From here on, the known input u is assumed to be zero here without any loss of generality.

Kalman Filters: state variable model

- The following assumptions are made

$$E[w(i)w^T(j)] = Q(i)\delta_{ij}$$

$$E[v(i)v^T(j)] = R(i)\delta_{ij}$$

$$E[w(i)v^T(j)] = 0 \text{ forall } i, j$$

- w is the state noise term, and contains modeling errors, errors in control input and other uncertainties in the system.
- v is the measurement noise term and models sensor noise.
- This model does not cover all situations: w and v may be correlated, noises may be colored, measurements may be "too" accurate (practically no measurement noise), ...
- Noises are typically assumed to be Gaussian (filter is optimal then) otherwise the filter is best linear estimator.

Kalman Filters: state variable model

- If $x(0)$, $w(k)$ and $v(k)$ are jointly Gaussian, so is $y(k)$.
- We may have: Scalar state and scalar measurements; scalar measurements and vector state; or vector measurements and vector state.
- F, G, B, H are known sequences of matrices.

Kalman Filtering Equations

- We group the equations comprising the Kalman Filter into two groups: prediction (time update) and correction (measurement update) equations; and the computation is then done in two stages.
- The Kalman Filter is defined by:
  (1) Prediction equations (prediction of the state of the system at time k using the measurements up to time $k-1$). The predicted state $\hat{x}(k|k-1) = E[x(k)|y(k-1), ..., y(0)]$ is given

$$\hat{x}(k|k-1) = F(k|k-1)\hat{x}(k-1|k-1)$$

  and prediction error covariance

$$P(k|k-1) = F(k|k-1)P(k-1|k-1)F^T(k|k-1)$$

$$+G(k|k-1)Q(k-1)G^T(k|k-1)$$

Kalman Filtering Equations

- $Q(k)$ is the state noise covariance matrix, and $w(k)$ is zero mean Gaussian sequence, $P(k|k-1)$ is prediction error covariance matrix and $P(k-1|k-1)$ is estimation error covariance matrix.
- (2) Correction equations use the new measurement to update the predicted state estimate to filtered state estimate $\hat{x}(k|k) = E[x(k)|y(k), ..., y(0)]$ is given

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)[y(k) - H(k)\hat{x}(k|k-1)]$$

and covariance matrix of filtered estimate of the state

$$P(k|k) = P(k|k-1) - K(k)H(k)P(k|k-1)$$

where Kalman gain

$$K(k) = P(k|k-1)H^T(k) \times [H(k)P(k|k-1)H^T(k) + R(k)]^{-1}$$

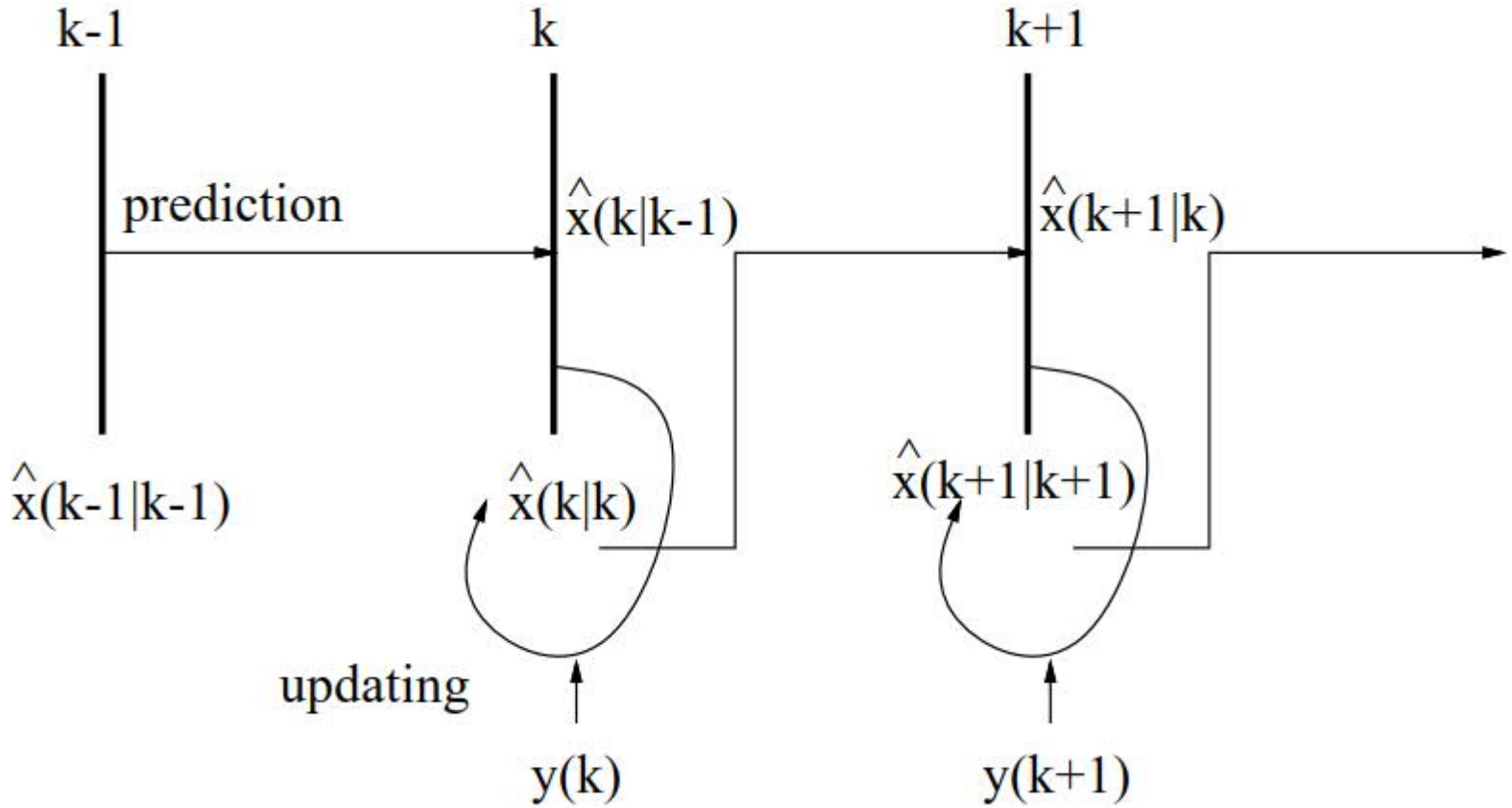
Kalman Filtering Equations

- $K(k)$ is the Kalman gain which specifies the amount the measurement prediction residual or innovation $\tilde{y}(k|k-1) = y(k) - H(k)\hat{x}(k|k-1)$ have to be multiplied to get the correction which updates the predicted state $\hat{x}(k|k-1)$ of $x(k)$ to new filtered state estimate $\hat{x}(k|k)$.
- Two stage computation in Kalman Filtering
- A filtered estimate of state is obtained from a predicted value $\hat{x}(k|k-1)$ by performing a correction step $K(k)\tilde{y}(k)$ when new measurement is obtained. The measurement residual $\tilde{y}(k)$ is

$$\tilde{y}(k) = y(k) - H(k)\hat{x}(k|k-1).$$

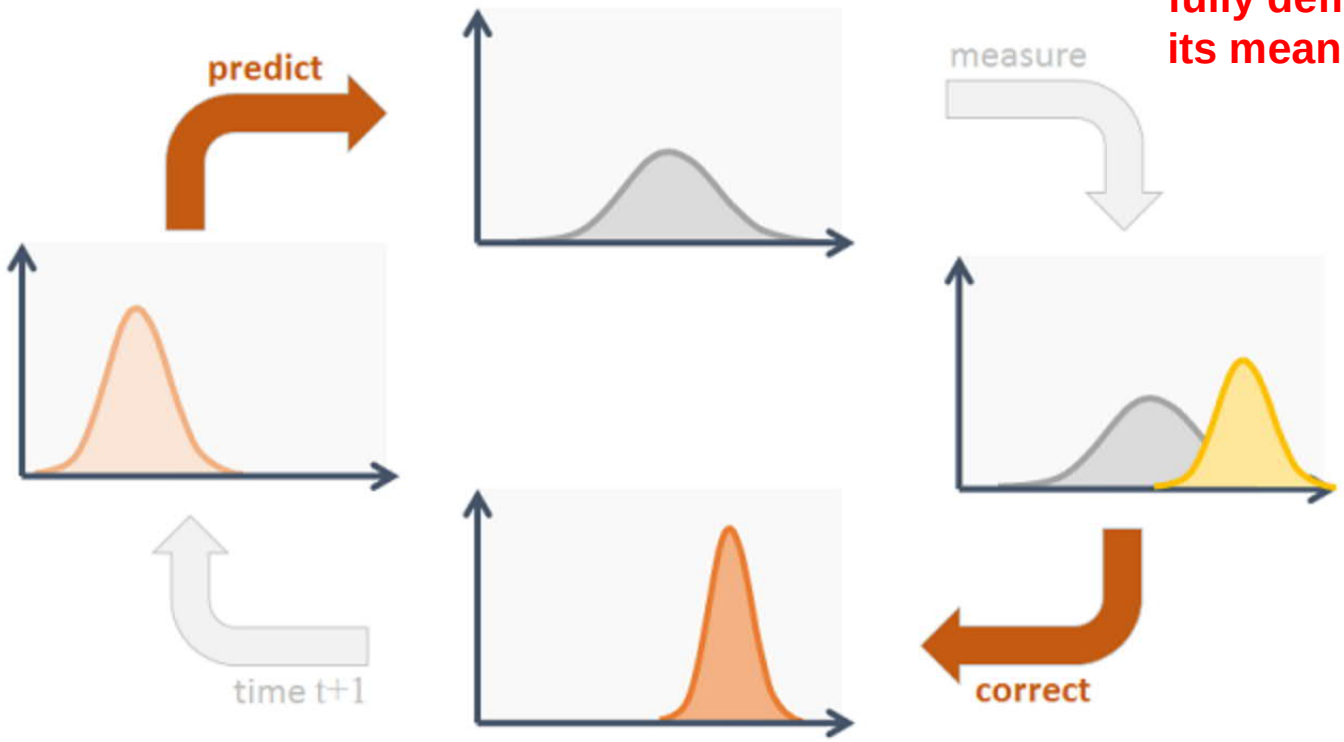
- The prediction step uses our previous estimate and our state model to predict the new state. The correction step uses the difference between predicted measurement $\hat{y}(k|k-1) = H(k)\hat{x}(k|k-1)$ and actual new measurement $y(k)$.

Kalman Filtering Equations

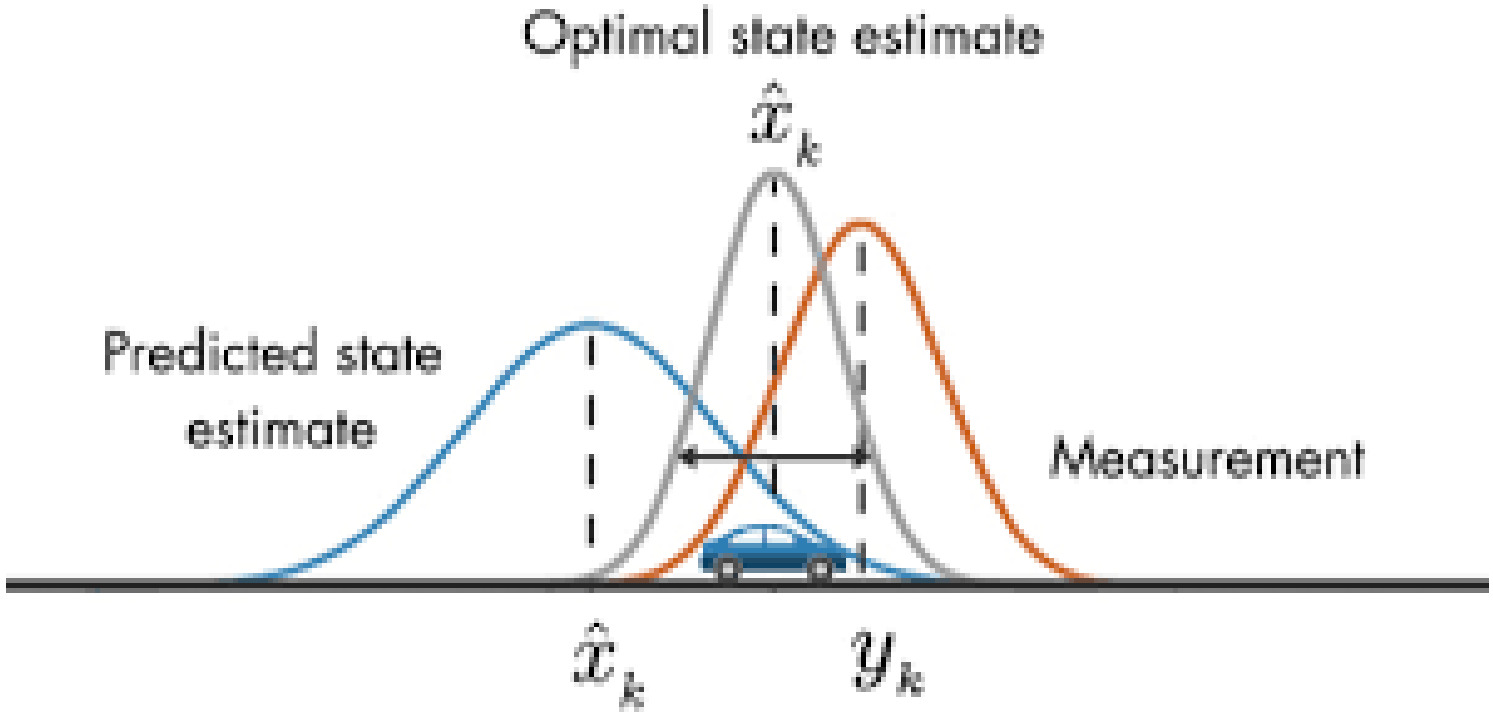


Kalman Filtering Equations

Gaussian pdf
fully defined by
its mean and covariance



Kalman Filtering Equations



Kalman Filtering Equations

- Prediction (a priori) error $e(k|k-1) = x(k) - \hat{x}(k|k-1)$ where $\hat{x}(k|k-1)$ means that estimate of state at time k is produced using data upto time $k-1$. Covariance matrix of prediction error is $P(k|k-1) = E[e(k|k-1)e^T(k|k-1)]$.
- filtering (a posteriori) error is $e(k|k) = x(k) - \hat{x}(k|k)$ Covariance matrix of filtering error is $P(k|k) = E[e(k|k)e^T(k|k)]$
- We can obtain a prediction of the next measurement by using the measurement equation $\hat{y}(k|k-1) = H(k)\hat{x}(k|k-1)$.

Prediction Equations Proof

- Proofs are based on the assumption that state and measurement noises and initial state are Gaussian. This will lead to an optimal estimator. Otherwise we will end up with best linear estimator.
- Proofs for the prediction equations

$$\hat{x}(k|k-1) = E[x(k)|y(k-1),...,y(0)]$$

$$= E[\{F(k|k-1)x(k-1) + G(k|k-1)w(k-1)\}|y(k-1),...,y(0)]$$

$$= F(k|k-1)E[x(k-1)|y(k-1),...,y(0)] + G(k|k-1)E[w(k-1)|y(k-1),...,y(0)]$$

$$= F(k|k-1)\hat{x}(k-1|k-1) + G(k|k-1)E[w(k-1)|y(k-1),...,y(0)]$$

- F and G can be moved in front of the expectation operation based on the linearity of expectation and $\hat{x}(k-1|k-1) = E[x(k-1)|y(k-1),...,y(0)]$ is the other original definition.

Prediction Equations Proof

- $E[w(k-1)|y(k-1),...,y(0)] = E[w(k-1)] = 0$. Let $i = 1,,,,,k-1$ so $y(i)$'s are determined by $x(i)$'s and $v(i)$'s that are independent of $w(k-1)$ so the conditioning can be removed. Moreover $w(n)$ were assumed to be zero mean and Gaussian. Hence, we get the original equation

$$\hat{x}(k|k-1) = F(k|k-1)\hat{x}(k-1|k-1)$$

- The proof for the prediction error covariance matrix

$$P(k|k-1) = Cov(x(k)|y(k-1),...,y(0)) =$$

$$= Cov(\{F(k|k-1)x(k-1)+G(k|k-1)w(k-1)\}|y(k-1),...,y(0)) =$$

$$= Cov(F(k|k-1)x(k-1)|y(k-1),...,y(0))$$

$$+Cov(G(k|k-1)w(k-1)|y(k-1),...,y(0)) =$$

$$= Cov(F(k|k-1)x(k-1)|y(k-1),...,y(0))+Cov(G(k|k-1)w(k-1))$$

Prediction Equations Proof

- The same independence property as earlier is used to remove the conditioning for w. We use the property of covariance matrices: $Cov(UX) = UCov(X)U^T$ to rewrite prediction error covariance

$$P(k|k-1) = F(k|k-1)Cov(x(k-1)|y(k-1),...,y(0))F^T(k|k-1)$$
$$+G(k|k-1)Cov(w(k-1))G^T(k|k-1)$$
$$= F(k|k-1)P(k-1|k-1)F^T(k|k-1)$$
$$+G(k|k-1)Q(k-1)G^T(k|k-1)$$

Correction Equations Proof

- In the measurement equation

$$y(k) = H(k)x(k) + v(k)$$

  y and x are jointly Gaussian and v is Gaussian and independent from x and y.

- The proofs for the correction equations are based on induction. First let us show that the equations hold for $k = 0$. The same assumptions for noises are used again: x is distributed as $N(\mu(0), \Sigma(0))$, and v is distributed as $N(0, R(0))$. Find conditional expectation $E[x(0)|y(0)]$:

$$\hat{x}(0|0) = E[x(0)|y(0)]$$

$$= \mu(0) + P(0)H^T(0)(H(0)P(0)H^T(0) + R(0))^{-1}(y(0) - H(0)\mu(0))$$

$$= \hat{x}(0|-1) + K(0)(y(0) - H(0)\hat{x}(0|-1))$$

  where $K(0) = P(0)H^T(0)(H(0)P(0)H^T(0) + R(0))^{-1}$

Correction Equations Proof

- The filtering error covariance is

$$P(0|0) = P(0) - P(0)H^T(0)(H(0)P(0)H^T(0) + R(0))^{-1}H(0)P(0)$$

$$= P(0|0-1) - K(0)H(0)P(0|0-1)$$

- Assume that the correction equations are valid for $k = m-1$ where $m \geq 1$. $x(m)$ and $y(0), ..., y(m-1)$ may be obtained by linear transformations of Gaussians $x(0)$ and $w(0), ..., w(m-1)$ and $v(0), ..., v(m-1)$. As a result, $x(m)$ and $y(0), ..., y(m-1)$ are jointly Gaussian.
- $x(m)$ given $y(m-1), ..., y(0)$ is Gaussian $(\hat{x}(m|m-1), P(m|m-1))$ and $v(m)$ is independent of $y(m-1), ..., y(0)$. $v(m)$ and $x(m)$ are conditionally independent given $y(m-1), ..., y(0)$

Correction Equations Proof

- Given $y(m-1), ..., y(0)$ equation $y(m) = H(m)x(m) + v(m)$ is of same form as the measurement equation for $k = 0$. Using this we can easily show that update in correction equation hold for all $k = 0, 1, 2, ...$
- Find conditional expectation of $x(m)$ given $y(m)$, we will get $\hat{x}(m|m)$, given $y(0), .., y(m-1)$ which is the conditional expectation of $x(m)$ given $y(0), .., y(m)$

$$\hat{x}(m|m) = E[x(m)|y(m)] = \hat{x}(m|m-1)$$

$$+P(m|m-1)H^T(m)(H(m)P(m|m-1)H^T(m) + R(m))^{-1}$$

$$\times[y(m) - H(m)\hat{x}(m|m-1)]$$

$$= \hat{x}(m|m-1) + K(m)[y(m) - H(m)\hat{x}(m|m-1)]$$

- Using similar arguments, for $P(m|m)$ we obtained:

$$P(m|m) = P(m|m-1) - K(m)H(m)P(m|m-1)$$

Innovations

- The measurement residual or prediction error is called Innovation

$$\tilde{y}(k|k-1) = y(k) - \hat{y}(k|k-1) = y(k) - H(k)\hat{x}(k|k-1)$$

(innovations sequence/process)

- The innovations are a zero mean Gaussian white noise process with covariance

$$E[\tilde{y}(k|k-1)\tilde{y}^T(k|k-1)].$$

Gaussianity is obtained from the fact that linear transformations of Gaussians $y(k), x(k|k-1)$ are Gaussians.

- The mean of the innovation sequence is

$$E[\tilde{y}(k)] = E[y(k) - E[y(k)|y(0), ..., y(k-1)]] = E[y(k)] - E[y(k)] = 0$$

Innovations

- The covariance of the innovation sequence is:

$$E[\tilde{y}(k)\tilde{y}^T(l)] = P_{\tilde{y}\tilde{y}}\delta_{kl}$$

  i.e., they are mutually uncorrelated and they are also independent because they are jointly Gaussian.

- As a result Kalman filter producs white innovations sequence which is equivalent to the original observation sequence in a sense that it contains the same statistical information.

- Innovations can be written $y(k) = \hat{y}(k|k-1) + \tilde{y}(k|k-1)$ where the first term on RHS is the part of $y(k)$ that can be predicted from the past and the second term (innovation) is the part that cannot be predicted. Therefore, innovations are the new information brought into system.

EX: wireless comms channel tracking using KF

- Communication channels are often characterized being linear but not necessary time invariant (e.g., fading multipath channel).
- The medium can be treated as a linear filter which alters an impulse at the input to a appear as a continuous waveform at the output.
- An appropriate channel model is a tapped delay line model. The input-output relationship may be expressed by

$$z(k) = \sum_{n=0}^{p-1} h_k(n) u(k-n)$$

$u(k) = 0$ for $k < 0$. Here $u(k)$ could be known training symbols (e.g. mid-amble in GSM) or decisions fed back by the receiver.

EX: wireless comms channel tracking using KF

- We have an FIR filter with time varying tap coefficients and we want to estimate $h_k(n)$ based on noisy measurements.

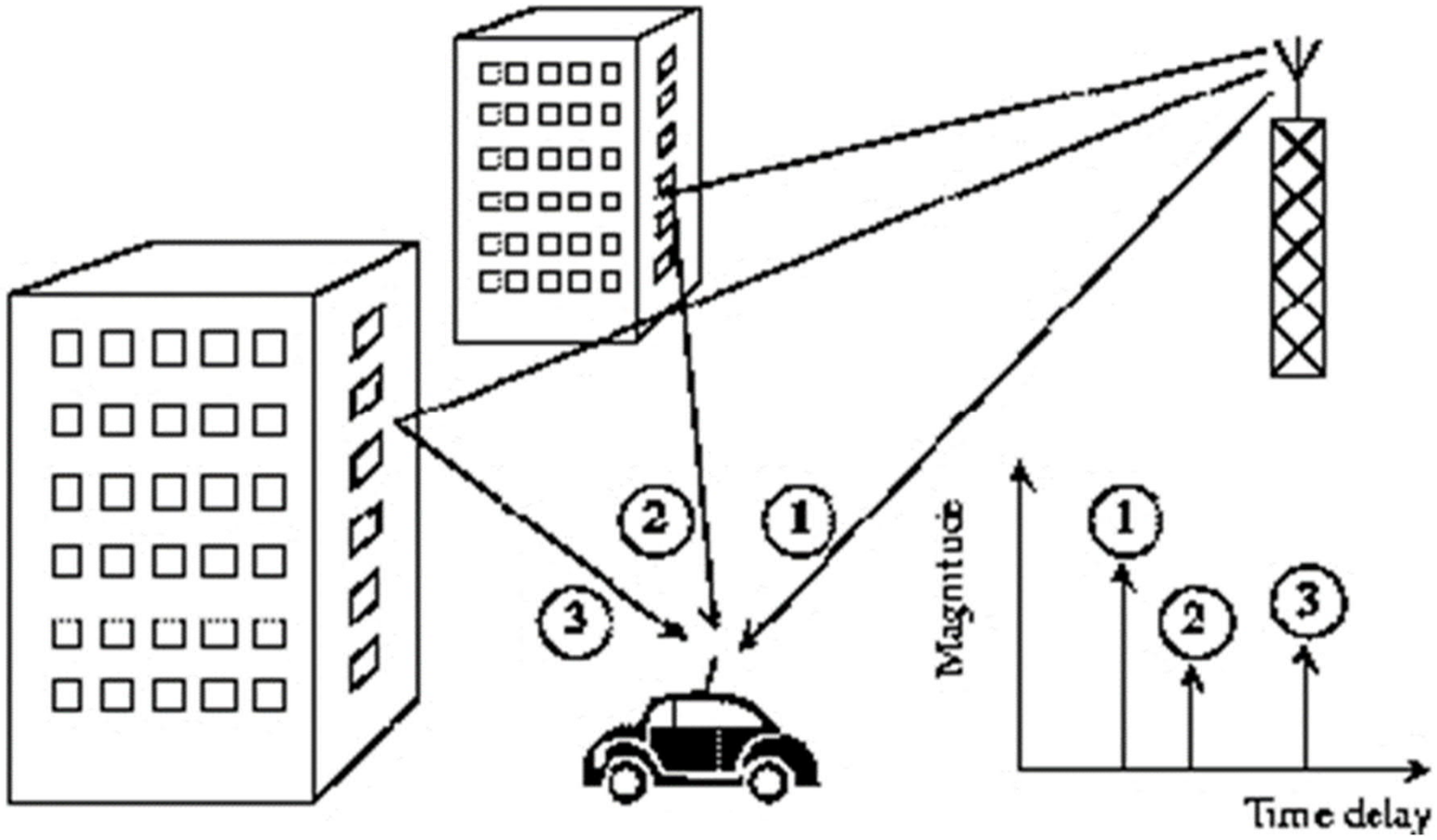
$$y(k) = \sum_{n=0}^{p-1} h_k(n)u(k-n) + v(k),$$

where $v(k)$ is observation noise.

- Suppose we have 2 taps, i.e., $p = 2$. The observations are
time instant $k = 0$:
$y(0) = h_0(0)u(0) + h_0(1)u(-1) + v(0) = h_0(0)u(0) + v(0)$
time $k = 1$: $y(1) = h_1(0)u(1) + h_1(1)u(0) + v(1)$
and time $k = 2$: $y(2) = h_2(0)u(2) + h_2(1)u(1) + v(2)$

EX: wireless comms channel tracking using KF

- so for each discrete time index k we have two new unknowns $h_k(0), h_k(1)$ which makes the determination of the channel tap weights a nasty job.
- The tap coefficients do not change rapidly in subsequent time steps (coherence time).
- We will exploit the correlation between subsequent values of the same tap weight.
- The tap weights are RV's that can be described employing Gauss-Markov model and the problem formulated using state-variable model.
- The tap weights are assumed to be mutually uncorrelated and consequently independent due to Gaussianity.


2
1
3
Magnitude
1
2
3
Time delay

EX: wireless comms channel tracking using KF

- Consequently, the state vector (channel) evolves as:

$$\mathbf{h}(k) = F\mathbf{h}(k-1) + \mathbf{w}(k),$$

where F is a known $p \times p$ matrix, $\mathbf{h}(n) = [h_k(0), ..., h_k(p-1)]$ and $\mathbf{w}(k)$ is vector white Gaussian noise with covariance Q $(p \times p)$.

- The measurement equation is given by

$$y(k) = \mathbf{u}^T(k)\mathbf{h}(k) + v(k),$$

where $\mathbf{u}^T(k) = [u(k)\ u(k-1)\ u(k-p+1)]$ and $v(k)$ is WGN with variance σ^2.

- Now we will estimate the tap coefficients using the Kalman filter. Our measurements are scalars and our state is a vector.

EX: wireless comms channel tracking using KF

- The prediction equations are

$$\hat{\mathbf{h}}(k|k-1) = F(k|k-1)\hat{\mathbf{h}}(k-1|k-1)$$

and

$$P(k|k-1) = F(k|k-1)P(k-1|k-1)F^T(k|k-1) + Q(k)$$

where $Q(k) = E[\mathbf{w}(k)\mathbf{w}^T(k)]$ (known control input is zero!).

- The correction (update) equations are

$$\hat{\mathbf{h}}(k|k) = \hat{\mathbf{h}}(k|k-1) + K(k)[y(k) - \mathbf{u}^T(k)\hat{\mathbf{h}}(k|k-1)]$$

and

$$P(k|k) = P(k|k-1) - K(k)\mathbf{u}^T(k)P(k|k-1)$$

EX: wireless comms channel tracking using KF

- and the Kalman gain

$$K(k) = P(k|k-1)\mathbf{u}(k)[\mathbf{u}^T(k)P(k|k-1)\mathbf{u}(k) + \sigma^2]^{-1}$$

- Now this system is initialized by $\hat{\mathbf{h}}(-1|-1) = \mu_h$ and $P(-1|-1) = P_h$.

- In our model we had $p = 2$, i.e., a 2 tap FIR filter and the matrices we know in our state equation are

$$F = \begin{bmatrix} 0.99 & 0 \\ 0 & 0.999 \end{bmatrix} \qquad Q = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix}$$

- We will use starting values $\hat{\mathbf{h}}(-1|-1) = 0$ and $P(-1|-1) = 100I$. Large initial P is chosen in order to avoid trusting too much on the initial state estimate which may have arbitrary value.

Kalman Predictor

- One-step Kalman Predictor equation:

$$\hat{x}(k+1|k) = F(k+1|k)\hat{x}(k|k-1) + K(k)[y(k) - H(k)\hat{x}(k|k-1)]$$

Predictor gain $K(k)$:

$$K(k) = F(k+1|k)P(k|k-1)H^T(k) \times [H(k)P(k|k-1)H^T(k) + R(k)]^{-1}$$

and Prediction error covariance:

$$P(k+1|k) = [F(k+1|k)K(k)H(k)]P(k|k-1)F^T(k+1|k) + Q(k)$$

Remarks on Kalman Filter Performance

- Filter performance can be analyzed before actually making measurements.
- The gain matrix $K(k)$ can be considered to minimize
  Trace(P(k|k)), i.e. estimation variance

- Intuitive interpretation for the gain is that the gain is large if the state prediction is inaccurate and the measurement is accurate ($R(k)$ in the matrix to be inverted is close to zero then, measurement is trusted and system model bad) and the gain is small if the state prediction is accurate and the the measurement is inaccurate (good system model and bad measurement, $P(k|k-1)$ close to zero, $R(k)$ is larger).
- Initial estimates: based on measurements: measurement error covariance and mean are pretty easy whereas state error covariance is more difficult (contains the modeling uncertainty). Sometimes one can use a poor model by using large state noise covariance matrix.

Remarks on Kalman Filter Performance

- If the noise covariance matrices are constant the equations for covariances are independent of measurements, hence they can be iterated before any measurements are obtained or processed.
- Riccatti difference equations written as a pair of coupled equations: The prediction error covariance matrix is

$$P(k+1|k)$$

$$= F(k+1|k)P(k|k)F^T(k+1|k) + G(k+1|k)Q(k)G^T(k+1|k)$$

where the estimation error covariance matrix $P(k|k)$ is given by

$$P(k|k)$$

$$= P(k|k-1) - K(k)H(k)P(k|k-1) = [I - K(k)H(k)]P(k|k-1)$$

Remarks on Kalman Filter Performance

- The matrix Riccatti equation for $P(k+1|k)$ allows for computing directly the updated value $P(k+1|k)$ given the old value $P(k|k-1)$:

$$P(k+1|k) = [F(k+1|k) - K(k)H(k)]P(k|k-1)$$

$$\times[F(k+1|k) - K(k)H(k)]^T + Q(k) + K(k)R(k)K^T(k)$$

$$= F(k+1|k)P(k|k-1)[I - H(k)^T[H(k)P(k|k-1)H(k)^T + R(k)]^{-1}$$

$$\times H(k)P(k|k-1)]F(k+1|k) + G(k+1|k)Q(k)G^T(k+1|k)$$

- Matrix inversion in the gain computation is in general not a problem since the dimension of the matrix to be inverted depends on the dimension of the measurement vector $y(k)$. If it is a probem, applying matrix inversion lemma should be considered (Covariance vs. Information form of KF).

Remarks on Kalman Filter Performance

- Two forms: Information and covariance form. The covariance form propagates the covariance matrix whereas the information form propagates the inverse of the covariance matrix which is related to Fisher Information Matrix. Information form is hence suitable for studying the performance using information theoretical tools.
- It is a sufficient and necessary condition for Kalman filter to be optimal that the innovations are zero mean and white sequence.
- If the assumption on zero mean Gaussian noises hold, Kalman filter is the Minimum Mean Square Error estimator. Otherwise it is the best linear estimator (Linear MMSE) (when only 2 moments are known).
- Relation to RLS, (Sayed and Kailath 1994, IEEE Signal Processing Magazine)

Square-Root Kalman Filter

- In expression for $P(k|k)$ one subtracts two non-negative definite matrices and the result may not be non-negative definite covariance matrix. Hence $P(k|k)$ may be presented as a product of its square root matrices. Square root matrices are updated and the product of two square root matrices (product of a square matrix and its hermite transpose) is always non-negative definite. As a result a numerically more stable algorithm is obtained.
- Square roots may be computed using Cholesky decomposition $P = LL^T$ where $L = P^{1/2}$ or UD factorization.
- In addition, Givens rotations are needed.

Extended Kalman Filter (EKF)

- If the process is nonlinear or the obervations are nonlinearly related to the state, the Kalman filter may not be used anymore.
- In state equation $x(k) = F(k|k-1)x(k-1)$ is replaced by nonlinear model $x(k) = F(k, x(k-1))$ and in measurement equation $y(k) = H(k)x(k)$ is replaced by $y(k) = H(k, x(k))$
- The extended Kalman filter algorithm is used in such cases. It linearizes the model (about the new estimate) and applies the linear Kalman filter then.
- As a result it is not optimal filtering anymore and the performance of the filter may not be determined in advantage. No convergence can be established either, hence EKF have to be considered an *ad hoc* estimation technique.
- The quality of the linearization (about the current estimate) is an important factor in obtaining a good performance.

Extended Kalman Filter (EKF)

- The linearization is similar to Taylor series approach. Partial derivatives of the state and measurement equations are used
- The equations appear as follows:

$$x(k) = \mathcal{F}(k|k-1)x(k-1)$$

$$+[F(k, \hat{x}(k-1|k-1)) - \mathcal{F}(k|k-1)\hat{x}(k-1|k-1)]$$

$$y(k) = \mathcal{H}(k)x(k) + [H(k, \hat{x}(k|k-1)) - \mathcal{H}(k)\hat{x}(k|k-1)]$$

where $\mathcal{F}$ and $\mathcal{H}$ are Jacobian matrices

$$\mathcal{F} = \frac{\partial F}{\partial x} \text{ and respectively } \mathcal{H} = \frac{\partial \mathrm{H}}{\partial \mathrm{x}}$$

where derivatives are evaluated at $\hat{x}(k-1|k-1)$ and $\hat{x}(k|k-1)$ respectively.

Extended Kalman Filter (EKF)

- Now the Extended Kalman Filtering equations in prediction stage appear as

$$\hat{x}(k|k-1) = F(k, \hat{x}(k-1|k-1)) = \mathcal{F}(k|k-1)\hat{x}(k-1|k-1)$$

$$+[F(k, \hat{x}(k-1|k-1)) - \mathcal{F}(k|k-1)\hat{x}(k-1|k-1)]$$

and the prediction error covariance matrix

$$P(k|k-1) = \mathcal{F}(k|k-1)P(k-1|k-1)\mathcal{F}^T(k|k-1)$$

$$+G(k|k-1)Q(k)G^T(k|k-1)$$

The equations in the correction stage are as follows: the updated state estimate is

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \mathcal{K}(k)[y(k) - H(k, \hat{x}(k|k-1))]$$

Extended Kalman Filter (EKF)

- The estimation error covariance matrix

$$P(k|k) = P(k|k-1) - \mathcal{K}(k)\mathcal{H}(k)P(k|k-1)$$

where the Kalman gain $\mathcal{K}(k)$ is

$$\mathcal{K}(k) = P(k|k-1)\mathcal{H}^T(k) \times [\mathcal{H}(k)P(k|k-1)\mathcal{H}^T(k) + R(k)]^{-1}.$$

- This is a linear Kalman filter for the model and it operates similarly to the Kalman filter. Th linear terms in the Kalman filter are replaced by approximate terms in the EKF.

Extended Kalman Filter (EKF)

- Application examples include, delay tracking in CDMA, frequency offset tracking in OFDM systems, maneuvering target tracking in radar.
- Let the state equation be

$$\begin{bmatrix} x_a(k+1) \\ x_b(k+1) \end{bmatrix} = \begin{bmatrix} x_a(k) + 0.9x_b^2(k) \\ kx_b(k) - 0.8x_a(k)x_b(k) \end{bmatrix} + \begin{bmatrix} w_a(k) \\ w_b(k) \end{bmatrix}$$

and measurement equation

$$y(k) = x_a(k)x_b^2(k) + v(k)$$

i.e., the model is nonlinear and EKF have to be used.

Extended Kalman Filter (EKF)

- Therefore we have

$$F(k, x(k)) = \begin{bmatrix} x_a(k) + 0.9x_b^2(k) \\ kx_b(k) - 0.8x_a(k)x_b(k) \end{bmatrix}$$

and

$$H(k, x(k)) = x_a(k)x_b^2(k)$$

- Construct the Jacobian matrices

$$\mathcal{F} = \frac{\partial F}{\partial x} = \frac{\partial F(k, x)}{\partial x}$$

where evaluation is performed at $x = \hat{x}(k|k)$

Extended Kalman Filter (EKF)

- For $H()$

$$\mathcal{H} = \frac{\partial H}{\partial x}$$

where evaluation takes place at $x = \hat{x}(k|k-1)$.

- Consequently

$$\mathcal{F} = \frac{\partial F}{\partial x} = \begin{bmatrix} 1 & 1.8x_b \\ -0.8x_b & k - 0.8x_a \end{bmatrix}$$

which is then evaluated yielding

$$\mathcal{F}(k+1|k) = \begin{bmatrix} 1 & 1.8\hat{x}_b(k|k) \\ -0.8\hat{x}_b(k|k) & k - 0.8\hat{x}_a(k|k) \end{bmatrix}$$

Extended Kalman Filter (EKF)

- Similarly for the measurement equation

$$\mathcal{H} = \frac{\partial H}{\partial x} = \begin{bmatrix} x_b^2 & 2x_a x_b \end{bmatrix}$$

- It gives

$$\mathcal{H}(k) = \begin{bmatrix} \hat{x}_b^2(k|k-1) & 2\hat{x}_a(k|k-1)\hat{x}_b(k|k-1) \end{bmatrix}$$

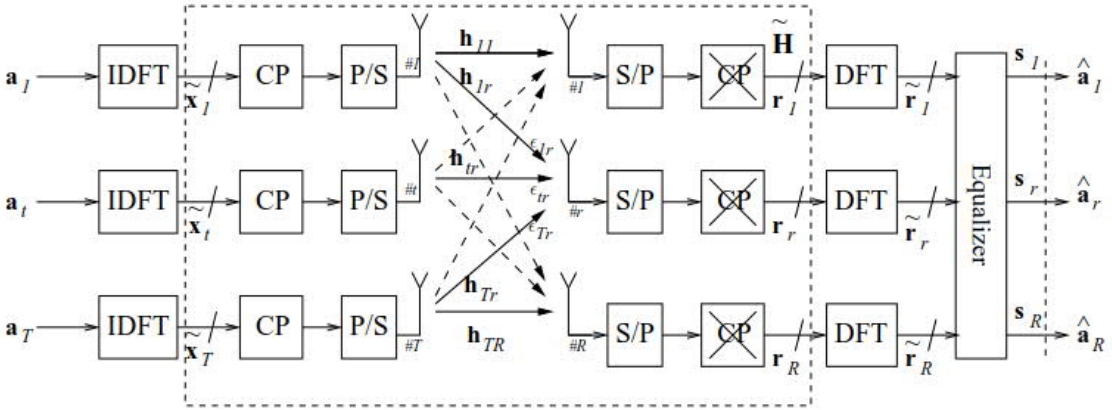
- Now the first order Taylor series approximations about the current estimates appear as

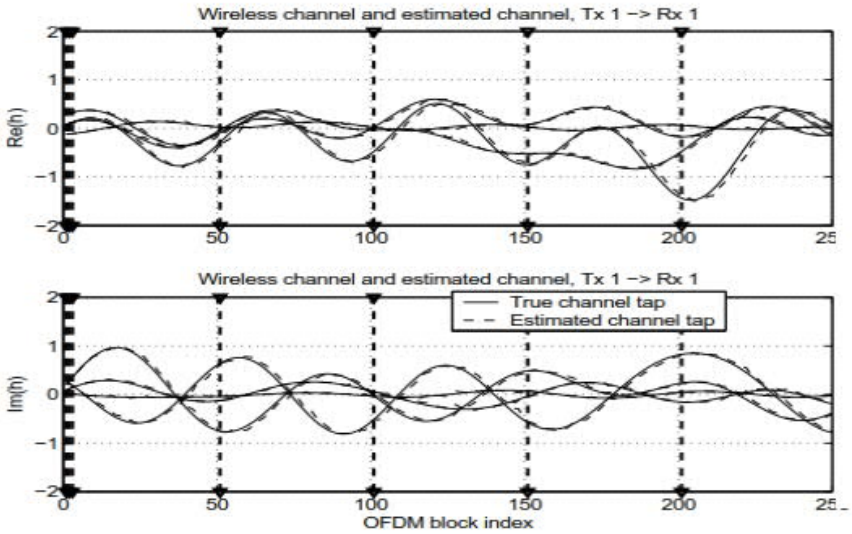
$$F(k, x(k)) \approx F(k, \hat{x}(k|k)) + \mathcal{F}(k+1|k)(x(k) - \hat{x}(k|k))$$

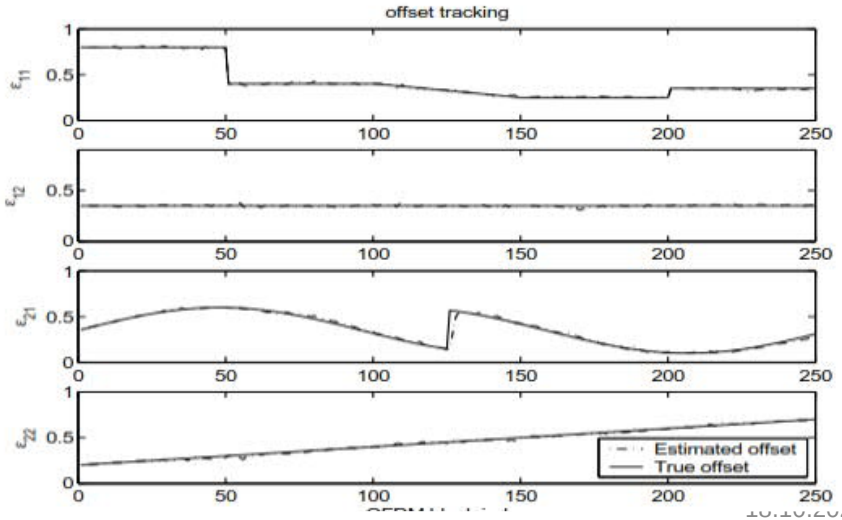
and

$$H(k, x(k)) \approx H(k, \hat{x}(k|k-1)) + \mathcal{H}(k)(x(k) - \hat{x}(k|k-1))$$

EX: MIMO channel and carrier offset tracking with EKF







Divergence of Kalman Filter

- The actual estimation error variance significantly exceeds the values theoretically predicted by $\mathbf{v}$. The error may become unbounded even if the error variance in Kalman algorithm is vanishingly small.
- Main reasons for divergence:
  - inaccuracies in the modeling process used in determining state and measurement models;
  - failure of linearization; lack of knowledge of physical problem;
  - too simplifying assumptions for mathematical tractability;
  - errors in modeling noise variances and mean;
  - round-off errors $\Rightarrow$ error covariance matrix may loose its positive definitiveness or symmetry.

Divergence of Kalman Filter

- Raise the order of the problem, i.e., take into account unaccounted input. This is OK, if computational complexity is not a problem.
- Square root filtering when round-off errors cause divergence

Recent advances in sequential estimation

- Mainly for nonlinear systems, non-Gaussian noises and cases where the a posteriori distribution is not necessarily a symmetric unimodal distribution
- Particle filters also known as Sequential Monte Carlo methods (SMC), are sophisticated model estimation techniques based on simulation.
- Unscented Kalman Filter (UKF): The Kalman Filter propagates a Gaussian rv. through the system dynamics. In the EKF, the state distribution is approximated by a Gaussian rv, which is then propagated analytically through the first-order (Taylor series) linearization of the nonlinear system. This can introduce large errors in the true posterior mean and covariance of the transformed rv, which may lead to degraded performance and sometimes divergence.

Unscented Kalman Filter (UKF) idea

- UKF is based on the idea that it is easier to approximate a probability distribution that it is to approximate an arbitrary nonlinear function or transformation
- The unscented transformation is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation
- This problem is solved by using a deterministic sampling approach.
- The state distribution is represented using a minimal set of carefully chosen sample points, called ***sigma*** points.
- Each sigma point is then propagated through the nonlinearity yielding in the end a cloud of transformed points.
- The new estimates of mean and covariance are computed for the transformed points using their weighted mean and covariance

Unscented Kalman Filter (UKF)…

- The UKF addresses this problem by using a deterministic sampling approach. The state distribution is again approximated by a Gaussian rv, but is now represented using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the rv, and when propagated through the true non-linear system, captures the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any nonlinearity. The EKF, in contrast, only achieves first-order accuracy. The computational complexity of the UKF is the same order as that of the EKF.
- Selected $2L$ points X_i are called Sigma points that are propagated through the true nonlinear transform $g()$ to get a-posterior sigma points Y_i
- Their weighted mean and covariance are computed
- Gaussian assumption is crucial!
- No need to calculate Jacobians as in EKF

Unscented Transform

2L+1 sigma points *X* and their weights *W*

i^{th} row of matrix sqrt

$$\begin{aligned}
\mathcal{X}_0 &= \bar{\mathbf{x}} \\
\mathcal{X}_i &= \bar{\mathbf{x}} + \left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)_i \qquad i = 1,\dots,L \\
\mathcal{X}_i &= \bar{\mathbf{x}} - \left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)_{i-L} \qquad i = L+1,\dots,2L \\
W_0^{(m)} &= \lambda/(L+\lambda) \\
W_0^{(c)} &= \lambda/(L+\lambda) + (1-\alpha^2+\beta) \\
W_i^{(m)} &= W_i^{(c)} = 1/\{2(L+\lambda)\} \qquad i = 1,\dots,2L
\end{aligned} \tag{15}$$

$\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter.

$\mathcal{Y}_i = g(\mathcal{X}_i) \quad i = 0,\dots,2L$

α is the spread of sigma points,
β contains prior knowledge of distribution of x
κ is secondary scaling parameter

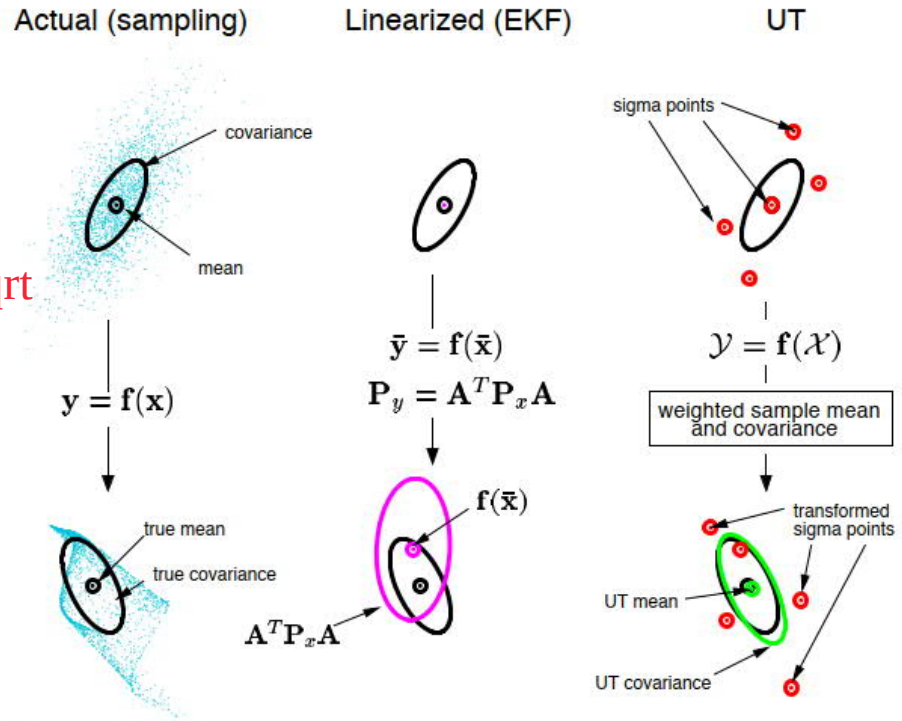


Figure 1: Example of the UT for mean and covariance propagation. a) actual, b) first-order linearization (EKF), c) UT.

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i$$

$$\mathbf{P_y} \approx \sum_{i=0}^{2L} W_i^{(c)} \{\mathcal{Y}_i - \bar{\mathbf{y}}\}\{\mathcal{Y}_i - \bar{\mathbf{y}}\}^T$$

Mean and covariance of posterior sigma points