III.C - Linear Transformations: Optimal Filtering

FIR Wiener Filter

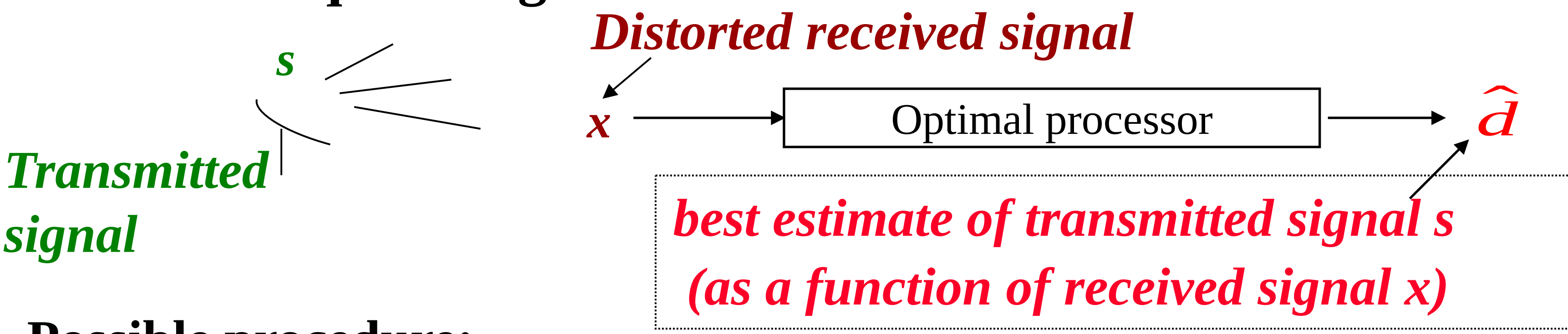
❖ Optimal Filtering Problem:

- How to estimate one signal from another.
- In many applications desired signal is not observable directly (convolved with another signal or distorted by noise).
- Examples:
  - Information signal transmitted over channel gets corrupted with noise.
  - Image recorded by system is subject to distortions.
  - Directional antenna array is vulnerable to string jammers in other directions due to sidelobe leakage, etc.

In this course:

- Emphasis on least square techniques to estimate/recover signal (i.e., case $\|\cdot\|_2$).
- Other types of norms could be used to solve problems

❖ Mean Square Signal Estimation



Possible procedure:

Mean square estimation, i.e.,

$$\text{minimize } \xi = E\left\{\left|s - \hat{d}(\underline{x})\right|^2\right\}$$

$$\text{leads to } \hat{d}(\underline{x}) = E\left[s \mid \underline{x}\right]$$

(proof given in Appendix A)

- **conditional mean**!, usually nonlinear in $\underline{x}$ [exception when $\underline{x}$ and s are jointly normal Gauss Markov theorem]
- Complicated to solve,
- Restriction to Linear Mean Square Estimator (LMS), estimator of s is **forced** to be a linear function of measurements $\underline{x}$: $\rightarrow \hat{d} = \underline{h}^H \underline{x}$
- Solution via Wiener Hopf equations using orthogonality principle

❖ Orthogonality Principle

Use LMS Criterion: estimate s by $\hat{d} = \underline{h}^H \underline{x}$
where weights $\{h_i\}$ minimize MS error:

$$\sigma_e^2 = E\left\{\left|\underline{s} - \hat{d}\left(\underline{x}\right)\right|^2\right\}$$

Theorem: Let error $e = s - \hat{d}$

$\underline{h}$ minimizes the MSE quantity σ_{e}^2 if $\underline{h}$ is chosen such that

$$E\left\{ex_i^*\right\} = E\left\{x_i^* e\right\} = 0, \;\; \forall_i = 1, \cdots, N$$

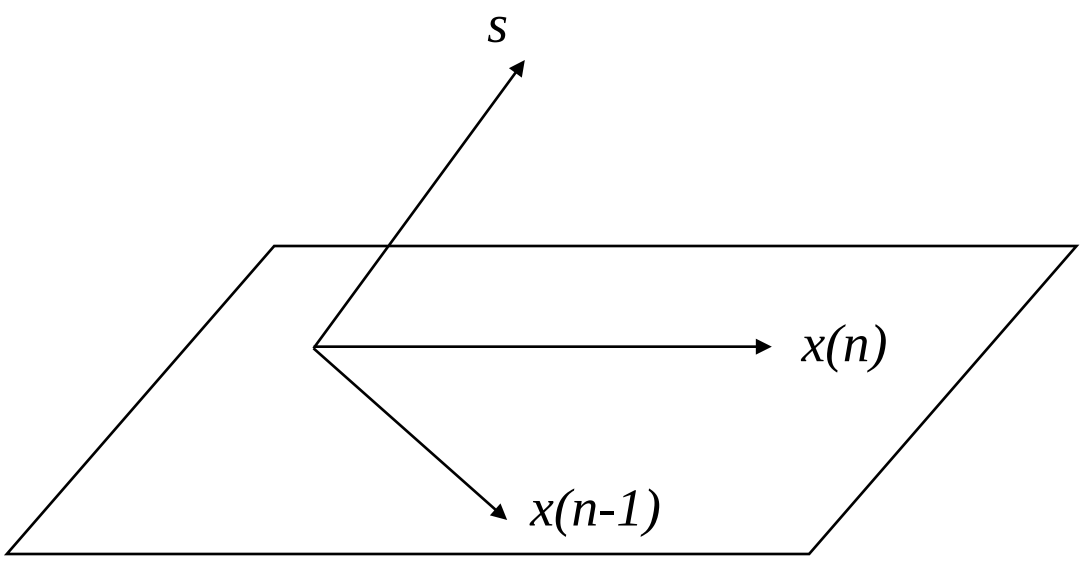
i.e., the error e is orthogonal to the observations $x_i, i = 1..., N$ used to compute the filter output.

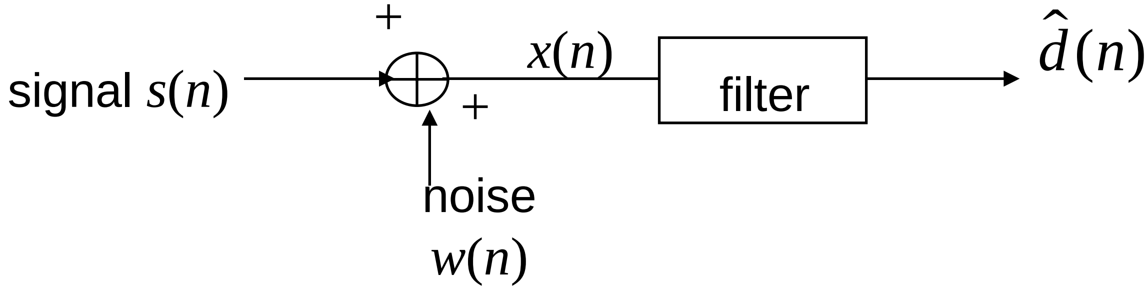
Corollary: minimum MSE obtained: $\sigma_{e_{\min}}^2 = E\left\{se^*\right\}$ where e is the minimum error obtained for the optimum filter vector.

(Proof given in Appendix B)

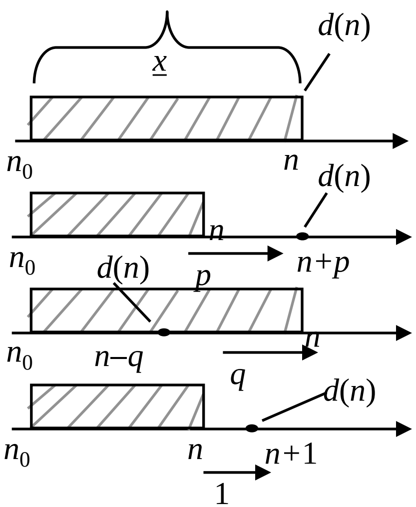
$$P = 2$$

$$\hat{d}(n) = h_0 x(n) + h_1 x(n-1)$$





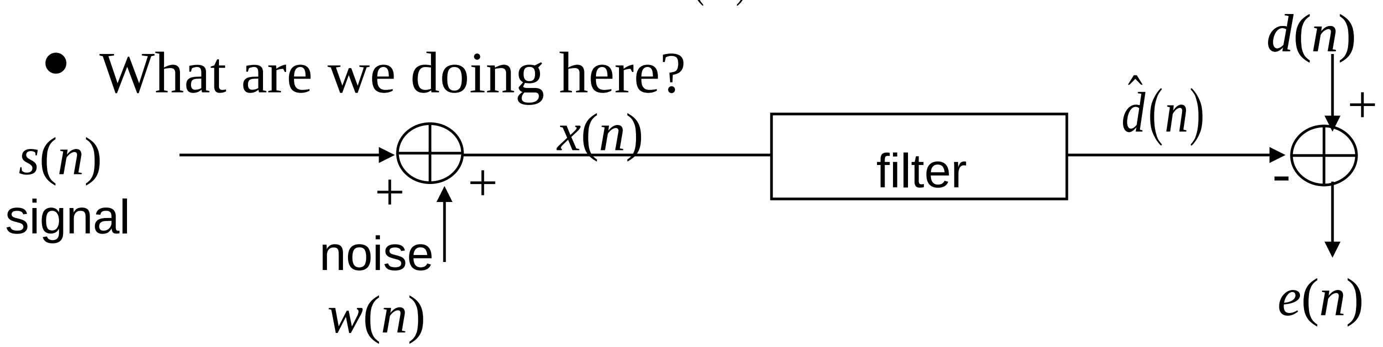
Typical Wiener Filtering Problems



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	$x(n) = s(n-1)$	$d(n) = s(n)$

❖ FIR Wiener Filtering Concepts

- Filter criterion used: minimization of mean square error between $d(n)$ and $\hat{d}(n)$.
- What are we doing here?



We want to design a filter (in the generic sense can be: filter, smoother, predictor) so that:

$$\hat{d}(n) = \sum_{k=0}^{P-1} h^*(k)x(n-k)$$

How $d(n)$ is defined specifies the operation done:

- filtering: $d(n)=s(n)$
- predicting: $d(n)=s(n+p)$
- smoothing: $d(n)=s(n-p)$

❖ How to find h_k?

Minimize the MSE: $E\left\{\left|d(n)-\hat{d}(n)\right|^2\right\}$

$$\sum_{k=0}^{P-1} h_k^* x(n-k) = \underline{h}^H \underline{x}$$

$$\underline{h} = \left[h_0, \qquad h_{P-1}\right]^T, \quad \underline{x} = \left[x(n), \qquad x(n-P+1)\right]^T$$

Wiener filter is a linear filter $\Rightarrow$ orthogonality principle applies

$$\Rightarrow E\left\{x(n-i)e^*(n)\right\} = 0, \quad \forall i = 0,\ldots,P-1$$

$$E\left\{x(n-i)\left[d(n)-\sum_{k=0}^{P-1} h_k^* x(n-k)\right]^*\right\} = 0, \quad \forall i = 0,\ldots,P-1$$

$$\Rightarrow r_{xd}(-i) - \sum_{k=0}^{P-1} h_k R_x(k-i) = 0, \quad \forall i = 0,\ldots,P-1$$

$$r_{dx}^{\ *}(i) = \sum_{k=0}^{P-1} h_k R_x(k-i), \quad \forall i = 0, \ldots, P-1$$

Matrix form:

$$\begin{array}{r} i=0 \;\Rightarrow \\ i=1 \;\Rightarrow \\ \\ \\ \end{array} \begin{bmatrix} r_{dx}^{*}(0) \\ r_{dx}^{*}(1) \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} R_x(0) & R_x(1) & \cdots & R_x(P-1) \\ R_x(-1) & R_x(0) & \cdots & R_x(P-2) \\ \vdots & & & \\ R_x(-P+1) & & & R_x(0) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{P-1} \end{bmatrix}$$

Note: different notation than in [Therrien, section 7.3]!

❖ **Minimum MSE** (MMSE)

obtained when $\underline{h}$ is obtained from solving WH equations.

For best $\underline{h}$ obtained:

$$\sigma_{e_{\min}}^2 = E\left\{\left|e_{\min}\right|^2\right\} = E\left\{(d(n) - \hat{d}(n))e_{\min}^*\right\}$$

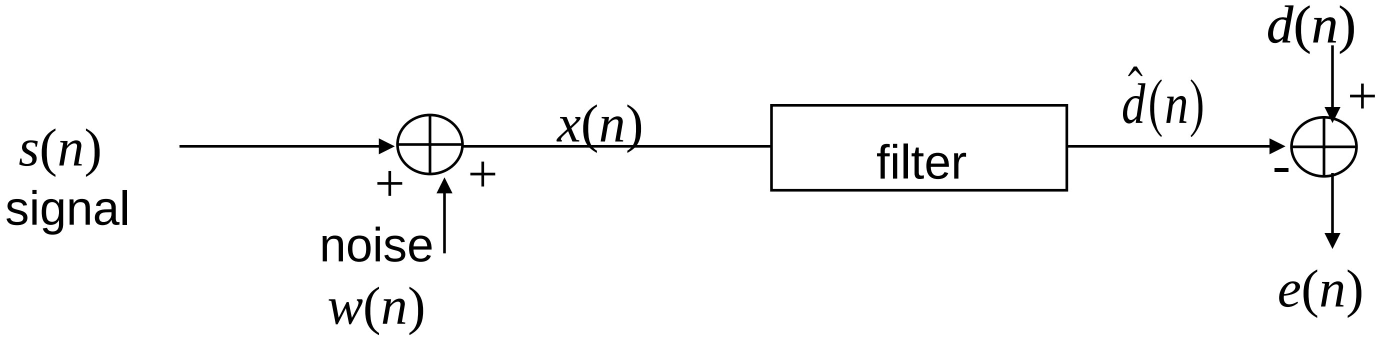
$$= E\left\{d(n)e_{\min}{}^*(n)\right\}$$

$$= E\left\{d(n)\left(d(n) - \sum_{k=0}^{P-1} h_k^* x(n-k)\right)^*\right\}$$

$$= R_d(0) - \sum_{k=0}^{P-1} h_k\, r_{dx}(k)$$

$$\boxed{\sigma_{e_{\min}}^2 = R_d(0) - \underline{h}^T\, \underline{r}_{dx}}$$

Summary: FIR Wiener Filter Equations



- FIR Wiener filter is a FIR filter such that:

$$\hat{d}(n) = \sum_{k=0}^{P-1} h_k^* x(n-k)$$

where $\sigma_e^2 = E\left[\left|d(n) - \hat{d}(n)\right|^2\right]$ is minimum.

- How $d(n)$ is defined **specifies** the specific type of Wiener filter designed:

  filtering:

  smoothing:

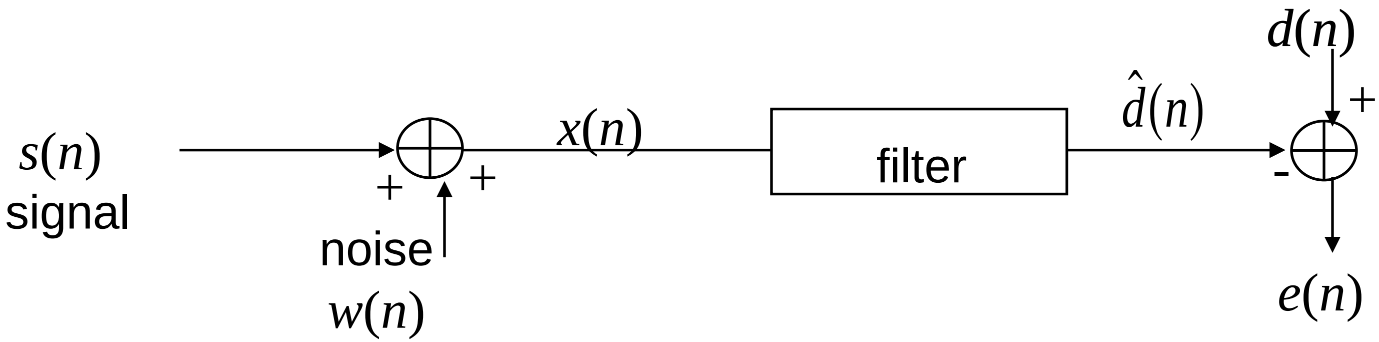
  predicting:

- $\rightarrow$ W-H eqs:
- MMSE:

$$\begin{cases} R_x \underline{h} = \underline{r}_{dx}^* \Rightarrow \underline{h}_{opt} = R_x^{-1} \underline{r}_{dx}^* \\ \sigma_{e_{\min}}^2 = R_d(0) - \underline{h}_{opt}^T \underline{r}_{dx} = R_d(0) - \underline{r}_{dx}^T \underline{h}_{opt} \end{cases}$$

❖ Example 1: Wiener filter (filter case: *d*(*n*) = *s*(*n*) & white noise)

Assume *x(n)* is defined by



$s(n)$, $w(n)$ uncorrelated

$w(n)$ white noise, zero mean $\qquad R_w(n) = 2\delta(n)$

$s(n)$ $\qquad R_s(n) = 2\,(0.8)^{|n|}$

Filter length	Filter coefficients	MMSE
2	[0.405, 0.238]	0.81
3	[0.382, 0.2, 0.118]	0.76
4	[0.377, 0.191, 0.01, 0.06]	0.7537
5	[0.375, 0.188, 0.095, 0.049, 0.029]	0.7509
6	[0.3751, 0.1877, 0.0941, 0.0476, 0.0249, 0.0146]	0.7502
7	[0.3750, 0.1875, 0.0938, 0.0471, 0.0238, 0.0125, 0.0073]	0.7501
8	[0.3750, 0.1875, 0.038, 0.049, 0.0235, 0.0119, 0.0062, 0.0037	0.75

❖ Example 2: Application to Wiener filter (filter case: $d(n) = s(n)$ & colored noise)

$s(n)$, $w(n)$ uncorrelated, and zero-mean

$w(n)$ noise with $R_w(n) = 2\,(0.5)^{|n|}$

$s(n)$ signal with $R_s(n) = 2\,(0.8)^{|n|}$

Filter length	Filter coefficients	MMSE
2	[0.4156, 0.1299]	0.961
3	[0.4122, 0.0750, 0.0878]	0.9432
4	[0.4106, 0.0737, 0.0508 0.0595]	0.9351
5	[0.4099, 0.0730, 0.0499, 0.0344, 0.0403]	0.9314
6	[0.4095, 0.0728, 0.0495, 0.0338, 0.0233, 0.0273]	0.9297
7	[0.4094, 0.0726, 0.0493, 0.0335, 0.0229, 0.0158, 0.0185]	0.9289
8	[0.4093, 0.0726, 0.0492, 0.0334, 0.0227, 0.0155, 0.0107, 0.0125]	0.9285

❖ Example 3: Application of Wiener filter to one-step linear prediction

- tracking of moving series
- forecasting of system behavior
- data compression
- telephone transmission

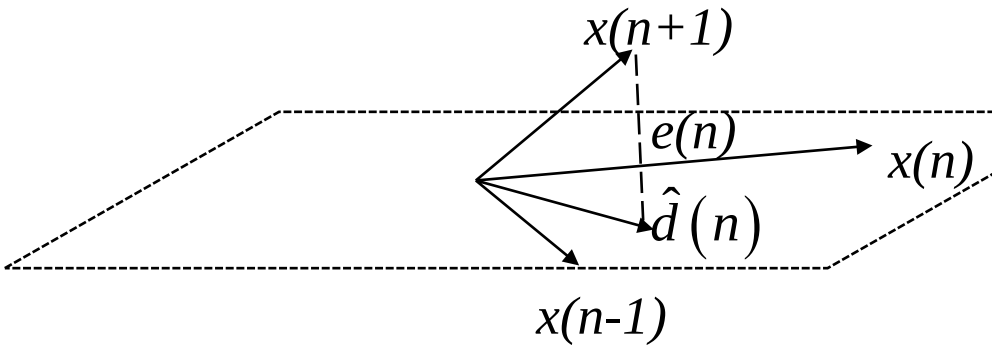
- <u>W-H equations</u>

$$\underline{h}_{\text{opt}} = R_x^{-1}\underline{r}_{dx}^*$$

where $$\hat{d}(n) = \sum_{\ell=0}^{P-1} h_l^* x(n-\ell)$$

$$d(n) = \quad ?$$

- <u>Geometric interpretation</u>: Assume a filter of length 2



$\hat{d}(n) =$

$e(n) =$

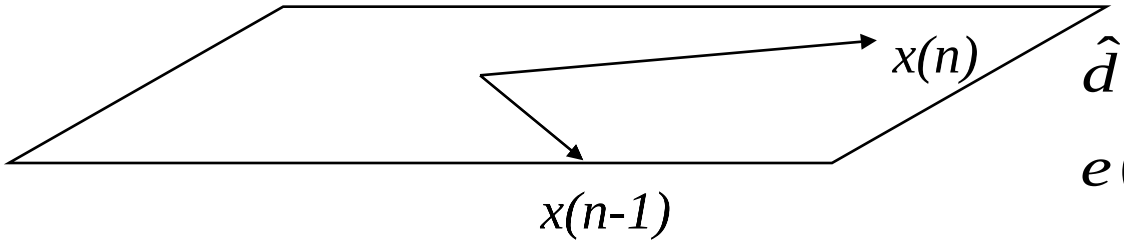
e(n) is the error between true value *x(n+1)* and predicted value for *x(n+1)* based on *x(n)* and *x(n-1)*

→ represents the new information in *x(n+1)* which is not contained in *x(n)* and *x(n-1)*

→ *e(n)* is called the **innovation process** corresponding to *x*(*n*)

- Geometric interpretation: Assume $x(n+1)$ only has **NO** new information (i.e., information in $x(n+1)$ is that already contained in $x(n)$ and $x(n-1)$. Filter of length 2.

Plot $x(n+1), \hat{d}(n), e(n)$

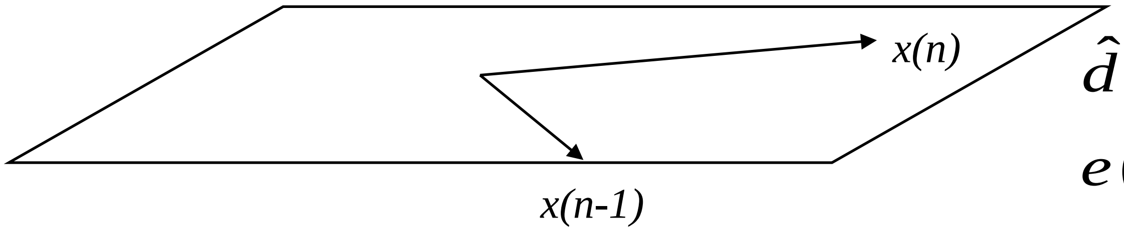


$\hat{d}(n) =$

$e(n) =$

- <u>Geometric interpretation</u>: Assume *x(n+1)* only has new information (i.e., information in *x(n+1)* is **NOT** contained at all in *x(n)* and *x(n-1)*). Filter of length 2.

Plot $x(n+1), \hat{d}(n), e(n)$



$\hat{d}(n) =$

$e(n) =$

❖ 1-step ahead predictor scenario

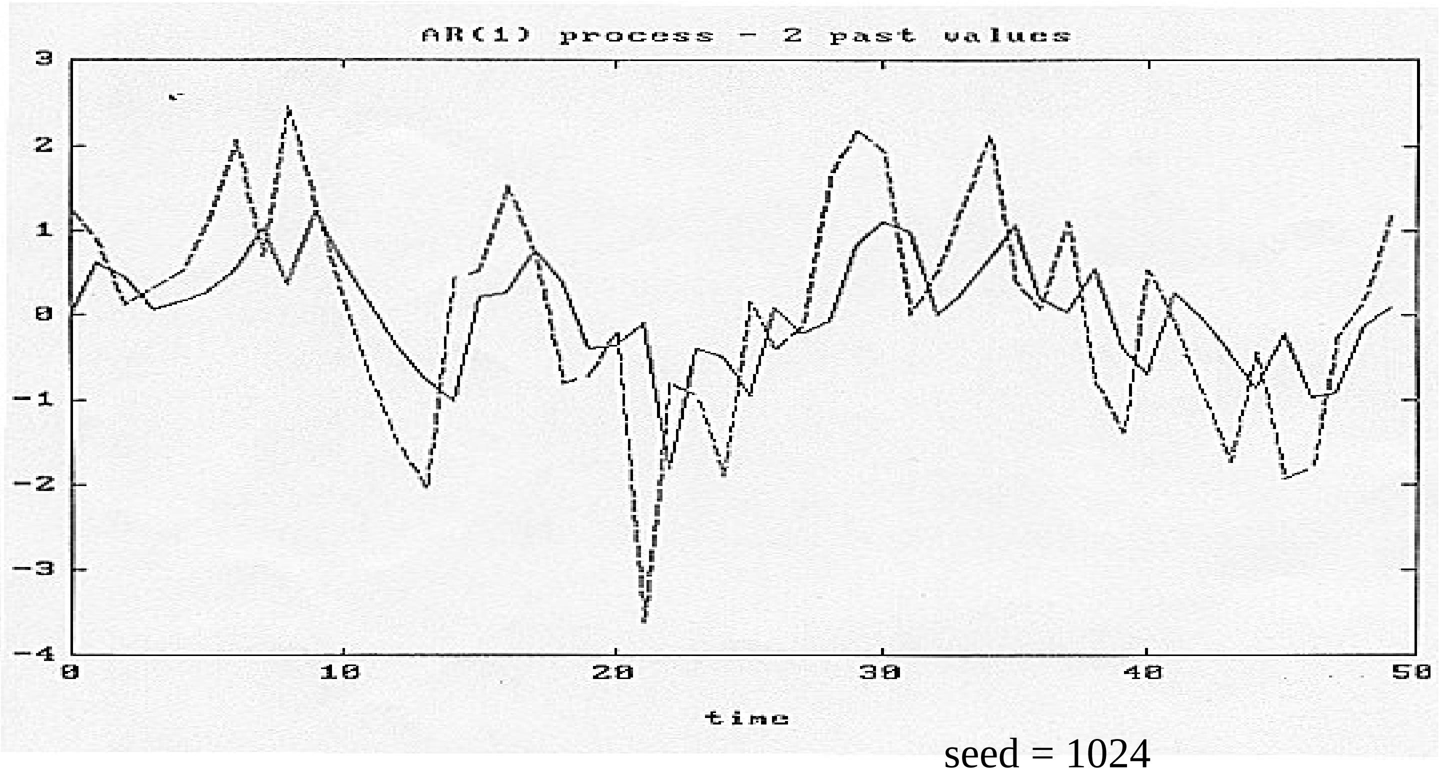
RP $x(n)$ defined as $x(n) = x(n-1) + v(n)$ $|a| < 1$

$v(n)$ is white noise. 1-step predictor of length 2.

------- AR (1) process

——— Predictor

$a = 0.5$



seed = 1024

$$\hat{x}(n) = +a_1 x(n-1) + a_2 x(n-2)$$

$$\boldsymbol{a} = \begin{bmatrix} +0.5 \\ 0 \end{bmatrix}$$

❖ Link between Predictor behavior & input signal behavior

1) Case 1*:* $s(n)$ = process with correlation

$$R_s(k) = \delta(k) + 0.5\delta(k-1) + 0.5\delta(k+1)$$

Investigate performances of N-step predictor as a function of changes N

2) Case 2*:* $s(n)$ = process with correlation

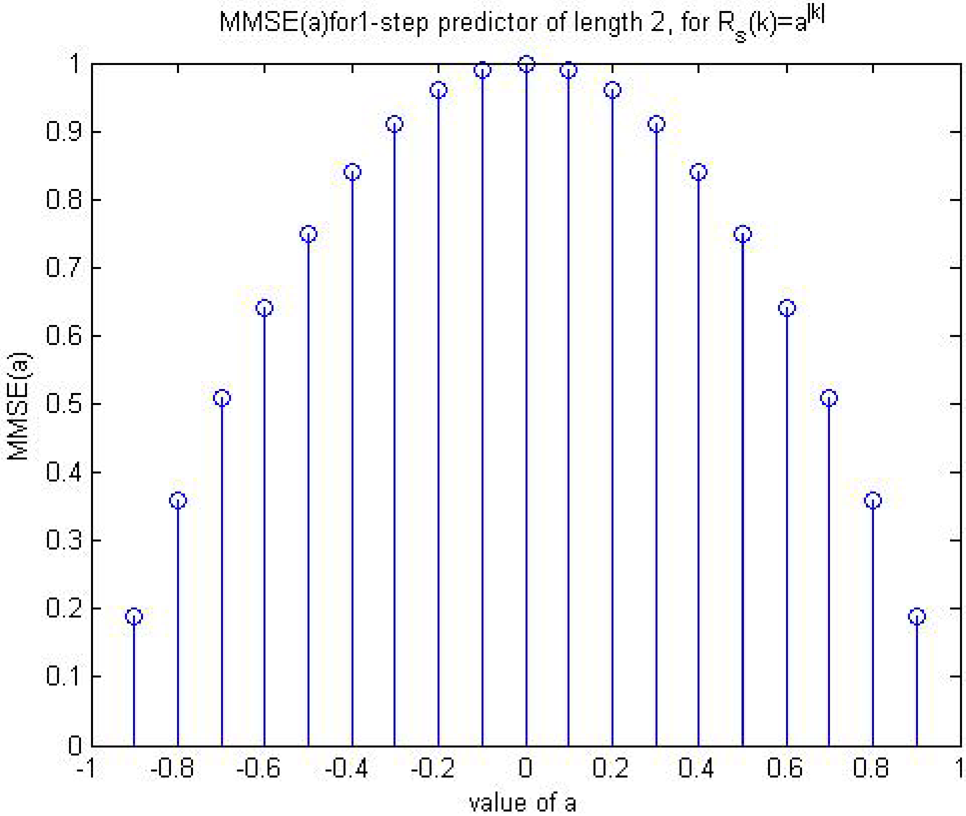
$$R_s(k) = a^{|k|}, |a| < 1$$

Investigate performances of predictor as a function of changes in a

1) Case 1*:* $s(n)$ = wss process with

$$R_s(k) = \delta(k) + 0.5\delta(k-1) + 0.5\delta(k+1)$$

2) Case 2: $s(n)$ = wss process with $R_s(k) = a^{|k|}, |a| < 1$




```
% EC3410 -  MPF
% Compute FIR filter coefficients for
% a 1-step ahead predictor of length 2
% for correlation sequence of type R(k)=a^{|k|}
% Compute and plot resulting MMSE value
A=[-0.9:0.1:0.9];
for k0=1:length(A)
   a=A(k0);
for k=1:3
   rs(k)=a^(k-1);
end
Rs=toeplitz(rs(1:2));
rdx=[rs(2);rs(3)];
h(:,k0)=Rs\rdx;
mmse(k0)=rs(1)-h(:,k0)'*rdx;
end
stem(A,mmse)
xlabel('value of a')
ylabel('MMSE(a)')
title('MMSE(a)for1-step predictor of length 2, …
        for R_s(k)=a^{|k|}')
```

❖ **Example 4:**

$s(n)$ = process with

$$R_s(n) = 2(0.8)^{|n|}$$

$w(n)$ = white noise, zero mean

$$R_w(n) = 2\delta(n)$$

$s(n)$, $w(n)$ uncorrelated

Design the 1-step ahead predictor of length 2.
Compute MMSE.

1- step ahead Predictor			Filter
Length	**Coefficients**	**MMSE**	**Filter MMSE**
2	[0.3238, 0.1905]	1.2381	0.81
3	[0.3059, 01.6, 0.0941]	1.2094	0.76
4	[0.3015, 0.1525, 0.0798, 0.0469]	1.2023	0.7537
5	[0.3004, 0.1506, 0.0762, 0.0762, 0.04, 0.0234]	1.2006	0.7509
6	[0.3001, 0.1502, 0.0753, 0.0381, 0.0199, 0.0199]	1.2001	0.7502
7	[0.3, 0.15, 0.0751, 0.0376, 0.0190, 0.001, 0.0059]	1.2	0.7501
8	[0.3, 0.15, 0.075, 0.0375, 0.0188, 0.0095, 0.0050, 0.003]	1.2	0.75

N-step ahead Predictor			Filter
Length	**1-step ahead MMSE**	**2-step ahead MMSE**	**Filter MMSE**
2	1.2381	1.5124	0.81
3	1.2094	1.494	0.76
4	1.2023	1.4895	0.7537
5	1.2006	1.4884	0.7509
6	1.2001	1.4881	0.7502
7	1.2	1.4880	0.7501
8	1.2	1.4880	0.75

❖ **Example 5:**

$$R_s(n) = 2(0.8)^{|n|}$$

$$R_w(n) = 2(0.5)^{|n|}$$

$s(n)$ = process with

$w(n)$ = wss noise, zero mean

$s(n)$, $w(n)$ uncorrelated

- Design the 1-step ahead predictor of length 2
- Design 1-step back smoother of length 2

1-step ahead predictor (Col. Noise)		
Length	**Coefficients**	**MMSE**
2	[0.3325, 0.1039]	1.3351
3	[0.3297, 0.06, 0.0703]	1.3237
4	[0.3285, 0.0589, 0.0406, 0.0476]	1.3185
5	[0.3279, 0.0584, 0.04, 0.0275, 0.0322]	1.3161
6	[0.3276, 0.0582, 0.0396, 0.0270, 0.0186, 0.0218]	1.315
7	[0.3275, 0.0581, 0.0394, 0.0268, 0.0183, 0.0126, 0.0148]	1.3145
8	[0.3275, 0.0581, 0.0394, 0.0267, 0.018, 0.0124, 0.0085, 0.0100]	1.3142

Length	MMSE (Col. Noise)								
	N-step ahead Predictor				N-step back Smoother				Filter
	1-step	2-step	3-step	4-step	1-step	2-step	3-step	4-step	
2	1.3351	1.5744	1.7276	1.8257	0.961	1.3351	1.5744	1.7276	0.961
3	1.3237	1.5672	1.7230	1.8227	0.925	0.9432	1.3237	1.5672	0.9432
4	1.3185	1.5638	1.7208	1.8213	0.9085	0.9085	0.9351	1.3185	0.9351
5	1.3161	1.5623	1.7199	1.8207	0.9009	0.8926	0.9009	0.9314	0.9314
6	1.315	1.5616	1.7194	1.8204	0.8975	0.8853	0.8853	0.8975	0.9297
7	1.3145	1.5613	1.7192	1.8203	0.8959	0.8819	0.8781	0.8819	0.9289
8	1.3142	1.5611	1.7191	1.8202	0.8952	0.8804	0.8748	0.8748	0.9285

Comments

❖ Example 6:

$s(n) = cos(\omega_0 n+\phi),\ \phi \sim U[0,2\pi]$

- Design the 1-step ahead predictor of length 2, and associated MMSE
- Design 1-step back smoother of length 2, and associated MMSE

❖ Weiner Filters and Error Surfaces

$$\boxed{R_x\, \underline{h}_{\text{opt}} = \underline{r}^*_{dx}}$$

Recall $\underline{h}_{\text{opt}}$ computed from

$$\sigma_e^2 = E\left\{\left|d(n) - \underline{h}^H \underline{x}\right|^2\right\} = E\left\{\left(d(n) - \underline{h}^H \underline{x}\right)\left(d(n) - \underline{h}^H \underline{x}\right)^*\right\}$$

$$= R_d(0) + \underline{h}^H E\left\{\underline{x}\underline{x}^H\right\}\underline{h} - 2\,\text{Real}\left(\underline{h}^T \underline{r}_{dx}\right)$$

↳ for real signals $d(n)$, $x(n)$

$$\boxed{\sigma_e^2 = R_d(0) + \underline{h}^T R_x \underline{h} - 2\underline{h}^T \underline{r}_{dx}}$$

using the fact that

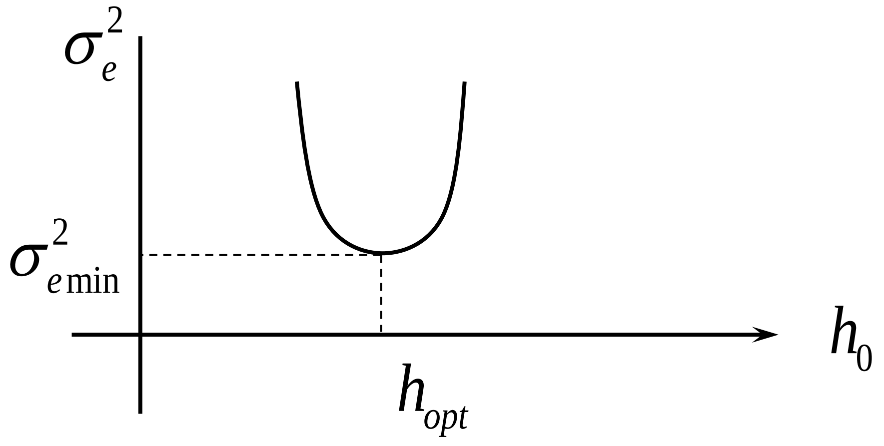
$$\left(\underline{h}^H \underline{x}\right)^* = \underline{x}^H \underline{h}$$

$$d(n)\,\underline{h}^H \underline{x} = \underline{h}^H d(n)\,\underline{x}$$

$$\boxed{\sigma_e^2 = R_d(0) + \underline{h}^T R_x \underline{h} - 2\underline{h}^T \underline{r}_{dx}}$$

- <u>For filter length 1</u> $\underline{h} = h_0$ $\underline{x} = x(n)$

$$\boxed{\sigma_e^2 = R_d(0) + h_0^{\,2} R_x(0) - 2h_0 r_{dx}(0)}$$



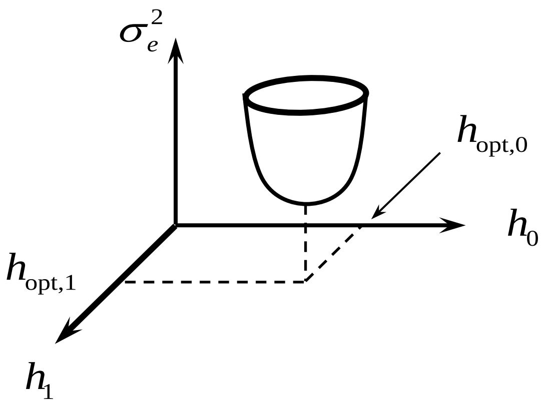
$$\boxed{\sigma_e^2 = R_d(0) + \underline{h}^T R_x \underline{h} - 2\underline{h}^T \underline{r}_{dx}}$$

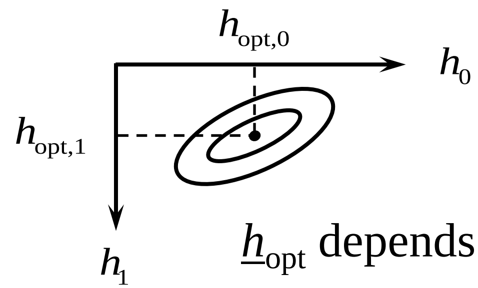
- For filter length $P = 2$

$$\underline{h} = \left[h_0, h_1\right]^T; \;\; \underline{x} = \left[x(n), x(n-1)\right]^T$$

$$\begin{aligned}
\sigma_e^2 &= R_d(0) + \underline{h}^T R_x \underline{h} - 2\underline{h}^T \underline{r}_{dx} \\
&= R_d(0) + \left[h_0, h_1\right] \begin{bmatrix} R_x(0) & R_x(1) \\ R_x(1) & R_x(0) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} \\
&\quad - 2\left[h_0, h_1\right] \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \end{bmatrix}
\end{aligned}$$

$$\boxed{\begin{aligned} \Rightarrow \;\; \sigma_e^2 = A_0 h_0^2 + A_1 h_1^2 + A_2 h_1 + A_3 h_0 \\ + A_4 h_0 h_1 + R_d(0) \end{aligned}}$$





$\underline{h}_{\text{opt}}$ depends on $\underline{r}_{dx}$ and R_x

$$\sigma_e^2 = R_d(0) + \underline{h}^T R_x \underline{h} - 2\underline{h}^T \underline{r}_{dx}$$

shape of σ_e^2 depends on R_x information: $\lambda(R_x)$ & eigenvectors

moves σ_e^2 up and down

$$\frac{\partial \sigma_e}{\partial \underline{h}} = 2R_x \underline{h} - 2\underline{r}_{dx}$$

$$R_x \underline{h} = \underline{r}_{dx}$$

R_x: specifies shape of $\sigma_e^2(\underline{h})$

$\underline{r}_{dx}$: specifies where the bowl is in the 3-d plane but

doesn't change the shape of the bowl

$R_d(0)$: moves bowl up and down in 3-d plane but doesn't change shape or location of bowl

❑ Correlation matrix Eigenvalue Spread Impact on Error Surface Shape

$\longrightarrow$ see plots

Eigenvector Directions for 2 × 2 Toeplitz Correlation Matrix

$$R_x = \begin{bmatrix} R_x(0) & R_x(1) \\ R_x(1) & R_x(0) \end{bmatrix} \rightarrow \begin{matrix}\text{normalize} \\ \text{correlation}\end{matrix} \quad \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix}$$

eigenvalues of R_x

$$(1-\lambda)^2 - a^2 = 0 \quad \Rightarrow \quad \lambda = \begin{cases} 1-a \\ 1+a \end{cases}$$

eigenvectors

$$\begin{pmatrix} 1-\lambda & a \\ a & 1-\lambda \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} = 0$$

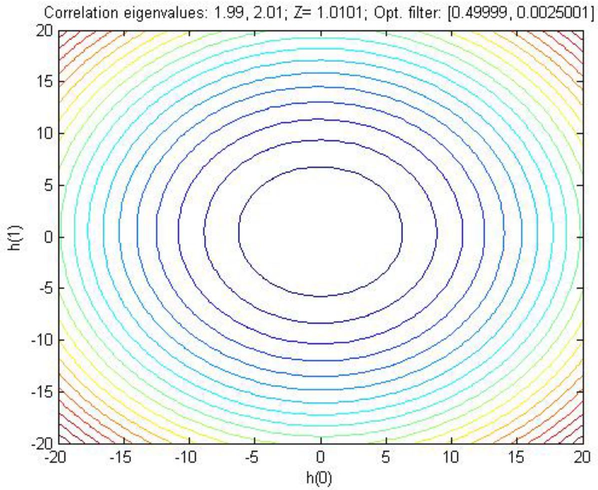
$$\Rightarrow \quad (1-\lambda)u_{11} + au_{12} = 0$$

$$\lambda_1 = 1-a \quad \Rightarrow \quad \left(\cancel{1} - \cancel{1} + a\right)u_{11} + au_{12} = 0$$

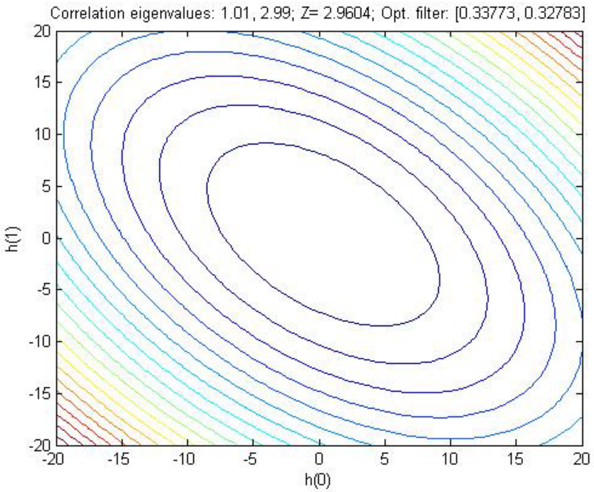
$$\Rightarrow \quad u_{11} = -u_{12} \qquad \Rightarrow \underline{u}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\lambda_2 = 1+a \quad \Rightarrow \quad \underline{u}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

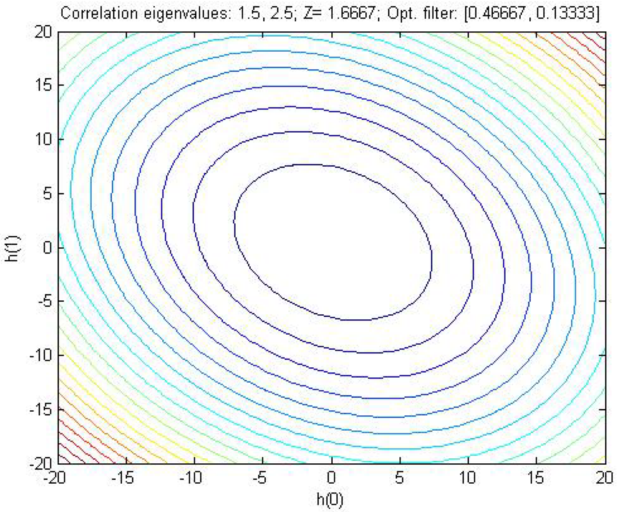
Error surface shape and eigenvalue ratios



a=0.1

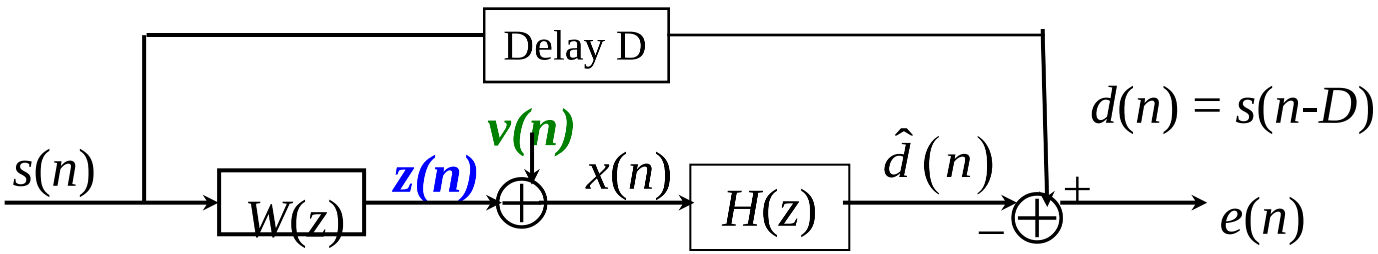


a=-0.99



a=-0.5

❖ Application to Channel Equalization



Goal: Implement the equalization filter H(z) as a stable causal FIR filter

- Goal: Recover *s(n)* by estimating channel distortion (applications in communications, control, etc.)
- Information Available:

$$x(n) = z(n) + v(n)$$

channel output **additive noise due to sensors**

$$d(n) = s(n-D)$$

s(n): original data samples

- Assumptions: 1) $v(n)$ is stationary, zero-mean, uncorrelated with $s(n)$.

  2) $v(n)$ = 0 & D=0

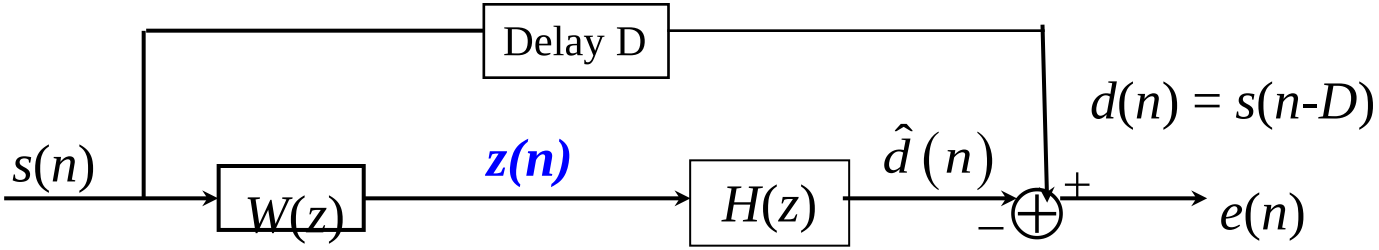
Assume: $W(z) = 0.2798 + z^{-1} + 0.2798z^{-2}$

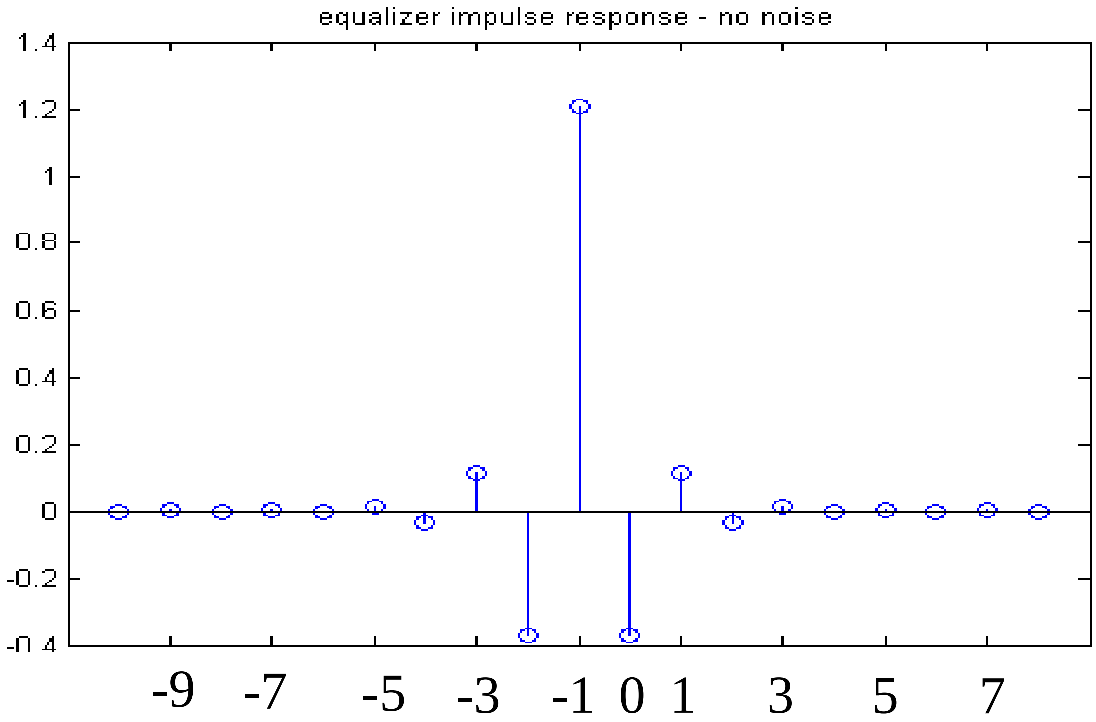
Questions:

1) Assume $v(n) = 0$ & D=0. Identify the type of filter (FIR/IIR) needed to cancel channel distortions. Identify resulting *H(z)*

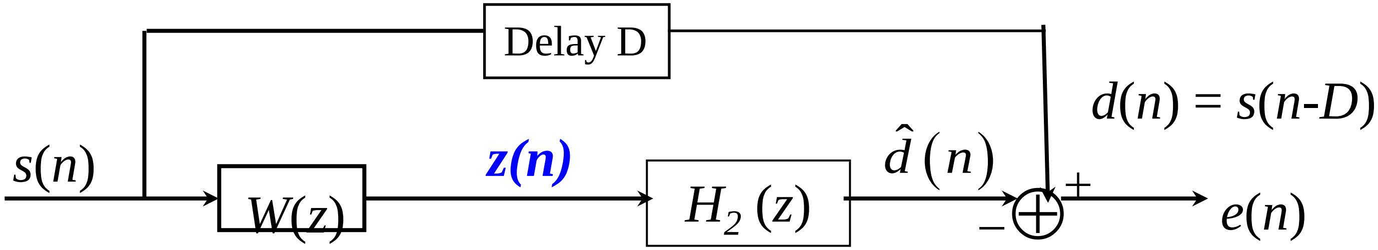
2) Identify whether the equalization filter is causal and stable.

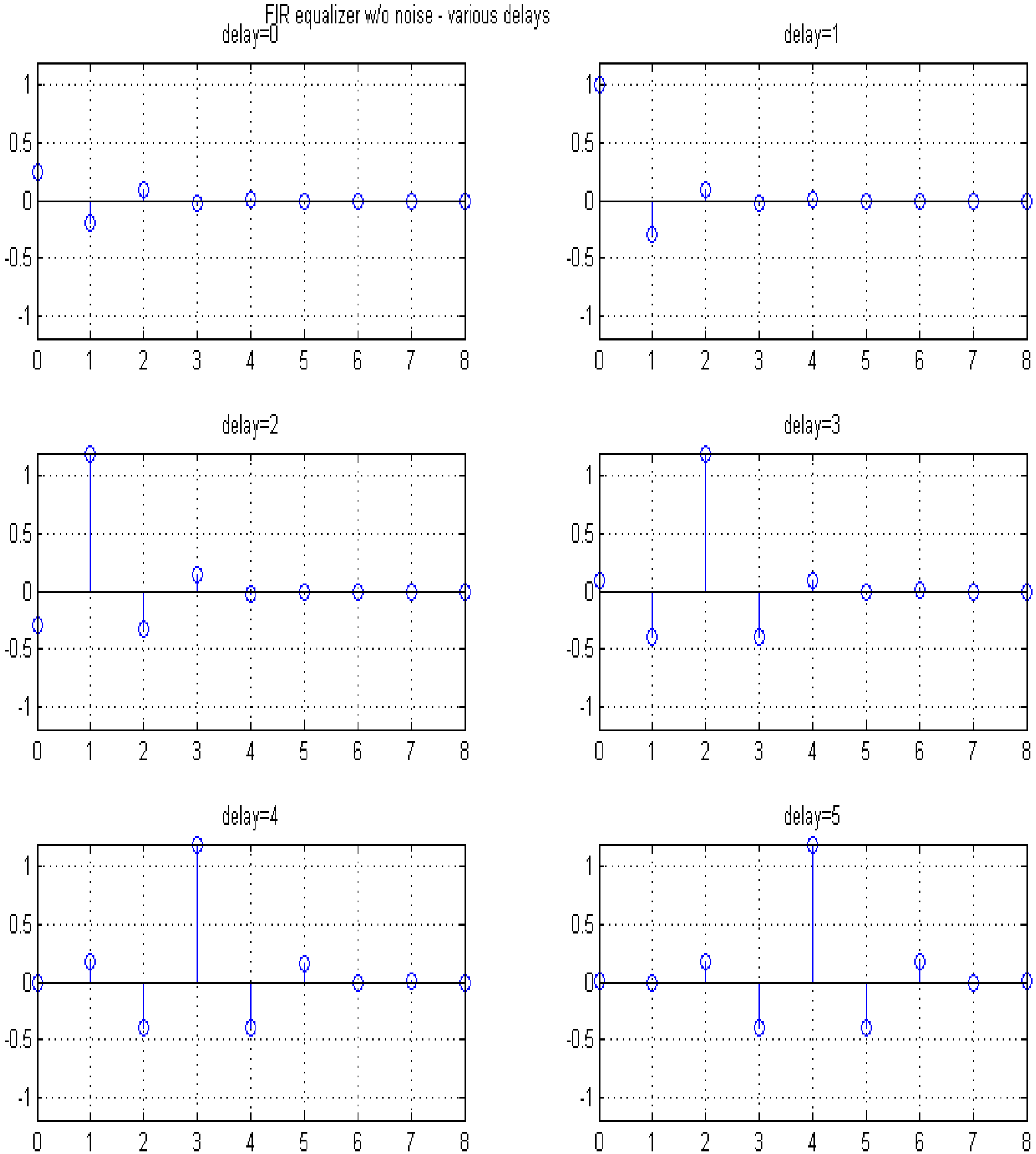
3) Assume $v(n) = 0$ & D≠0. Identify resulting $H_2(z)$ in terms of *H(z)*.

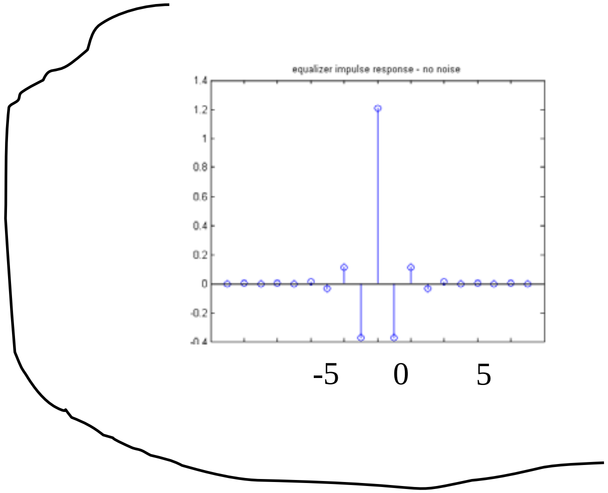



equalizer impulse response - no noise
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-9 -7 -5 -3 -1 0 1 3 5 7

Assume D≠0

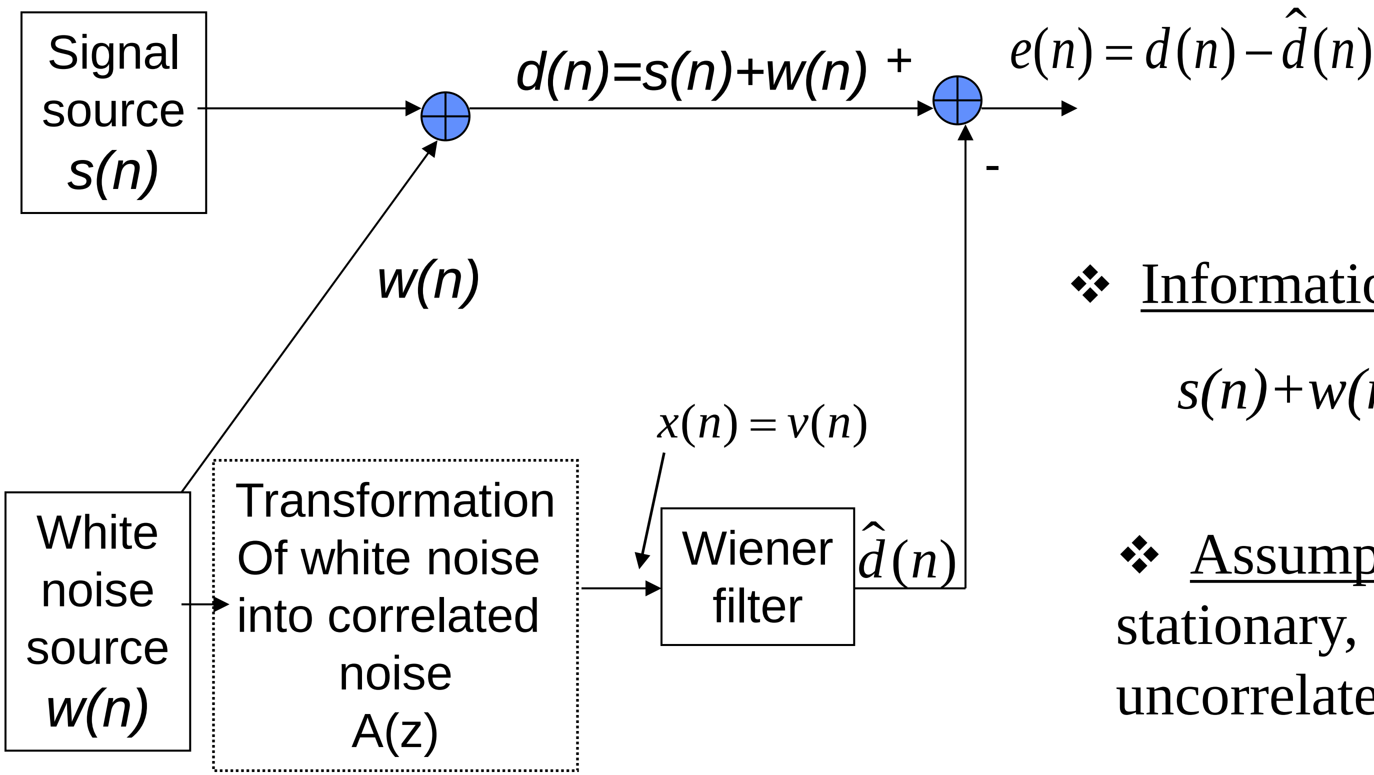



FIR equalizer w/o noise - various delays
delay=0
delay=1
delay=2
delay=3
delay=4
delay=5


equalizer impulse response - no noise
-5
0
5

❖ Application to Noise Cancellation

Goal: Recover *s(n)* by compensating for the noise distortion while having only access to the related noise distortion signal *v(n)* (applications in communications, control, etc.)



❖ Information Available:

$s(n)+w(n)$ & $v(n)$

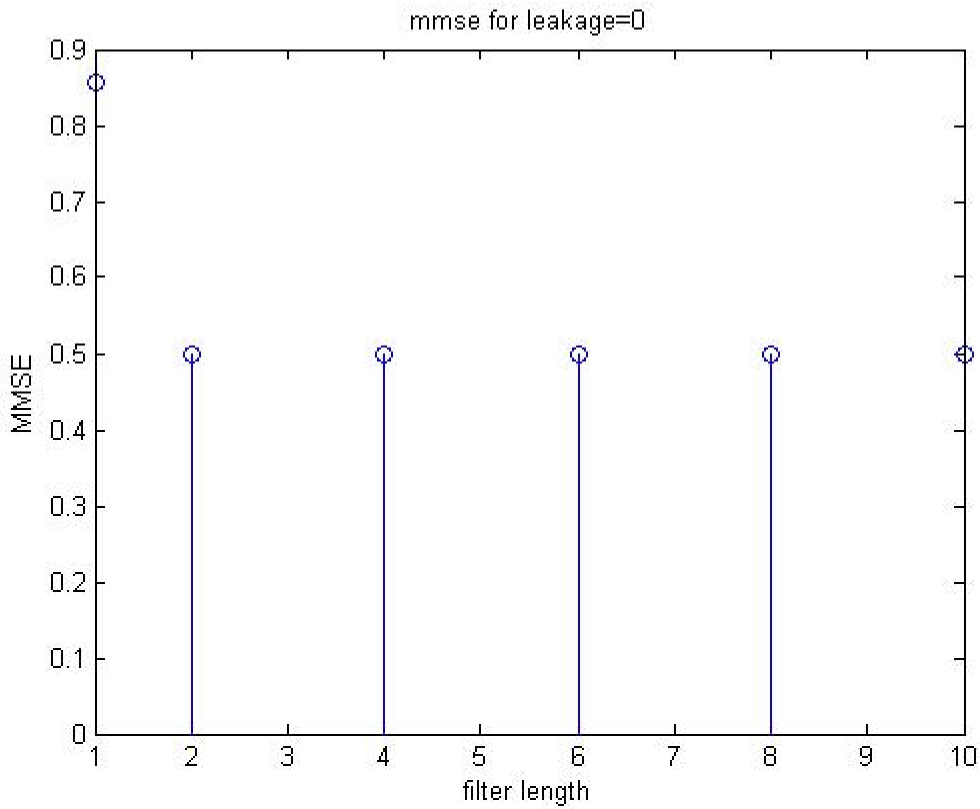
❖ Assumption: $w(n)$ is stationary, zero-mean, uncorrelated with $s(n)$.

Assume: $v(n) = av(n-1) + w(n), \quad a = 0.6,$

$w(n)$ white noise with variance σ_w^2

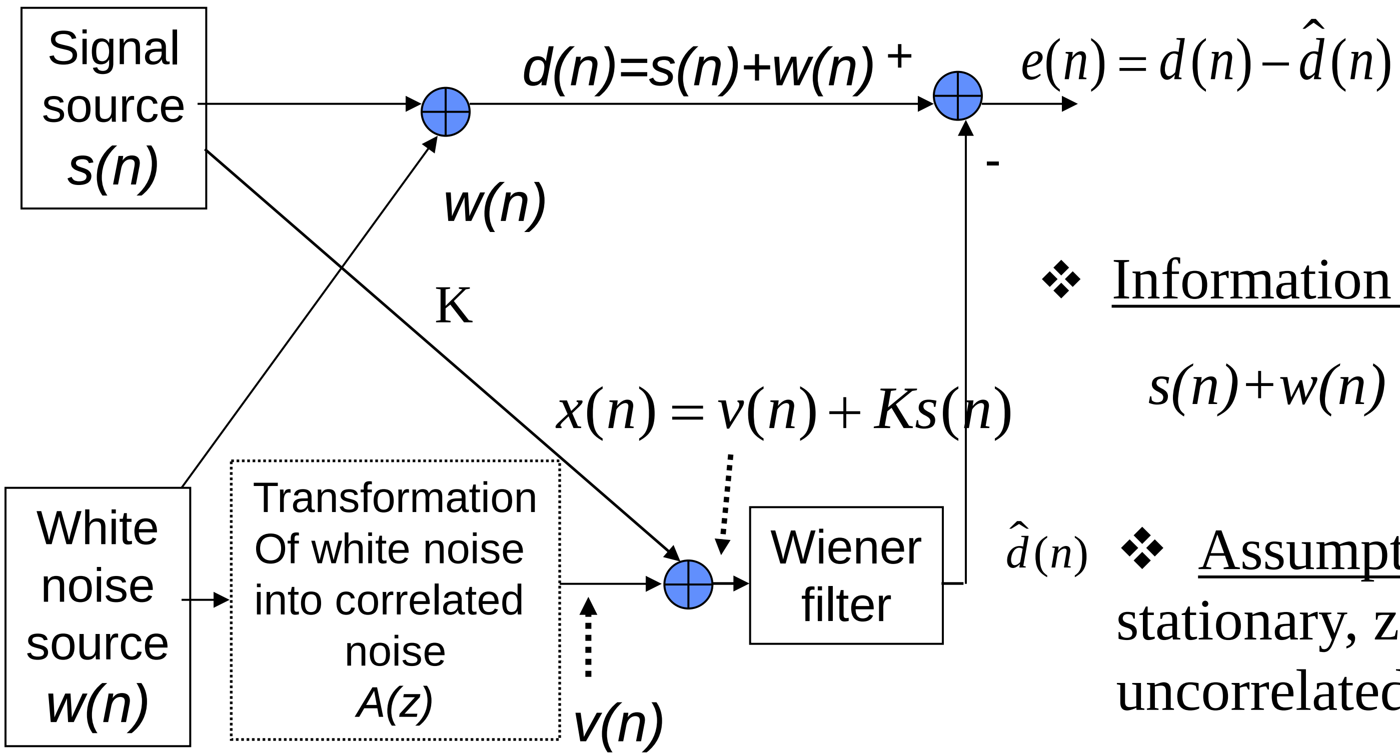
$s(n) = \sin(\omega_0 n + \phi), \quad \phi \sim U[0, 2\pi]$

Compute the FIR Wiener filter of length 2 and evaluate filter performances



Results Interpretation

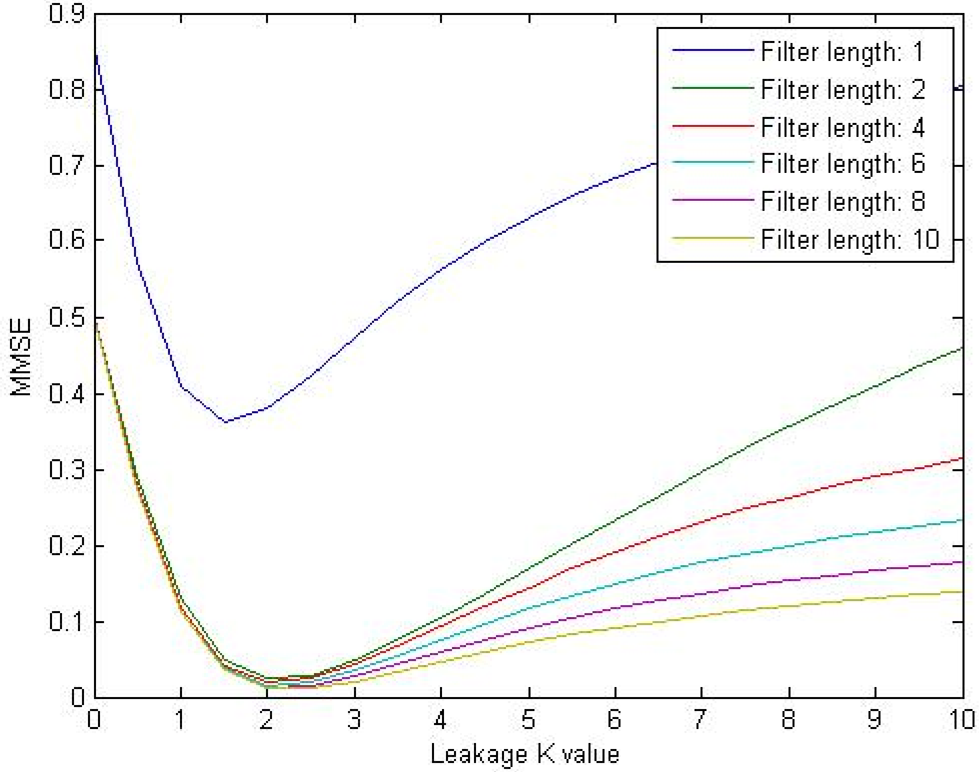
❖ Application to Noise Cancellation (with information leakage)



❖ <u>Information Available</u>:

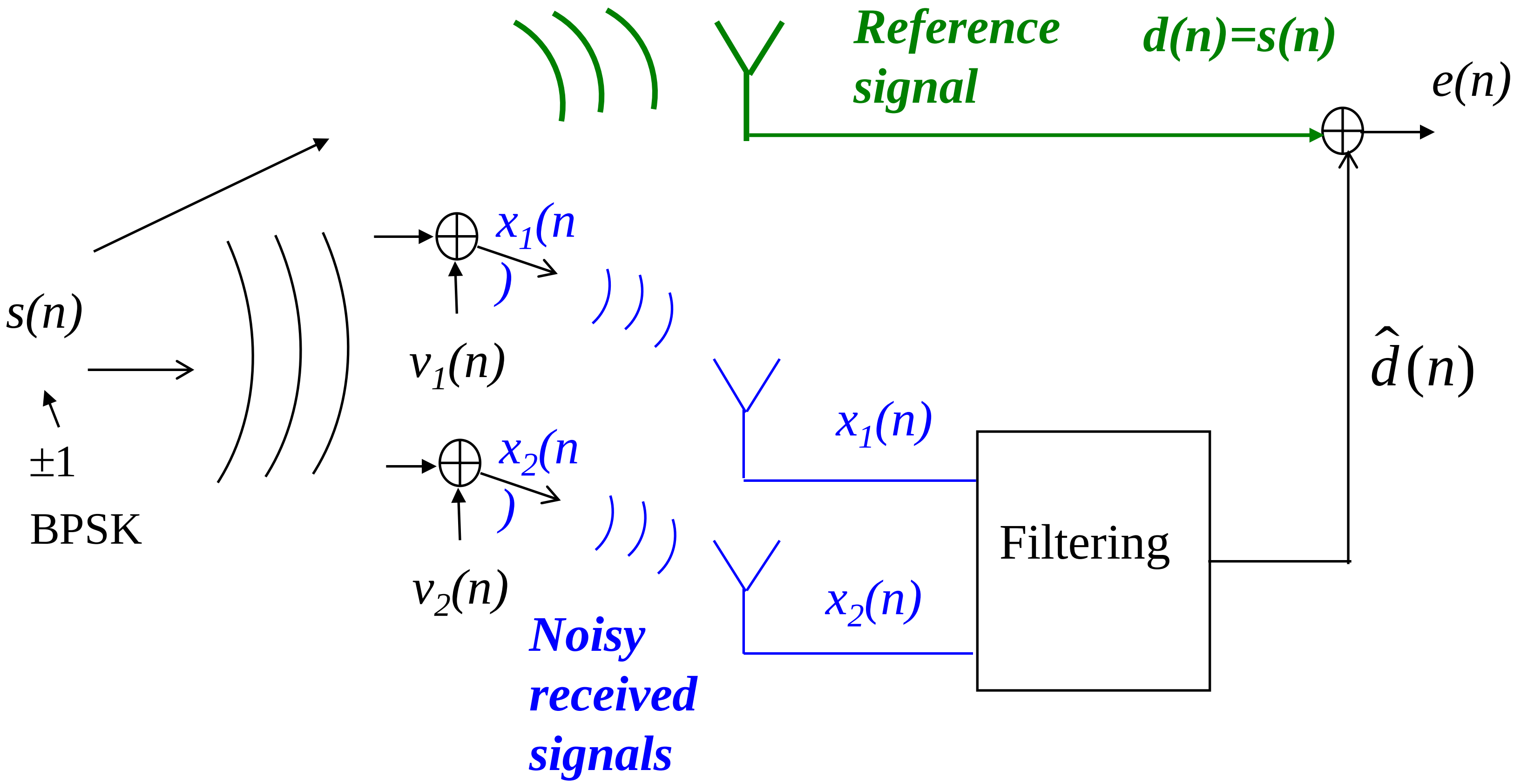
$s(n)+w(n)$ & $v(n)+Ks(n)$

❖ <u>Assumption</u>: $w(n)$ is stationary, zero-mean, uncorrelated with $s(n)$.



Results Interpretation

❖ Application to Spatial Filtering



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

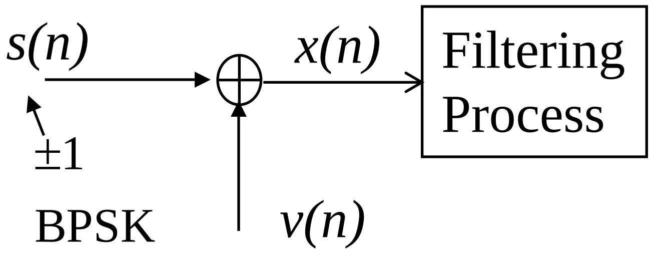
❖ <u>Assumption</u>: *$v_1(n)$, $v_2(n)$ zero mean wss white noise RPs independent of each other and of $s(n)$.*

❖ <u>Goal</u>: Denoise received signal

❖ Application to Spatial Filtering, cont'

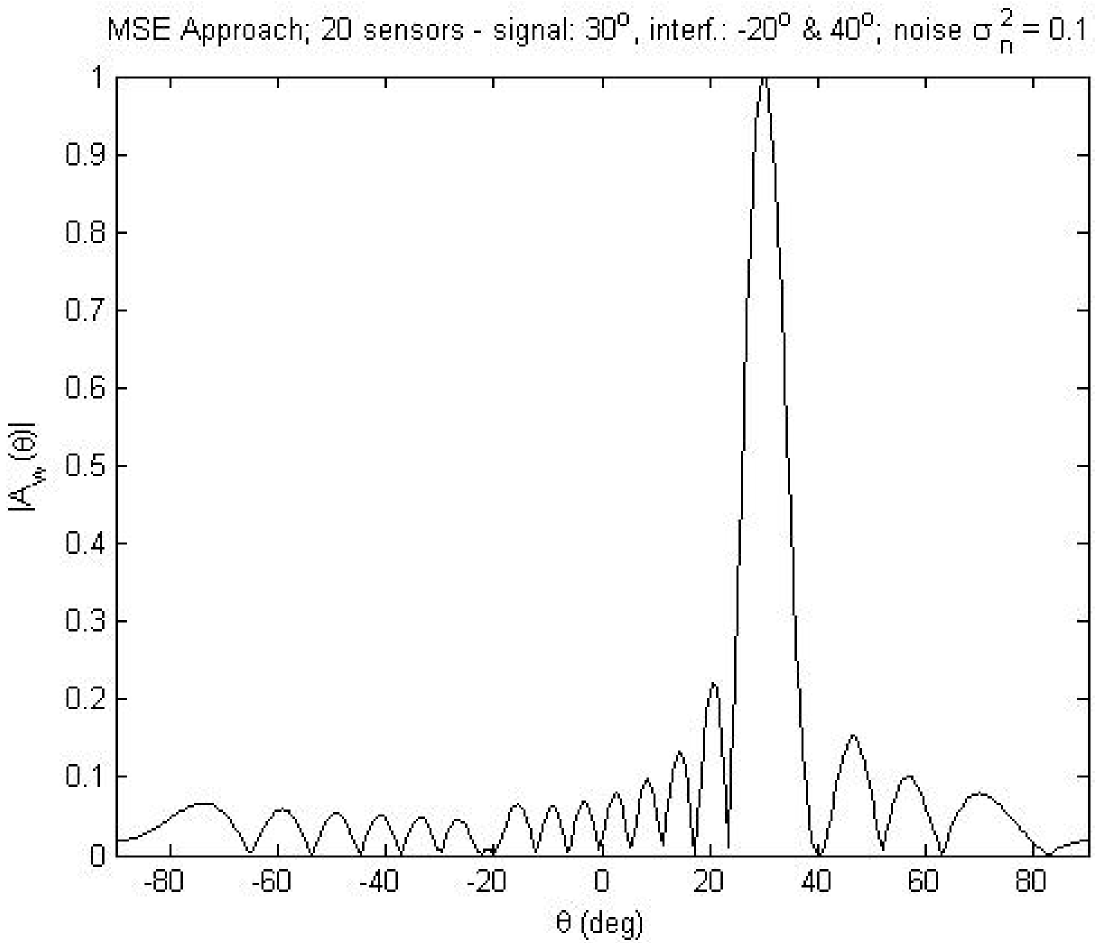
Application to Spatial Filtering, cont'

Did we gain anything by using multiple receivers?



- <u>Assumption</u>: *v(n) zero mean RP independent of s(n).*
- <u>Goal</u>: Compute filter coefficients, filter output, and MMSE.

❑ **Example:** Gain Pattern at filter output



Example:
N-element array,
- desired signal at $\theta_0 = 30°$,
- interferences at $\theta_1 = -20°$, $\theta_2 = 40°$
- Noise power: 0.1

Array steering vector

$$A_w(\theta) = \frac{\left| \underline{w}^H \underline{a}(\theta) \right|^2}{\underline{w}^H \underline{w}}$$

Appendices

Appendix A: Derivation of proof for Mean Square estimate derivation (p. 3)

Proof:

$$\xi = E\left\{\left|s-\hat{d}(\underline{x})\right|^2\right\}$$

$$= E\left\{\left(s-\underbrace{\hat{d}(\underline{x})\right)\left(s-\hat{d}(\underline{x})\right)^T}_{L(s,\hat{s}(\underline{x}))}\right\}$$

Define $L\left(s,\hat{s}(\underline{x})\right)$: loss function

$$\xi = \iint L\left(s,\hat{d}(\underline{x})\right) f\left(s|\underline{x}\right) f\left(\underline{x}\right) d\underline{x}^M ds, \quad \text{Bayes Rule}$$

$$= \int \underbrace{\left[\int L\left(s,\hat{d}(\underline{x})\right) f\left(s|\underline{x}\right) ds\right]}_{K} \underbrace{f\left(\underline{x}\right)}_{\geq 0} d^M \underline{x}$$

ξ is minimized if K is minimized for each value of $\underline{x}$

Problem: find $\hat{d}(\underline{x})$ so that K is minimum

$$\frac{\partial K}{\partial \hat{d}} = \frac{\partial}{\partial \hat{d}}\left[\int \left(s-\hat{d}(\underline{x})\right)^2 f\left(s|\underline{x}\right) ds\right]$$

$$= -2\int \left(s-\hat{d}(\underline{x})\right) f\left(s|\underline{x}\right) ds$$

$$\frac{\partial K}{\partial \hat{d}} = 0 \Rightarrow \int \left(s-\hat{d}(\underline{x})\right) f\left(s|\underline{x}\right) ds = 0$$

$$\Rightarrow \int s f\left(s|\underline{x}\right) ds = \int \hat{d}(\underline{x}) f\left(s|\underline{x}\right) ds$$

$$\Rightarrow \int s f\left(s|\underline{x}\right) ds = \hat{d}(\underline{x}) \int f\left(s|\underline{x}\right) ds$$

$$\Rightarrow \int s f\left(s|\underline{x}\right) ds = \hat{d}(\underline{x}) \times 1$$

$$\Rightarrow E[s\,|\,\underline{x}] = \hat{d}(\underline{x})$$

For a minimum we need: $\dfrac{\partial^2 K}{\partial \hat{d}^2} > 0$

Note:

$$\frac{\partial^2 K}{\partial \hat{d}^2} = (-2)\,\partial\left(\int\left(s-\hat{d}\left(\underline{x}\right)\right)f\left(s\middle|\underline{x}\right)ds\right)/\,\partial\hat{d}$$

$$=(-2)\int -f\left(s\middle|\underline{x}\right)ds$$

$$=(2)\times 1 > 0$$

Appendix B: Proof of corollary of orthogonality principle

Proof:

$$e = s - \underline{h}^H \underline{x} = s - \left(\underline{h} + \hat{\underline{h}} - \hat{\underline{h}}\right)^H \underline{x}$$

where $\hat{\underline{h}}$ is the weight vector defined so that the orthogonality principle holds. Resulting error is called $\hat{e} = s - \hat{\underline{h}}^H \underline{x}$

$$\begin{aligned}
\sigma_e^2 &= E\left\{|e|^2\right\} = E\left\{\left(s - \hat{\underline{h}}^H \underline{x} + \left(\hat{\underline{h}} - \underline{h}\right)^H \underline{x}\right)(\qquad\qquad\qquad)^H\right\} \\
&= E\left\{\left(\hat{e} + \left(\hat{\underline{h}} - \underline{h}\right)^H \underline{x}\right)(\qquad\qquad\qquad)^H\right\} \\
&= E\left\{\left|\hat{e}\right|^2 + \left(\hat{\underline{h}} - \underline{h}\right)^H \underline{x}\hat{e}^H + \left|\left(\hat{\underline{h}} - \underline{h}\right)^H \underline{x}\right|^2 + \hat{e}\underline{x}^H\left(\hat{\underline{h}} - \underline{h}\right)\right\} \\
&= E\left\{\left|\hat{e}\right|^2\right\} + \left(\hat{\underline{h}} - \underline{h}\right)^H \cancel{E\left\{\underline{x}\hat{e}^H\right\}} + E\left\{\left|\left(\hat{\underline{h}} - \underline{h}\right)^H \underline{x}\right|^2\right\} + \cancel{E\left\{\hat{e}\underline{x}^H\right\}}\left(\hat{\underline{h}} - \underline{h}\right) \\
&= E\left\{\left|\hat{e}\right|^2\right\} = E\left\{(s - \hat{\underline{h}}^H \underline{x})\hat{e}^H\right\} = E\left\{s\hat{e}^H\right\} - \hat{\underline{h}}^H \cancel{E\left\{\underline{x}\hat{e}^H\right\}} \\
&= E\left\{s\hat{e}^H\right\} = E\left\{s\hat{e}^*\right\}
\end{aligned}$$

References

[1]: *Discrete Random Signals and Statistical Signal Processing*, C. Therrien, Prentice Hall, 1992
[2]: *Statistical and Adaptive Signal Processing*, D. Manolakis, V. Ingle & S. Kogon, Artech House, 2005
[3]: *Adaptive Filters*, 2nd Ed., S. Haykin, Prentice Hall, 1991