Chapter 5:
Wiener Filter

- Wiener filter
  - Principle of orthogonality
  - Wiener-Hopf equations
  - Error performance surface
- Linearly Constrained Minimum Variance Filter (LCMV Filter)
- Examples: Fixed Weight Beamforming

References

[1] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, 1996.

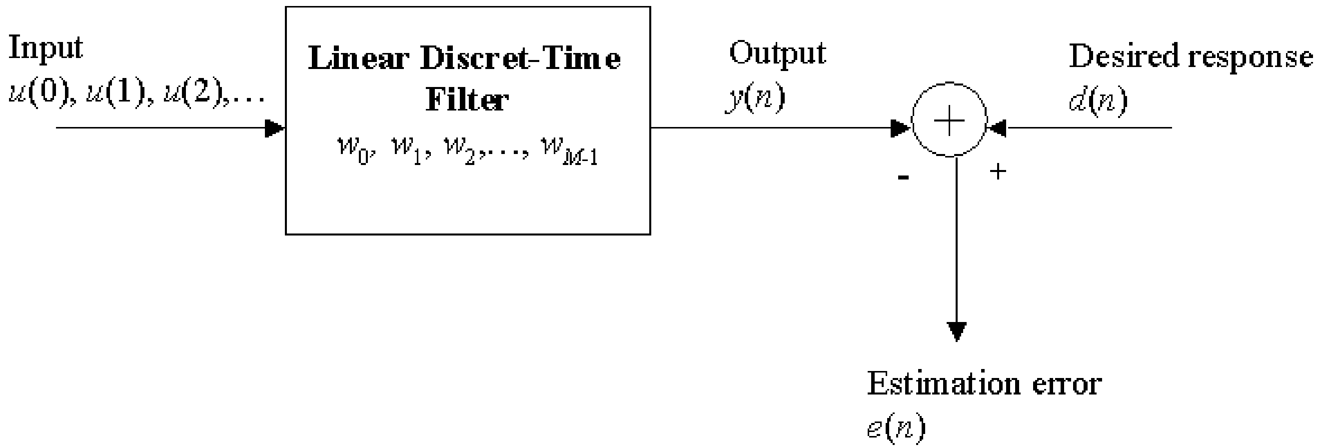
[2] Steven M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[3] Alan V. Oppenheim, Ronald W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, 1989.

[4] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1991.

5. Wiener Filter: Linear Optimum Filtering

- Block diagram representation of the optimum filtering problem:



- The goal of the optimum filter is to provide an estimate of the desired response that is "as close as possible".
- Questions :
  - Is the filter impulse response finite or infinite?
  - What statistical criterion is used for optimization?

5. Wiener Filter: Statistical Criteria for Optimization

- Typical choices
  - Mean-squared value of the error
  - Expectation of the absolute value of the error
  - Expectation of third or higher order powers of the absolute value of the error.

- The first choice is most preferred as it leads to a mathematically tractable solution.

- **Minimum Mean-Squared Error (MMSE) criteria**
  - Design the linear discrete time filter such that the mean-squared value of the estimation error is minimum.

5. WF: Mean-Squared Error (Cost Function)

Mean-Squared Error (Cost Function):

$$J = E\left[e(n)e^*(n)\right] = E\left[\left|e(n)\right|^2\right]$$

where

$$e(n) = d(n) - y(n)$$

$$y(n) = \sum_{k=0}^{M-1} w_k^* u(n-k) = \mathbf{w}^H \mathbf{u}(n)$$

$$\mathbf{u}(n) = \begin{bmatrix} u(n) \\ u(n-1) \\ \dots \\ u(n-M+1) \end{bmatrix}$$

Question: Condition for $J \to J_{\min}$?

5. WF: Principle of Orthogonality (1)

- Taking derivative of J with respect to w^*:

$$\frac{\partial J}{\partial \mathbf{w}^*} = \frac{\partial}{\partial \mathbf{w}^*}\left(E\left[e(n)e^*(n)\right]\right) = E\left[-\mathbf{u}(n)e^*(n)\right]$$

Let $\frac{\partial J}{\partial \mathbf{w}^*} = 0 \Rightarrow E\left[\mathbf{u}(n)e^*_{opt}(n)\right] = \mathbf{0}$: **Principle of orthogonality**

- Moreover: $E\left[y(n)e^*(n)\right] = \mathbf{w}^H E\left[\mathbf{u}(n)e^*(n)\right]$

with optimum condition: $E\left[\mathbf{u}(n)e^*_{opt}(n)\right] = \mathbf{0}$

then $E\left[y_{opt}(n)e^*_{opt}(n)\right] = \mathbf{0}$

and $y_{opt}(n) + e_{opt}(n) = d(n)$

5. WF: Principle of Orthogonality (2)

- The **principle of orthogonality** states that:
  - The necessary and sufficient condition for the cost function J to attain its minimum value is that the estimation error and the input values are orthogonal to each other.

- The corollary:
  - When the filter operates in its optimum condition, the output of the filter is orthogonal to the estimation error.

5. WF: Wiener-Hopf Equations (1)

- The MMSE criterion for designing the optimal filter leads to a set of equations given by:

$$E\left[u(n-k)e^{*}(n)\right]=0, \qquad k=0,1,2,\ldots,M-1$$

then $$\sum_{i=0}^{M} w_{oi} E\left[u(n-k)u^*(n-i)\right]=E\left[u(n-k)d^*(n)\right],$$

where *r(k)* is the auto-correlation function of the input, *p(k)* is the cross-correlation between the input and the desired response, and $w_{opt,i}$ is the i^{th} optimal weight value.

- These equations are known as the **Wiener - Hopf equations** and form the basis for adaptive filtering algorithms.

5. WF: Wiener-Hopf Equations (2)

- In matrix form:

$$\mathbf{R}\mathbf{w}_{opt} = \mathbf{p}$$

where

$$\mathbf{R} = E\left[\mathbf{u}(n)\mathbf{u}^H(n)\right] \in C^{M\times M}, \quad \mathbf{R} = \mathbf{R}^H$$

$$\mathbf{p} = E\left[\mathbf{u}(n)d^*(n)\right] \in C^M$$

then $$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$$

5. WF: Error Performance Surface

- The **cost function** J for the transversal filter can be written as:

$$J(\mathbf{w}) = E\left[e(n)e^{*}(n)\right] = \sigma_d^2 - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w}$$

where σ_d^2 is the variance of the desired signal.

In **optimum condition**, we obtain **MMSE**:

$$J_{\min} = J(\mathbf{w}_{opt}) = \sigma_d^2 - \mathbf{p}^H\mathbf{w}_{opt} = \sigma_d^2 - \mathbf{p}^H\mathbf{R}^{-1}\mathbf{p}$$

- The cost function is a second order function of the tap weights. It can be visualized as a bowl-shaped (M+1) dimensional surface with M degrees of freedom.

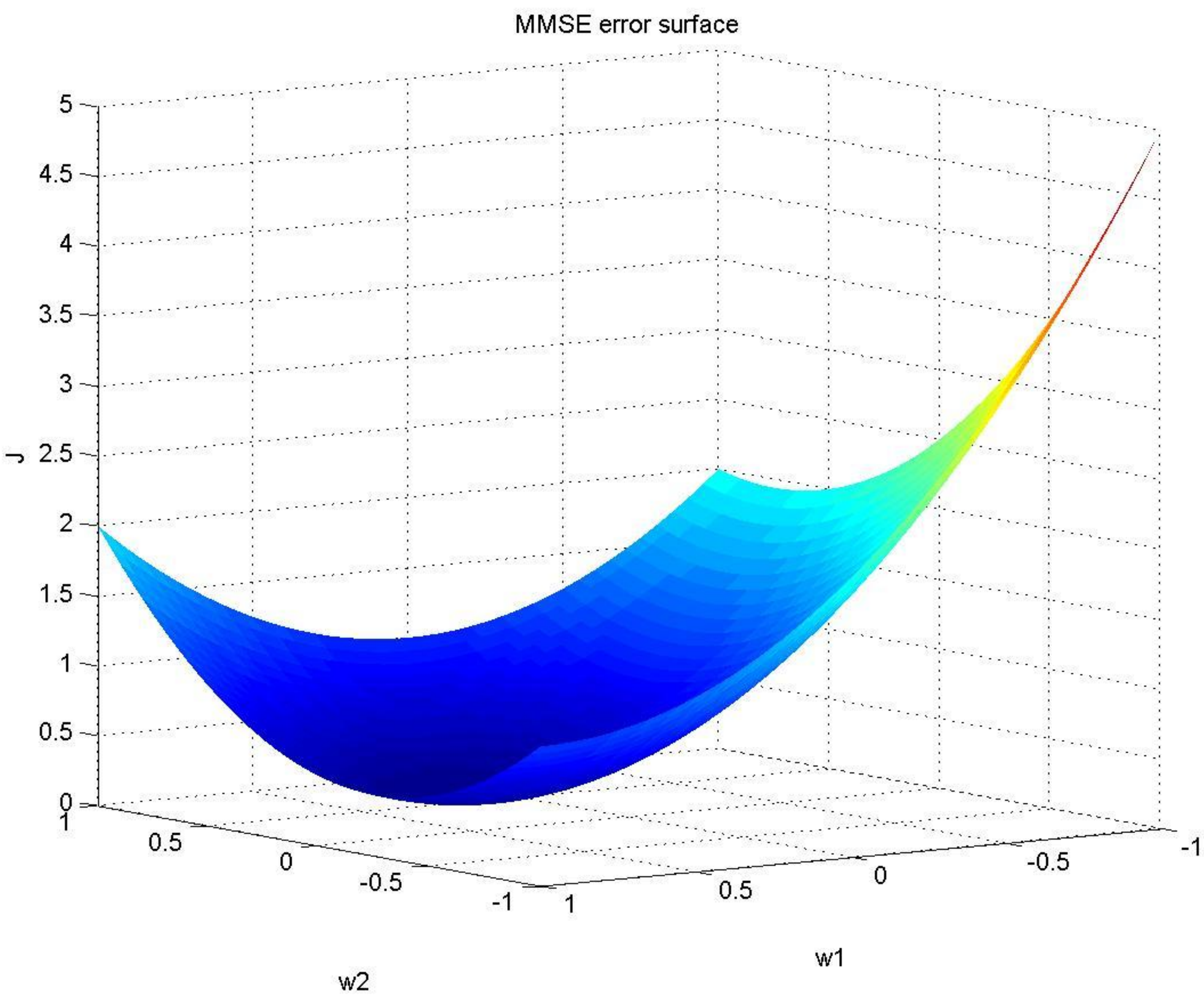
5. WF: Example 1 (1)

- In a Wiener filtering problem, the correlation matrix and the cross-correlation vector are:

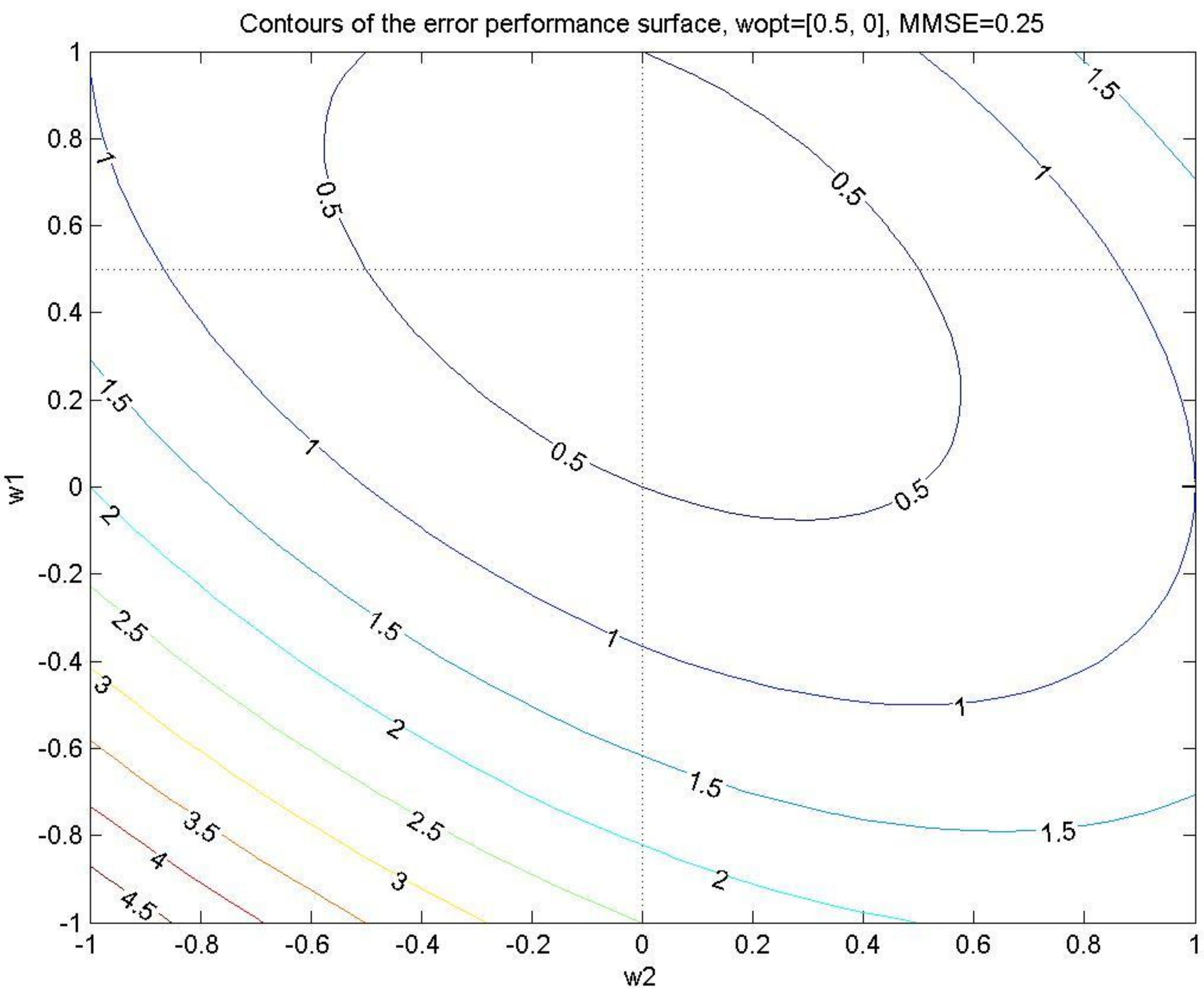
$$\mathbf{R} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \ \mathbf{p} = [0.5 \quad 0.25]^T$$

  - Evaluate the tap weights of the Wiener filter.
  - Express the cost function in terms of the weights.
  - Plot the error performance surface assuming that the variance of the desired input is 0.5. What is the minimum mean squared error?

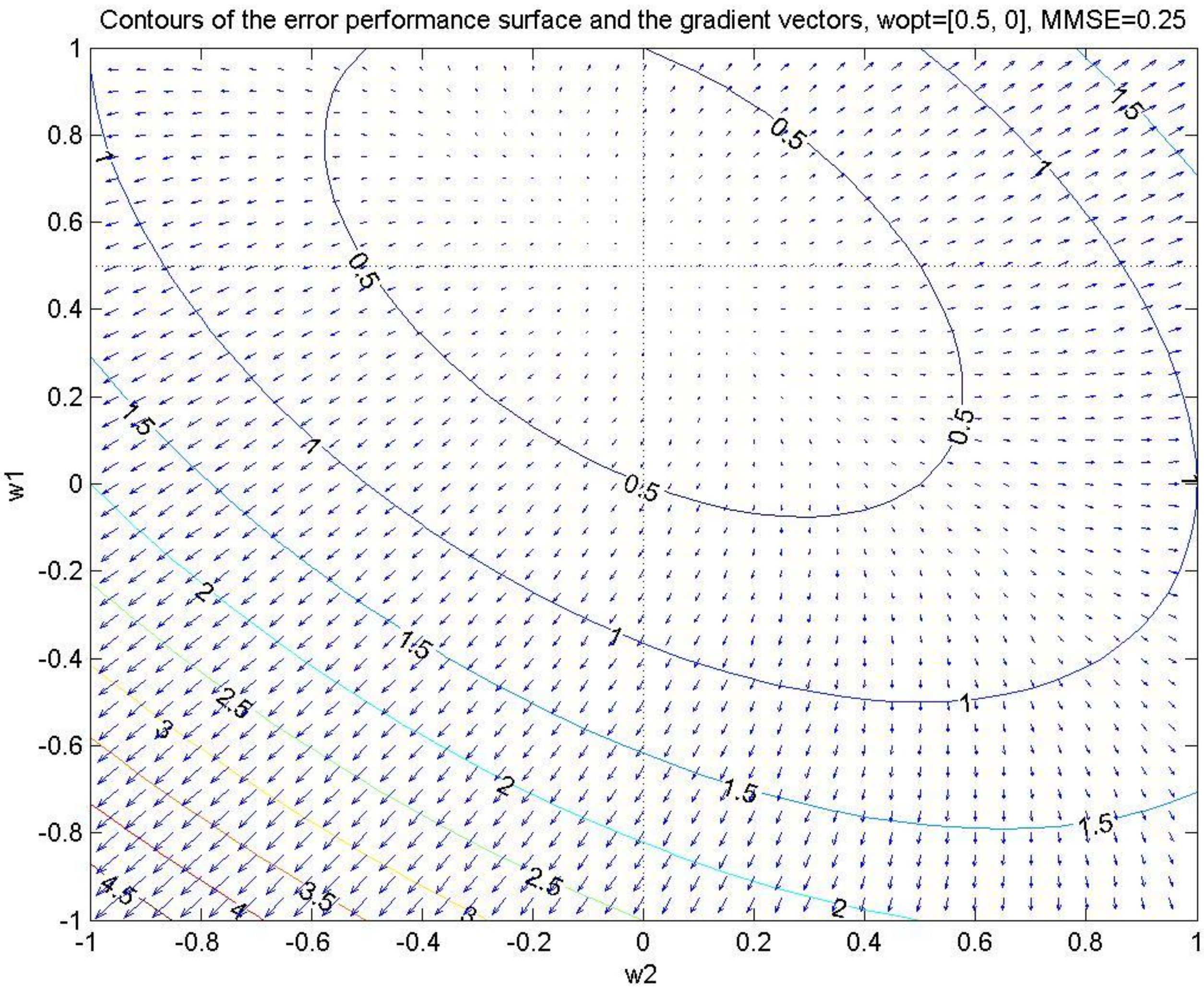
5. WF: Example 1 (2)



5. WF: Example 1 (3)

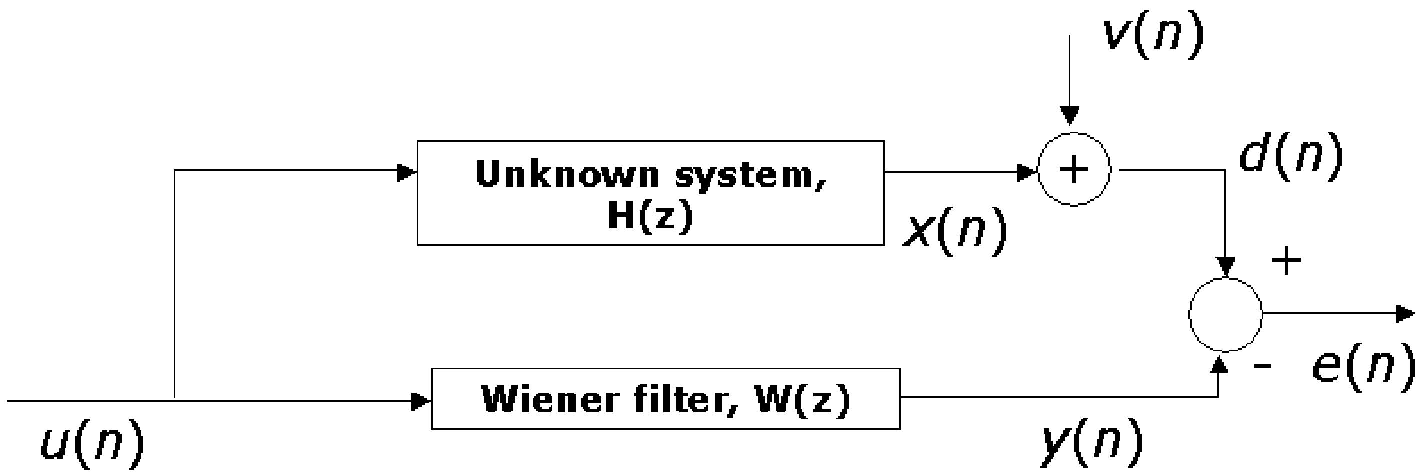


5. WF: Example 1 (4)



5. WF: Example 2

- Consider the **system identification** problem shown below:



Both $u(n)$ and $v(n)$ are zero-mean, white noise sequences with variances 0.5 and 0.1 respectively.

If $H(z) = 1\text{-}0.5z^{-1}+0.25z^{-2}$, find the optimum Wiener solution for filter lengths of 1, 2, 3, and 4. In each case, calculate the minimum mean squared error.

5. WF: Constrained Optimization

- The Wiener-Hopf equations are sometimes referred to as **unconstrained** optimization.

- In some situations, a constraint is placed on finding the solution with minimum mean squared error.

- The solution, termed **Linearly Constrained Minimum Variance** (**LCMV**), is found using Lagrange multipliers.

5. WF: LCMV Filter (1)

- Output of the filter: $y(n) = \mathbf{w}^H \mathbf{u}(n)$

  Assumed a complex sinusoidal excitation: $u(n) = e^{j\omega_0 Tn}$

  then:
  $$y(n) = \mathbf{w}^H e^{j\phi_0 n}\mathbf{a}(\phi_0) = e^{j\phi_0 n}\mathbf{w}^H \mathbf{a}(\phi_0)$$

  where $\phi_0 = \omega_0 T, \quad \mathbf{a}(\phi_0) = \left[1, e^{-j\phi_0}, \ldots, e^{-j\phi_0 (M-1)}\right]$

- The **linear constraint**: $\mathbf{w}^H \mathbf{a}(\phi_0) = g^*$, enforcing a:
  - certain value g^* of the transfer function of the filter at frequency $\omega_0=\phi_0/\mathrm{T}$ (*case of temporal frequency*),
  - certain antenna gain at an angle of arrival θ_0 with $\phi_0=2\pi\Delta\sin\theta_0/\lambda$, where Δ is inter-antenna element spacing and λ is wavelength of the incomming signal (*case of spatial frequency*),

  while trying to find a weight vector $\mathbf{w}_{opt}$, which minimizes the output power $E[y(n)y^*(n)]$

5. WF: LCMV Filter (2)

- It means:

$$J(\mathbf{w}) = E\left[y(n)y^*(n)\right] = E\left[\mathbf{w}^H\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}\right] = \mathbf{w}^H\mathbf{R}\mathbf{w} \rightarrow J_{\min}$$

$$\text{subject to} \qquad \mathbf{w}^H\mathbf{a}(\phi_0) = g^*$$

- Using **Lagrange multipliers** (see Reference [1]), we obtain finally:

$$\mathbf{w}_{opt} = \frac{g\mathbf{R}^{-1}\mathbf{a}(\phi_0)}{\mathbf{a}^H(\phi_0)\mathbf{R}^{-1}\mathbf{a}(\phi_0)}$$

- When g=1 $\Rightarrow$ **Minimum Variance Distortionless Response** (**MVDR**), with

$$\mathbf{w}_{opt} = \frac{\mathbf{R}^{-1}\mathbf{a}(\phi_0)}{\mathbf{a}^H(\phi_0)\mathbf{R}^{-1}\mathbf{a}(\phi_0)}$$

5. Smart Antennas (1)

- Traditional array antennas, where the main beam is steered to directions of interest, are called **phased arrays**, **beamsteered arrays**, or **scanned arrays**. The beam is steered via phase shifters oftenly implemented at RF frequencies. This general approach to phase shifting has been referred to as electronic beamsteering because of the attempt to change the phase of the current directly at each antenna element.

- Modern beamsteered array antennas, where the pattern is shaped according to certain optimum criteria, are called **smart antennas**. Smart antennas have alternatively been called **digital beamformed (DBF) arrays** or **adaptive arrays** (when adaptive algorithms are employed). The term smart implies the use of digital signal processing in order to shape the beam pattern according to certain conditions. Since an antenna pattern (or beam) is formed by digital signal processing, this process is often referred to as **digital beamforming**.

5. Smart Antennas (2)

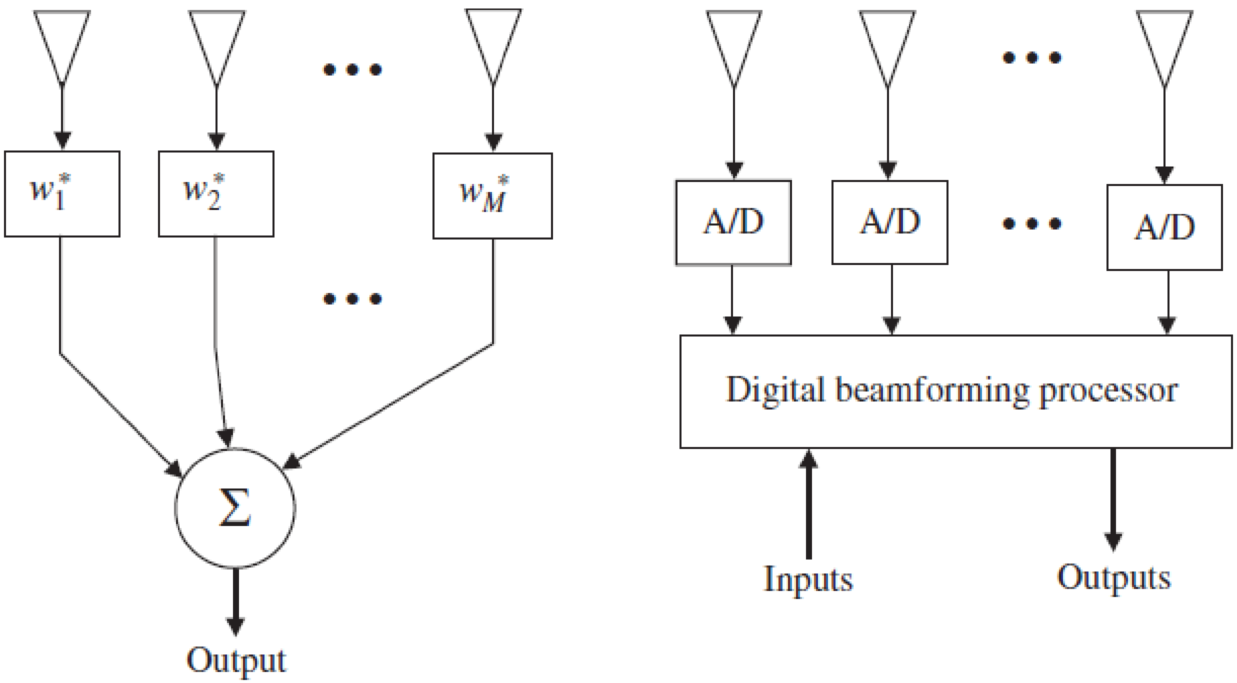
- Smart antennas can be applied for improved radar systems, improved system capacities with mobile wireless, and improved wireless communications through the implementation of **space division multiple access (SDMA).**

- Smart antenna patterns are controlled via algorithms based upon certain criteria. These criteria could be maximizing the **signal-tointerference ratio (SIR)**, **minimizing the variance, minimizing the means-quare error (MSE), steering toward a signal of interest,** or **nulling the interfering signals**.

- The implementation of these algorithms can be performed electronically through analog devices but it is generally more easily performed using digital signal processing. This requires that the array outputs be digitized through the use of an A/D converter. This digitization can be performed at either IF or baseband frequencies.

5. Smart Antennas (3)

- The main advantage of digital beamforming is that phase shifting and array weighting can be performed on the digitized data rather than by being implemented in hardware. If the parameters of operation are changed or the detection criteria are modified, the beamforming can be changed by simply changing an algorithm rather than by replacing hardware.



5. Smart Antennas (4)

- There are two types of digital beamforming:
  - **Fixed weight beamforming.**
  - **Adaptive beamforming.**

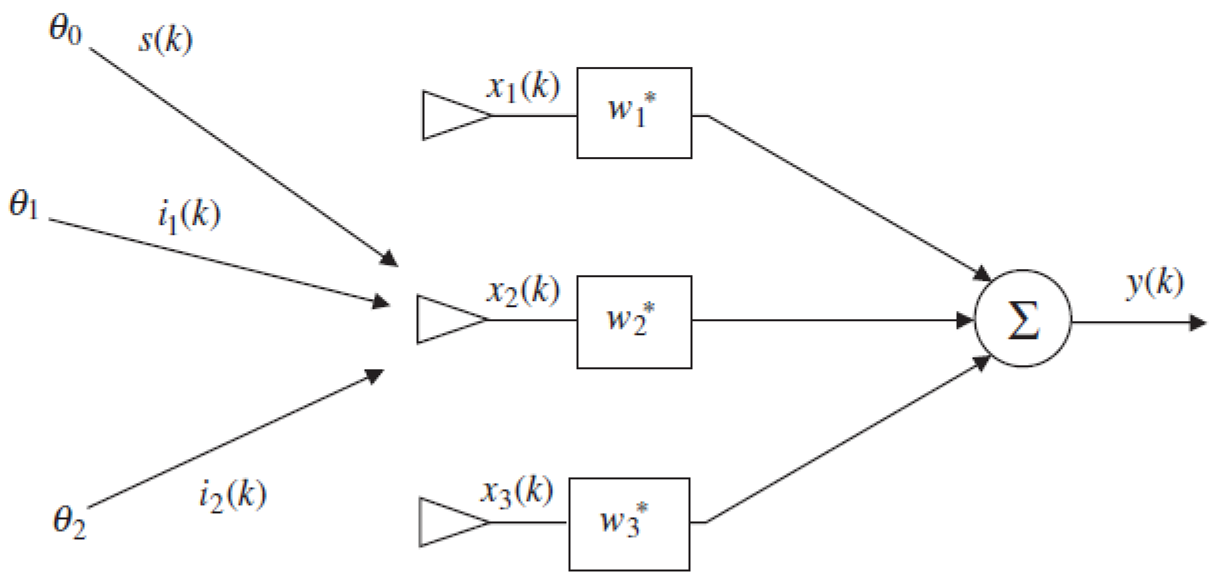
In this chapter, we consider algorithms for fixed weight beamforming. Algorithms for adaptive beamforming will be considered in chapter 8.

5. Fixed Weight Beamforming (1)

- **<u>Maximum signal-to-interference ratio</u>:**

  One criterion which can be applied to enhancing the received signal and minimizing the interfering signals is based upon maximizing the SIR.

  <u>Example</u>: 3-element array with one fixed known desired source and two fixed undesired interferers. All signals are assumed to operate at the same carrier frequency.



5. Fixed Weight Beamforming (2)

The required complex weights w_1, w_2, and w_3 can be determined as

$$\mathbf{w}^H = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} = \mathbf{u}_1^T \mathbf{A}^H \left(\mathbf{A}\mathbf{A}^H + \sigma_n^2 \mathbf{I} \right)^{-1}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_0 & \mathbf{a}_1 & \mathbf{a}_2 \end{bmatrix}$$

is matrix of steering vectors, and $\mathbf{u}_1 = [1 \;\; 0 \;\; \ldots \; 0]^T$. The steering vector for each source is given by

$$\mathbf{a} = \begin{bmatrix} e^{-jkd\sin\theta} & 1 & e^{jkd\sin\theta} \end{bmatrix}^T$$

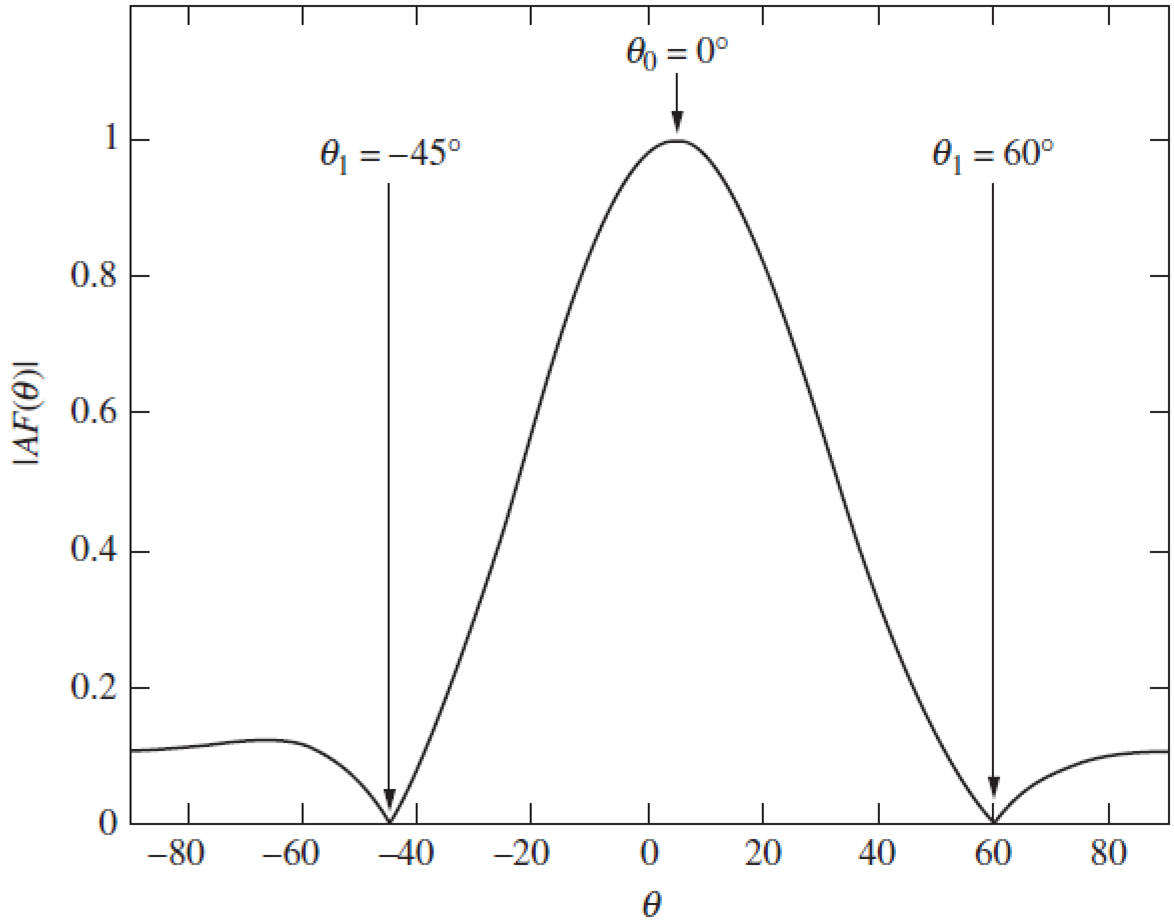
and $\sigma_n^{\,2}$ is the noise variace (noise power).

<u>Example</u>: if the desired signal is arriving from $\theta_0 = 0^\circ$, while $\theta_1 = -45^\circ$ and $\theta_2 = 60^\circ$, the necessary weights can be calculated to be

$$\begin{bmatrix} w_1^* \\ w_2^* \\ w_3^* \end{bmatrix} = \begin{bmatrix} 0.28 - j0.07 \\ 0.45 \\ 0.28 + j0.07 \end{bmatrix}$$

5. Fixed Weight Beamforming (3)

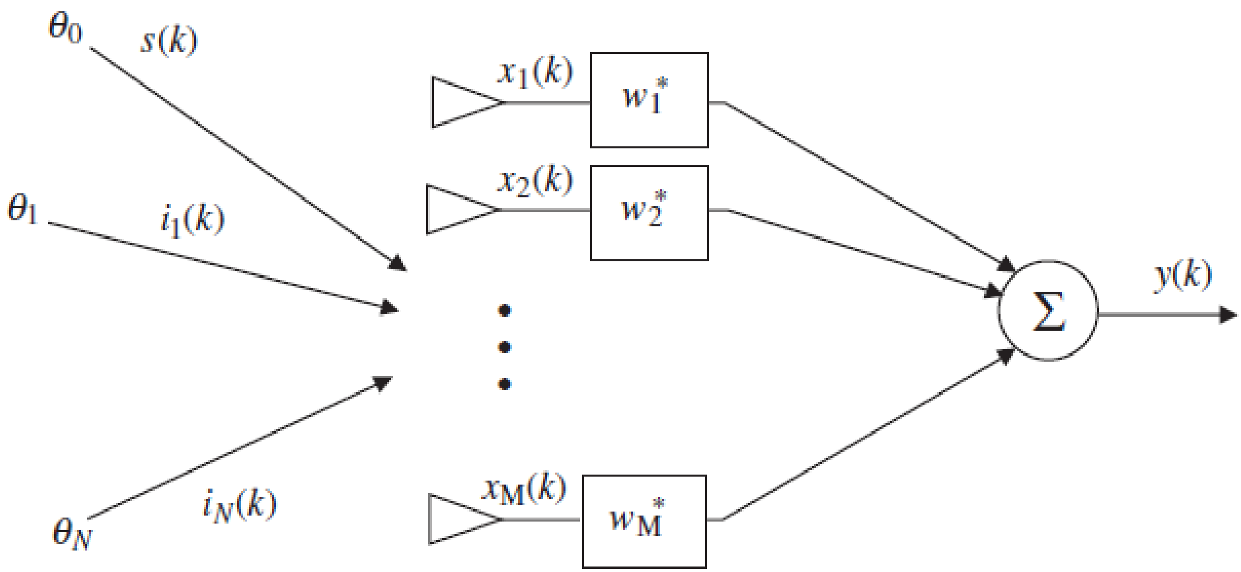
The array factor is plotted as



5. Fixed Weight Beamforming (4)

The general case for max SIR:

It shows one desired signal arriving from the angle θ_0 and N interferers arriving from angles $\theta_1, \ldots, \theta_N$. The signal and the interferers are received by an array of M elements with M potential weights. Each received signal at element m also includes additive Gaussian noise. Time is represented by the *kth* time sample.



5. Fixed Weight Beamforming (5)

The array output y can be given in the following form:

$$y = \mathbf{w}^H \mathbf{x}(k)$$

where

$$\mathbf{x}(k) = \mathbf{a}_0 s(k) + \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \end{bmatrix} \begin{bmatrix} i_1(k) \\ i_2(k) \\ \vdots \\ i_N(k) \end{bmatrix} + \mathbf{n}(k)$$

$$= \mathbf{x}_s(k) + \mathbf{x}_i(k) + \mathbf{n}(k)$$

and $\mathbf{w} = [w_1 \ \ w_2 \ \ \ldots \ \ w_M]^T$ is array weights; $\mathbf{x}_s(k)$: desired signal vector; $\mathbf{x}_i(k)$: interfering signals vector; $\mathbf{n}(k)$: zero-mean Gaussian noise for each channel; and $\mathbf{a}_i$: M-element array steering vector for the θ_i direction of arrival.

5. Fixed Weight Beamforming (6)

Therefore, the array output can be re-written as

$$y = \mathbf{w}^H \mathbf{x}(k) = \mathbf{w}^H \left[\mathbf{x}_s(k) + \mathbf{x}_i(k) + \mathbf{n}(k)\right] = \mathbf{w}^H \left[\mathbf{x}_s(k) + \mathbf{u}(k)\right]$$

where $\mathbf{u}(k) = \mathbf{x}_i(k) + \mathbf{n}(k)$ is undesired signal.

The SIR is defined as the ratio of the desired signal power divided by the undesired signal power

$$SIR = \frac{\sigma_s^2}{\sigma_u^2} = \frac{\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w}}$$

where $\mathbf{R}_{ss} = E[\mathbf{x}_s \mathbf{x}_s^H]$ is signal correlation matrix; $\mathbf{R}_{ii} = E[\mathbf{x}_i \mathbf{x}_i^H]$ is correlation matrix for interferers; and $\mathbf{R}_{nn} = E[\mathbf{n}\mathbf{n}^H]$ is correlation matrix for noise. The SIR can be maximized by taking the derivative with respect to $\mathbf{w}$ and setting the result equal to zero, then we obtain

5. Fixed Weight Beamforming (7)

This equation is an eigenvector equation with SIR being the eigenvalues.The maximum SIR (SIR_{max}) is equal to the largest eigenvalue λ_{max} for the Hermitian matrix $\mathbf{R}^{-1}{}_{uu}\mathbf{R}_{ss}$. The eigenvector associated with the largest eigenvalue is the optimum weight vector $\mathbf{w}_{opt}$. Thus

$$\mathbf{R}_{uu}^{-1}\mathbf{R}_{ss}\mathbf{w}_{SIR} = \lambda_{\max}\mathbf{w}_{opt} = SIR_{\max}\mathbf{w}_{SIR}$$

The final optimized weight for max SIR is

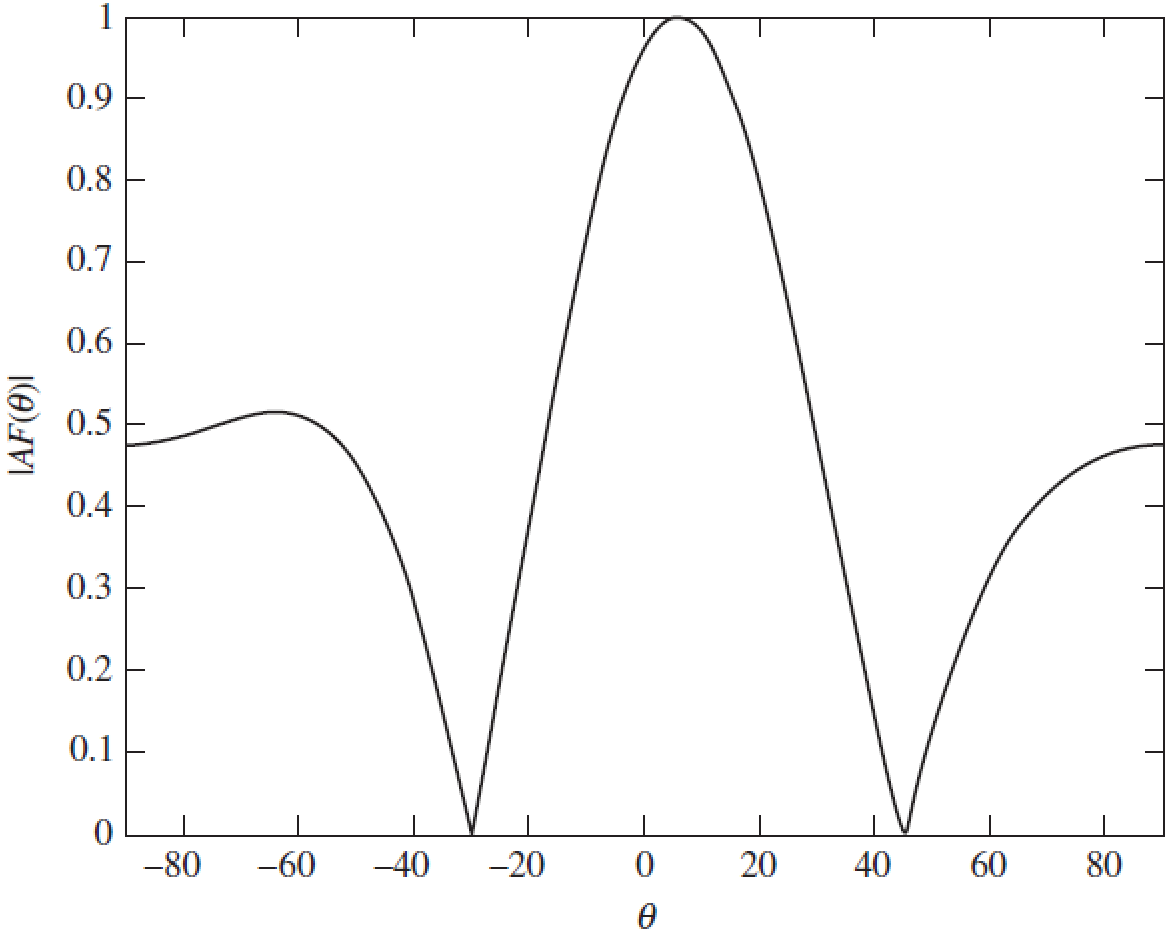
$$\mathbf{w}_{SIR} = \beta\mathbf{R}_{uu}^{-1}\mathbf{a}_0$$

where

$$\beta = \frac{E[|s|^2]}{SIR_{\max}}\mathbf{a}_0^H\mathbf{w}_{SIR}$$

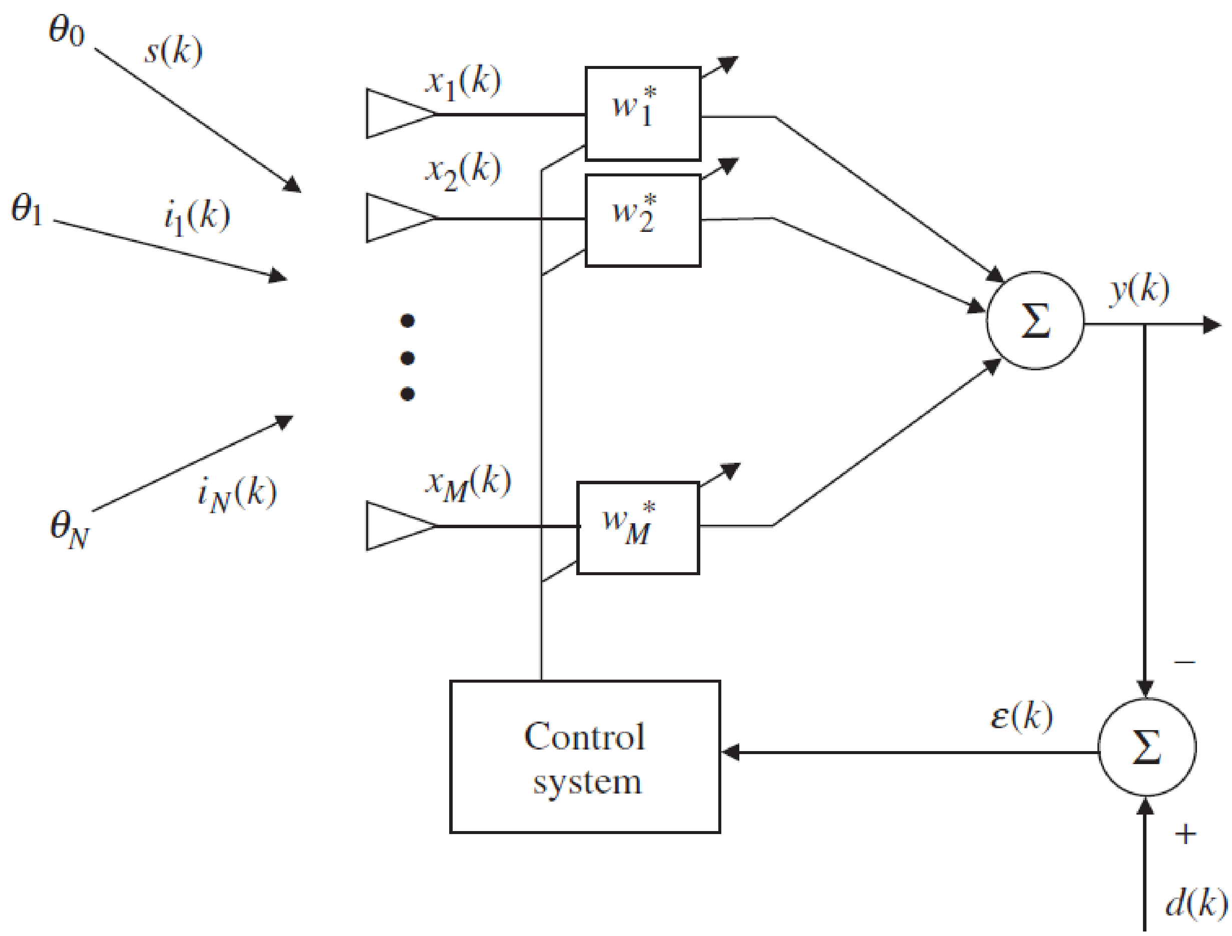
<u>Example</u>: M = 3-element array with spacing $d = 0.5\lambda$ has a noise variance $\sigma^2{}_n = 0.001$, a desired received signal arriving at $\theta_0 = 30^\circ$, and two interferers arriving at angles $\theta_1 = -30^\circ$ and $\theta_2 = 45^\circ$. Assume that the signal and interferer amplitudes are constant.

5. Fixed Weight Beamforming (8)



5. Fixed Weight Beamforming (9)

- **Minimum mean-square error**



5. Fixed Weight Beamforming (10)

The signal $d(k)$ is the reference signal. Preferably the reference signal is either identical to the desired signal $s(k)$ or it is highly correlated with $s(k)$ and uncorrelated with the interfering signals $i_n(k)$. If $s(k)$ is not distinctly different from the interfering signals, the minimum mean square technique will not work properly. The signal $\varepsilon(k)$ is the error signal such that

$$\varepsilon(k) = d(k) - \mathbf{w}^H \mathbf{x}(k)$$

The mean-square error is given by

$$E\left[|\varepsilon|^2\right] = E\left[|d|^2\right] - 2\mathbf{w}^H \mathbf{r} + \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}$$

where

$$\mathbf{r} = E\left[d^* \mathbf{x}\right] = E\left[d^* (\mathbf{x}_s + \mathbf{x}_i + \mathbf{n})\right]$$

$$\mathbf{R}_{xx} = E\left[\mathbf{x}\mathbf{x}^H\right] = \mathbf{R}_{ss} + \mathbf{R}_{uu}$$

$$\mathbf{R}_{ss} = E\left[\mathbf{x}_s \mathbf{x}_s{}^H\right]$$

$$\mathbf{R}_{uu} = \mathbf{R}_{ii} + \mathbf{R}_{nn}$$

5. Fixed Weight Beamforming (11)

The optimum weights provide the minimum mean-square error, the optimum Wiener solution given as

$$\mathbf{w}_{MSE} = \mathbf{R}_{xx}^{-1}\mathbf{r}$$

If we allow the reference signal d to be equal to the desired signal s, and if s is uncorrelated with all interferers, we may simplify as

$$\mathbf{r} = E[s\mathbf{x}] = S\mathbf{a}_0$$

where

$$S = E\left[\left|s\right|^2\right]$$

The optimum weights can then be identified as

$$\mathbf{w}_{MSE} = S\mathbf{R}_{xx}^{-1}\mathbf{a}_0$$

5. Fixed Weight Beamforming (12)

- **<u>Minimum variance - MV</u>** (or **<u>minimum variance distortionless response - MVDR</u>**):

  The goal of the minimum variance method is to minimize the array output noise variance. The weighted array output is given by

  $$y = \mathbf{w}^H \mathbf{x} = \mathbf{w}^H \mathbf{a}_0 s + \mathbf{w}^H \mathbf{u}$$

  In order to ensure a distortionless response, we must also add the constraint that

  $$\mathbf{w}^H \mathbf{a}_0 = 1$$

  Finally, the minimum variance optimum weights can be obtained as

  $$\mathbf{w}_{MV} = \frac{\mathbf{R}_{uu}^{-1}\mathbf{a}_0}{\mathbf{a}_0^H \mathbf{R}_{uu}^{-1}\mathbf{a}_0}$$

  where

  $$\mathbf{R}_{uu} = \mathbf{R}_{ii} + \mathbf{R}_{nn}$$

  is the correlation matrix of unwanted signals and noise.

Chapter 6:
Linear Prediction

- Forward Linear Prediction.
- Backward Linear Prediction.
- Levinson-Durbin Algorithm.

6. Linear Prediction: Definitions (1)

- (*M*+1) samples of time series (Stationary DTSt process) :

$$u(n), u(n-1), \dots, u(n-M)$$

- Linear prediction of order *M*-**Forward Prediction:** Predicting (estimating) the future value $u(n)$ using the M samples $u(n\text{-}1)$, $u(n\text{-}2)$,..., $u(n\text{-}M)$

$$\hat{u}(n) = w_1^* u(n-1) + w_2^* u(n-2) + \dots + w_M^* u(n-M)$$
$$= \sum_{k=1}^{M} w_k^* u(n-k) = \mathbf{w}^H \mathbf{u}(n-1)$$

$\mathbf{u}(n\text{-}1)$: **tap inputs**

- Predictor vector (tap weight vector) of order *M*-**Forward Predictor:**

$$\mathbf{w} = \left[w_1, w_2, \dots, w_M\right]^T$$
$$\mathbf{a}_M = \left[1, -w_1, -w_2, \dots, -w_M\right]^T = \left[a_{M,0}, a_{M,1}, \dots, a_{M,M}\right]^T$$

6. Linear Prediction: Definitions (2)

- Prediction error of order M-**Forward Prediction Error:**

$$f_M(n) = u(n) - \hat{u}(n) = u(n) - \mathbf{w}^H \mathbf{u}(n-1) = \mathbf{a}_M^H \begin{bmatrix} u(n) \\ \mathbf{u}(n-1) \end{bmatrix} = \mathbf{a}_M^H \mathbf{u}(n)$$

$\Rightarrow$ Mean-squared prediction error: $P_M = E\left[\left|f_M(n)\right|^2\right]$

if tap inputs $u(n\text{-}1)$ have zero-mean, then P_M called **forward prediction error power.**

- Correlation matrix of tap inputs:

$$\mathbf{R} = E\left[\mathbf{u}(n-1)\mathbf{u}^H(n-1)\right] = \begin{bmatrix} r(0) & r(1) & \ldots & r(M-1) \\ r^*(1) & r(0) & \ldots & r(M-2) \\ \ldots & \ldots & \ldots & \ldots \\ r^*(M-1) & r^*(M-2) & \ldots & r(0) \end{bmatrix}$$

6. Linear Prediction: Definitions (3)

- Cross correlation vector between tap inputs $\mathbf{u}(n\text{-}1)$ and the future value $u(n)$:

$$\mathbf{r} = E\left[\mathbf{u}(n-1)u^*(n)\right] = \begin{bmatrix} r^*(1) \\ r^*(2) \\ \dots \\ r^*(M) \end{bmatrix} = \begin{bmatrix} r(-1) \\ r(-2) \\ \dots \\ r(-M) \end{bmatrix}$$

- Variance of $u(n)$ equals $r(0)$ since $u(n)$ has zero-mean
- Vector reversing (backward):

$$\mathbf{r}^B = \begin{bmatrix} r^*(M) \\ r^*(M-1) \\ \dots \\ r^*(1) \end{bmatrix}$$

6. LP: Optimal Forward Linear Prediction

- Optimal criterion:

$$J(\mathbf{w}) = E\left[\left|f_M(n)\right|^2\right] = E\left[\left|u(n) - \mathbf{w}^H \mathbf{u}(n-1)\right|^2\right]$$

$$J(\mathbf{a}_M) = E\left[\left|f_M(n)\right|^2\right] = E\left[\left|\mathbf{a}_M^H \begin{bmatrix} u(n) \\ \mathbf{u}(n-1) \end{bmatrix}\right|^2\right]$$

- Optimal solution: based on optimal Wiener filter design with:
  - output filter: $\mathbf{w}^H\mathbf{u}(n\text{-}1)$
  - desired signal: $d(n)=u(n)$
  - Therefore, **optimal predictor vector**: $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{r}$
  - Similar to Chapter 5, we obtain **forward prediction error power**: $P_M = J_{\min}(\mathbf{w}) = r(0) - \mathbf{r}^H \mathbf{w}_{opt}$

6. LP: Augmented Wiener-Hopf Equations (1)

- **Optimum predictor vector** and **optimal prediction error power** satisfy:

$$r(0) - \mathbf{r}^H \mathbf{w}_{opt} = P_M$$
$$\mathbf{R}\mathbf{w}_{opt} - \mathbf{r} = \mathbf{0}$$

or in matrix form: $\begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_{opt} \end{bmatrix} = \begin{bmatrix} P_M \\ \mathbf{0} \end{bmatrix}$

but $\mathbf{a}_M = \left[1, -\mathbf{w}_{opt}\right]^T$. Then, augmented Wiener-Hopf equations for **optimal forward prediction error filter** are:

$$\begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r^*(1) & r(0) & \dots & r(M-1) \\ \dots & \dots & \dots & \dots \\ r^*(M) & r^*(M-1) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_{M,0} \\ a_{M,1} \\ \dots \\ a_{M,M} \end{bmatrix} = \begin{bmatrix} P_M \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

When $\mathbf{R}_{M+1}$ is nonsingular and $a_{M,0}=1$, there are unique solutions $\mathbf{a}_M$ and P_M. See Example 2, p. 247, [1].

6. LP: Optimal Backward Linear Prediction (1)

- Linear prediction of order M-**Backward Prediction:** Predicting (estimating) the past value $u(n\text{-}M)$ using the M samples $u(n)$, $u(n\text{-}1)$,..., $u(n\text{-}M\text{+}1)$:

$$\hat{u}^b(n-M) = g_1^* u(n) + g_2^* u(n-1) + \ldots + g_M^* u(n-M+1)$$
$$= \sum_{k=1}^{M} g_k^* u(n-k+1) = \mathbf{g}^H \mathbf{u}(n)$$

where **g** is **backward predictor vector.**

- Prediction error of order M-**Backward Prediction Error:**

$$b_M(n) = u(n-M) - \hat{u}^b(n-M) = u(n-M) - \mathbf{g}^H \mathbf{u}(n)$$

- Optimum criterion

$$J^b(\mathbf{g}) = E\left[\left|b_M(n)\right|^2\right] = E\left[\left|u(n-M) - \mathbf{g}^H \mathbf{u}(n)\right|^2\right]$$

6. LP: Optimal Backward Linear Prediction (2)

- **Optimal backward predictor vector:**

$$\mathbf{g}_{opt} = \mathbf{R}^{-1}\mathbf{r}^{B} = \mathbf{w}_{opt}^{B}$$

- **Optimal backward prediction error power**:

$$P_M = J_{\min}^{b}(\mathbf{g}) = r(0) - (\mathbf{r}^{B})^{H}\mathbf{g}_{opt}$$

- The **optimal backward predictor filter** solution and the **optimal backward prediction error power** satisfy:

$$\begin{aligned} \mathbf{R}\mathbf{g}_{opt} - \mathbf{r}^{B} &= \mathbf{0} \\ r(0) - (\mathbf{r}^{B})^{H}\mathbf{g}_{opt} &= P_M \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} \mathbf{R} & \mathbf{r}^{B} \\ (\mathbf{r}^{B})^{H} & r(0) \end{bmatrix} \begin{bmatrix} -\mathbf{g}_{opt} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ P_M \end{bmatrix}$$

Let $\mathbf{c}_M = \begin{bmatrix} -\mathbf{g}_{opt} & 1 \end{bmatrix}^{T}$

6. LP: Optimal Backward Linear Prediction (3)

⇒ Finally, we obtain the augmented Wiener-Hopf equations for optimal backward prediction error filter:

$$\begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r^*(1) & r(0) & \dots & r(M-1) \\ \dots & \dots & \dots & \dots \\ r^*(M) & r^*(M-1) & \dots & r(0) \end{bmatrix} \begin{bmatrix} c_{M,0} \\ c_{M,1} \\ \dots \\ c_{M,M} \end{bmatrix} =$$

$$= \begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r^*(1) & r(0) & \dots & r(M-1) \\ \dots & \dots & \dots & \dots \\ r^*(M) & r^*(M-1) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_{M,M} \\ a_{M,M-1} \\ \dots \\ a_{M,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ P_M \end{bmatrix}$$

6. LP: Levinson-Durbin Algorithm (1)

- **Purpose:** Recursive method with computational efficiency for computing the prediction error filter vectors ($\mathbf{a}_M$, $\mathbf{c}_M$) and the prediction error power (P_M) by solving the augmented Wiener-Hopf equations.

- Using the solution of the augmented Wiener-Hopf equations for a prediction error filter of order m-1 to compute the corresponding solution for a prediction error filter of order m (m=1, 2, ..., M; M is the **final order** of the filter).

- All variables are with a subscript expressing the order of the predictor: $\mathbf{R}_m$, $\mathbf{r}_m$, $\mathbf{a}_m$, $\mathbf{w}_{opt,m}$

6. LP: Levinson-Durbin Algorithm (2)

- Some order recursive equations can be written:

$$\mathbf{r}_{m+1} = \left[r(1), \quad r(2), \quad \ldots \quad r(m), \quad r(m+1) \right]^T = \begin{bmatrix} \mathbf{r}_m \\ r(m+1) \end{bmatrix}$$

$$\mathbf{R}_{m+1} = \begin{bmatrix} r(0) & \mathbf{r}_m^H \\ \mathbf{r}_m & \mathbf{R}_m \end{bmatrix} = \begin{bmatrix} \mathbf{R}_m & \mathbf{r}_m^B \\ (\mathbf{r}_m^B)^H & r(0) \end{bmatrix}$$

See more in [1].

6. LP: Levinson-Durbin Algorithm (3)

Summary of First Form:

Given the values of the autocorrelation function (for the lags $k = 0, 1, \ldots M$) $r(0), r(1),\ldots, r(M)$. These values can be estimated from the input data $u(1), u(2), \ldots, u(N)$, where N is total length of the input data, $N>>M$; by **time average**:

$$r(k) = \frac{1}{N}\sum_{n=k+1}^{N} u(n)u^*(n-k)$$

1. Initialize $\Delta_0 = r(1)$, $P_0 = r(0)$
2. For $m=1,\ldots, M$

2.1 $\Gamma_m = -\dfrac{\Delta_{m-1}}{P_{m-1}}$: **Reflection coefficients**

2.2 $a_{m,0} = 1; \quad a_{m,k} = a_{m-1,k} + \Gamma_m a^*_{m-1,m-k} \quad , \quad k = 1,\ldots, m$

2.3 $\Delta_m = r(m+1) + \sum_{k=1}^{m} a_{m,k} r(m+1-k)$

2.4 $P_m = P_{m-1}(1-\Gamma_m^2)$

6. LP: Levinson-Durbin Algorithm (4)

Summary of Second Form:

Given $r(0)$ and $\Gamma_1, \Gamma_2, ..., \Gamma_M$

1. Initialize $P(0) = r(0)$

2. For $m=1, ..., M$

   2.1 $a_{m,0} = 1; \quad a_{m,k} = a_{m-1,k} + \Gamma_m a^*_{m-1,m-k} \;, \quad k = 1,..., m$

   2.2 $P_m = P_{m-1}(1 - \Gamma_m^2)$

See Example 2, p. 260, [1]

6. LP: Inverse Levinson-Durbin Algorithm

- **Inverse problem:** Given prediction error filter vector $a_{M,1}$, $a_{M,2}$,..., $a_{M,M}$, solve for the corresponding set of reflection coefficients Γ_1, Γ_2,..., Γ_M.
  - Starting with set of $\{a_{M,k}\}$
  - Using: $$a_{m-1,k} = \frac{a_{m,k} - a_{m,m}a_{m,m-k}}{1-(a_{m,m})^2}, \quad k = 1,\ldots,m$$
    with $m=M$, M-1, ..., 2; to compute the vectors of the corresponding prediction error filters of order M-1, M-2,..., 1.
  - Finally, using: $\Gamma_m = a_{m,m}, \quad m = M, M-1,\ldots,1$
    to determine the desired set of Γ_M, Γ_{M-1},..., Γ_1.

See Example 3, p. 261, [1]