Microphone-Array Signal Processing

José A. Apolinário Jr. and Marcello L. R. de Campos

{apolin},{mcampos}@ieee.org

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming
4. Adaptive Beamforming
5. DoA Estimation with Microphone Arrays

3. Optimal Beamforming

3.1 Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.

Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.
- We assume that the appropriate statistics are known.

Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.
- We assume that the appropriate statistics are known.
- Our objective of interest is to estimate the waveform of a plane-wave impinging on the array in the presence of noise and interfering signals.

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.
- We assume that the appropriate statistics are known.
- Our objective of interest is to estimate the waveform of a plane-wave impinging on the array in the presence of noise and interfering signals.
- Even if a particular beamformer developed in this chapter has good performance, it does not guarantee that its adaptive version (next chapter) will. However, if the performance is poor, it is unlikely that the adaptive version will be useful.

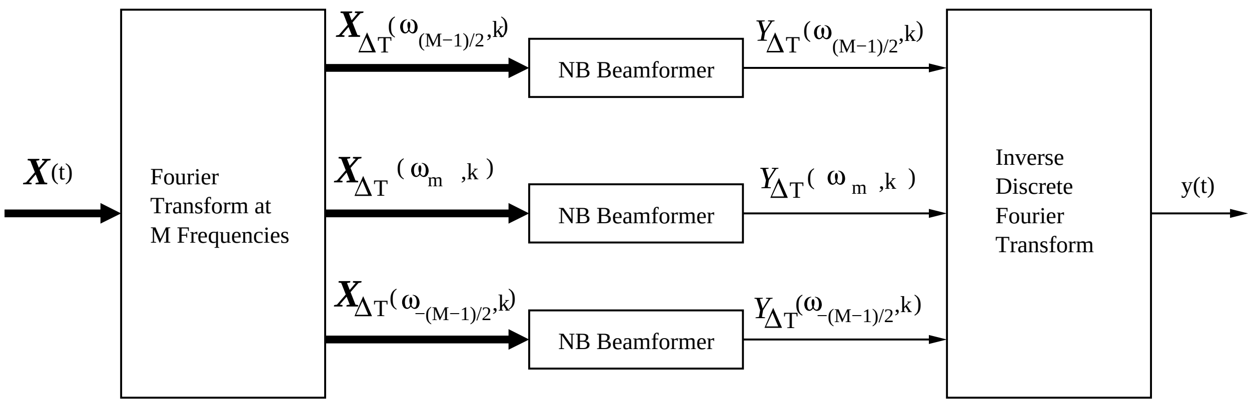
3.2 Optimal Beamformers

MVDR Beamformer

Snapshot model in the frequency domain:

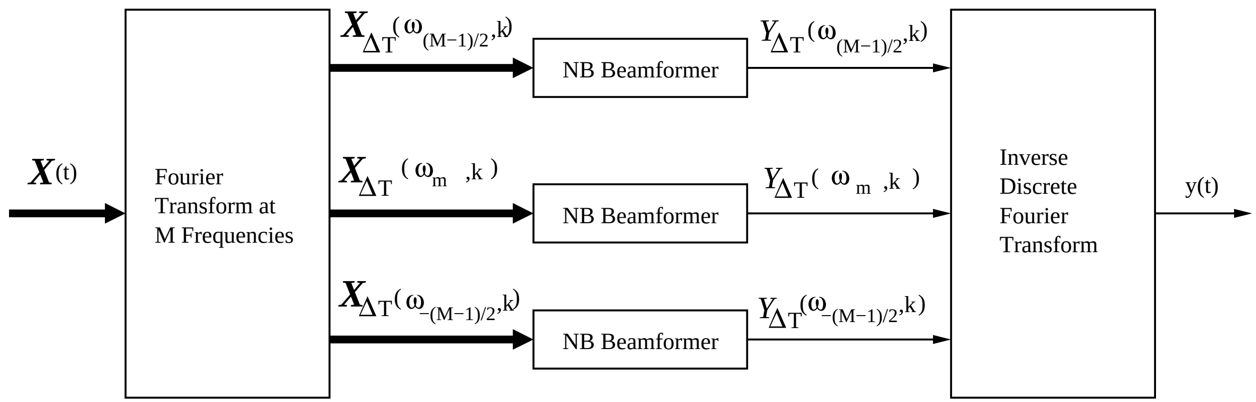
Snapshot model in the frequency domain:

- In many applications, we implement a beamforming in the frequency domain ($\omega_m = \omega_c + m\frac{2\pi}{\Delta T}$ and M varies from $-\frac{M-1}{2}$ to $\frac{M-1}{2}$ if odd and from $-\frac{M}{2}$ to $\frac{M}{2} - 1$ if even).



Snapshot model in the frequency domain:

- In many applications, we implement a beamforming in the frequency domain ($\omega_m = \omega_c + m\frac{2\pi}{\Delta T}$ and M varies from $-\frac{M-1}{2}$ to $\frac{M-1}{2}$ if odd and from $-\frac{M}{2}$ to $\frac{M}{2} - 1$ if even).



- In order to generate these vectors, divide the observation interval T in K disjoint intervals of duration ΔT: $(k-1)\Delta T \leq t < k\Delta T, k = 1, \cdots, K$.

- ΔT must be significantly greater than the propagation time across the array.

- ΔT must be significantly greater than the propagation time across the array.
- ΔT also depends on the bandwidth of the input signal.

- ΔT must be significantly greater than the propagation time across the array.
- ΔT also depends on the bandwidth of the input signal.
- Assume an input signal with BW B_s centered in f_c

- ΔT must be significantly greater than the propagation time across the array.
- ΔT also depends on the bandwidth of the input signal.

- Assume an input signal with BW B_s centered in f_c
- In order to develop the frequency-domain snapshot model for the case in which the desired signals and the interfering signals can de modeled as plane waves, we have two cases: desired signals are deterministic or samples of a random process.

MVDR Beamformer

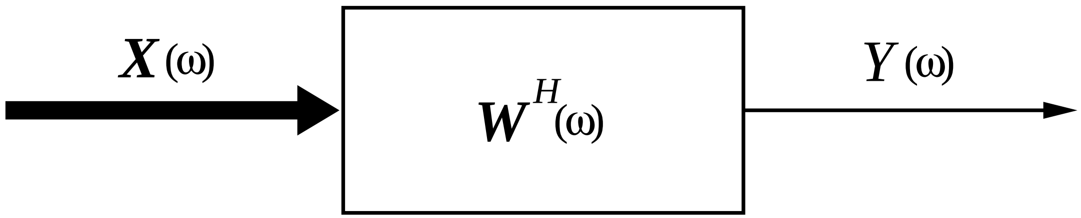
- Let’s assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.

- Let's assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.
- Frequency-domain snapshot consists of signal plus noise: $\boldsymbol{X}(\omega) = \boldsymbol{X}_s(\omega) + \boldsymbol{N}(\omega)$

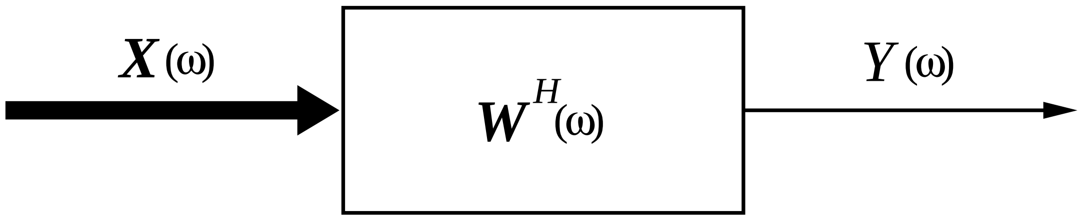
- Let's assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.
- Frequency-domain snapshot consists of signal plus noise: $\boldsymbol{X}(\omega) = \boldsymbol{X}_s(\omega) + \boldsymbol{N}(\omega)$
- The signal vector can be written as $\boldsymbol{X}_s(\omega) = F(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ where $F(\omega)$ is the frequency-domain snapshot of the source signal and $\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ is the array manifold vector for a plane-wave with wavenumber $\boldsymbol{k}_s$.

- Let's assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.
- Frequency-domain snapshot consists of signal plus noise: $\boldsymbol{X}(\omega) = \boldsymbol{X}_s(\omega) + \boldsymbol{N}(\omega)$
- The signal vector can be written as $\boldsymbol{X}_s(\omega) = F(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ where $F(\omega)$ is the frequency-domain snapshot of the source signal and $\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ is the array manifold vector for a plane-wave with wavenumber $\boldsymbol{k}_s$.
- The noise snapshot is a zero-mean random vector $\boldsymbol{N}(\omega)$ with spectral matrix given by $\boldsymbol{S}_{\boldsymbol{n}}(\omega) = \boldsymbol{S}_{\boldsymbol{c}}(\omega) + \sigma_\omega^2 \boldsymbol{I}$

- We process $\boldsymbol{X}(\omega)$ with the $1 \times N$ operator $\boldsymbol{W}^H(\omega)$:



- We process $\boldsymbol{X}(\omega)$ with the $1 \times N$ operator $\boldsymbol{W}^H(\omega)$:



- Distortionless criterion (in the absence of noise):

$$
\begin{aligned}
Y(\omega) &= F(\omega) \\
&= \boldsymbol{W}^H(\omega)\boldsymbol{X}_s(\omega) = F(\omega)\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) \\
\Longrightarrow & \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) = 1
\end{aligned}
$$

MVDR Beamformer

- In the presence of noise, we have:

$$Y(\omega) = F(\omega) + Y_n(\omega)$$

MVDR Beamformer

- In the presence of noise, we have:

$$Y(\omega) = F(\omega) + Y_n(\omega)$$

- The mean square of the output noise is:

$$E[|Y_n(\omega)|^2] = \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}(\omega)$$

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) = 1$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s) = 1$$

- Using the method of Lagrange multipliers, we define the following cost function to be minimized

$$F = \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}\omega$$

$$+\lambda\left[\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s) - 1\right] + \lambda^*\left[\boldsymbol{v}^H(\omega:\boldsymbol{k}_s)\boldsymbol{W}(\omega) - 1\right]$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) = 1$$

- Using the method of Lagrange multipliers, we define the following cost function to be minimized

$$F = \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}\omega$$
$$+ \lambda\left[\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) - 1\right] + \lambda^*\left[\boldsymbol{v}^H(\omega : \boldsymbol{k}_s)\boldsymbol{W}(\omega) - 1\right]$$

- ...and the result (suppressing ω and $\boldsymbol{k}_s$) is

$$\boldsymbol{W}^H_{mvdr} = \Lambda_s \boldsymbol{v}^H \boldsymbol{S}_n^{-1} \text{ where } \Lambda_s = \left[\boldsymbol{v}^H \boldsymbol{S}_n^{-1} \boldsymbol{v}\right]^{-1}$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) = 1$$

- Using the method of Lagrange multipliers, we define the following cost function to be minimized

$$F = \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}\omega$$
$$+ \lambda\left[\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) - 1\right] + \lambda^*\left[\boldsymbol{v}^H(\omega : \boldsymbol{k}_s)\boldsymbol{W}(\omega) - 1\right]$$

- ...and the result (suppressing ω and $\boldsymbol{k}_s$) is

$$\boldsymbol{W}^H_{mvdr} = \Lambda_s \boldsymbol{v}^H \boldsymbol{S}_n^{-1} \text{ where } \Lambda_s = \left[\boldsymbol{v}^H \boldsymbol{S}_n^{-1} \boldsymbol{v}\right]^{-1}$$

- This result is referred to as MVDR or *Capon* Beamformer.

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\nabla_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

- From the definition above, it is easy to show that: $\nabla_{\boldsymbol{w}}(\boldsymbol{b}^T\boldsymbol{w}) = \nabla_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{b}) = \boldsymbol{b}$

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

- From the definition above, it is easy to show that: $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{b}^T\boldsymbol{w}) = \boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{b}) = \boldsymbol{b}$

- Also $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}) = \boldsymbol{R}^T\boldsymbol{w} + \boldsymbol{R}\boldsymbol{w}$

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

- From the definition above, it is easy to show that: $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{b}^T\boldsymbol{w}) = \boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{b}) = \boldsymbol{b}$

- Also $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}) = \boldsymbol{R}^T\boldsymbol{w} + \boldsymbol{R}\boldsymbol{w}$

- which, when $\mathbf{R}$ is symmetric, leads to $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}) = 2\boldsymbol{R}\boldsymbol{w}$

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.
- The gradient becomes $\boldsymbol{\nabla}_{\mathbf{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial a_0} + j\frac{\partial \xi}{\partial b_0} \\ \frac{\partial \xi}{\partial a_1} + j\frac{\partial \xi}{\partial b_1} \\ \vdots \\ \frac{\partial \xi}{\partial a_{N-1}} + j\frac{\partial \xi}{\partial b_{N-1}} \end{bmatrix}$

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.
- The gradient becomes $\nabla_{\mathbf{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial a_0} + j\frac{\partial \xi}{\partial b_0} \\ \frac{\partial \xi}{\partial a_1} + j\frac{\partial \xi}{\partial b_1} \\ \vdots \\ \frac{\partial \xi}{\partial a_{N-1}} + j\frac{\partial \xi}{\partial b_{N-1}} \end{bmatrix}$
- $\cdots$ which corresponds to $\nabla_{\boldsymbol{w}}\xi = \nabla_{\boldsymbol{a}}\xi + j\nabla_{\boldsymbol{b}}\xi$

Constrained Optimal Filtering

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.
- The gradient becomes $\nabla_{\mathbf{w}}\xi = \begin{bmatrix} \frac{\partial\xi}{\partial a_0} + j\frac{\partial\xi}{\partial b_0} \\ \frac{\partial\xi}{\partial a_1} + j\frac{\partial\xi}{\partial b_1} \\ \vdots \\ \frac{\partial\xi}{\partial a_{N-1}} + j\frac{\partial\xi}{\partial b_{N-1}} \end{bmatrix}$
- $\cdots$ which corresponds to $\nabla_{\boldsymbol{w}}\xi = \nabla_{\boldsymbol{a}}\xi + j\nabla_{\boldsymbol{b}}\xi$
- Let us define the *derivative* $\frac{\partial}{\partial \mathbf{w}}$ (with respect to $\mathbf{w}$):

$$\frac{\partial}{\partial \mathbf{w}} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} - j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} - j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} - j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- Therefore, $\boldsymbol{\nabla}_{\mathbf{w}}\xi = \boldsymbol{\nabla}_{\mathbf{a}}\xi + j\boldsymbol{\nabla}_{\mathbf{b}}\xi$ is equivalent to $2\frac{\partial \xi}{\partial \mathbf{w}^*}$.

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial a_0} + j \frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j \frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j \frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- Therefore, $\boldsymbol{\nabla}_{\mathbf{w}} \xi = \boldsymbol{\nabla}_{\mathbf{a}} \xi + j \boldsymbol{\nabla}_{\mathbf{b}} \xi$ is equivalent to $2 \frac{\partial \xi}{\partial \mathbf{w}^*}$.
- The complex gradient may be slightly tricky if compared to the simple real gradient. For this reason, we exemplify the use of the complex gradient by calculating $\boldsymbol{\nabla}_{\mathbf{w}} E[|e(k)|^2]$.

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- Therefore, $\boldsymbol{\nabla}_{\mathbf{w}}\xi = \boldsymbol{\nabla}_{\mathbf{a}}\xi + j\boldsymbol{\nabla}_{\mathbf{b}}\xi$ is equivalent to $2\frac{\partial \xi}{\partial \mathbf{w}^*}$.

- The complex gradient may be slightly tricky if compared to the simple real gradient. For this reason, we exemplify the use of the complex gradient by calculating $\boldsymbol{\nabla}_{\mathbf{w}} E[|e(k)|^2]$.

- $\boldsymbol{\nabla}_{\mathbf{w}} E[e(k)e^*(k)] = E\{e^*(k)[\boldsymbol{\nabla}_{\mathbf{w}} e(k)] + e(k)[\boldsymbol{\nabla}_{\mathbf{w}} e^*(k)]\}$

- We compute each gradient ...

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}} e(k) =& \boldsymbol{\nabla}_{\mathbf{a}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] + j\boldsymbol{\nabla}_{\mathbf{b}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] \\ =& - \mathbf{x}(k) - \mathbf{x}(k) = -2\mathbf{x}(k)\end{aligned}$$

- We compute each gradient ...

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}} e(k) =& \boldsymbol{\nabla}_{\mathbf{a}}[d(k) - \mathbf{w}^H\mathbf{x}(k)] + j\boldsymbol{\nabla}_{\mathbf{b}}[d(k) - \mathbf{w}^H\mathbf{x}(k)] \\ =& -\mathbf{x}(k) - \mathbf{x}(k) = -2\mathbf{x}(k)\end{aligned}$$

- and

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}} e^*(k) =& \boldsymbol{\nabla}_{\mathbf{a}}[d^*(k) - \mathbf{w}^T\mathbf{x}^*(k)] + j\boldsymbol{\nabla}_{\mathbf{b}}[d^*(k) - \mathbf{w}^T\mathbf{x}^*(k)] \\ =& -\mathbf{x}^*(k) + \mathbf{x}^*(k) = \mathbf{0}\end{aligned}$$

Constrained Optimal Filtering

- We compute each gradient ...

$$\begin{aligned}\nabla_{\mathbf{w}} e(k) =& \nabla_{\mathbf{a}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] + j \nabla_{\mathbf{b}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] \\ =& -\mathbf{x}(k) - \mathbf{x}(k) = -2\mathbf{x}(k)\end{aligned}$$

- and

$$\begin{aligned}\nabla_{\mathbf{w}} e^*(k) =& \nabla_{\mathbf{a}}[d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)] + j \nabla_{\mathbf{b}}[d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)] \\ =& -\mathbf{x}^*(k) + \mathbf{x}^*(k) = \mathbf{0}\end{aligned}$$

- such that the final result is

$$\begin{aligned}\nabla_{\mathbf{w}} E[e(k) e^*(k)] =& -2E[e^*(k)\mathbf{x}(k)] \\ =& -2E[\mathbf{x}(k)[d(k) - \mathbf{w}^H \mathbf{x}(k)]^*\} \\ =& -2\underbrace{E[\mathbf{x}(k) d^*(k)]}_{\mathbf{p}} + 2\underbrace{E[\mathbf{x}(k)\mathbf{x}^H(k)]}_{\mathbf{R}} \mathbf{w}\end{aligned}$$

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.
- When a set of linear constraints involving the coefficient vector of an adaptive filter is imposed, the resulting problem (LCAF)—admitting the MSE as the objective function—can be stated as minimizing $E[|e(k)|^2]$ subject to $\mathbf{C}^H\mathbf{w} = \mathbf{f}$.

Constrained Optimal Filtering

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.
- When a set of linear constraints involving the coefficient vector of an adaptive filter is imposed, the resulting problem (LCAF)—admitting the MSE as the objective function—can be stated as minimizing $E[|e(k)|^2]$ subject to $\mathbf{C}^H\mathbf{w} = \mathbf{f}$.
- The output of the processor is $y(k) = \mathbf{w}^H\mathbf{x}(k)$.

Constrained Optimal Filtering

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.
- When a set of linear constraints involving the coefficient vector of an adaptive filter is imposed, the resulting problem (LCAF)—admitting the MSE as the objective function—can be stated as minimizing $E[|e(k)|^2]$ subject to $\mathbf{C}^H\mathbf{w} = \mathbf{f}$.
- The output of the processor is $y(k) = \mathbf{w}^H\mathbf{x}(k)$.
- It is worth mentioning that the most general case corresponds to having a reference signal, $d(k)$. It is, however, usual to have no reference signal as in Linearly-Constrained Minimum-Variance (LCMV) applications. In LCMV, if $\mathbf{f} = 1$, the system is often referred to as Minimum-Variance Distortionless Response (MVDR).

- Using Lagrange multipliers, we form

$$\xi(k) = E[e(k)e^*(k)] + \boldsymbol{\mathcal{L}}_R^T Re[\mathbf{C}^H\mathbf{w} - \mathbf{f}] + \boldsymbol{\mathcal{L}}_I^T Im[\mathbf{C}^H\mathbf{w} - \mathbf{f}]$$

Constrained Optimal Filtering

- Using Lagrange multipliers, we form
$\xi(k) = E[e(k)e^*(k)] + \boldsymbol{\mathcal{L}}_R^T Re[\mathbf{C}^H\mathbf{w} - \mathbf{f}] + \boldsymbol{\mathcal{L}}_I^T Im[\mathbf{C}^H\mathbf{w} - \mathbf{f}]$
- We can also represent the above expression with a complex $\boldsymbol{\mathcal{L}}$ given by $\boldsymbol{\mathcal{L}}_R + j\boldsymbol{\mathcal{L}}_I$ such that

$$\begin{aligned}
\xi(k) &= E[e(k)e^*(k)] + Re[\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f})] \\
&= E[e(k)e^*(k)] + \frac{1}{2}\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \frac{1}{2}\boldsymbol{\mathcal{L}}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*)
\end{aligned}$$

Constrained Optimal Filtering

- Using Lagrange multipliers, we form

$$\xi(k) = E[e(k)e^*(k)] + \boldsymbol{\mathcal{L}}_R^T Re[\mathbf{C}^H\mathbf{w} - \mathbf{f}] + \boldsymbol{\mathcal{L}}_I^T Im[\mathbf{C}^H\mathbf{w} - \mathbf{f}]$$

- We can also represent the above expression with a complex $\boldsymbol{\mathcal{L}}$ given by $\boldsymbol{\mathcal{L}}_R + j\boldsymbol{\mathcal{L}}_I$ such that

$$\begin{aligned}\xi(k) &= E[e(k)e^*(k)] + Re[\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f})] \\ &= E[e(k)e^*(k)] + \frac{1}{2}\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \frac{1}{2}\boldsymbol{\mathcal{L}}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*)\end{aligned}$$

- Noting that $e(k) = d(k) - \mathbf{w}^H\mathbf{x}(k)$, we compute:

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}}\xi(k) &= \boldsymbol{\nabla}_{\mathbf{w}}\left\{E[e(k)e^*(k)] + \frac{1}{2}\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \frac{1}{2}\boldsymbol{\mathcal{L}}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*)\right\} \\ &= E[-2\mathbf{x}(k)e^*(k)] + \mathbf{0} + \mathbf{C}\boldsymbol{\mathcal{L}} \\ &= -2E[\mathbf{x}(k)d^*(k)] + 2E[\mathbf{x}(k)\mathbf{x}^H(k)]\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}}\end{aligned}$$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$
- Which leads to $\mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}(2\mathbf{p} - \mathbf{C}\boldsymbol{\mathcal{L}})$

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$
- Which leads to $\mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}(2\mathbf{p} - \mathbf{C}\boldsymbol{\mathcal{L}})$
- If we pre-multiply the previous expression by $\mathbf{C}^H$ and use $\mathbf{C}^H\mathbf{w} = \mathbf{f}$, we find $\boldsymbol{\mathcal{L}}$:
  $\boldsymbol{\mathcal{L}} = 2(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{p} - \mathbf{f})$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$

- Which leads to $\mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}(2\mathbf{p} - \mathbf{C}\boldsymbol{\mathcal{L}})$

- If we pre-multiply the previous expression by $\mathbf{C}^H$ and use $\mathbf{C}^H\mathbf{w} = \mathbf{f}$, we find $\boldsymbol{\mathcal{L}}$:
$\boldsymbol{\mathcal{L}} = 2(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{p} - \mathbf{f})$

- By replacing $\boldsymbol{\mathcal{L}}$, we obtain the *Wiener solution* for the linearly constrained adaptive filter:
$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$

- The optimal solution for LCAF:
  $$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$$

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$
- Yet with $d(k) = 0$ but $f = 1$ (MVDR)
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$
- Yet with $d(k) = 0$ but $f = 1$ (MVDR)
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}$
- For this case, $d(k) = 0$, the cost function is termed minimum output energy (MOE) and is given by
  $E[|e(k)|^2] = \mathbf{w}^H\mathbf{R}\mathbf{w}$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$
- Yet with $d(k) = 0$ but $f = 1$ (MVDR)
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}$
- For this case, $d(k) = 0$, the cost function is termed minimum output energy (MOE) and is given by
  $E[|e(k)|^2] = \mathbf{w}^H\mathbf{R}\mathbf{w}$
- Also note that in case we do not have constraints ($\mathbf{C}$ and $\mathbf{f}$ are nulls), the optimal solution above becomes the *unconstrained* Wiener solution $\mathbf{R}^{-1}\mathbf{p}$.

We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that

$$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$

We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that
$$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}]\begin{bmatrix}\bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L\end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$
- Matrix $\mathbf{B}$ is usually called the *Blocking Matrix* and we recall that $\mathbf{C}^H\mathbf{w} = \mathbf{g}$ such that
$\mathbf{C}^H\mathbf{w} = \mathbf{C}^H\mathbf{C}\bar{\mathbf{w}}_U - \mathbf{C}^H\mathbf{B}\bar{\mathbf{w}}_L = \mathbf{g}$.

We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that
  $$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$
- Matrix $\mathbf{B}$ is usually called the *Blocking Matrix* and we recall that $\mathbf{C}^H\mathbf{w} = \mathbf{g}$ such that
  $\mathbf{C}^H\mathbf{w} = \mathbf{C}^H\mathbf{C}\bar{\mathbf{w}}_U - \mathbf{C}^H\mathbf{B}\bar{\mathbf{w}}_L = \mathbf{g}$.
- If we impose the condition $\mathbf{B}^H\mathbf{C} = \mathbf{0}$ or, equivalently, $\mathbf{C}^H\mathbf{B} = \mathbf{0}$, we will have $\bar{\mathbf{w}}_U = (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g}$.

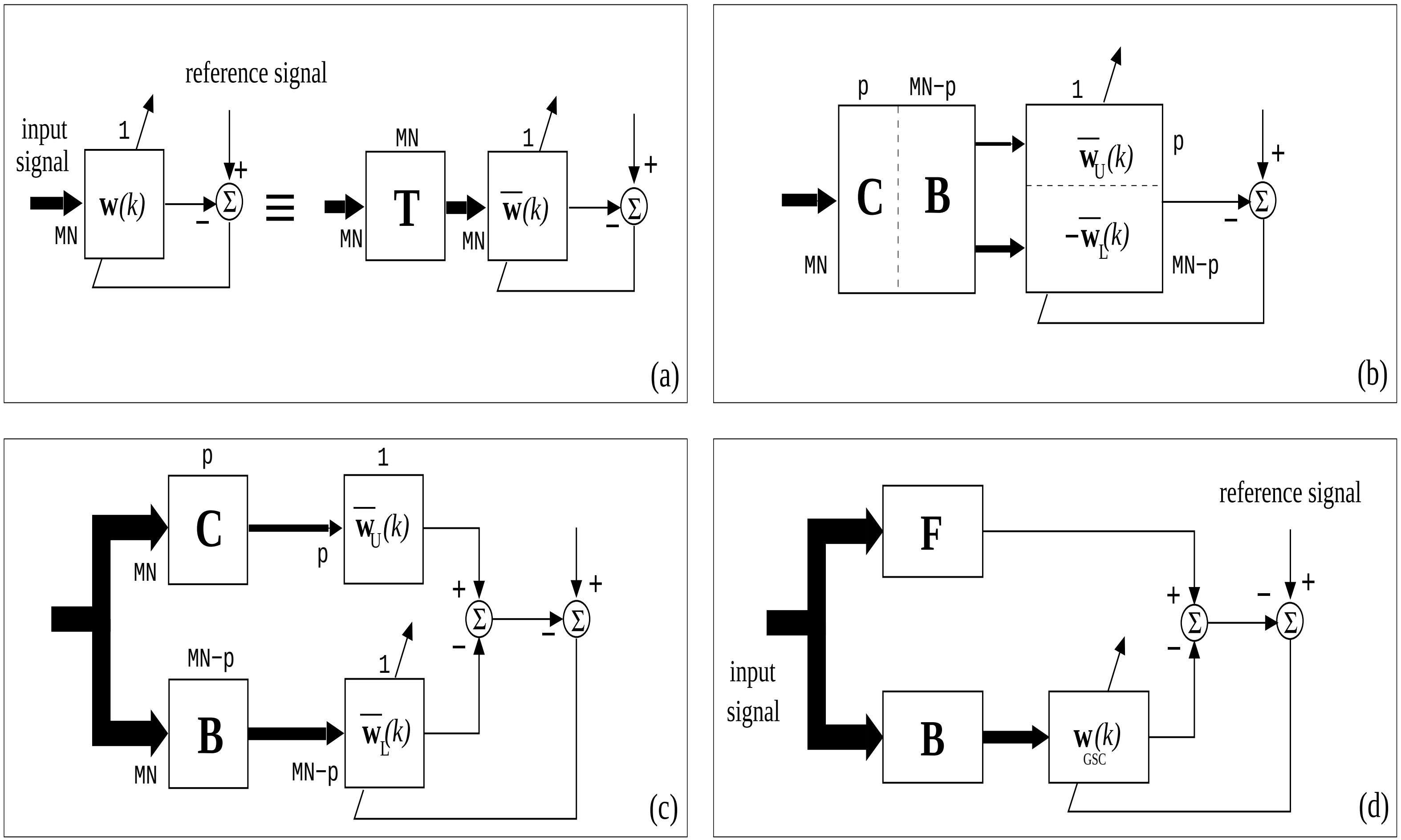
We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that

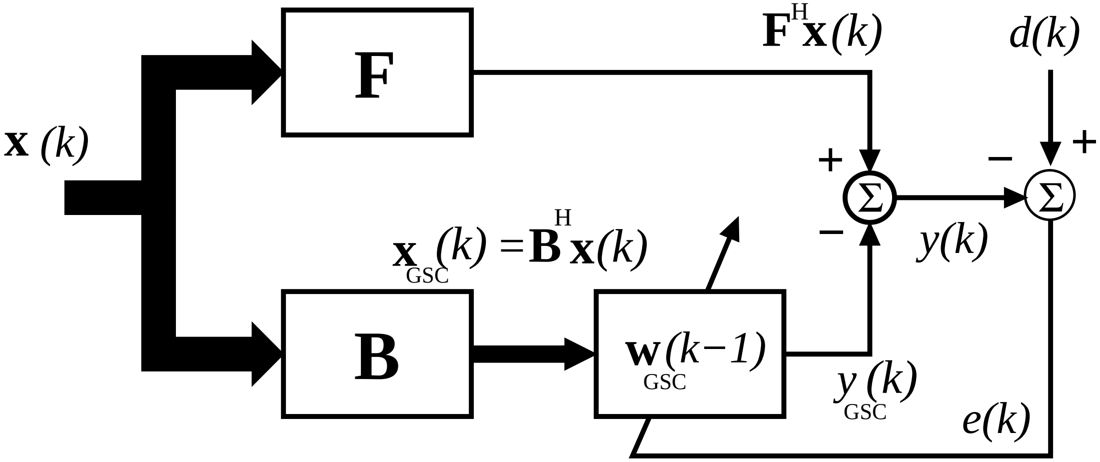
$$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$

- Matrix $\mathbf{B}$ is usually called the *Blocking Matrix* and we recall that $\mathbf{C}^H\mathbf{w} = \mathbf{g}$ such that $\mathbf{C}^H\mathbf{w} = \mathbf{C}^H\mathbf{C}\bar{\mathbf{w}}_U - \mathbf{C}^H\mathbf{B}\bar{\mathbf{w}}_L = \mathbf{g}$.
- If we impose the condition $\mathbf{B}^H\mathbf{C} = \mathbf{0}$ or, equivalently, $\mathbf{C}^H\mathbf{B} = \mathbf{0}$, we will have $\bar{\mathbf{w}}_U = (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g}$.
- $\bar{\mathbf{w}}_U$ is fixed and termed the quiescent weight vector; the minimization process will be carried out only in the lower part, also designated $\mathbf{w}_{GSC} = \bar{\mathbf{w}}_L$.

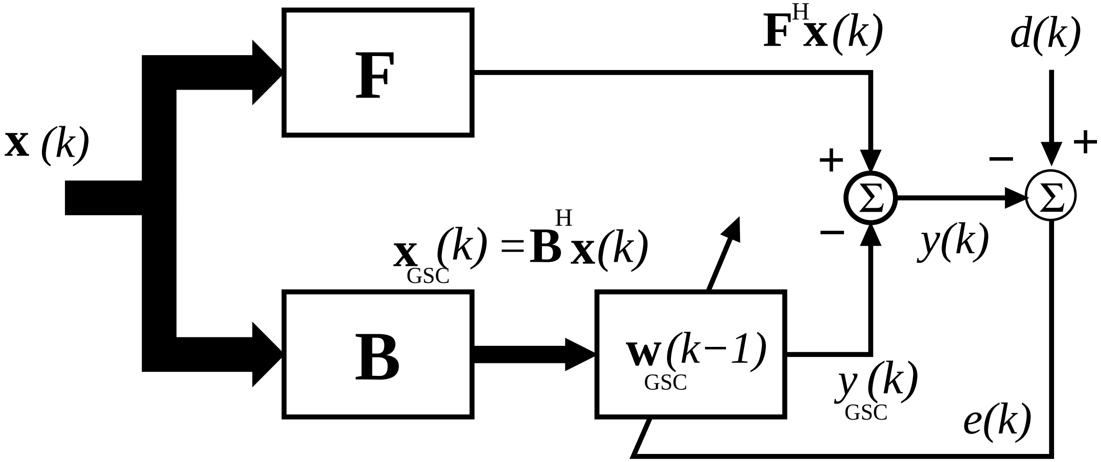
- It is shown below how to split the transformation matrix into two parts: a fixed path and an *adaptive* path.



- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).

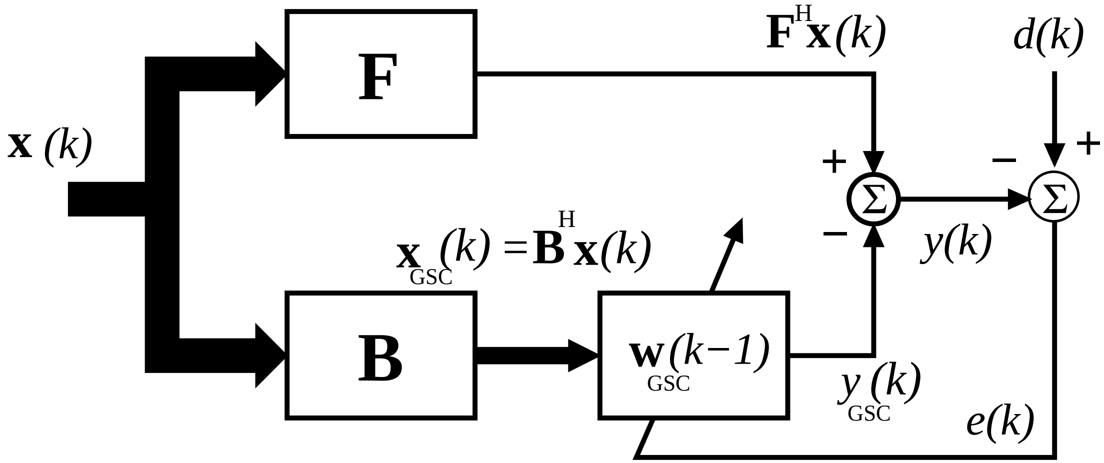


- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



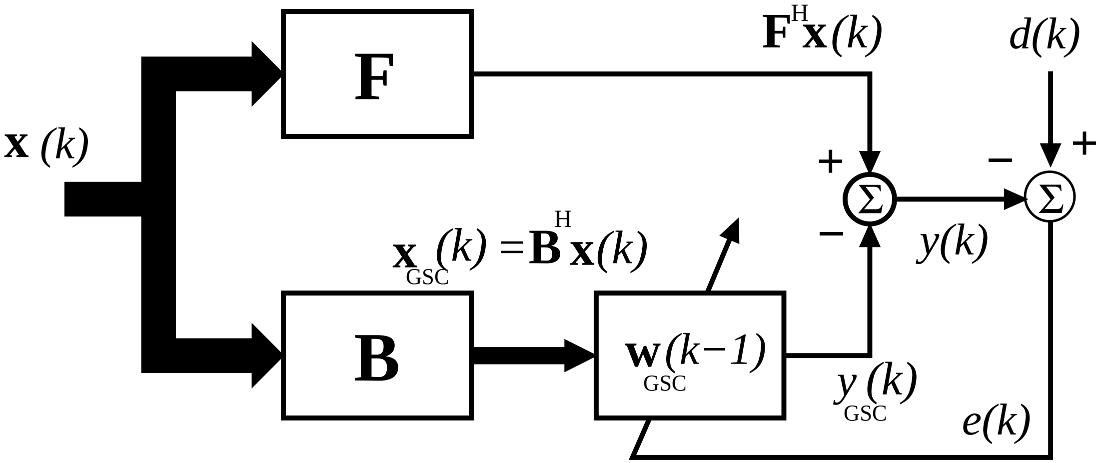
- It is always possible to have the overall equivalent coefficient vector which is given by $\mathbf{w} = \mathbf{F} - \mathbf{B}\mathbf{w}_{GSC}$.

- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



- It is always possible to have the overall equivalent coefficient vector which is given by $\mathbf{w} = \mathbf{F} - \mathbf{B}\mathbf{w}_{GSC}$.
- If we pre-multiply last equation by $\mathbf{B}^H$ and isolate $\mathbf{w}_{GSC}$, we find $\mathbf{w}_{GSC} = -(\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\mathbf{w}$.

- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



- It is always possible to have the overall equivalent coefficient vector which is given by $\mathbf{w} = \mathbf{F} - \mathbf{B}\mathbf{w}_{GSC}$.
- If we pre-multiply last equation by $\mathbf{B}^H$ and isolate $\mathbf{w}_{GSC}$, we find $\mathbf{w}_{GSC} = -(\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\mathbf{w}$.
- Knowing that $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ and that $\mathbf{T}^H\mathbf{T} = \boldsymbol{I}$, it follows that $\boldsymbol{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H = \mathbf{B}(\mathbf{B}^H\mathbf{B})\mathbf{B}^H$.

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$
- From the figure, it is clear that:
$\mathbf{R}_{GSC} = E[\mathbf{x}_{GSC}\mathbf{x}_{GSC}^H] = E[\mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{B}] = \mathbf{B}^H\mathbf{R}\mathbf{B}$

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$
- From the figure, it is clear that: $\mathbf{R}_{GSC} = E[\mathbf{x}_{GSC}\mathbf{x}_{GSC}^H] = E[\mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{B}] = \mathbf{B}^H\mathbf{R}\mathbf{B}$
- The cross-correlation vector is given as:

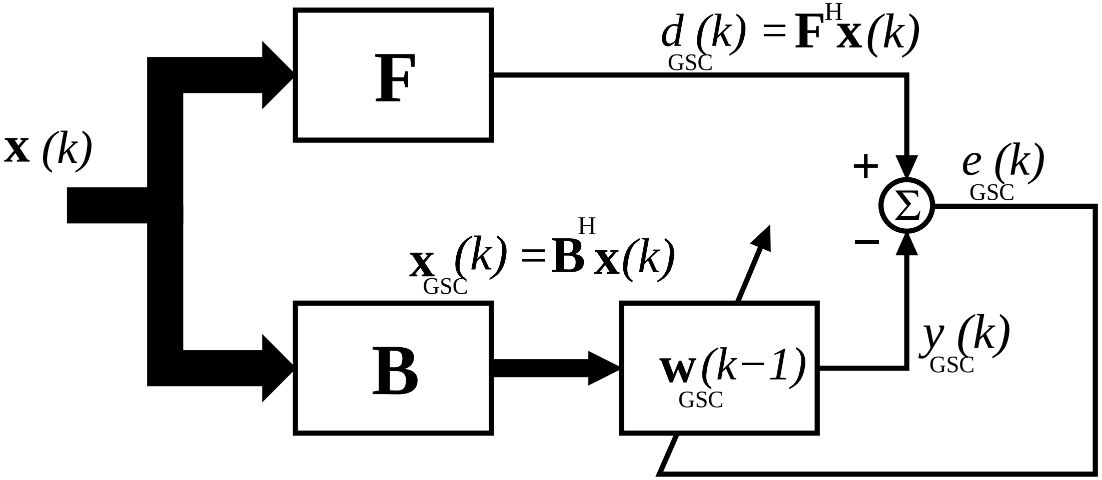
$$
\begin{aligned}
\mathbf{p}_{GSC} &= E[d_{GSC}^*\mathbf{x}_{GSC}] \\
&= E\{[\mathbf{F}^H\mathbf{x} - d]^*[\mathbf{B}^H\mathbf{x}]\} \\
&= E[-\mathbf{B}^H d^*\mathbf{x} + \mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{F}] \\
&= -\mathbf{B}^H\mathbf{p} + \mathbf{B}^H\mathbf{R}\mathbf{F}
\end{aligned}
$$

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$
- From the figure, it is clear that: $\mathbf{R}_{GSC} = E[\mathbf{x}_{GSC}\mathbf{x}_{GSC}^H] = E[\mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{B}] = \mathbf{B}^H\mathbf{R}\mathbf{B}$
- The cross-correlation vector is given as:

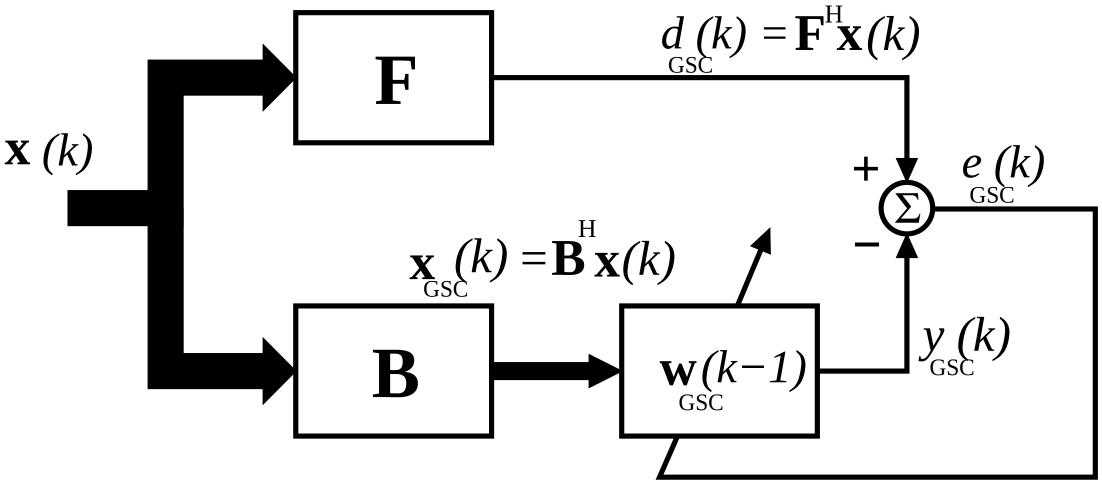
$$\begin{aligned}\mathbf{p}_{GSC} &= E[d_{GSC}^*\mathbf{x}_{GSC}] \\ &= E\{[\mathbf{F}^H\mathbf{x} - d]^*[\mathbf{B}^H\mathbf{x}]\} \\ &= E[-\mathbf{B}^H d^*\mathbf{x} + \mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{F}] \\ &= -\mathbf{B}^H\mathbf{p} + \mathbf{B}^H\mathbf{R}\mathbf{F}\end{aligned}$$

- $\cdots$ and $\mathbf{w}_{GSC-OPT} = (\mathbf{B}^H\mathbf{R}\mathbf{B})^{-1}(-\mathbf{B}^H\mathbf{p} + \mathbf{B}^H\mathbf{R}\mathbf{F})$

- A common case is when $d(k) = 0$:

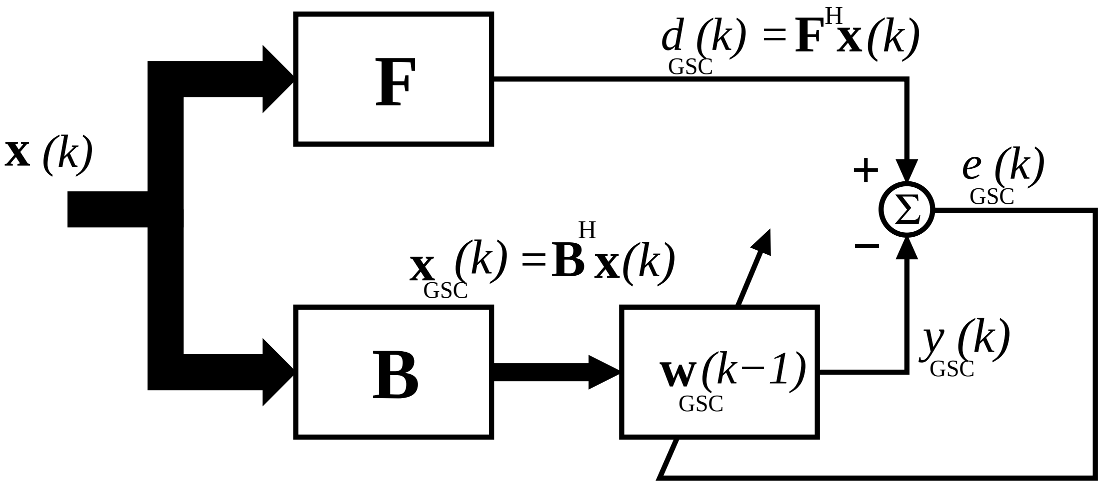


- A common case is when $d(k) = 0$:



- We have dropped the negative sign that should exist according to the notation used. Although we define $e_{GSC}(k) = -e(k)$, the inversion of the sign in the error signal actually results in the same results because the error function is always based on the absolute value.

- A common case is when $d(k) = 0$:



- We have dropped the negative sign that should exist according to the notation used. Although we define $e_{GSC}(k) = -e(k)$, the inversion of the sign in the error signal actually results in the same results because the error function is always based on the absolute value.
- In this case, the optimum filter $\mathbf{w}_{OPT}$ is: $\mathbf{F} - \mathbf{B}\mathbf{w}_{GSC-OPT} = \mathbf{F} - \mathbf{B}(\mathbf{B}^H\mathbf{R}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{R}\mathbf{F} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$ (LCMV solution)

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.
- Since the only need for B is having its columns forming a basis orthogonal to the constraints, $\mathbf{B}^H\mathbf{C} = \mathbf{0}$, a myriad of options are possible.

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.
- Since the only need for B is having its columns forming a basis orthogonal to the constraints, $\mathbf{B}^H\mathbf{C} = \mathbf{0}$, a myriad of options are possible.
- Let us recall the paper by Griffiths and Jim where the term GSC was coined; let

$$\mathbf{C}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.
- Since the only need for $\mathbf{B}$ is having its columns forming a basis orthogonal to the constraints, $\mathbf{B}^H\mathbf{C} = \mathbf{0}$, a myriad of options are possible.
- Let us recall the paper by Griffiths and Jim where the term GSC was coined; let

$$\mathbf{C}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- With simple constraint matrices, simple blocking matrices satisfying $\mathbf{B}^T\mathbf{C} = \mathbf{0}$ are possible.

- For this particular example, the paper presents two possibilities. The first one (orthogonal) is:

- For this particular example, the paper presents two possibilities. The first one (orthogonal) is:

$$\mathbf{B}_1^T =$$

$$\begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}$$

- And the second possibility (non-orthogonal) is:

- And the second possibility (non-orthogonal) is:

$$\mathbf{B}_2^T =$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

- **SVD**: the blocking matrix can be produced with the following Matlab command lines,

```
[U,S,V]=svd(C);
B3=U(:,p+1:M*N); % p=N in this case
```

- **SVD**: the blocking matrix can be produced with the following Matlab command lines,

```
[U,S,V]=svd(C);
B3=U(:,p+1:M*N); % p=N in this case
```

$\mathbf{B}_3^T$ is given by:

$$\begin{bmatrix} -0.50 & -0.17 & -0.17 & 0.83 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & -0.42 & 0.08 & 0.08 & 0.75 & -0.25 & -0.25 & -0.25 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & -0.42 & 0.08 & 0.08 & -0.25 & 0.75 & -0.25 & -0.25 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & -0.42 & 0.08 & 0.08 & -0.25 & -0.25 & 0.75 & -0.20 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & -0.42 & 0.08 & 0.08 & -0.25 & -0.25 & -0.25 & 0.75 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & 0.75 & -0.25 & -0.25 & -0.25 \\ 0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & 0.75 & -0.25 & -0.25 \\ 0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & -0.25 & 0.75 & -0.25 \\ 0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & -0.25 & -0.25 & 0.75 \end{bmatrix}$$

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- B_4 was identical to B_3 (SVD).

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- $\mathbf{B}_4$ was identical to $\mathbf{B}_3$ (SVD).
- Two other possibilities are: the one presented in [Tseng Griffiths 88] where a decomposition procedure is introduced in order to offer an effective implementation structure and the other one concerned to a narrowband BF implemented with GSC where $\mathbf{B}$ is combined with a wavelet transform [Chu Fang 99].

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- $\mathbf{B}_4$ was identical to $\mathbf{B}_3$ (SVD).
- Two other possibilities are: the one presented in [Tseng Griffiths 88] where a decomposition procedure is introduced in order to offer an effective implementation structure and the other one concerned to a narrowband BF implemented with GSC where $\mathbf{B}$ is combined with a wavelet transform [Chu Fang 99].
- Finally, a new efficient linearly constrained adaptive scheme which can also be visualized as a GSC structure can be found in [Campos&Werner&Apolinário IEEE-TSP Sept. 2002].