Adaptive and Array Signal Processing

Univ.-Prof. Dr. techn. Dr. h. c. Josef A. Nossek

October 12, 2015

Technische Universität München
Institute for Circuit Theory and Signal Processing
Univ.-Prof. Dr. techn. Josef A. Nossek

2nd Edition 2015

Contact: Josef.A.Nossek@tum.de
Editor: Univ.-Prof. Dr. techn. Josef A. Nossek,
Institute for Circuit Theory and Signal Processing, Technische Universität München
Internal reference number: TUM-LNS-TR-15-05
Print: Fachschaft Elektrotechnik und Informationstechnik e.V., München

Contents

1. Introduction and Motivation

A *filter* is an electronic circuit used to process signals, with the goal of removing undesired components in the signal and/or enhancing desired components in the signal. Technically speaking, filtering refers to the extraction of information about a desired quantity at time t by using data measured up to and including time t [1]. A filter can be classified according to its characteristics as either:

- passive or active,
- analog or digital,
- discrete-time (sampled) or continuous-time,
- linear or non-linear and
- infinite impulse response (IIR) or finite impulse response (FIR).

In this course we will focus on *digital* filters which are discrete-time by nature. A digital filter works by performing discrete mathematical operations on a sampled version of the signal.

We can further classify a digital filter as either a fixed filter or an adaptive filter. As the name implies, the parameters (coefficients) of a fixed filter cannot be adapted depending on the input signal. This is a viable approach if the statistics of the signal to be processed are known *a priori*. However, when this information is not known completely, an optimum fixed filter cannot be designed beforehand. In this scenario of unknown statistics, an adaptive filter can be employed since the filter *adapts* to the given environment, being able to filter a desired signal from the input signal with unknown statistics. The use of an adaptive filter offers an attractive solution to the filtering problem as it usually provides a significant improvement in performance over the use of a fixed filter designed by conventional methods [1]. At each iteration, the parameters of the filter are updated according to the filter parameters in the previous iteration, the input signal and some further information depending on the specific filter. If the scenario is stationary, successive iterations of such an algorithm converge to an optimum solution. In a non-stationary scenario, the algorithm is to some extent able to the track the time variations of the statistics of the input signal, under the assumption that the variations are sufficiently slow. As the name of this course implies, we will be specifically dealing with *adaptive digital* filters.

In general, an adaptive filter can be represented as depicted in Fig. 1, where $\mathbf{u}[n]$, $\mathbf{y}[n]$, $\mathbf{d}[n]$ and $\mathbf{e}[n]$ represent the discrete-time input signal, the discrete-time output signal, the desired signal and the error, respectively. The error $\mathbf{e}[n]$ denotes the difference between the actual output and the desired output. In addition, we have that $\mathbf{w}[n]$ is the weight vector which holds the set of the filter parameters that can be adapted to drive the error to zero. Furthermore, we have that the current state of the adaptive filter is denoted by $\mathbf{x}[n]$ and is stored in the memory of the filter. These signals can be real or complex valued scalars or vectors. However, the vectors $\mathbf{y}[n]$, $\mathbf{d}[n]$ and $\mathbf{e}[n]$ must have the same dimensions in order that the summation shown in Fig. 1 is consistent. The signals can be summarized as follows:

- $\mathbf{u}[n]$ = input signal of the adaptive filter
- $\mathbf{y}[n]$ = output signal of the adaptive filter
- $\mathbf{d}[n]$ = desired or reference signal for the output
- $\mathbf{e}[n]$ = error between actual output and desired signal,

with $\mathbf{w}[n]$ as the set of filter coefficients at time instant n.

Notation: The notation that will be employed in this script will be the following. The vectors will be represented as small boldface letters; meanwhile, the matrices will be given by capital boldface letters. Since the signals are discrete-time we have that n is an integer.

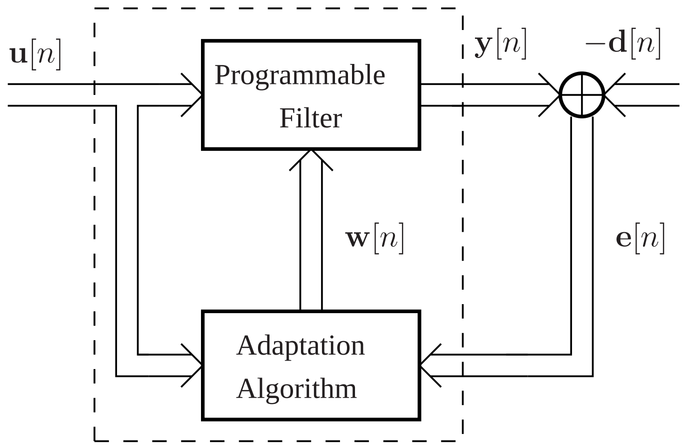


Fig. 1.1. Adaptive Filter

1.1 Application Areas

Due to the satisfactory performance of adaptive filters in a scenario with unknow statistics, adaptive filters have been successfully applied in diverse fields with different applications [1]. For each application we have an input vector $\mathbf{u}[n]$ and an output vector $\mathbf{y}[n]$ with which an estimation error $\mathbf{e}[n]$ with respect to a desired signal $\mathbf{d}[n]$ is computed. The computed error is in turn used to calculate the values of the adjustable parameters or coefficients $\mathbf{w}[n]$ of the adaptive filter. The main difference between the different applications where an adaptive filter can be applied is the way the reference or desired signal is extracted. In the following we take a look at some examples.

1.1.1 System Identification

In this application, the unknown plant is a system to be identified by the adaptive filter. Prior knowledge about the inner structure of the plant is of course desirable. Such knowledge would enable us to choose a proper model structure in our adaptive filter, the parameters of which are adapted to match the output of the adaptive filter to the output of the plant exciting both with the same input.

An example of system identification in communications is the problem of channel estimation. In such a case, the plant is the channel and with system identification we are then attempting to estimate the channel. The training sequence is given by $\mathbf{u}$ in Fig. 1.2, which is known at the receiver.

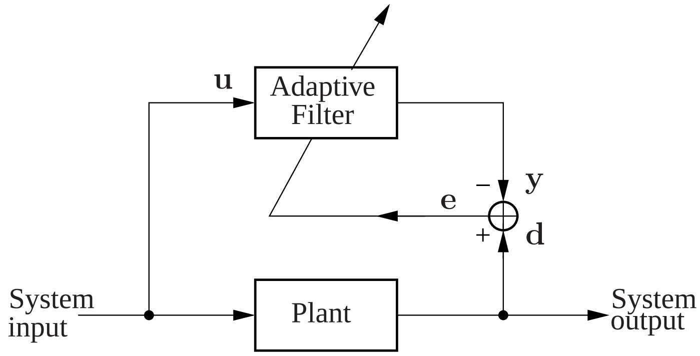


Fig. 1.2. System Identification

1.1.2 Inverse Modeling

In inverse modeling, the adaptive filter tries to undo the effect of the plant apart of some delay. The cascade of the plant and the adaptive filter is a distortionless system, i.e. the adaptive filter is inverting the plant.

Let us take again an example in communications. Once again we have that the plant is the channel but now by cascading the filter with the plant we are performing inverse modeling for this plant. This is commonly referred to as adaptive equalization in communications. As depicted in Fig. 1.3, the training sequence is the system input in this case.

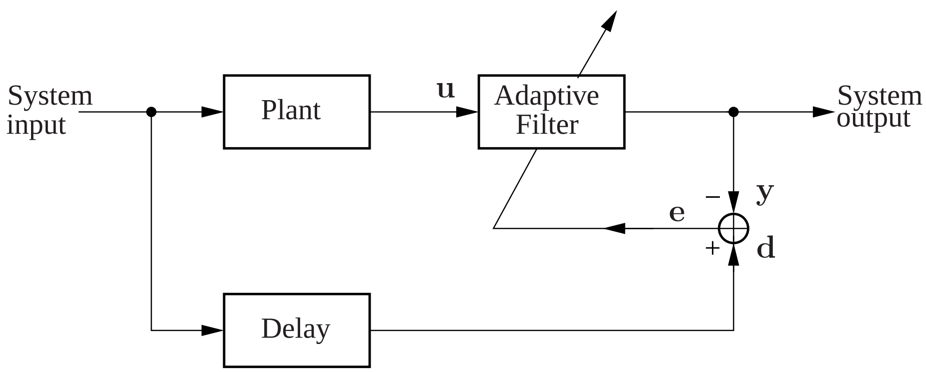


Fig. 1.3. Inverse Modeling

1.1.3 Prediction

Now we will take another focus on the application of adaptive filters. Let us assume that we are interested in providing the best prediction of a given sample based on previous samples. If there is correlation between the time samples one can use prediction. The predicted signal would be the system output 2 and the prediction error would be the system output 1 shown in Fig. 1.4. An example where prediction is done is encoding the speech in GSM.

1.1.4 Interference Cancellation

Another application of adaptive filters is the mitigation of unknown interference which usually corrupts our desired signal. The parameters of the adaptive filter are chosen such that the cancel-

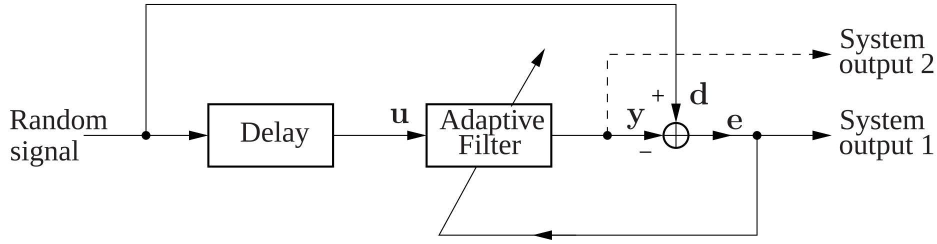


Fig. 1.4. Prediction

lation of the interference is optimized in some sense. In Fig. 1.5, we depict a general example for interference cancellation. The plant is now the path, through which the interference leaks into the primary signal of interest. The adaptive filter, if appropriately adjusted, compensates for this interference. The interference can also be an echo in a long distance telephone connection. Such a situation is depicted in Fig. 1.6, the boxes marked N are balancing impedances. Fig. 1.7 gives more details with an adaptive filter for echo compensation on the site of speaker A. Fig. 1.8 shows another form of interference cancellation, where the interference may be acoustic noise, which contaminates a voice signal picked up by a microphone (primary sensor). If we are able to pick up the noise only with a reference sensor, we could cancel the noise component in the voice signal.

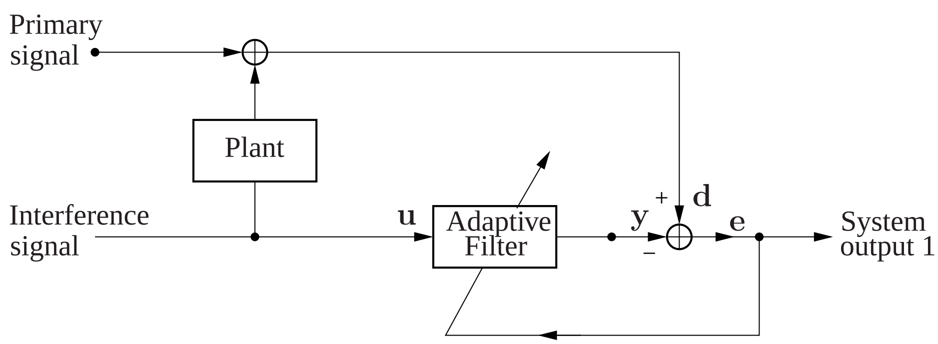


Fig. 1.5. Interference Cancellation

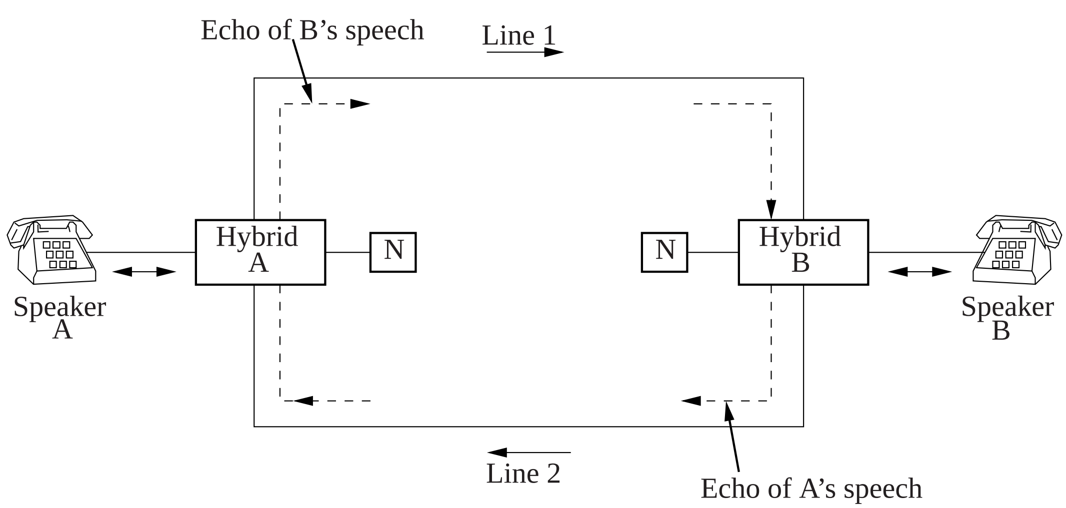


Fig. 1.6. Long-distance telephone circuit

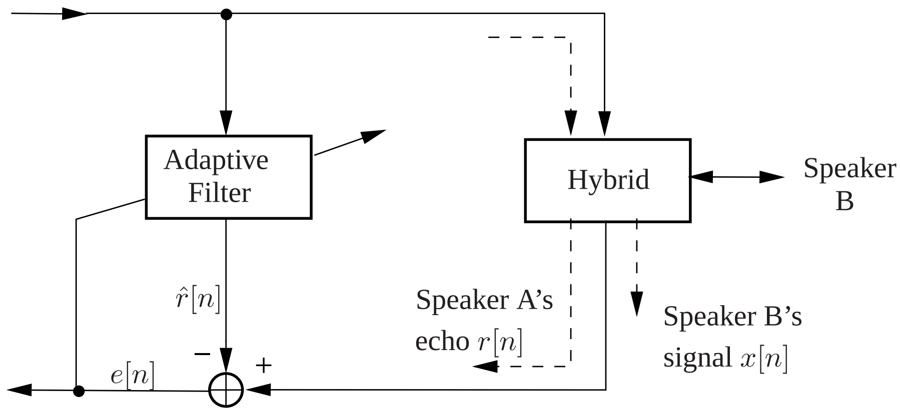


Fig. 1.7. Signal definitions for echo cancellation

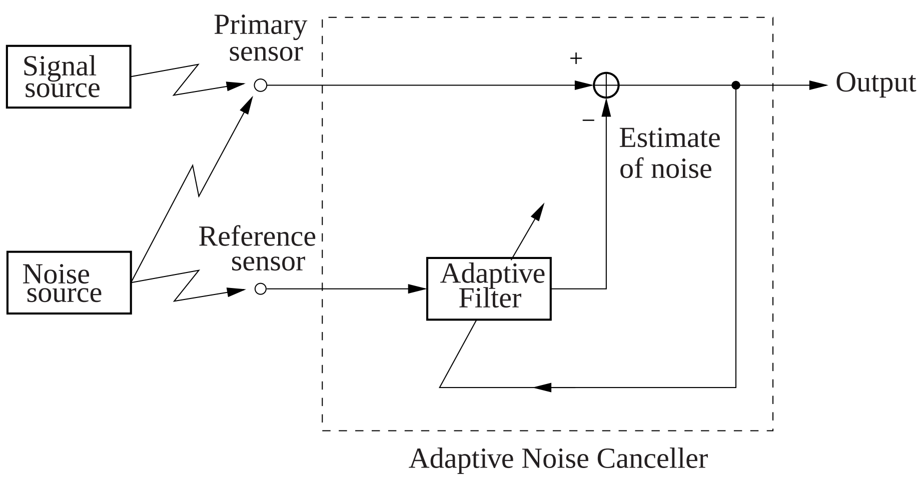


Fig. 1.8. Adaptive noise cancellation

1.2 Adaptive Equalization

As seen before, adaptive filters can be employed in order to perform adaptive equalization, which is an important topic in communications via time-variant channels, which we encounter e.g. in mobile communications. Fig. 1.9 gives an abstract block diagram of such an application.

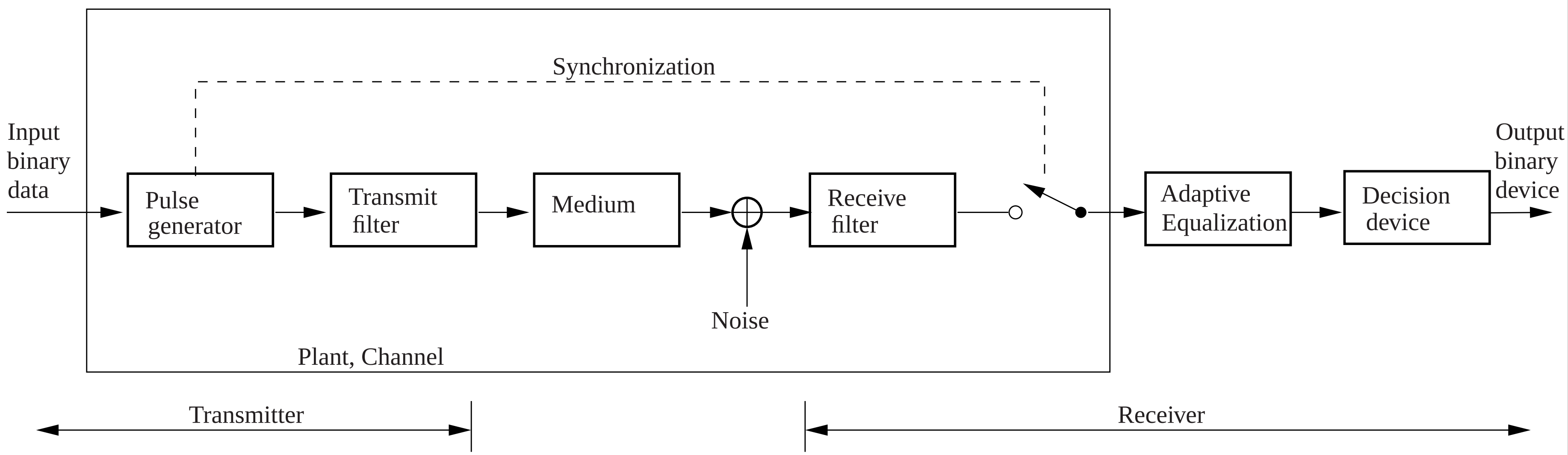


Fig. 1.9. Block Diagram of a Baseband Data Transmission System with Equalization

1.3 Single Channel Adaptive Equalization

Let us now consider the case of single channel temporal adaptive equalization. In Fig. 1.10 we have a closer look at the adaptive equalizer which is a linear, discrete-time filter with finite duration of its impulse response (FIR). Other names for these kinds of filters are transversal filter, non-recursive filter and moving average (MA) filter. As stated before the M filter coefficients are given by $w_0, w_1, \ldots, w_{M-1}$. The z^{-1} represents the delay in the shift registers and the $u[n]$,$y[n]$,$e[n]$ and $d[n]$ are the input signal, output signal, error estimate and desired signal at time n, respectively.

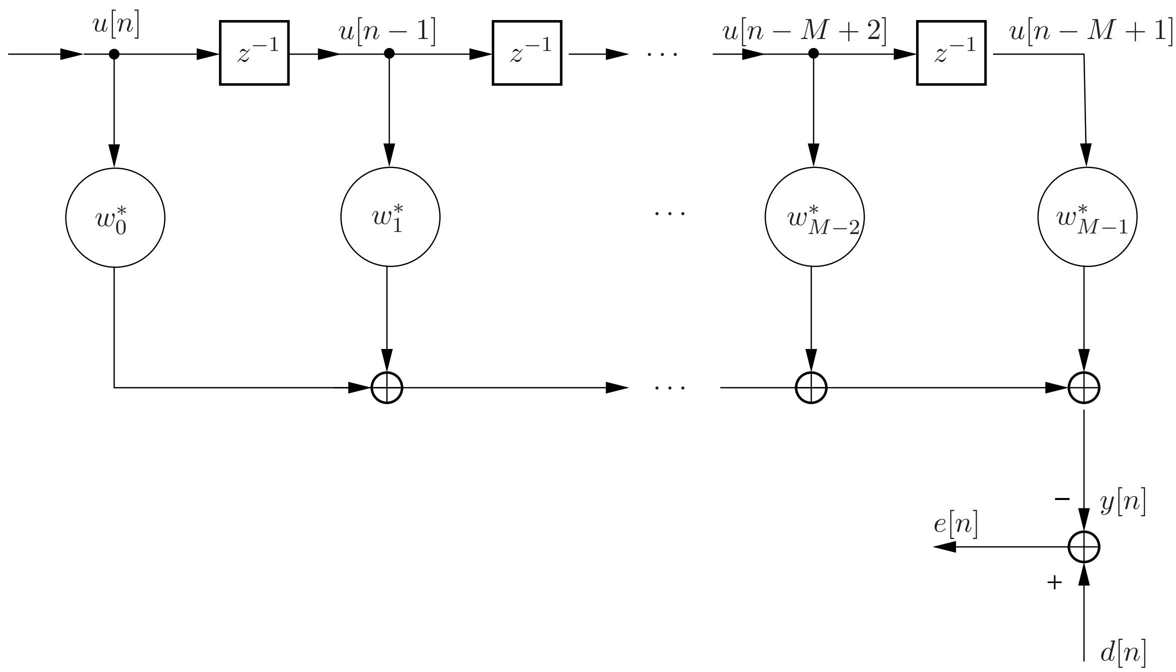


Fig. 1.10. Temporal Adaptive Equalization

First, we collect the signals present at the inputs of the various stages of the shift register into an M-dimensional vector:

$$\mathbf{u}[n] = \begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix} \in \mathbb{C}^M, \tag{1.1}$$

whereas the filter parameters are stacked in

$$\mathbf{w}^* = \begin{bmatrix} w_0^* \\ w_1^* \\ \vdots \\ w_{M-1}^* \end{bmatrix} \in \mathbb{C}^M. \tag{1.2}$$

For the filter output we can write

$$y[n] \quad = \quad \sum_{k=0}^{M-1} w_k^* u[n-k] = \mathbf{w}^{\mathrm{H}} \cdot \mathbf{u}[n], \tag{1.3}$$

and its complex conjugate is then given by

$$y^*[n] \quad = \quad \mathbf{u}^{\mathrm{H}}[n] \cdot \mathbf{w}. \tag{1.4}$$

Let us recall that $\mathbf{w}^{\mathrm{H}} = (\mathbf{w}^{\mathrm{T}})^* = (\mathbf{w}^*)^{\mathrm{T}} = [w_0^*, w_1^*, \ldots, w_{M-1}^*]$, where $(\bullet)^{\mathrm{H}}$ is the Hermitian operation, which is the complex conjugate and transposition of a matrix or vector.

Next, we collect $N+1$ output samples, given by (1.4), from time instant n to instant $n+N$:

$$\begin{aligned}
\mathbf{y}^*[n] \quad &= \begin{bmatrix} y^*[n] \\ y^*[n+1] \\ \vdots \\ y^*[n+N] \end{bmatrix} = \begin{bmatrix} u^*[n] & u^*[n-1] & \ldots & u^*[n-M+1] \\ u^*[n+1] & u^*[n] & \ldots & u^*[n-M+2] \\ \vdots & \vdots & \ddots & \vdots \\ u^*[n+N] & u^*[n+N-1] & \ldots & u^*[n+N-M+1] \end{bmatrix} \cdot \mathbf{w} \\
&= \begin{bmatrix} \mathbf{u}^{\mathrm{H}}[n] \\ \mathbf{u}^{\mathrm{H}}[n+1] \\ \vdots \\ \mathbf{u}^{\mathrm{H}}[n+N] \end{bmatrix} \cdot \mathbf{w} = \mathbf{U}^{\mathrm{H}} \cdot \mathbf{w},
\end{aligned} \tag{1.5}$$

where $\mathbf{U}^{\mathrm{H}} \in \mathbb{C}^{(N+1)\times M}$ with $\mathbf{U} = [\mathbf{u}[n], \mathbf{u}[n+1], \ldots, \mathbf{u}[n+N]]$. In addition, we have that $e[n] = d[n] - y[n]$ and that $e^*[n] = d^*[n] - y^*[n]$ and let us denote $\mathbf{e}[n]$ as a collection of $N+1$ error samples

$$\mathbf{e}[n] = \begin{bmatrix} e[n] \\ e[n+1] \\ \vdots \\ e[n+N] \end{bmatrix} \in \mathbb{C}^{N+1}. \tag{1.6}$$

Furthermore, let us collect $N+1$ samples of the desired signal

$$\mathbf{d}[n] = \begin{bmatrix} d[n] \\ d[n+1] \\ \vdots \\ d[n+N] \end{bmatrix} \in \mathbb{C}^{N+1}. \tag{1.7}$$

The problem to be solve can be posed in the following way:

$$\boxed{\text{Find a } \mathbf{w} \text{ such that } \|\mathbf{e}[n]\|_2^2 \text{ is minimal!}}$$

Depending on the value of $N+1$ (the number of equations, the number of input vectors) and M (the number of degrees of freedom in our weight vector), we have the following three possibilities:

- $N+1 = M$: If $N+1$ is equal to M, we can make the error zero, i.e. $\|\mathbf{e}[n]\|_2^2 = 0$ and $\mathbf{d}[n] = \mathbf{y}[n]$, and hence, $\mathbf{d}^*[n] = \mathbf{U}^{\mathrm{H}} \cdot \mathbf{w}$.
- $N+1 < M$: If $N+1$ is less than M, we can still make the error zero like the previous case, i.e. $\|\mathbf{e}[n]\|_2^2 = 0$. However, if $(N+1) < M$, the solution $\mathbf{w}$ is not uniquely determined since we have fewer equations than unknowns and we can impose additional restrictions on $\mathbf{w}$ (e.g. minimum norm $\|\mathbf{w}\|$) to arrive at a specific solution. For $M = N+1$ and rank$\{\mathbf{U}^{\mathrm{H}}\} = M$ we have an unique solution already.
- $N+1 > M$: For the usual case of $N+1 > M$, we have an overdetermined system of linear equations and in general, we will not have zero error! Nevertheless, we aim at minimizing $\|\mathbf{e}[n]\|_2^2$!

With $\mathbf{e}^* = \mathbf{d}^* - \mathbf{U}^{\mathrm{H}}\mathbf{w}$ we have that the squared norm of this error vector is

$$\begin{aligned} \|\mathbf{e}\|_2^2 &= \mathbf{e}^{\mathrm{H}} \cdot \mathbf{e} = (\mathbf{e}^*)^{\mathrm{H}} \cdot \mathbf{e}^* \\ &= (\mathbf{d}^* - \mathbf{U}^{\mathrm{H}}\mathbf{w})^{\mathrm{H}} \cdot (\mathbf{d}^* - \mathbf{U}^{\mathrm{H}}\mathbf{w}) \\ &= (\mathbf{d}^*)^{\mathrm{H}} \cdot \mathbf{d}^* - \mathbf{w}^{\mathrm{H}}\mathbf{U}\mathbf{d}^* - \mathbf{d}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{U}\mathbf{U}^{\mathrm{H}}\mathbf{w} \\ &= \|\mathbf{d}\|_2^2 - \mathbf{w}^{\mathrm{H}}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} \end{aligned} \tag{1.8}$$

where $\mathbf{R} = \mathbf{U}\mathbf{U}^{\mathrm{H}} = \mathbf{R}^{\mathrm{H}} \in \mathbb{C}^{M\times M}$ and $\mathbf{p} = \mathbf{U}\mathbf{d}^* \in \mathbb{C}^M$. This is one cost function which we will minimize by choosing an appropriate $\mathbf{w}$. We denote the cost function as $J(\mathbf{w})$:

$$J(\mathbf{w}) = \|\mathbf{e}\|_2^2 = \|\mathbf{d}\|_2^2 - \mathbf{w}^{\mathrm{H}}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} \in \mathbb{R}, \tag{1.9}$$

which is a real valued cost function of a complex vector $\mathbf{w}$. In order to find the minimum we then need to

$$\min_{\mathbf{w}} J(\mathbf{w}). \tag{1.10}$$

This amounts to compute the derivative of $J(\mathbf{w})$ given in (1.9) and set the derivative to zero to find the $\mathbf{w}_{\text{opt}}$ which minimizes $J(\mathbf{w})$.

1.4 Multichannel Adaptive Beamforming

In the previous section, we considered the case of temporal adaptive equalization. In this section we analyze an example of spatial adaptive equalization. Another interesting application for adaptive equalization is multichannel adaptive beamforming, which instead of temporal filtering is basically spatial filtering. In Fig. 1.11 a *uniform linear array* (ULA) is depicted on which a planar wavefront impinges with an angle θ. This wavefront is spatially sampled by the sensors of the antenna array and a weighted sum of these samples constitutes the output $y[n]$. The weighting vector is called beamforming vector. As in the previous section, the w_k for $k = 0, \dots, M-1$, represent the filter coefficients and the $y[n]$ is the output at time instant n. However, the input is now sampled in space and hence, $u_k[n]$ denotes the received signal at the k-th receive antenna at the time instant n.

In Fig. 1.11, the angle of arrival of the impinging wavefront is given by θ. Additionally, the distance between adjacent antenna in the ULA is Δ and therefore, there is a delay between two adjacent antennas receiving the impinging wavefront. We denote this delay as τ. The azimuthal *angle of arrival* (AoA, *direction of arrival* (DoA)) can be converted to an electric phase angle ϕ via the delay τ between two sensors. To this end, we have that the distance that the signal travels during the delay τ is

$$c \cdot \tau = \Delta \cdot \sin\theta, \tag{1.11}$$

where c is the speed of light

$$c = \lambda \cdot f_{\mathrm{c}} = \lambda \cdot \frac{\omega_{\mathrm{c}}}{2\pi} = \lambda \cdot \frac{\omega_{\mathrm{c}}}{2\pi}, \tag{1.12}$$

with λ, f_{c} and ω as the wavelength of the impinging wavefront or signal, the carrier frequency of the impinging wavefront and the angular frequency, respectively. Hence, taking (1.11) and (1.12) the delay is

$$\tau = \frac{2\pi}{\lambda\omega_{\mathrm{c}}} \cdot \Delta \cdot \sin\theta = \Big|_{\Delta=\frac{\lambda}{2}} \frac{\pi}{\omega_{\mathrm{c}}} \sin\theta, \tag{1.13}$$

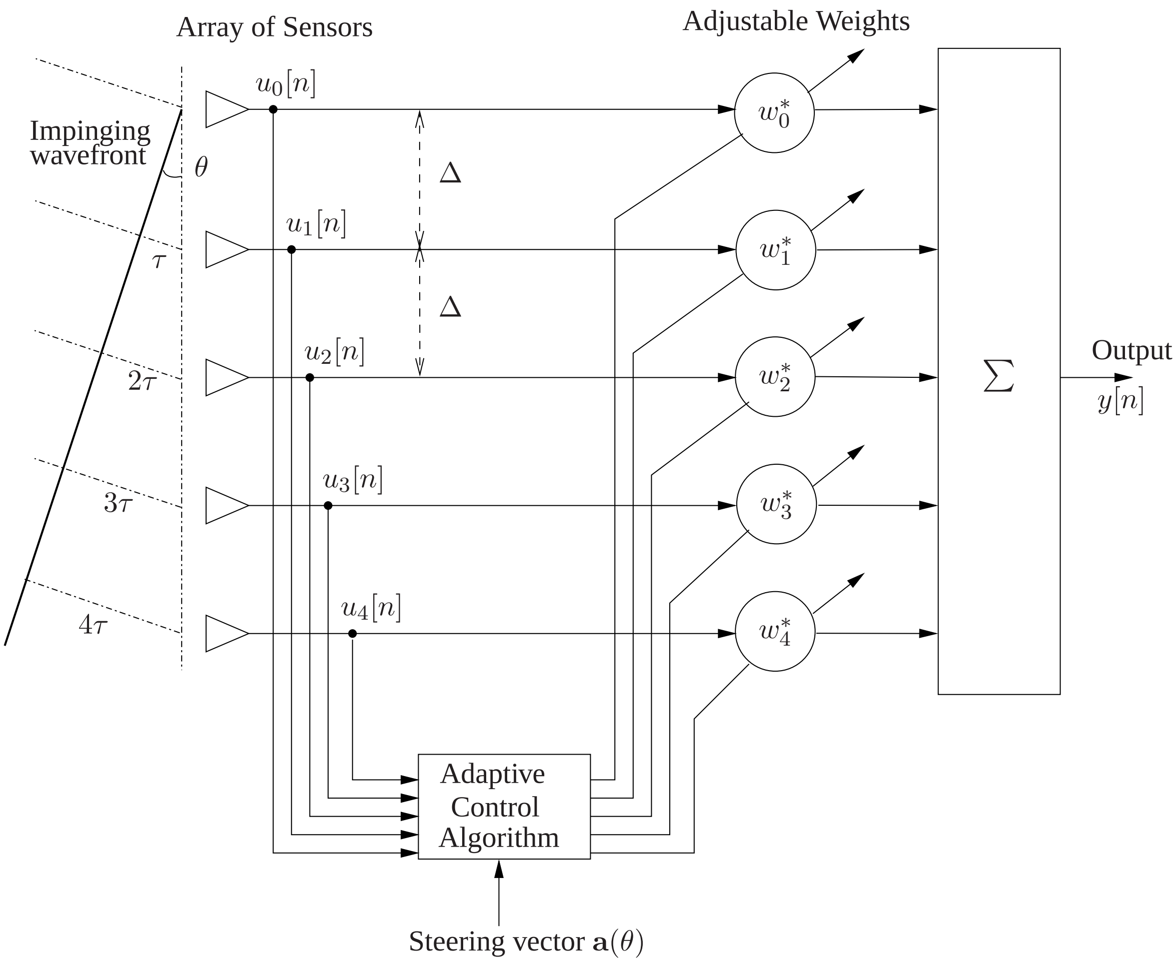


Fig. 1.11. Adaptive beamformer for an array of five sensors

where in the last equality we have assumed that distance between adjacent antennas is half a wavelength of the impinging wavefront, i.e. $\Delta = \frac{\lambda}{2}$ [1]. Using (1.13), the electric phase angle is

$$\phi = \omega_c \tau = \pi \sin\theta. \tag{1.14}$$

We have the input vector

$$\mathbf{u}[n] = \begin{bmatrix} u_0[n] \\ u_1[n] \\ \vdots \\ u_{M-1}[n] \end{bmatrix} \in \mathbb{C}^M, \tag{1.15}$$

and the filter vector

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix} \in \mathbb{C}^M. \tag{1.16}$$

[1] In this course, unless otherwise stated we will usually make this assumption when we have a ULA.

The output is given by

$$y[n] = \sum_{k=0}^{M-1} u_k[n] w_k^* = \mathbf{w}^{\mathrm{H}} \cdot \mathbf{u}[n], \tag{1.17}$$

$$y^*[n] = \mathbf{u}^{\mathrm{H}}[n] \cdot \mathbf{w}. \tag{1.18}$$

Let us collect again $N+1$ snapshots in one output vector of the beamformer,

$$\mathbf{y}^*[n] = \begin{bmatrix} y^*[n] \\ y^*[n+1] \\ \vdots \\ y^*[n+N] \end{bmatrix} = \mathbf{U}^{\mathrm{H}} \cdot \mathbf{w} \tag{1.19}$$

with

$$\mathbf{U}^{\mathrm{H}} = \begin{bmatrix} \mathbf{u}^{\mathrm{H}}[n] \\ \mathbf{u}^{\mathrm{H}}[n+1] \\ \vdots \\ \mathbf{u}^{\mathrm{H}}[n+N] \end{bmatrix} \in \mathbb{C}^{(N+1)\times M} \tag{1.20}$$

At the moment we are lacking a desired signal $d[n]$, which we can use to drive the weight vector to an optimal solution $\mathbf{w}_{\mathrm{opt}}$!

Assume, we have several planar wavefronts (far-field approximation) incident on the sensor array arriving at angles $\theta_1, \theta_2, \ldots, \theta_d$. Let the wavefront with *angle of arrival* (AoA) θ_d be the desired one and consider all the other wavefronts as interferers which should be suppressed. θ_d is the only AoA, which we know a priori. Let us first calculate the contribution of this desired wavefront to the beamformer output. To this end, we assume that the received signal impinging on the first antenna of the array is $d[n]$. Therefore, we have that the desired signal impinging on the k-th antenna of the ULA at time n is denoted by $d_k[n]$ and for $k = 0, \ldots, M-1$, given by

$$\begin{aligned} d_0[n] &= d[n] \\ d_1[n] &= d[n-\tau] \approx d[n]\mathrm{e}^{-\mathrm{j}\phi_d} \\ d_2[n] &= d[n-2\tau] \approx d[n]\mathrm{e}^{-\mathrm{j}2\phi_d} \\ &\vdots \\ d_{M-1}[n] &= d[n-(M-1)\tau] \approx d[n]\mathrm{e}^{-\mathrm{j}(M-1)\phi_d} \end{aligned}$$

where the approximation $\approx$ comes from the narrowband assumption, i.e the envelope of the signal of our wavefront is approximately constant for several multiples of τ.

We put this array measurement of our desired signal into vector form

$$\mathbf{d}[n] = \begin{bmatrix} d_0[n] \\ d_1[n] \\ \vdots \\ d_{M-1}[n] \end{bmatrix} \approx \begin{bmatrix} 1 \\ \mathrm{e}^{-\mathrm{j}\phi_\mathrm{d}} \\ \mathrm{e}^{-\mathrm{j}2\phi_\mathrm{d}} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}(M-1)\phi_\mathrm{d}} \end{bmatrix} \cdot d[n] = \mathbf{a}(\theta_\mathrm{d}) \cdot d[n], \tag{1.21}$$

where we have substituted

$$\mathbf{a}(\theta_d) = \begin{bmatrix} 1 \\ \mathrm{e}^{-\mathrm{j}\phi_\mathrm{d}} \\ \mathrm{e}^{-\mathrm{j}2\phi_\mathrm{d}} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}(M-1)\phi_\mathrm{d}} \end{bmatrix}, \tag{1.22}$$

where $\mathbf{a}(\theta_\mathrm{d})$ is the so-called *array steering vector* evaluated at the angle of arrival θ_d.

Let us denote the output signal due to the desired wavefront as

$$d[n] = \mathbf{w}^\mathrm{H} \cdot \mathbf{d}[n] = \mathbf{w}^\mathrm{H} \cdot \mathbf{a}(\theta_\mathrm{d}) \cdot d[n] = d[n],$$

i.e. we require the beamformer output due to the desired wavefront to be $d[n]$. This means that $\mathbf{w}^\mathrm{H} \cdot \mathbf{a}(\theta_\mathrm{d}) = 1$. Besides this constraint we would like to minimize the output power of the beamformer, since this will minimize the interference power at the output.

The total output signal $y[n] = \mathbf{w}^\mathrm{H}[n] \cdot \mathbf{u}[n]$ and assuming D impinging wavefronts on the array, we have that $\mathbf{u}[n] = \sum_{i=1}^{D} \mathbf{u}_i[n]$ as the sum of the D impinging wavefronts where our desired signal is $\mathbf{u}_d[n] = \mathbf{d}[n]$. For each impinging wavefront $i = 1, \ldots, D$ the received signal of the ULA is

$$\mathbf{u}_i[n] = \begin{bmatrix} 1 \\ \mathrm{e}^{-\mathrm{j}\phi_i} \\ \mathrm{e}^{-\mathrm{j}2\phi_i} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}(M-1)\phi_i} \end{bmatrix} \cdot u_i[n] = \mathbf{a}(\theta_i) \cdot u_i[n], \tag{1.23}$$

and the collection of $N+1$ samples of the output is

$$\mathbf{y}^*[n] = \begin{bmatrix} y^*[n] \\ y^*[n+1] \\ \vdots \\ y^*[n+N] \end{bmatrix} = \begin{bmatrix} \mathbf{u}^\mathrm{H}[n] \cdot \mathbf{w} \\ \mathbf{u}^\mathrm{H}[n+1] \cdot \mathbf{w} \\ \vdots \\ \mathbf{u}^\mathrm{H}[n+N] \cdot \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^\mathrm{H}[n] \\ \mathbf{u}^\mathrm{H}[n+1] \\ \vdots \\ \mathbf{u}^\mathrm{H}[n+N] \end{bmatrix} \cdot \mathbf{w} = \mathbf{U}^\mathrm{H} \cdot \mathbf{w}, \tag{1.24}$$

where recall that $\mathbf{U}$ is given by (1.20). The power at the output is

$$\begin{aligned} \|\mathbf{y}[n]\|_2^2 &= \mathbf{y}^\mathrm{H}[n] \cdot \mathbf{y}[n] = \mathbf{y}[n]^\mathrm{T} \cdot \mathbf{y}^*[n] \\ &= \mathbf{w}^\mathrm{H}\mathbf{U} \cdot \mathbf{U}^\mathrm{H}\mathbf{w} = \mathbf{w}^\mathrm{H}\mathbf{R}\mathbf{w}, \end{aligned} \tag{1.25}$$

with $\mathbf{R} = \mathbf{U}\mathbf{U}^\mathrm{H}$. Therefore, the problem can be stated as follows:

$$\boxed{\min_{\mathbf{w}} \ \|\mathbf{y}[n]\|_2^2 = \min_{\mathbf{w}} \ \mathbf{w}^\mathrm{H}\mathbf{R}\mathbf{w}, \quad \text{subject to} \quad \mathbf{w}^\mathrm{H} \cdot \mathbf{a}(\theta_\mathrm{d}) = 1.}$$

The problem at hand is a quadratic minimization subject to a linear constraint, i.e. a *linearly constrained least squares* (LCLS) problem! In next chapter, we will see how we can solve such a problem.

2. Mathematical Background

As we have seen in Section 1.3 and 1.4, we need to find extreme points of a real valued scalar cost function, which in general is a function of a complex vector. This optimization can be an unconstrained or a constrained one. In order to find extreme points, we have to compute derivatives with respect to complex vectors.

2.1 Gradients

Let us start with computing derivatives with respect to vectors in the field of reals

$$\begin{aligned} \mathbf{x} &\in \mathbb{R}^n \\ f(\mathbf{x}) &\in \mathbb{R} \\ f &: \mathbb{R}^n \to \mathbb{R} \end{aligned}$$

$$\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} = \mathrm{grad} f(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x})$$

$$\mathrm{d}f = \left(\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right)^{\mathrm{T}} \cdot \mathrm{d}\mathbf{x}.$$

If $f(\mathbf{x})$ is linear

$$f(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{x} = \mathbf{x}^{\mathrm{T}}\mathbf{a},$$

where $\mathbf{a} \in \mathbb{R}^n$, then we have

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} = \mathbf{a},$$

since

$$f(\mathbf{x}) = \sum_{k=1}^{n} a_k \cdot x_k,$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ and $\mathbf{a} = [a_1, a_2, \ldots, a_n]^{\mathrm{T}}$ and we have that

$$\frac{\partial f(\mathbf{x})}{\partial x_k} = a_k.$$

If $f(\mathbf{x})$ is quadratic

$$f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x},$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, we have

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} &= \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\left(\mathbf{x}^{\mathrm{T}}\mathbf{v}\right)\Big|_{\mathbf{v}\,=\,\mathbf{A}\mathbf{x}\,=\,\mathrm{const}} + \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\left(\mathbf{u}^{\mathrm{T}}\mathbf{x}\right)\Big|_{\mathbf{u}^{\mathrm{T}}\,=\,\mathbf{x}^{\mathrm{T}}\mathbf{A}\,=\,\mathrm{const}} \\
&= \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\left(\mathbf{x}^{\mathrm{T}}\mathbf{v}\right) + \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\left(\mathbf{x}^{\mathrm{T}}\mathbf{u}\right) \\
&= \mathbf{v} + \mathbf{u} \\
&= \left(\mathbf{A} + \mathbf{A}^{\mathrm{T}}\right)\mathbf{x} \\
&= \Big|_{\mathbf{A}=\mathbf{A}^{\mathrm{T}}}\ 2\mathbf{A}\mathbf{x}.
\end{aligned}$$

2.2 Differentiation with respect to a Complex Vector

Let us start with different ways, we can write a scalar cost function, which is a function of a complex vector $\mathbf{z}$.

$$\begin{aligned}
\mathbf{z} &= \mathbf{x} + \mathrm{j}\mathbf{y} \\
\mathbf{z} &\in \mathbb{C}^n \\
\mathbf{x}, \mathbf{y} &\in \mathbb{R}^n \\
\mathbf{z}^* &= \mathbf{x} - \mathrm{j}\mathbf{y}
\end{aligned}$$

We can write the function either as a function of $\mathbf{z}$, or alternatively as a function of $\mathbf{x}$ and $\mathbf{y}$ or eventually as a function of $\mathbf{z}$ and $\mathbf{z}^*$:

$$\begin{aligned}
h(\mathbf{z}) &: \mathbb{C}^n \to \mathbb{C} \\
f(\mathbf{x}, \mathbf{y}) &: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{C} \\
g(\mathbf{z}, \mathbf{z}^*) &: \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}.
\end{aligned}$$

Let us pick a simple example for illustration:

$$\begin{aligned}
h(\mathbf{z}) &= \|\mathbf{z}\|_2^2 \\
f(\mathbf{x}, \mathbf{y}) &= \mathbf{x}^{\mathrm{T}}\mathbf{x} + \mathbf{y}^{\mathrm{T}}\mathbf{y} \\
g(\mathbf{z}, \mathbf{z}^*) &= (\mathbf{z}^*)^{\mathrm{T}}\mathbf{z} \\
h(\mathbf{z}) &= f(\mathbf{x}, \mathbf{y}) = g(\mathbf{z}, \mathbf{z}^*).
\end{aligned}$$

The definition of a gradient enables us to compute the increment of the function due to an increment of the vector argument:

$$\begin{aligned}
\mathrm{d}h &= \left(\frac{\mathrm{d}h}{\mathrm{d}\mathbf{z}}\right)^{\mathrm{T}} \cdot \mathrm{d}\mathbf{z} \\
\mathrm{d}f &= \left(\frac{\partial f}{\partial \mathbf{x}}\right)^{\mathrm{T}} \cdot \mathrm{d}\mathbf{x} + \left(\frac{\partial f}{\partial \mathbf{y}}\right)^{\mathrm{T}} \cdot \mathrm{d}\mathbf{y} \\
\mathrm{d}g &= \left(\frac{\partial g}{\partial \mathbf{z}}\right)^{\mathrm{T}} \cdot \mathrm{d}\mathbf{z} + \left(\frac{\partial g}{\partial \mathbf{z}^*}\right)^{\mathrm{T}} \cdot \mathrm{d}\mathbf{z}^*.
\end{aligned}$$

Let us assume that $f, g \in \mathbb{C}$ are differentiable, therefore, $\frac{\partial f}{\partial \mathbf{x}}, \frac{\partial f}{\partial \mathbf{y}}$ and $\frac{\partial g}{\partial \mathbf{z}}, \frac{\partial g}{\partial \mathbf{z}^*}$ exist and thus, $\mathrm{d}f$ and $\mathrm{d}g$ can be computed and $\mathrm{d}f = \mathrm{d}g$!

$\frac{\mathrm{d}h}{\mathrm{d}\mathbf{z}}$ exists only, if $h(\mathbf{z})$ is analytic. That means that assuming $h = h_\mathrm{R} + \mathrm{j}h_\mathrm{I}$ ($\mathrm{Re}\{h\} = h_\mathrm{R}$ and $\mathrm{Im}\{h\} = h_\mathrm{I}$), then the Cauchy-Riemann equations must hold:

$$\begin{aligned} \frac{\partial h_\mathrm{R}}{\partial \mathbf{x}} &= \frac{\partial h_\mathrm{I}}{\partial \mathbf{y}} \quad \text{and} \\ \frac{\partial h_\mathrm{R}}{\partial \mathbf{y}} &= -\frac{\partial h_\mathrm{I}}{\partial \mathbf{x}} \end{aligned}$$

Many functions encountered in signal processing are not analytic, e.g. $\mathbf{w}^\mathrm{H}\mathbf{R}\mathbf{w}$. In fact, cost functions are real valued and, thus, not analytic. Therefore we have to drop the possibility to work with h. It is easy to show, that

$$\left.\begin{aligned} \frac{\partial f}{\partial \mathbf{x}} &= \frac{\partial g}{\partial \mathbf{z}} + \frac{\partial g}{\partial \mathbf{z}^*} \\ \frac{\partial f}{\partial \mathbf{y}} &= \mathrm{j}\left(\frac{\partial g}{\partial \mathbf{z}} - \frac{\partial g}{\partial \mathbf{z}^*}\right) \end{aligned}\right\} \Leftrightarrow \left\{\begin{aligned} \frac{\partial g}{\partial \mathbf{z}} &= \frac{1}{2}\left(\frac{\partial f}{\partial \mathbf{x}} - \mathrm{j}\frac{\partial f}{\partial \mathbf{y}}\right) \\ \frac{\partial g}{\partial \mathbf{z}^*} &= \frac{1}{2}\left(\frac{\partial f}{\partial \mathbf{x}} + \mathrm{j}\frac{\partial f}{\partial \mathbf{y}}\right) \end{aligned}\right. .$$

Since f is real valued, both $\frac{\partial f}{\partial \mathbf{x}}$ and $\frac{\partial f}{\partial \mathbf{y}}$ are also real valued and, therefore,

$$\frac{\partial g}{\partial \mathbf{z}} = \left(\frac{\partial g}{\partial \mathbf{z}^*}\right)^*$$

$$\begin{aligned} \mathrm{d}g &= \left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} \mathrm{d}\mathbf{z} + \left(\frac{\partial g}{\partial \mathbf{z}^*}\right)^\mathrm{T} \mathrm{d}\mathbf{z}^* \\ &= \left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} \mathrm{d}\mathbf{z} + \left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{H} \mathrm{d}\mathbf{z}^* \\ &= \left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} \mathrm{d}\mathbf{z} + \left(\left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} \mathrm{d}\mathbf{z}\right)^* \\ &= 2\mathrm{Re}\left\{\left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} \mathrm{d}\mathbf{z}\right\} \\ &= 2\mathrm{Re}\left\{\left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{H} \mathrm{d}\mathbf{z}^*\right\}. \end{aligned}$$

Additionally, note that

$$\mathrm{d}g \stackrel{!}{=} \mathrm{d}f,$$

since

$$\begin{aligned} \mathrm{d}g &= \left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} \mathrm{d}\mathbf{z} + \left(\frac{\partial g}{\partial \mathbf{z}^*}\right)^\mathrm{T} \mathrm{d}\mathbf{z}^* \\ &= \left(\left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} + \left(\frac{\partial g}{\partial \mathbf{z}^*}\right)^\mathrm{T}\right) \mathrm{d}\mathbf{x} + \mathrm{j}\left(\left(\frac{\partial g}{\partial \mathbf{z}}\right)^\mathrm{T} - \left(\frac{\partial g}{\partial \mathbf{z}^*}\right)^\mathrm{T}\right) \mathrm{d}\mathbf{y} \\ &= \left(\frac{\partial f}{\partial \mathbf{x}}\right)^\mathrm{T} \mathrm{d}\mathbf{x} + \left(\frac{\partial f}{\partial \mathbf{y}}\right)^\mathrm{T} \mathrm{d}\mathbf{y}. \end{aligned}$$

Therefore, only one derivative, either $\frac{\partial g}{\partial \mathbf{z}}$ or $\frac{\partial g}{\partial \mathbf{z}^*}$ must be computed to have the full gradient information! To compute stationary points of f or g, it is sufficient to set $\frac{\partial g}{\partial \mathbf{z}^*} = 0$. The direction of the steepest descent (gradient descent) is given by $-\mathrm{d}g$. To this end, we need to find for which direction $\mathrm{d}\mathbf{z}$, with a given length $\|\mathrm{d}\mathbf{z}\|$, we will obtain a maximal $\mathrm{d}g$.

$$\mathrm{d}g = 2\mathrm{Re}\left\{ \left(\frac{\partial g}{\partial \mathbf{z}}\right)^{\mathrm{H}} \mathrm{d}\mathbf{z}^* \right\} \stackrel{!}{=} \max .$$

From this and based on the Cauchy-Schwarz inequality, we conclude that the steepest descent will be achieved if $\Delta\mathbf{z} = -\mu\frac{\partial g}{\partial \mathbf{z}^*}$ or $\Delta\mathbf{z}^* = -\mu\frac{\partial g}{\partial \mathbf{z}}$, with $\mu \in \mathbb{R}_+$.

Look at the problem in Section 1.3

$$\begin{aligned} \mathrm{J}(\mathbf{w},\mathbf{w}^*) &= \sigma_{\mathrm{d}}^2 - \mathbf{w}^{\mathrm{H}}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} \\ \frac{\partial \mathrm{J}}{\partial \mathbf{w}^*} &= -\mathbf{p} + \mathbf{R}\mathbf{w}, \end{aligned}$$

with $\mathbf{R} = \mathbf{R}^{\mathrm{H}}$ and with the stationary point at

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

The steepest descent is

$$\mathrm{d}\mathbf{w} = \mu(\mathbf{p} - \mathbf{R}\mathbf{w}),$$

and

$$\mathrm{dJ} = 2\mathrm{Re}\left\{ \left((\mathbf{R}\mathbf{w} - \mathbf{p})^{\mathrm{H}}\, \mathrm{d}\mathbf{w}^* \right) \right\} .$$

These calculations are much simpler than to work with $f(\mathbf{x},\mathbf{y})$. This method is called *Wirtinger calculus*.

2.3 Quadratic Optimization with Linear Constraints

In Section 1.4 we have an optimization problem, which is constrained by a linear equation. Let us generalize this to a set of linear constraints.

$$\min_{\mathbf{w}} f(\mathbf{w}) \qquad \text{subject to} \qquad \mathbf{c}(\mathbf{w}) = \mathbf{S}^{\mathrm{H}}\mathbf{w} - \mathbf{g} = \mathbf{0}$$

where $\mathbf{w} \in \mathbb{C}^M$, $\mathbf{c}, \mathbf{g} \in \mathbb{C}^K$, $\mathbf{S}^{\mathrm{H}} \in \mathbb{C}^{K\times M}$ $K < M$. In order to accommodate the constraints, we augment the cost function and obtain the Lagrangian function $L(\mathbf{w},\boldsymbol{\lambda})$, which we minimize over $\mathbf{w}$ and maximize over $\boldsymbol{\lambda}$, the vector of Lagrangian multipliers:

$$\max_{\boldsymbol{\lambda}}\left\{ \min_{\mathbf{w}} \left[L(\mathbf{w},\boldsymbol{\lambda}) = \underbrace{f(\mathbf{w})}_{\in\mathbb{R}} + \underbrace{\boldsymbol{\lambda}^{\mathrm{H}}\overbrace{(\mathbf{S}^{\mathrm{H}}\mathbf{w} - \mathbf{g})}^{\mathbf{c}(\mathbf{w})} + \boldsymbol{\lambda}^{\mathrm{T}}\overbrace{(\mathbf{S}^{\mathrm{T}}\mathbf{w}^* - \mathbf{g}^*)}^{\mathbf{c}^*(\mathbf{w})}}_{=2\mathrm{Re}\left\{\boldsymbol{\lambda}^{\mathrm{H}}(\mathbf{S}^{\mathrm{H}}\mathbf{w}-\mathbf{g})\right\}} \right] \right\}$$

First we take the derivative with respect to $\mathbf{w}^*$

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}^*} &= \frac{\partial f}{\partial \mathbf{w}^*} + \frac{\partial}{\partial \mathbf{w}^*}\left(\boldsymbol{\lambda}^{\mathrm{H}}\cdot\mathbf{c}(\mathbf{w}) + \mathbf{c}^{\mathrm{H}}(\mathbf{w})\cdot\boldsymbol{\lambda}\right) \\ &= \frac{\partial f}{\partial \mathbf{w}^*} + \mathbf{S}\boldsymbol{\lambda} \stackrel{!}{=} \mathbf{0}. \end{aligned}$$

Next we differentiate with respect to $\boldsymbol{\lambda}^*$

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{\lambda}^*} &= \frac{\partial}{\partial \boldsymbol{\lambda}^*}\left(\boldsymbol{\lambda}^{\mathrm{H}} \cdot \mathbf{c}(\mathbf{w}) + \mathbf{c}^{\mathrm{H}}(\mathbf{w}) \cdot \boldsymbol{\lambda}\right) \\ &= \mathbf{c}(\mathbf{w}) \stackrel{!}{=} \mathbf{0}.\end{aligned}$$

From where we have

$$\begin{aligned}\frac{\partial f}{\partial \mathbf{w}^*} &= -\mathbf{S}\boldsymbol{\lambda} \\ \mathbf{S}^{\mathrm{H}}\mathbf{w} - \mathbf{g} &= \mathbf{0},\end{aligned}$$

respectively.

Choosing as a simple example cost function

$$f(\mathbf{w}) = \mathbf{w}^{\mathrm{H}}\mathbf{w}.$$

Then

$$\frac{\partial f}{\partial \mathbf{w}^*} = \mathbf{w}$$

and

$$\frac{\partial L}{\partial \mathbf{w}^*} = \mathbf{0},$$

from where we have

$$\mathbf{w} = -\mathbf{S}\boldsymbol{\lambda}.$$

and

$$-\mathbf{S}^{\mathrm{H}}\mathbf{S}\boldsymbol{\lambda} - \mathbf{g} = \mathbf{0}.$$

Computing $\boldsymbol{\lambda}$

$$\boldsymbol{\lambda} = -(\mathbf{S}^{\mathrm{H}}\mathbf{S})^{-1}\mathbf{g},$$

and plugging this in the previous equation we then have

$$\mathbf{w}_{\text{opt}} = \mathbf{S}(\mathbf{S}^{\mathrm{H}}\mathbf{S})^{-1}\mathbf{g} = (\mathbf{S}^{\mathrm{H}})^{+}\mathbf{g},$$

where $(\mathbf{S}^{\mathrm{H}})^{+}$ is the so-called pseudo-inverse of $\mathbf{S}^{\mathrm{H}}$, which will be discussed later on.

2.4 Stochastic Processes

2.4.1 Characterization: Mean, Autocorrelation, Autocovariance, Variance

A stochastic process is not just a single function of time, but is represented by - in theory - an infinite number of different realizations of the said process. The set of realizations is called ensemble. The expectation operator $\mathrm{E}[\bullet]$ takes the average over the different realizations (ensemble average), i.e. across the process.

If the process is stationary we then have that the mean, autocorrelation and autocovariance as

$$\begin{aligned}\mu[n] &= \mu \\ r[n, n-k] &= r[k] \\ c[n, n-k] &= c[k],\end{aligned}$$

Table 2.1. General Definitions of Mean, Autocorrelation and Autocovariance

Characterization	**Representation**	**Expression**
Mean-value function	$\mu[n]$	$\mathrm{E}\left[u[n]\right]$
Autocorrelation function	$r[n,n-k]$	$\mathrm{E}\left[u[n]u^*[n-k]\right]$
Autocovariance function	$c[n,n-k]$	$\mathrm{E}\left[\left(u[n]-\mu[n]\right)\left(u[n-k]-\mu(n-k)\right)^*\right]$ $r[n,n-k]-\mu[n]\mu^*[n-k]$

respectively. The autocovariance function can be expressed by the autocorrelation function and the mean value function:

$$\begin{aligned} c[n,n-k] &= \mathrm{E}\left[\left(u[n]-\mu[n]\right)\left(u[n-k]-\mu[n-k]\right)^*\right] \\ &= \mathrm{E}\left[u[n]u^*[n-k]\right]-\mathrm{E}\left[\mu[n]u^*[n-k]\right]-\mathrm{E}\left[u[n]\mu^*[n-k]\right]+\mathrm{E}\left[\mu[n]\mu^*[n-k]\right] \\ &= r[n,n-k]-\mu[n]\mathrm{E}\left[u^*[n-k]\right]-\mu^*[n-k]\mathrm{E}\left[u[n]\right]+\mu[n]\mu^*[n-k] \\ &= r[n,n-k]-\mu[n]\mu^*[n-k]-\mu^*[n-k]\mu[n]+\mu[n]\mu^*[n-k] \\ &= r[n,n-k]-\mu[n]\mu^*[n-k]. \end{aligned}$$

For $k=0$ we have that

$$\begin{aligned} r[0] &= \mathrm{E}\left[|u[n]|^2\right] \\ c[0] &= \sigma^2, \end{aligned}$$

where $r[0]$ and $c[0]$ are the mean square value and the variance, respectively. A summary of these general definitions is shown in Table 2.1.

2.4.1.1 Time averages (averages along the process)

If the process is ergodic, then we have the expected value over an ensemble of time averages is the same as the expected value over the process

$$\mathrm{E}[\hat{\mu}[n]] = \mu,$$

where $\hat{\mu}[n]$ is an unbiased estimate of μ. In addition,

$$\begin{aligned} \mathrm{E}[|\mu-\hat{\mu}[n]|^2] &= 0 \\ \mathrm{E}[|r[k]-\hat{r}(k,N)|^2] &= 0, \end{aligned}$$

mean ergodic and correlation ergodic, in the mean square error sense, respectively. A summary with the definitions for the time averages is shown in Table 2.2.

The stochastic processes which we will work with are stationary, ergodic and zero-mean.

2.4.2 Correlation Matrix

With the signal vectors we have to work with correlation matrices.

$$\mathbf{u}[n] = \begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix}$$

Table 2.2. Time averages: Mean, Autocorrelation and Autocovariance

Characterization	Representation	Expression
Mean-value function	$\hat{\mu}[N]$	$\frac{1}{N}\sum_{n=0}^{N-1} u[n]$
Autocorrelation function	$\hat{r}(k,N)$	$\frac{1}{N}\sum_{n=0}^{N-1} u[n]u^*[n-k] \quad 0 \le k \le N-1$
Autocovariance function	$\hat{c}(k,N)$	$\hat{r}(k,N) - \hat{\mu}[n]\hat{\mu}^*[n] \quad 0 \le k \le N-1$

$$\mathbf{R} = \mathrm{E}\left[\mathbf{u}[n]\cdot\mathbf{u}^{\mathrm{H}}[n]\right] = \mathrm{E}\left[\begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix} \cdot [u^*[n], u^*[n-1], \cdots, u^*[n-M+1]]\right]$$

$$\mathbf{R} = \begin{bmatrix} r[0] & r[1] & \cdots & r[M-1] \\ r[-1] & r[0] & \cdots & r[M-2] \\ \dots & \dots & \ddots & \dots \\ r[-M+1] & r[-M+2] & \cdots & r[0] \end{bmatrix} \in \mathbb{C}^{M\times M},$$

which is a matrix that is both Toeplitz and Hermitian. A matrix is Toeplitz if the entries along the diagonals are the same. A matrix is Hermitian if $\mathbf{R} = \mathbf{R}^{\mathrm{H}} = \left(\mathbf{R}^{\mathrm{T}}\right)^*$, which leads to having $r[-k] = r^*[k]$. In addition, $\mathbf{R}$ is nonnegative definite, which means that $\mathbf{x}^{\mathrm{H}}\mathbf{R}\mathbf{x} \geq 0 \; \forall \mathbf{x}$. This is shown in the following. Let us denote

$$\begin{aligned} y &= \mathbf{x}^{\mathrm{H}} \cdot \mathbf{u}[n] \\ y^* &= \mathbf{u}^{\mathrm{H}}[n] \cdot \mathbf{x}, \end{aligned}$$

where $\mathbf{x}$ is a an arbitrary vector of appropriate dimension. Then we have

$$\begin{aligned} \mathrm{E}\left[yy^*\right] &= \mathrm{E}\left[|y|^2\right] \geq 0 \\ &= \mathrm{E}\left[\mathbf{x}^{\mathrm{H}}\cdot\mathbf{u}[n]\cdot\mathbf{u}^{\mathrm{H}}[n]\cdot\mathbf{x}\right] \\ &= \mathbf{x}^{\mathrm{H}}\cdot\mathrm{E}\left[\mathbf{u}[n]\cdot\mathbf{u}^{\mathrm{H}}[n]\right]\cdot\mathbf{x} \\ &= \mathbf{x}^{\mathrm{H}}\cdot\mathbf{R}\cdot\mathbf{x} \geq 0 \end{aligned}$$

which concludes the proof.

Example: Assume we have a complex exponential signal plus noise. The noise is assumed to be zero-mean white noise, i.e. $\mathrm{E}\left[\nu[n]\right] = 0$ and in addition

$$\mathrm{E}\left[\nu[n]\nu^*[n-k]\right] = \begin{cases} \sigma_\nu^2 & k = 0 \\ 0 & k \neq 0 \end{cases}.$$

We have the signal expressed as follows

$$u[n] = \alpha \cdot \mathrm{e}^{\mathrm{j}\omega T n} + \nu[n].$$

Then the autocorrelation function is given by

$$\begin{aligned} r[k] &= \mathrm{E}\left[u[n]u^*[n-k]\right] \\ &= \mathrm{E}\left[(\alpha \cdot \mathrm{e}^{\mathrm{j}\omega T n} + \nu[n])(\alpha^* \cdot \mathrm{e}^{-\mathrm{j}\omega T(n-k)} + \nu^*[n-k])\right] \\ &= \mathrm{E}\left[|\alpha|^2 \mathrm{e}^{\mathrm{j}\omega T k} + \alpha^* \mathrm{e}^{-\mathrm{j}\omega T(n-k)} \cdot \nu[n] + \alpha \mathrm{e}^{\mathrm{j}\omega T n} \cdot \nu^*[n-k] + \nu[n]\nu^*[n-k]\right] \\ &= |\alpha|^2 \mathrm{e}^{\mathrm{j}\omega T k} + \alpha^* \mathrm{e}^{-\mathrm{j}\omega T(n-k)} \cdot \underbrace{\mathrm{E}\left[\nu[n]\right]}_{0} + \alpha \mathrm{e}^{\mathrm{j}\omega T n} \cdot \underbrace{\mathrm{E}\left[\nu^*[n-k]\right]}_{0} + \mathrm{E}\left[\nu[n]\nu^*[n-k]\right], \end{aligned}$$

resulting in

$$r[k] = \begin{cases} |\alpha|^2 + \sigma_\nu^2 & k = 0 \\ |\alpha|^2 \mathrm{e}^{\mathrm{j}\omega k T} & k \neq 0 \end{cases} .$$

Thus, the correlation matrix is

$$\mathbf{R} = |\alpha|^2 \cdot \begin{bmatrix} 1 + \frac{1}{\mathrm{SNR}} & \mathrm{e}^{\mathrm{j}\omega T} & \cdots & \mathrm{e}^{\mathrm{j}\omega T(M-1)} \\ \mathrm{e}^{-\mathrm{j}\omega T} & 1 + \frac{1}{\mathrm{SNR}} & \cdots & \mathrm{e}^{\mathrm{j}\omega T(M-2)} \\ \vdots & \vdots & \ddots & \cdots \\ \mathrm{e}^{-\mathrm{j}\omega T(M-1)} & \mathrm{e}^{-\mathrm{j}\omega T(M-2)} & \cdots & 1 + \frac{1}{\mathrm{SNR}} \end{bmatrix},$$

where $\mathrm{SNR} = \frac{|\alpha|^2}{\sigma_\nu^2}$.

2.5 Linear Equations

Many problems in science in general and in signal processing in particular come down to solving sets of linear equations. Therefore let us review the most important aspects of this topic from a geometric point of view.

Given a matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ and a vector $\mathbf{b} \in \mathbb{C}^m$, we are looking for a vector $\mathbf{x} \in \mathbb{C}^n$, which ”as good as possible” fulfills

$$\mathbf{A}\mathbf{x} \approx \mathbf{b}.$$

The rank of a matrix is the number of linearly independent column(row)-vectors [2]

$$\text{rank}(\mathbf{A}) = r \leq \min(m, n).$$

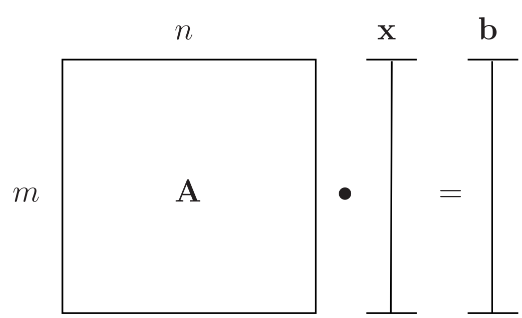
The columnspace or image or span$(\mathbf{A})$ is the vector space spanned by the column vectors of $\mathbf{A}$

$$\mathcal{S} = \text{span}(\mathbf{A}) = \text{image}(\mathbf{A}) = \{\mathbf{y} | \mathbf{y} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \mathbb{C}^n\}.$$

If $\mathbf{b} \in \text{span}(\mathbf{A})$, then there exists an $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{b}$, otherwise we have only $\mathbf{A}\mathbf{x} \approx \mathbf{b}$. The dimension of the columnspace of $\mathbf{A}$ is equal to the rank of $\mathbf{A}$

$$\dim\left(\text{span}(\mathbf{A})\right) = \text{rank}(\mathbf{A}) = r.$$

Now we will consider three different cases concerning the size of $\mathbf{A}$. Let us start with the most familiar, but not necessarily most important case, i.e. with a square full rank matrix : $r = m = n$, i.e. the number of equations is equal to the number of unknowns and all equations are linearly independent.

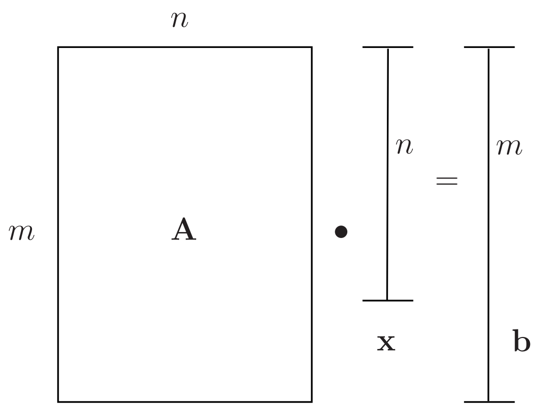


There exists a unique inverse $\mathbf{A}^{-1}$ such that

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}.$$

This does not mean, that we should compute the solution by calculating the inverse $\mathbf{A}^{-1}$. There are many more algorithms available to solve such a set of linear equations like Gaussian Elimination, LU-, LR-, QR-, QL-decomposition or conjugate gradient (CG) descent, which depending on the specific setting should be employed.

Next let us assume that $m > n$, i.e. more equations than unknowns (overdetermined system of equations), but still $r = n$, i.e. a full rank tall matrix:



In this case we can premultiply from the left with $\mathbf{A}^{\mathrm{H}}$ getting
$$\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{H}}\mathbf{b},$$

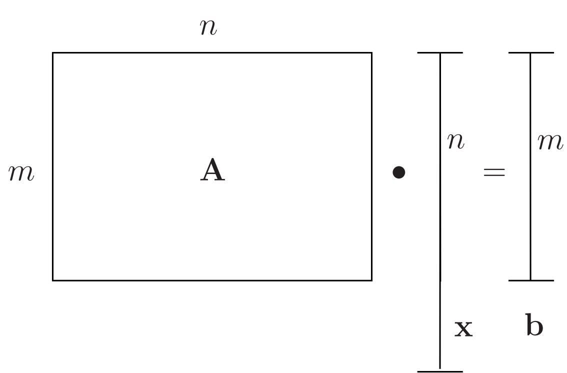
where $\mathbf{A}^{\mathrm{H}}\mathbf{A} \in \mathbb{C}^{n\times n}$ now is a full rank square matrix, which has a unique standard inverse leading to the following solution

$$\mathbf{x} = \left(\mathbf{A}^{\mathrm{H}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{b} = \mathbf{A}^{+}\mathbf{b} = \mathbf{x}_{\mathrm{LS}},$$

which is the so-called least squares solution, as we shall see later. This solution is unique and can be computed with QR-, QL-decompositions or CG-descent. Anyway, we have to be aware that

$$\mathbf{A}\mathbf{x}_{\mathrm{LS}} \neq \mathbf{b}.$$

Looking finally on the case $m < n$ (underdetermined system of equations) and still $\mathbf{A}$ being a full rank flat matrix $m = r$:



$\dim(\mathrm{span}\mathbf{A}) = m$
if $\mathbf{b} \in \mathrm{span}\mathbf{A} \Rightarrow$
$\exists\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}$
which is not unique!

Since there is no unique solution but a manifold of solutions, we can choose one of these from the $(n - r)$-dimensional solution space, e.g. by minimizing the norm of the solution vector:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \text{subject to} \quad \|\mathbf{x}\|_2 \longrightarrow \min .$$

Before we proceed let us look at a specific example discussing the three aforementioned different cases. Assume noisefree measurements taken from a ULA with m sensors exposed to n

impinging planar wavefronts. If the signal from the i-th impinging wavefront is given by x_i then the measurement is

$$\begin{aligned}\mathbf{b} &= \sum_{i=1}^{n} x_i \cdot \mathbf{a}_i \\ \mathbf{b} &= \mathbf{A}\mathbf{x},\end{aligned}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^{\mathrm{T}}$ where $\mathbf{A}$ is the array steering matrix

$$\begin{aligned}\mathbf{A} &= [\mathbf{a}_1(\theta_1), \mathbf{a}_2(\theta_2), \cdots, \mathbf{a}_n(\theta_n)] \quad \text{with} \\ \mathbf{a_i}(\theta_\mathbf{i}) &= \begin{bmatrix} 1 \\ \mathrm{e}^{\mathrm{j}\mu_i} \\ \mathrm{e}^{\mathrm{j}2\mu_i} \\ \vdots \\ \mathrm{e}^{\mathrm{j}(m-1)\mu_i} \end{bmatrix}, \quad \mu_i = 2\pi \cdot \frac{d}{\lambda} \cdot \sin\theta_i,\end{aligned}$$

where μ_i are the spatial frequencies.

For the moment assume that we know $\mathbf{A}$ (through DoA estimation with ESPRIT or MUSIC, which we will deal with later on) and it is our goal to reconstruct the vector $\mathbf{x}$ of impinging wavefronts from our array measurements $\mathbf{b}$. Since we have assumed that there is no noise, $\mathbf{b} \in \operatorname{span}(\mathbf{A})$ holds. We of course assume that all n wavefronts impinge from different directions, therefore $\mathbf{A}$ is full rank.

$m = n$: there is a unique solution reconstructing the n planar wavefronts.

$m > n$: $(m - n)$ antenna elements are superfluous, we can drop them.

$m < n$: the wavefield, consisting of the superposition of n planar wavefronts cannot be reconstructed uniquely!

But since there is noise $\mathbf{n}$ we have

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{n},$$

where $\mathbf{b}$ in general in none of these cases is in the column space of $\mathbf{A}$, and therefore $\mathbf{b} \approx \mathbf{A}\mathbf{x}$!

Let us first have a closer look at $m > n = r$, i.e. all wavefronts have distinct DoA's and we have more antenna elements than wavefronts. Therefore we have

$$\mathbf{A}\mathbf{x} - \mathbf{b} = -\mathbf{n} \quad \text{and} \quad \mathbf{x}_{\mathrm{LS}} = \operatorname*{argmin}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

$$\begin{aligned}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 &= (\mathbf{A}\mathbf{x} - \mathbf{b})^{\mathrm{H}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ &= \mathbf{x}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{b} - \mathbf{b}^{\mathrm{H}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{H}}\mathbf{b} = \|\mathbf{n}\|_2^2 \\ \frac{\partial\|\mathbf{n}\|_2^2}{\partial\mathbf{x}^*} &= \mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{x} - \mathbf{A}^{\mathrm{H}}\mathbf{b} = \mathbf{0} \Rightarrow \mathbf{x}_{\mathrm{LS}} = \left(\mathbf{A}^{\mathrm{H}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{b} = \mathbf{A}^{+}\mathbf{b},\end{aligned}$$

where $\mathbf{A}^{+} \in \mathbb{C}^{n\times m}$ and $\mathbf{A}^{+}\mathbf{A} = \mathbf{1}_n$.

If $r < n$ (i.e several wavefronts are impinging from the same direction and, therefore, are indistinguishable.) $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is rank deficient and its standard inverse does not exist.

To be able to handle such a situation, it is necessary to introduce the *Singular Value Decomposition* (SVD) of a matrix. Any matrix $\mathbf{A}$ can be decomposed into the product of three matrices

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}, \mathbf{A} \in \mathbb{C}^{m \times n}, \quad \text{rank}(\mathbf{A}) = r \leq \min(m, n),$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, i.e.

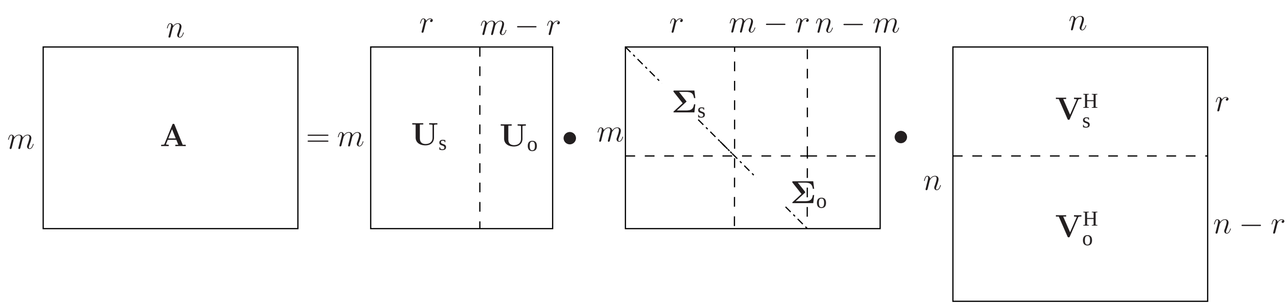
$$\mathbf{U}\mathbf{U}^{\mathrm{H}} = \mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{1}_m, \mathbf{V}\mathbf{V}^{\mathrm{H}} = \mathbf{V}^{\mathrm{H}}\mathbf{V} = \mathbf{1}_n \quad \text{and}$$

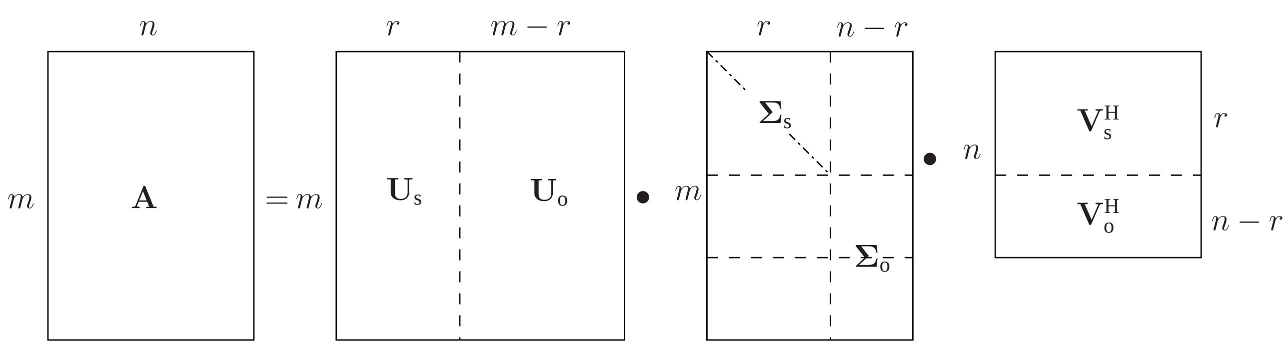
$$\mathbf{\Sigma} = \text{diag}\{\sigma_i\}_{i=1}^{\min(m,n)} \in (\mathbb{R}_+ \cup \{0\})^{m \times n}$$

with

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_{\min(m,n)} = 0.$$

The following two pictures try to visualize the situation.





We also see that

$$\mathbf{A} = \mathbf{U}_{\mathrm{s}}\mathbf{\Sigma}_{\mathrm{s}}\mathbf{V}_{\mathrm{s}}^{\mathrm{H}} + \mathbf{U}_{\mathrm{o}}\mathbf{\Sigma}_{\mathrm{o}}\mathbf{V}_{\mathrm{o}}^{\mathrm{H}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{H}}$$

and $\text{rank}(\mathbf{u}_i \cdot \mathbf{v}_i^{\mathrm{H}}) = 1$.

Now we are able to express the generalized inverse $\mathbf{A}^+$ with the aid of the SVD. Let us first start with the case $m > n = r$, which we already have handled satisfactorily. For $n = r$, there is no $\mathbf{\Sigma}_{\mathrm{o}}$ and

$$\boldsymbol{\Sigma} \;=\; m \begin{array}{c} n \\ \left[\begin{array}{c} \boldsymbol{\Sigma}_\mathrm{s} \\ \hdashline \phantom{\boldsymbol{\Sigma}_\mathrm{s}} \end{array}\right] \end{array} \begin{array}{l} n = r \\ m - r \end{array} \qquad \text{with } \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0.$$

The generalized inverse $\mathbf{A}^+$ is obtained by

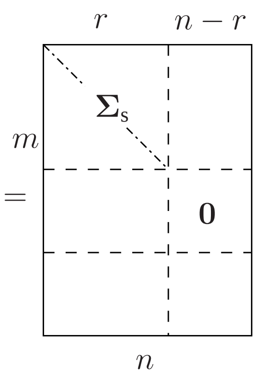
$$\mathbf{A}^+ = \mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^\mathrm{H} \quad \text{with} \qquad \boldsymbol{\Sigma}^+ \;=\; n \begin{array}{cc} n & m-n \\ \left[\begin{array}{c:c} \boldsymbol{\Sigma}_\mathrm{s}^{-1} & \phantom{\boldsymbol{\Sigma}_\mathrm{s}^{-1}} \end{array}\right] \end{array} .$$

We can easily verify this with the following calculation

$$\begin{aligned}
\left(\mathbf{A}^\mathrm{H}\mathbf{A}\right)^{-1}\mathbf{A}^\mathrm{H} &= \left(\mathbf{V}\boldsymbol{\Sigma}^\mathrm{T}\underbrace{\mathbf{U}^\mathrm{H}\mathbf{U}}_{\mathbf{1}_m}\boldsymbol{\Sigma}\mathbf{V}^\mathrm{H}\right)^{-1}\mathbf{V}\boldsymbol{\Sigma}^\mathrm{T}\mathbf{U}^\mathrm{H} \\
&= \left(\mathbf{V}\underbrace{\boldsymbol{\Sigma}^\mathrm{T}\boldsymbol{\Sigma}}_{\boldsymbol{\Sigma}_\mathrm{s}^2}\mathbf{V}^\mathrm{H}\right)^{-1}\mathbf{V}\boldsymbol{\Sigma}^\mathrm{T}\mathbf{U}^\mathrm{H} \\
&= \left(\mathbf{V}\boldsymbol{\Sigma}_\mathrm{s}^2\mathbf{V}^\mathrm{H}\right)^{-1}\mathbf{V}\boldsymbol{\Sigma}^\mathrm{T}\mathbf{U}^\mathrm{H} \\
&= \mathbf{V}\boldsymbol{\Sigma}_\mathrm{s}^{-2}\underbrace{\mathbf{V}^\mathrm{H}\mathbf{V}}_{\mathbf{1}_n}\boldsymbol{\Sigma}^\mathrm{T}\mathbf{U}^\mathrm{H} \\
&= \mathbf{V}\boldsymbol{\Sigma}_\mathrm{s}^{-2}\boldsymbol{\Sigma}^\mathrm{T}\mathbf{U}^\mathrm{H} \\
&= \mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^\mathrm{H} \\
&= \mathbf{A}^+ \quad \text{q.e.d.}
\end{aligned}$$

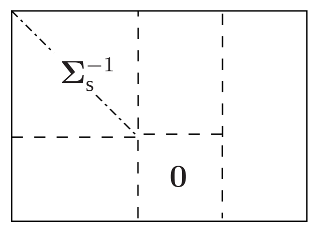
For $r < n$ the situation is a little bit more involved. For $m > n > r$, $\boldsymbol{\Sigma}$ reads as follows

$$\mathbf{\Sigma} =$$



For the generalized inverse $\mathbf{A}^+$ we proceed as follows

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^{\mathrm{H}} \quad \text{with} \qquad \mathbf{\Sigma}^+ =$$

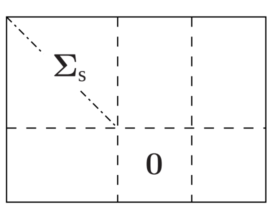


The obtained $\mathbf{x}$ with $\mathbf{x} = \mathbf{A}^+\mathbf{b}$ is the solution vector with the smallest norm. We will show this by asking the question how to choose $\mathbf{\Sigma}^+$ in order to minimize $\|\mathbf{n}\|_2^2$, $\mathbf{n} = \mathbf{A}\mathbf{x} - \mathbf{b}$?

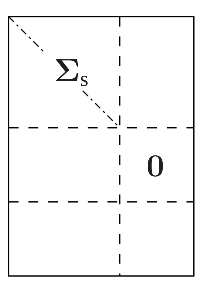
$$\begin{aligned}
\|\mathbf{n}\|_2^2 &= (\mathbf{A}\mathbf{x} - \mathbf{b})^{\mathrm{H}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \\
&= \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{x} - \mathbf{b}\right)^{\mathrm{H}}\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{x} - \mathbf{b}\right) \\
&= \mathbf{x}^{\mathrm{H}}\mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{x} - \mathbf{x}^{\mathrm{H}}\mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}}\mathbf{b} - \mathbf{b}^{\mathrm{H}}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{x} + \mathbf{b}^{\mathrm{H}}\mathbf{b}
\end{aligned}$$

$$\frac{\partial\|\mathbf{n}\|_2^2}{\partial\mathbf{x}^*} = \mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{x} - \mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}}\mathbf{b} = \mathbf{0}.$$

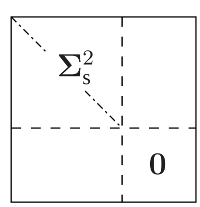
For $\mathbf{x} = \mathbf{A}^+\mathbf{b} = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^{\mathrm{H}}\mathbf{b}$ the above derivative should be zero. Let us first look at $\mathbf{\Sigma}^{\mathrm{T}}\mathbf{\Sigma}$:



$$\bullet$$



$$=$$



$$= \mathbf{\Sigma}_{\mathrm{S}_0}^2 .$$

Therefore we have

$$\begin{aligned}\frac{\partial\|\mathbf{n}\|_2^2}{\mathbf{x}^*} &= \Big|_{\mathbf{x}=\mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^{\mathrm{H}}\mathbf{b}} \mathbf{V}\boldsymbol{\Sigma}_{\mathrm{S}_0}^2\underbrace{\mathbf{V}^{\mathrm{H}}\mathbf{V}}_{\mathbf{1}_n}\boldsymbol{\Sigma}^+\mathbf{U}^{\mathrm{H}}\mathbf{b} - \mathbf{V}\boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}}\mathbf{b} \\ &= \mathbf{V}\left(\boldsymbol{\Sigma}_{\mathrm{S}_0}^2\boldsymbol{\Sigma}^+ - \boldsymbol{\Sigma}^{\mathrm{T}}\right)\mathbf{U}^{\mathrm{H}}\mathbf{b} = \mathbf{0},\end{aligned}$$

Thus

$$\frac{\partial\|\mathbf{n}\|_2^2}{\mathbf{x}^*} = \Big|_{\mathbf{x}=\mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^{\mathrm{H}}\mathbf{b}} \mathbf{0}$$

implies that

$$\boldsymbol{\Sigma}_{\mathrm{S}_0}^2 \cdot \boldsymbol{\Sigma}^+ = \boldsymbol{\Sigma}^{\mathrm{T}}$$

$$\left[\begin{array}{c|c} \boldsymbol{\Sigma}_{\mathrm{s}}^2 & \\ \hline & \mathbf{0} \end{array}\right] \bullet \left[\begin{array}{c|c|c} \boldsymbol{\Sigma}_{\mathrm{s}}^{-1} & & \\ \hline & * & \end{array}\right] = \left[\begin{array}{c|c|c} \boldsymbol{\Sigma}_{\mathrm{s}} & & \\ \hline & \mathbf{0} & \end{array}\right] .$$

For the don't care entry "*" we can choose whatever we want, the derivative will be zero. Therefore, let us use this degree of freedom to minimize the norm of the solution $\mathbf{x}$!

$$\mathbf{x} = \mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^{\mathrm{H}}\mathbf{b} \Rightarrow (\mathbf{V}^{\mathrm{H}}\mathbf{x}) = \boldsymbol{\Sigma}^+(\mathbf{U}^{\mathrm{H}}\mathbf{b})$$

$$\begin{aligned}\|\mathbf{V}^{\mathrm{H}}\mathbf{x}\|_2^2 &= \|\mathbf{x}\|_2^2 \\ \|\mathbf{U}^{\mathrm{H}}\mathbf{b}\|_2^2 &= \|\mathbf{b}\|_2^2\end{aligned}$$

since both $\mathbf{V}$ and $\mathbf{U}$ are unitary matrices.

To minimize $\|\mathbf{x}\|_2^2$ as many entries in $\boldsymbol{\Sigma}^+$ as possible should be zero! Therefore, we choose

$$\boldsymbol{\Sigma}^+ = \left[\begin{array}{c|c|c} \boldsymbol{\Sigma}_{\mathrm{s}}^{-1} & & \\ \hline & \mathbf{0} & \end{array}\right] .$$

Now we have a solution for computing generalized inverses, no matter how large m, n and r are. We simply invert only the nonzero singular values to obtain $\boldsymbol{\Sigma}^+$ and leave the zeroes as they are.

Given $\mathbf{A}$ and its SVD we now discuss the four fundamental subspaces of a matrix. We already had the column space of $\mathbf{A}$:

$$\mathcal{S} = \mathrm{span}(\mathbf{A}) = \{\mathbf{y}|\mathbf{y} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \mathbb{C}^n\}, \quad \dim(\mathcal{S}) = r.$$

Table 2.3. Four Fundamental Subspaces of a Matrix $\mathbf{A}$

Subspace	Representation	Dimension	Definition
Columnspace	$\mathrm{im}(\mathbf{A})$	$\dim(\mathrm{im}(\mathbf{A})) = r$	$\mathrm{im}(\mathbf{A}) = \{\mathbf{b} \mid \mathbf{b} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \mathbb{C}^n\}$
Nullspace	$\mathrm{null}(\mathbf{A})$	$\dim(\ker(\mathbf{A})) = n - r$	$\ker(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}$
Row space	$\mathrm{im}(\mathbf{A}^{\mathrm{H}})$	$\dim\left(\mathrm{im}(\mathbf{A}^{\mathrm{H}})\right) = r$	$\mathrm{im}(\mathbf{A}^{\mathrm{H}}) = \{\mathbf{b} \mid \mathbf{b} = \mathbf{A}^{\mathrm{H}}\mathbf{x}, \mathbf{x} \in \mathbb{C}^m\}$
Left Nullspace	$\ker(\mathbf{A}^{\mathrm{H}})$	$\dim\left(\ker(\mathbf{A}^{\mathrm{H}})\right) = m - r$	$\ker(\mathbf{A}^{\mathrm{H}}) = \{\mathbf{x} \mid \mathbf{A}^{\mathrm{H}}\mathbf{x} = \mathbf{0}\}$

$\mathbf{U}_{\mathrm{s}}$, the first r columns of $\mathbf{U}$ are a unitary basis of $\mathcal{S}$, which is also called signal subspace. Next we have the nullspace or kernel of $\mathbf{A}$:

$$\mathcal{N} = \ker(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0}\} \quad \dim(\mathcal{N}) = n - r.$$

$\mathbf{V}_{\mathrm{o}}$, the last $n - r$ rows of $\mathbf{V}^{\mathrm{H}}$ are a unitary basis of $\mathcal{N}$.

The columnspace of $\mathbf{A}^{\mathrm{H}}$ (also called left columnspace of $\mathbf{A}$) is:

$$\mathcal{S}_{\mathrm{l}} = \mathrm{span}(\mathbf{A}^{\mathrm{H}}) = \{\mathbf{y} \mid \mathbf{y} = \mathbf{A}^{\mathrm{H}}\mathbf{x}, \mathbf{x} \in \mathbb{C}^m\}, \quad \dim(\mathcal{S}_{\mathrm{l}}) = r.$$

$\mathbf{V}_{\mathrm{s}}$, the first r columns of $\mathbf{V}$ are a unitary basis of $\mathcal{S}_{\mathrm{l}}$. Finally, we have the nullspace of $\mathbf{A}^{\mathrm{H}}$ (left nullspace of $\mathbf{A}$):

$$\mathcal{N}_{\mathrm{l}} = \ker(\mathbf{A}^{\mathrm{H}}) = \{\mathbf{x} \mid \mathbf{A}^{\mathrm{H}}\mathbf{x} = \mathbf{0}\} \quad \dim(\mathcal{N}_{\mathrm{l}}) = m - r.$$

$\mathbf{U}_{\mathrm{o}}$, the last $m - r$ columns of $\mathbf{U}$ are a unitary basis of $\mathcal{N}_{\mathrm{l}}$, which is also called the noise subspace! The four fundamental subspace are summarized in Table 2.3.

In signal processing we will mainly deal with the signal and the noise subspace. It is interesting to note, that the SVD is not unique, but the four fundamental subspaces, for which the SVD gives unitary basis vectors, are unique. This will be shown in the following calculation:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} = \mathbf{U}\mathbf{\Phi}\mathbf{\Phi}^*\mathbf{\Sigma}\mathbf{\Psi}\mathbf{\Psi}^*\mathbf{V}^{\mathrm{H}}$$

with $\mathbf{\Phi} = \mathrm{diag}\{\mathrm{e}^{\mathrm{j}\phi_i}\}_{i=1}^m$ and $\mathbf{\Psi} = \mathrm{diag}\{\mathrm{e}^{\mathrm{j}\phi_i}\}_{i=1}^n$, $\mathbf{\Phi}\mathbf{\Phi}^* = \mathbf{1}_m$ and $\mathbf{\Psi}\mathbf{\Psi}^* = \mathbf{1}_n$. In addition, we have $\mathbf{\Phi}^*\mathbf{\Sigma}\mathbf{\Psi} = \mathbf{\Sigma}$, $\mathbf{U}\mathbf{\Phi} = \mathbf{U}'$, $\mathbf{V}\mathbf{\Psi} = \mathbf{V}'$ and $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} = \mathbf{U}'\mathbf{\Sigma}\mathbf{V}'^{\mathrm{H}}$.

Subspaces are uniquely characterized by projectors $\mathbf{P}$. Let $\mathbf{U}_{\mathrm{s}} \in \mathbb{C}^{m \times r}$ be a unitary basis of an r-dimensional subspace of $\mathbb{C}^m$, $m > r$. Then $\mathbf{U}_{\mathrm{s}}\mathbf{Q}$, $\mathbf{Q} \in \mathbb{C}^{r \times r}$ and unitary, is also a unitary basis of the same subspace. The projector onto this subspace is

$$\mathbf{P}_{\mathcal{S}} = \mathbf{A}\mathbf{A}^+ = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^{\mathrm{H}} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^+\mathbf{U}^{\mathrm{H}} =$$

$$= \left[\begin{array}{c:c} \mathbf{U}_{\mathrm{s}} & \mathbf{U}_{\mathrm{o}} \end{array}\right] \cdot \left[\begin{array}{c:c} \mathbf{1}_r & \\ \hdashline & \mathbf{0} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{U}_{\mathrm{s}}^{\mathrm{H}} \\ \hdashline \mathbf{U}_{\mathrm{o}}^{\mathrm{H}} \end{array}\right] = \mathbf{U}_{\mathrm{s}}\mathbf{U}_{\mathrm{s}}^{\mathrm{H}}.$$

Projectors are:

- Hermitian matrices: $\mathbf{P} = \mathbf{P}^{\mathrm{H}}$

- Idempotent matrices: $\mathbf{P} = \mathbf{P}^2$
- Rank deficient matrices: rank$(\mathbf{P}) < m$.

A change of the basis vectors for a given subspace $\mathbf{U}_\mathrm{s}' = \mathbf{U}_\mathrm{s}\mathbf{Q}$, $\mathbf{Q}$ unitary, does not change the projector

$$\mathbf{P}_\mathcal{S}' = \mathbf{U}_\mathrm{s}'\mathbf{U}_\mathrm{s}'^{,\mathrm{H}} = \mathbf{U}_\mathrm{s}\underbrace{\mathbf{Q}\mathbf{Q}^\mathrm{H}}_{\mathbf{1}_r}\mathbf{U}_\mathrm{s}^\mathrm{H} = \mathbf{U}_\mathrm{s}\mathbf{U}_\mathrm{s}^\mathrm{H} = \mathbf{P}_\mathcal{S}!$$

For tracking subspaces it is advantageous to work with projectors and not with basis vectors of the (slowly) changing subspace.

Now we can look at the connection between SVD and the well known *eigenvalue decomposition* (EVD), which exists only for square matrices

$$\mathbf{A} \in \mathbb{C}^{m\times m}: \quad \mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}, \quad \mathbf{\Lambda} = \mathrm{diag}\{\lambda_i\}_{i=1}^m.$$

(*Attention:* not every square matrix is diagonalizable. This leads to the so called Jordan forms, which we will not discuss here.)

The above transform from $\mathbf{A}$ to $\mathbf{\Lambda}$ (and vice versa) is called similarity transform. Similarity transforms leave the eigenvalues and the trace of a matrix invariant. If $\mathbf{A}$ is a normal matrix, then $\mathbf{Q}$ is a unitary matrix.

$$\mathbf{A} \text{ is a normal matrix} \Leftrightarrow \mathbf{A}^\mathrm{H}\mathbf{A} = \mathbf{A}\mathbf{A}^\mathrm{H}.$$

It is easy to see that hermitian matrices are normal. If $\mathbf{A}$ is hermitian and positive semidefinite, then we have

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathrm{H}, \quad \mathbf{\Lambda} = \mathrm{diag}\{\lambda_i\}_{i=1}^m, \quad \lambda_i \in \mathbb{R}_+ \cup \{0\}.$$

Additionally, if we arrange the eigenvalues such that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_m$, then the EVD and SVD of $\mathbf{A}$ are identical. Covariance matrices are always hermitian and positive semidefinite.

The column vectors $\mathbf{q}$ of $\mathbf{Q}$ are the so called eigenvectors of $\mathbf{A}$:

$$\mathbf{A}\mathbf{q} = \lambda\mathbf{q}.$$

This equation tells us that there are vectors $\mathbf{q}$, when multiplied by $\mathbf{A}$ do not change directions in space but will only be scaled by λ. To determine those vectors $\mathbf{q}$, we have to solve

$$(\mathbf{A} - \lambda\mathbf{1})\mathbf{q} = \mathbf{0},$$

which has nontrivial solutions only, if det$(\mathbf{A} - \lambda\mathbf{1}) = 0$, which determine the eigenvalues λ.

Assume that $\mathbf{A}$ is an estimated correlation matrix $\mathbf{R} = \mathbf{U}\mathbf{U}^\mathrm{H}$, which is positive semidefinite and hermitian. We will now show, that the eigenvalues of such a matrix are nonnegative.

$$\begin{aligned} \mathbf{R}\mathbf{q}_i &= \lambda_i\mathbf{q}_i \\ \mathbf{q}_i^\mathrm{H}\mathbf{R}\mathbf{q}_i &= \lambda_i\mathbf{q}_i^\mathrm{H}\mathbf{q}_i \\ &= \lambda_i\|\mathbf{q}_i\|_2^2, \end{aligned}$$

from which we obtain

$$\lambda_i = \frac{\mathbf{q}_i^\mathrm{H}\mathbf{R}\mathbf{q}_i}{\|\mathbf{q}_i\|_2^2} \geq 0,$$

since $\mathbf{q}_i^\mathrm{H}\mathbf{R}\mathbf{q}_i \geq 0$ because $\mathbf{R}$ is positive semidefinite.

If all eigenvalues of such a matrix are distinct, then all eigenvectors will be perpendicular to each other, i.e $\mathbf{q}_i^{\mathrm{H}}\mathbf{q}_j = 0\;\forall i \neq j$.

$$\begin{aligned}\mathbf{R}\mathbf{q}_i &= \lambda_i\mathbf{q}_i\\ \mathbf{q}_j^{\mathrm{H}}\mathbf{R}\mathbf{q}_i &= \lambda_i\mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i.\end{aligned}$$

Similarly we have

$$\begin{aligned}\mathbf{R}\mathbf{q}_j &= \lambda_j\mathbf{q}_j\\ \mathbf{q}_j^{\mathrm{H}}\mathbf{R} &= \lambda_j\mathbf{q}_j^{\mathrm{H}}\\ \mathbf{q}_j^{\mathrm{H}}\mathbf{R}\mathbf{q}_i &= \lambda_j\mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i.\end{aligned}$$

Then subtracting the last two results

$$\begin{aligned}\mathbf{q}_j^{\mathrm{H}}\mathbf{R}\mathbf{q}_i &= \lambda_i\mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i\\ -(\underline{\mathbf{q}_j^{\mathrm{H}}\mathbf{R}\mathbf{q}_i} &= \underline{\lambda_j\mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i})\\ 0 &= \underbrace{(\lambda_i - \lambda_j)}_{\neq 0}\cdot\mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i\\ 0 &= \mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i,\end{aligned}$$

from where we have that $\mathbf{q}_j^{\mathrm{H}}\mathbf{q}_i = 0$, which means that $\mathbf{q}_j$ and $\mathbf{q}_i$ are perpendicular!

By simply scaling all eigenvectors to unit norm, the matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m]$ is unitary! Moreover, we have

$$\begin{aligned}\mathrm{tr}(\mathbf{R}) &= \mathrm{tr}(\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{H}})\\ &= \mathrm{tr}(\mathbf{Q}^{\mathrm{H}}\mathbf{Q}\boldsymbol{\Lambda})\\ &= \mathrm{tr}(\boldsymbol{\Lambda})\\ &= \sum_{i=1}^{m}\lambda_i,\end{aligned}$$

where $\mathrm{tr}(\mathbf{R})$ is the trace of $\mathbf{R}$, i.e. the sum of the diagonal elements of $\mathbf{R}$.

The Rayleigh quotient of $\mathbf{R}$ is $\frac{\mathbf{x}^{\mathrm{H}}\mathbf{R}\mathbf{x}}{\mathbf{x}^{\mathrm{H}}\mathbf{x}}$.

$$\lambda_{\text{max}} = \max_{\substack{\mathbf{x}\in\mathbb{C}^m\\ \mathbf{x}\neq\mathbf{0}}} \frac{\mathbf{x}^{\mathrm{H}}\mathbf{R}\mathbf{x}}{\mathbf{x}^{\mathrm{H}}\mathbf{x}}$$

and

$$\lambda_{\text{min}} = \min_{\substack{\mathbf{x}\in\mathbb{C}^m\\ \mathbf{x}\neq\mathbf{0}}} \frac{\mathbf{x}^{\mathrm{H}}\mathbf{R}\mathbf{x}}{\mathbf{x}^{\mathrm{H}}\mathbf{x}}.$$

2.6 Eigenfilter (Generalized Matched Filter)

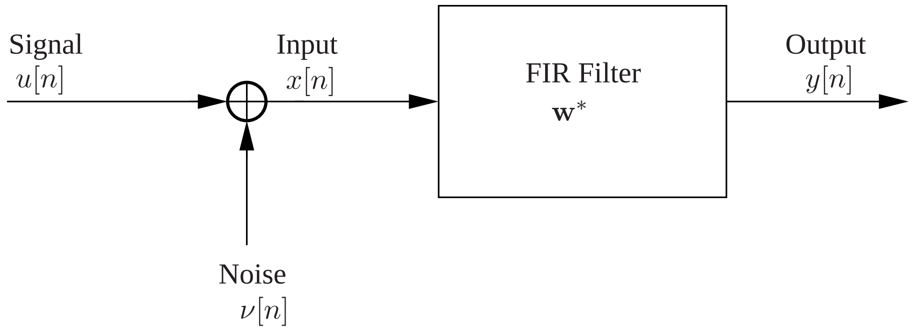


Fig. 2.1. Linear Filtering

Let us denote the following vectors as

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}, \quad \mathbf{u}[n] = \begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix}, \quad \boldsymbol{\nu}[n] = \begin{bmatrix} \nu[n] \\ \nu[n-1] \\ \vdots \\ \nu[n-M+1] \end{bmatrix},$$

where we have that the signal $u[n]$ is uncorrelated with the noise $\nu[n]$:

$$\mathrm{E}\left[u[n]\nu^*[m]\right] = 0 \quad \forall\ n, m.$$

From Figure 2.1, we have that the output $y[n]$ can be expressed as

$$y[n] = \mathbf{w}^{\mathrm{H}}\left(\mathbf{u}[n] + \nu[n]\right).$$

The power of the output can be written as

$$\begin{aligned} P &= \mathrm{E}\left[y[n]y^*[n]\right] \\ &= \mathrm{E}\left[\left(\mathbf{w}^{\mathrm{H}}(\mathbf{u}[n] + \boldsymbol{\nu}[n])\right)\left((\mathbf{u}^{\mathrm{H}}[n] + \boldsymbol{\nu}^{\mathrm{H}}[n])\mathbf{w}\right)\right] \\ &= \mathrm{E}\left[\left(\mathbf{w}^{\mathrm{H}}(\mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n] + \mathbf{u}[n]\boldsymbol{\nu}^{\mathrm{H}}[n] + \boldsymbol{\nu}[n]\mathbf{u}^{\mathrm{H}}[n] + \boldsymbol{\nu}[n]\boldsymbol{\nu}^{\mathrm{H}}[n]\right)\mathbf{w}\right] \\ &= \mathbf{w}^{\mathrm{H}}\left(\mathrm{E}\left[\mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n]\right] + \mathrm{E}\left[\boldsymbol{\nu}[n]\boldsymbol{\nu}^{\mathrm{H}}[n]\right]\right)\mathbf{w} \\ &= \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\sigma^2\mathbf{w} \\ &= P_{\mathrm{S}} + P_{\mathrm{N}}, \end{aligned}$$

where P_{S} and P_{N} are the power of the signal and the power of the noise, respectively. Thus, we can express the *signal to noise ratio* (SNR) at the output as

$$\mathrm{SNR} = \frac{P_{\mathrm{S}}}{P_{\mathrm{N}}} = \frac{\mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}}{\sigma^2\|\mathbf{w}\|^2}.$$

Let us now find the $\mathbf{w}_{\mathrm{opt}}$ that maximizes the SNR. To this end, let us differentiate the SNR expression with respect to $\mathbf{w}^*$ and set it to zero

$$\frac{\partial}{\partial \mathbf{w}^*}\left(\frac{\mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{w}}\right) = \frac{\mathbf{R}\mathbf{w} \cdot \mathbf{w}^{\mathrm{H}}\mathbf{w} - \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}\mathbf{w}}{(\mathbf{w}^{\mathrm{H}}\mathbf{w})^2} = \mathbf{0}.$$

In order that the numerator is equal to zero it must hold that

$$\begin{aligned}\mathbf{R}\mathbf{w} &= \frac{\mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{w}} \cdot \mathbf{w} \\ &= \lambda \cdot \mathbf{q},\end{aligned}$$

where $\mathbf{w}$ $(= \mathbf{q})$ is an eigenvector and $\frac{\mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{w}}$ $(= \lambda)$ is the corresponding eigenvalue of the matrix $\mathbf{R}$. Thus, the SNR is maximized for

$$\mathbf{w}_{\mathrm{opt}} = \mathbf{q}_{\mathrm{max}}$$

where $\mathbf{q}_{\mathrm{max}}$ is the eigenvector corresponding to the largest eigenvalue λ_{max} of $\mathbf{R}$, i.e.

$$\mathbf{R}\mathbf{q}_{\mathrm{max}} = \lambda_{\mathrm{max}}\mathbf{q}_{\mathrm{max}}.$$

The maximum SNR is then given by

$$\mathrm{SNR}_{\mathrm{max}} = \frac{\lambda_{\mathrm{max}}(\mathbf{R})}{\sigma^2}.$$

3. Adaptive Filters

3.1 Linear Optimum Filtering (Wiener Filters)

The optimum receive filters, which we investigate in this chapter, are linear discrete-time filters with finite duration impulse response (FIR). The reasons for these restrictions are as follows:

- **linearity:** allows for the principle of superposition and makes the mathematical analysis easy to handle.
- **discrete-time:** allows for implementation with digital VLSI hardware and firm/software.
- **FIR:** provides inherent stability.

When we talk about optimum filters, we have to give a criterion for optimization. The statistical criterion for optimization (adaptation), which we use here, is the mean square value of the estimation error, i.e. the expectation of the square difference between the desired and the actual filter output, i.e. the *mean square error* (MSE). This MSE should be minimized, i.e. we are aiming at a *minimum mean square error* (MMSE) solution. The reason for this choice is tractability of mathematics and practical performance.

The discrete-time system model is given in Fig 3.1

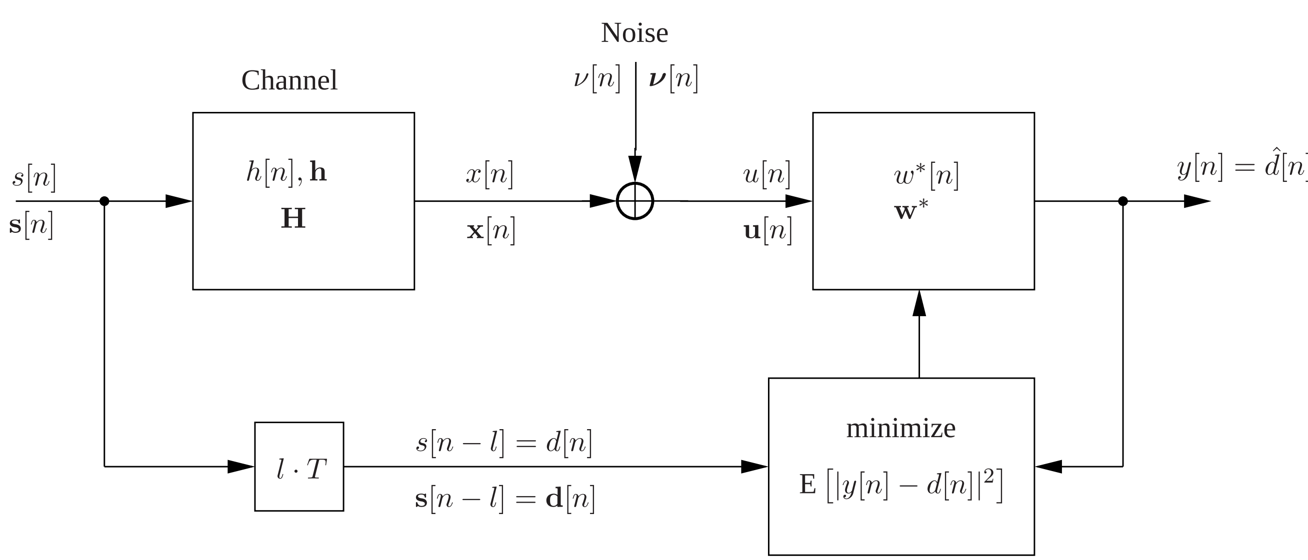


Fig. 3.1. Discrete Time System Model

The discrete-time channel impulse response is also modeled by an FIR filter:

$$h[n] = \sum_{k=0}^{K} h_k \cdot \delta[n-k],$$

where

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{else} \end{cases} .$$

In addition we have

$$\mathbf{h} = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_K \end{bmatrix} \in \mathbb{C}^{K+1}, \quad \mathbf{s}[n] = \begin{bmatrix} s[n] \\ s[n-1] \\ \vdots \\ s[n-N+1] \end{bmatrix} \in \mathbb{C}^{N},$$

and

$$x[n] = s[n] \star h[n] = \sum_{k=0}^{K} h_k \cdot s[n-k],$$

where "$\star$" denotes convolution and

$$\mathbf{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-M+1] \end{bmatrix} \in \mathbb{C}^{M}.$$

The matrix $\mathbf{H}$ can be written as

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_K & 0 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & \cdots & h_K & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & h_0 & h_1 & h_2 & \cdots & h_K \end{bmatrix} \in \mathbb{C}^{M \times N}$$

and is called convolutional matrix, which has Toeplitz structure. With $\mathbf{H}$ we can write

$$\mathbf{x}[n] \; = \; \mathbf{H} \cdot \mathbf{s}[n],$$

and

$$\mathbf{u}[n] = \mathbf{x}[n] + \boldsymbol{\nu}[n] = \mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n],$$

with $\mathbf{u}[n], \boldsymbol{\nu}[n] \in \mathbb{C}^M$. The output $y[n]$ is obtained by convolving $u[n]$ with $w^*[n]$

$$y[n] = u[n] \star w^*[n] = \mathbf{w}^{\mathrm{H}} \cdot \mathbf{u}[n] = \mathbf{w}^{\mathrm{H}}(\mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n]).$$

The error is defined as

$$e[n] = d[n] - y[n] = d[n] - \mathbf{w}^{\mathrm{H}}(\mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n]).$$

Its mean square value is

$$\mathrm{E}\left[|e[n]|^2\right] = \mathrm{E}\left[|y[n] - d[n]|^2\right],$$

from where we can find an optimum vector $\mathbf{w}_{\text{opt}}$ such that

$$\mathbf{w}_{\text{opt}} = \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathrm{E}\left[|e[n]|^2\right].$$

Therefore, the cost function can be written as

$$J(\mathbf{w}, \mathbf{w}^*) = \mathrm{E}\left[e[n] \cdot e^*[n]\right] = \mathrm{E}\left[(d[n] - \mathbf{w}^{\mathrm{H}}\mathbf{u}[n])(d^*[n] - \mathbf{u}^{\mathrm{H}}[n]\mathbf{w})\right].$$

Differentiating J with respect to $\mathbf{w}^*$ leads to

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

$$\frac{\partial J}{\partial \mathbf{w}^*} = \mathbf{0} \rightarrow \mathrm{E}\left[\mathbf{u}[n]e^*[n]\right] = \mathbf{0},$$

which means we can state that

$$\mathrm{E}\left[y[n]e^*[n]\right] = \mathbf{w}^{\mathrm{H}}\mathrm{E}\left[\mathbf{u}[n]e^*[n]\right] = 0!$$

The last two equations are known as the *principle of orthogonality*.

We can also write

$$\begin{aligned}
J &= \mathrm{E}\left[|d[n]|^2 - \mathbf{w}^{\mathrm{H}}\mathbf{u}[n]d^*[n] - \mathbf{u}^{\mathrm{H}}[n]d[n]\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{u}[n]\mathbf{u}^H[n]\mathbf{w}\right] \\
&= \sigma_d^2 - \mathbf{w}^{\mathrm{H}}\mathrm{E}\left[\mathbf{u}[n]d^*[n]\right] - \mathrm{E}\left[\mathbf{u}^{\mathrm{H}}[n]d[n]\right]\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathrm{E}\left[\mathbf{u}[n]\mathbf{u}^H[n]\right]\mathbf{w} \\
&= \sigma_d^2 - \mathbf{w}^{\mathrm{H}}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}
\end{aligned}$$

and arrive at

$$\frac{\partial J}{\partial \mathbf{w}^*} = -\mathbf{p} + \mathbf{R}\mathbf{w} = \mathbf{0} \rightarrow \mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$$

The minimum error computes to

$$\begin{aligned}
J_{\min}(\mathbf{w}_{\text{opt}}) &= \sigma_d^2 - \mathbf{w}_{\text{opt}}^{\mathrm{H}}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w}_{\text{opt}} + \mathbf{w}_{\text{opt}}^{\mathrm{H}}\mathbf{R}\mathbf{w}_{\text{opt}} \\
&= \sigma_d^2 - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p} + \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{p} \\
&= \sigma_d^2 - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p},
\end{aligned}$$

where it has been assumed that $\mathbf{R}$ is positive definitive, which means that $\mathbf{R}^{-1}$ exists. It is Hermitian anyway. The canonical form of the error surface $J(\mathbf{w})$ can be described as

$$\begin{aligned}
J(\mathbf{w}) &= \sigma_d^2 - \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{R}^{-1}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{R}\mathbf{w} + \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} \\
&= \sigma_d^2 - \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}_{\text{opt}} - \mathbf{w}_{\text{opt}}^{\mathrm{H}}\mathbf{R}\mathbf{w} + \mathbf{w}_{\text{opt}}^{\mathrm{H}}\mathbf{R}\mathbf{w}_{\text{opt}} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p} \\
&= \sigma_d^2 + (\mathbf{w}^{\mathrm{H}} - \mathbf{w}_{\text{opt}}^{\mathrm{H}})\mathbf{R}(\mathbf{w} - \mathbf{w}_{\text{opt}}) - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p} \\
&= J_{\min} + (\mathbf{w}^{\mathrm{H}} - \mathbf{w}_{\text{opt}}^{\mathrm{H}})\mathbf{R}(\mathbf{w} - \mathbf{w}_{\text{opt}}) \\
&= J_{\min} + \mathbf{\Delta}\mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{\Delta}\mathbf{w},
\end{aligned}$$

and with the eigenvalue decomposition of $\mathbf{R} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{H}}$ we have

$$J(\mathbf{w}) = J_{\text{min}} + \boldsymbol{\Delta}\mathbf{w}^{\mathrm{H}}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{H}}\boldsymbol{\Delta}\mathbf{w}.$$

Substituting $\mathbf{v} = \mathbf{Q}^{\mathrm{H}}\boldsymbol{\Delta}\mathbf{w}$ leads to

$$\begin{aligned} J(\mathbf{v}) &= J_{\text{min}} + \mathbf{v}^{\mathrm{H}}\boldsymbol{\Lambda}\mathbf{v} \\ &= J_{\text{min}} + \sum_{k=1}^{M} \lambda_k v_k v_k^* \\ &= J_{\text{min}} + \sum_{k=1}^{M} \lambda_k |v_k|^2, \end{aligned}$$

where $\lambda_k > 0$.

Now assume $h[n]$ and therefore $\mathbf{h}$ and $\mathbf{H}$ are known to the receiver (e.g. through the process of channel estimation). Let us rewrite

$$\begin{aligned} \mathrm{E}\left[\mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n]\right] &= \mathbf{R}_{\mathbf{uu}} \in \mathbb{C}^{M\times M} \\ \mathrm{E}\left[\mathbf{s}[n]\mathbf{s}^{\mathrm{H}}[n]\right] &= \mathbf{R}_{\mathbf{ss}} \in \mathbb{C}^{N\times N} \\ \mathrm{E}\left[\boldsymbol{\nu}[n]\boldsymbol{\nu}^{\mathrm{H}}[n]\right] &= \mathbf{R}_{\boldsymbol{\nu\nu}} \in \mathbb{C}^{M\times M} \end{aligned}$$

All three covariance matrices are Hermitian and positive definite. Therefore, all three have a standard inverse.

Since $\mathbf{u}[n] = \mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n]$, then

$$\begin{aligned} \mathbf{R}_{\mathbf{uu}} = \mathrm{E}\left[\mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n]\right] &= \mathrm{E}\left[(\mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n])(\mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n])^{\mathrm{H}}\right] \\ &= \mathbf{H}\mathbf{R}_{\mathbf{ss}}\mathbf{H}^{\mathrm{H}} + \mathbf{H}\cdot\underbrace{\mathrm{E}\left[\mathbf{s}[n]\boldsymbol{\nu}^{\mathrm{H}}[n]\right]}_{\mathbf{0}} + \underbrace{\mathrm{E}\left[\boldsymbol{\nu}[n]\mathbf{s}^{\mathrm{H}}[n]\right]}_{\mathbf{0}}\cdot\mathbf{H}^{\mathrm{H}} + \mathbf{R}_{\boldsymbol{\nu\nu}} \\ &= \mathbf{H}\mathbf{R}_{\mathbf{ss}}\mathbf{H}^{\mathrm{H}} + \mathbf{R}_{\boldsymbol{\nu\nu}}, \end{aligned}$$

and

$$\begin{aligned} \mathbf{p}_{\mathbf{u}d} &= \mathrm{E}\left[\mathbf{u}[n]d^*[n]\right] \\ &= \mathrm{E}\left[(\mathbf{H}\mathbf{s}[n] + \boldsymbol{\nu}[n])d^*[n]\right] \\ &= \mathbf{H}\mathrm{E}\left[\mathbf{s}[n]s^*[n-l]\right] + \underbrace{\mathrm{E}\left[\boldsymbol{\nu}[n]s^*[n-l]\right]}_{\mathbf{0}} \\ &= \mathbf{H}\mathrm{E}\left[\mathbf{s}[n]\mathbf{s}^{\mathrm{H}}[n]\cdot \mathrm{e}_{l+1}\right] \\ &= \mathbf{H}\mathbf{R}_{\mathbf{ss}}\mathrm{e}_{l+1}, \end{aligned}$$

where e_{l+1} is a vector with all entries equal to zero except the $(l+1)$th entry

$$\mathrm{e}_{l+1} = \begin{array}{c} 1 \\ 2 \\ \vdots \\ l+1 \\ \vdots \\ N \end{array} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}.$$

Therefore,

$$\begin{aligned}\mathbf{w}_{\text{opt}} &= \mathbf{R}^{-1}\mathbf{p} \\ &= \mathbf{R}_{\mathbf{uu}}^{-1}\mathbf{p}_{\mathbf{u}d} \\ &= \left(\mathbf{H}\mathbf{R}_{\mathbf{ss}}\mathbf{H}^{\mathrm{H}} + \mathbf{R}_{\boldsymbol{\nu\nu}}\right)^{-1}\mathbf{H}\mathbf{R}_{\mathbf{ss}}\mathrm{e}_{l+1}.\end{aligned}$$

With application of the *matrix inversion lemma*

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{D}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{D}\mathbf{A}^{-1},$$

where $\mathbf{A}$ and $\mathbf{C}$ must be square and full rank, and $\mathbf{B}$ and $\mathbf{D}$ must be of appropriate sizes, we can rewrite the optimum filter vector:

$$\begin{aligned}\mathbf{w}_{\text{opt}}^{\mathrm{H}} &= \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{R}_{\mathbf{ss}}\mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{R}_{\mathbf{ss}}\mathbf{H}^{\mathrm{H}}+\mathbf{R}_{\boldsymbol{\nu\nu}}\right)^{-1} \\ &= \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{R}_{\mathbf{ss}}\mathbf{H}^{\mathrm{H}}\left(\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1} - \mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\left(\mathbf{R}_{\mathbf{ss}}^{-1}+\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\right) \\ &= \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{R}_{\mathbf{ss}}\left(\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1} - \mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\left(\mathbf{R}_{\mathbf{ss}}^{-1}+\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\right) \\ &= \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{R}_{\mathbf{ss}}\left(\mathbf{1}_N - \mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\left(\mathbf{R}_{\mathbf{ss}}^{-1}+\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right)^{-1}\right)\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1} \\ &= \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{R}_{\mathbf{ss}}\left(\left(\mathbf{R}_{\mathbf{ss}}^{-1}+\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right) - \mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right)\left(\mathbf{R}_{\mathbf{ss}}^{-1}+\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1} \\ &= \mathrm{e}_{l+1}^{\mathrm{T}}\left(\mathbf{R}_{\mathbf{ss}}^{-1}+\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu\nu}}^{-1},\end{aligned}$$

and for white transmit symbols and white noise we have

$$\begin{aligned}\mathbf{R}_{\mathbf{ss}} &= \sigma_d^2\mathbf{1}_N \\ \mathbf{R}_{\boldsymbol{\nu\nu}} &= \sigma_\nu^2\mathbf{1}_M\end{aligned}$$

and the optimum filter vector is

$$\begin{aligned}\mathbf{w}_{\text{opt}}^{\mathrm{H}} &= \mathrm{e}_{l+1}^{\mathrm{T}}\left(\frac{1}{\sigma_d^2}\mathbf{1}_N + \frac{1}{\sigma_\nu^2}\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\frac{1}{\sigma_\nu^2} \\ &= \mathrm{e}_{l+1}^{\mathrm{T}}\left(\frac{\sigma_\nu^2}{\sigma_d^2}\mathbf{1}_N + \mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}},\end{aligned}$$

and from using $\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$ we had also that the optimum filter vector can be expressed as

$$\mathbf{w}_{\text{opt}}^{\mathrm{H}} = \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{H}^{\mathrm{H}}\left(\frac{\sigma_\nu^2}{\sigma_d^2}\mathbf{1}_M + \mathbf{H}\mathbf{H}^{\mathrm{H}}\right)^{-1}.$$

For very large SNR ($\frac{\sigma_\nu^2}{\sigma_d^2} \to 0$) the parenthesized matrices in the last two expressions will converge to either $(\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}$ or $(\mathbf{H}\mathbf{H}^{\mathrm{H}})^{-1}$, respectively. Since $\mathbf{H}$ is a flat matrix, $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ is not full rank and therefore does not have a standard inverse. But $\mathbf{H}\mathbf{H}^{\mathrm{H}}$ is full rank and has a standard inverse and we get

$$\lim_{\text{SNR}\to\infty}\mathbf{w}_{\text{opt}}^{\mathrm{H}} = \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{H}^{\mathrm{H}}\right)^{-1} = \mathrm{e}_{l+1}^{\mathrm{T}}\left(\left(\mathbf{H}^{\mathrm{H}}\right)^{+}\right)^{\mathrm{H}}.$$

We could equally well compute a generalized inverse of $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ making use of the SVD of $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}$:

$$\begin{aligned}
\left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{+}\mathbf{H}^{\mathrm{H}} &= \left(\mathbf{V}\boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}\right)^{+}\mathbf{V}\boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}} \\
&= \left(\mathbf{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}\right)^{+}\mathbf{V}\boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}} \\
&= \mathbf{V}\left(\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}\right)^{+}\boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{H}} \\
&= \mathbf{V}\boldsymbol{\Sigma}^{+}\mathbf{U}^{\mathrm{H}} \\
&= \mathbf{H}^{+}.
\end{aligned}$$

Therefore we also have

$$\mathbf{H}^{+} = \left(\left(\mathbf{H}^{\mathrm{H}}\right)^{+}\right)^{\mathrm{H}}.$$

Let us try to understand, what this $(\mathbf{w}'_{\mathrm{opt}})^{\mathrm{H}} = \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{H}^{+}$ actually achieves.

Assume in this noise-free case a single isolated impulse transmission:

$$\mathbf{s}[n] = \begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ N \end{array}\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{s}[n+1] = \begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ N \end{array}\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \mathbf{s}[n+N-1] = \begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ N \end{array}\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

The corresponding channel output vectors $\mathbf{x}[n] = \mathbf{u}[n]$ are therefore

$$\begin{aligned}
\mathbf{u}[n] &= \mathbf{H}\mathbf{s}[n] \\
\mathbf{u}[n+1] &= \mathbf{H}\mathbf{s}[n+1] \\
&\cdots \\
\mathbf{u}[n+N-1] &= \mathbf{H}\mathbf{s}[n+N-1].
\end{aligned}$$

Using the specific transmit vectors above, we get the following receive matrix:

$$\mathbf{u}[n] = \begin{array}{c} 1 \\ 2 \\ \vdots \\ M \end{array}\begin{bmatrix} h_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{u}[n+1] = \begin{array}{c} 1 \\ 2 \\ \vdots \\ M \end{array}\begin{bmatrix} h_1 \\ h_0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \mathbf{u}[n+N-1] = \begin{array}{c} 1 \\ 2 \\ \vdots \\ M \end{array}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ h_K \end{bmatrix}.$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}^{\mathrm{T}}[n] \\ \mathbf{u}^{\mathrm{T}}[n+1] \\ \mathbf{u}^{\mathrm{T}}[n+2] \\ \vdots \\ \mathbf{u}^{\mathrm{T}}[n+N-1] \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & \cdots & 0 \\ h_2 & h_1 & h_0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_K & h_{K-1} & h_{K-2} & \cdots & h_0 \\ 0 & h_K & h_{K-1} & \cdots & h_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & h_{K-1} \\ 0 & 0 & 0 & \cdots & h_K \end{bmatrix} \in \mathbb{C}^{N\times M}$$

This receive matrix $\mathbf{U}^{\mathrm{T}}$ is equal to $\mathbf{H}$ and we impose

$$\mathbf{w}^{\mathrm{H}}\mathbf{U}^{\mathrm{T}} = \mathbf{w}^{\mathrm{H}}\mathbf{H} = \mathrm{e}_{l+1}^{\mathrm{T}},$$

i.e. we require $\mathbf{w}^{\mathrm{H}}$ to equalize the channel such, that the impulse response of channel plus equalizer is a delayed impulse. With $\left(\mathbf{H}^{\mathrm{H}}\right)^{+}$ as the pseudoinverse of $\mathbf{H}^{\mathrm{H}}$ we have

$$\begin{aligned}
\mathbf{H}^{\mathrm{H}}\mathbf{w} &= \mathrm{e}_{l+1} \\
\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{w} &= \mathbf{H}\mathrm{e}_{l+1} \\
\mathbf{w}_{\mathrm{LS}} &= \underbrace{\left(\mathbf{H}\mathbf{H}^{\mathrm{H}}\right)^{-1}\mathbf{H}}_{(\mathbf{H}^{\mathrm{H}})^{+}}\mathrm{e}_{l+1} = \mathbf{w}'_{\mathrm{opt}} \\
\mathbf{w}_{\mathrm{LS}}^{\mathrm{H}} &= \mathrm{e}_{l+1}^{\mathrm{T}}\underbrace{\mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{H}^{\mathrm{H}}\right)^{-1}}_{\left((\mathbf{H}^{\mathrm{H}})^{+}\right)^{\mathrm{H}} = \mathbf{H}^{+}} = \lim_{\frac{\sigma_\nu^2}{\sigma_d^2}\to 0} \mathbf{w}_{\mathrm{opt}}^{\mathrm{H}}.
\end{aligned}$$

That is, we try to enforce zero *intersymbol interference* (ISI), but have not enough degrees of freedom to do so. Therefore we refrain to a least squares solution.

We could also drop some of the equations enforcing the $(N-1)$ zeros to have only M equations enforcing $(M-1)$ zeros and a one at delay l. This is equivalent to neglecting the ISI contribution from $(N-M)$ leading and/or lagging transmit symbols and can be expressed by a reduced receive matrix

$$\mathbf{U}_{\mathrm{r}}^{\mathrm{T}} = \mathbf{H}_{\mathrm{r}} \in \mathbb{C}^{M\times M}$$

by dropping the N_{post} first and N_{pre} last rows, such that

$$N_{\mathrm{r}} = N - (N_{\mathrm{pre}} + N_{\mathrm{post}}) = M$$

holds. Then $\mathbf{H}_{\mathrm{r}}$ is a square full rank matrix and

$$\mathbf{w}^{\mathrm{H}}\mathbf{H}_{\mathrm{r}} = \mathrm{e}_{l+1}^{\mathrm{T}}$$

can be exactly fulfilled with the unique solution

$$\mathbf{w}_{\mathrm{ZF}}^{\mathrm{H}} = \mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{H}_{\mathrm{r}}^{-1},$$

which zeroes out the ISI from adjacent symbols within the N_{r}-window and, therefore, is called *zero forcing* solution.

On the other hand, if $\frac{\sigma_\nu^2}{\sigma_d^2} \gg \max\{\lambda_i\}_{i=1}^{N}$ (low SNR region) is fulfilled, we get

$$\mathbf{w}_{\mathrm{opt}}^{\mathrm{H}} = \mathrm{e}_{l+1}^{\mathrm{T}} \cdot \mathbf{H}^{\mathrm{H}} \cdot \frac{\sigma_d^2}{\sigma_\nu^2}.$$

We will show in the following, that this $\mathbf{w}_{\mathrm{opt}}$ is the so called *matched filter*. Replace first $\mathbf{H}^{\mathrm{H}} \cdot \frac{\sigma_d^2}{\sigma_\nu^2}$ with $\mathbf{G}$ with

$$\begin{aligned}
\mathbf{G} &= \begin{bmatrix} \mathbf{g}_1^{\mathrm{H}} \\ \mathbf{g}_2^{\mathrm{H}} \\ \vdots \\ \mathbf{g}_N^{\mathrm{H}} \end{bmatrix} \in \mathbb{C}^{N\times M} \\
\mathrm{e}_{l+1}^{\mathrm{T}}\mathbf{G} &= \mathbf{g}_{l+1}^{\mathrm{H}}
\end{aligned}$$

and

$$y[n] = (\mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{H}\mathbf{s}[n] + \mathbf{g}_{l+1}^{\mathrm{H}}\boldsymbol{\nu}[n]).$$

The last term is the noise contribution with a variance

$$\begin{aligned} \mathrm{E}\left[|\mathbf{g}_{l+1}^{\mathrm{H}}\boldsymbol{\nu}[n]|^2\right] &= \mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{R}_{\boldsymbol{\nu}\boldsymbol{\nu}}\mathbf{g}_{l+1} \\ &= \sigma_\nu^2\|\mathbf{g}_{l+1}\|^2. \end{aligned}$$

The first term contains the desired signal contribution, stemming from $s[n-l] = d[n]$ and intersymbol interference from previous and subsequent symbols. Let us focus on the desired part, which is

$$\mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{H}\mathbf{e}_{l+1}s[n-l] = \mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{H}\mathbf{e}_{l+1}d[n],$$

the variance of which is

$$\mathrm{E}\left[\mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{H}\mathbf{e}_{l+1}d[n]d^*[n]\mathbf{e}_{l+1}^{\mathrm{T}}\mathbf{H}^{\mathrm{H}}\mathbf{g}_{l+1}\right] = \mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{H}\mathbf{e}_{l+1}\mathbf{e}_{l+1}^{\mathrm{T}}\mathbf{H}^{\mathrm{H}}\mathbf{g}_{l+1} \cdot \underbrace{\mathrm{E}\left[d[n]d^*[n]\right]}_{\sigma_d^2}.$$

With $N = 2M-1$ and $l+1 = M$ we have

$$\mathbf{H}\mathbf{e}_{l+1} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & \cdots & \boxed{h_K} & 0 & 0 & 0 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & h_3 & \cdots & h_K & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & h_K & 0 & 0 & 0 \\ 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & h_K & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & h_K & 0 \\ 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & h_K \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \mathbf{\Pi} \cdot \mathbf{h},$$

where

$$\mathbf{\Pi} = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 1 \\ 0 & \cdots & 0 & \iddots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \iddots & 0 & \cdots & 0 \\ 1 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

and $\mathbf{\Pi} \cdot \mathbf{\Pi} = \mathbf{1}$. With the variance of the desired signal becomes

$$(\mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{\Pi}\mathbf{h})(\mathbf{h}^{\mathrm{H}}\mathbf{\Pi}\mathbf{g}_{l+1})\sigma_d^2 = |\mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{\Pi}\mathbf{h}|^2\sigma_d^2.$$

The ratio of the desired signal variance and the noise variance at the filter output is

$$\mathrm{SNR} = \frac{|\mathbf{g}_{l+1}^{\mathrm{H}}\mathbf{\Pi}\mathbf{h}|^2}{\|\mathbf{g}_{l+1}\|_2^2} \cdot \frac{\sigma_d^2}{\sigma_\nu^2},$$

which will be maximal, if

$$\mathbf{g}_{l+1} = \alpha\mathbf{\Pi}\mathbf{h},$$

and the maximum SNR is

$$\begin{aligned}
\mathrm{SNR}_{\max} &= \frac{|\alpha^* \mathbf{h}^{\mathrm{H}} \boldsymbol{\Pi}\boldsymbol{\Pi}\mathbf{h}|^2}{\|\alpha \boldsymbol{\Pi}\mathbf{h}\|_2^2} \cdot \frac{\sigma_d^2}{\sigma_\nu^2} \\
&= \frac{|\alpha|^2 |\mathbf{h}^{\mathrm{H}}\mathbf{h}|^2}{|\alpha|^2 \|\mathbf{h}\|_2^2} \cdot \frac{\sigma_d^2}{\sigma_\nu^2} \\
&= \frac{\|\mathbf{h}\|_2^4}{\|\mathbf{h}\|_2^2} \cdot \frac{\sigma_d^2}{\sigma_\nu^2} \\
&= \|\mathbf{h}\|_2^2 \cdot \frac{\sigma_d^2}{\sigma_\nu^2},
\end{aligned}$$

which is independent of α! Then the optimum vector is

$$\begin{aligned}
\mathbf{w}_{\mathrm{opt}}^{\mathrm{H}} &= \mathbf{g}_{l+1}^{\mathrm{H}} \\
&= \frac{\sigma_d^2}{\sigma_\nu^2} \cdot \mathbf{h}^{\mathrm{H}} \boldsymbol{\Pi} \\
&= \frac{\sigma_d^2}{\sigma_\nu^2} \cdot \left[h_K^*, h_{K-1}^*, \cdots, h_1^*, h_0^* \right] \\
&= \frac{\sigma_d^2}{\sigma_\nu^2} \cdot \mathbf{h}_* \\
&= \mathbf{w}_{\mathrm{MF}}^{\mathrm{H}}.
\end{aligned}$$

Thus, we see that the MMSE solution converges in the high SNR regime to the ZF solution and in the low SNR regime to the MF solution.

3.2 Spatial Filtering

Here we are dealing with beamforming, which we will treat as a minimization problem with linear constraints. We will tackle that problem in two steps: first we will accommodate one linear constraint, an approach which is known as *minimum variance distortionless response* (MVDR) beamformer and second we will accommodate many linear constraints which is known as *general sidelobe canceller* (GSC). Although MVDR is a special case of GSC, both are treated in the literature separately.

3.2.1 Minimum Variance Distortionless Response (MVDR) Beamforming

Let us first look at a linearly constrained temporal minimum variance filter. Due to the FIR structure, the output of this filter with a complex harmonic excitation $u[n] = \mathrm{e}^{\mathrm{j}\omega_0 Tn}$ with frequency ω_0 can be written as

$$\begin{aligned} y[n] &= \mathbf{w}^{\mathrm{H}} \cdot \mathbf{u}[n] \\ &= \mathbf{w}^{\mathrm{H}} \cdot \begin{bmatrix} u[n] \\ u[n-1] \\ u[n-2] \\ \vdots \\ u[n-M+1] \end{bmatrix} \\ &= \mathbf{w}^{\mathrm{H}} \cdot \mathrm{e}^{\mathrm{j}\omega_0 Tn} \cdot \begin{bmatrix} 1 \\ \mathrm{e}^{-\mathrm{j}\omega_0 T} \\ \mathrm{e}^{-\mathrm{j}\omega_0 2T} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}\omega_0 (M-1)T} \end{bmatrix} \\ &= \mathrm{e}^{\mathrm{j}\omega_0 Tn} \cdot \mathbf{w}^{\mathrm{H}} \cdot \mathbf{a}(\phi_0), \end{aligned}$$

with $\phi_0 = \omega_0 T$. Note that $\mathbf{a}(\phi_0)$ is a Vandermonde vector which is identical to the array steering vector which has been introduced in Section 1.4, when dealing with spatial filtering, i.e with beamforming. Our aim is now to design the filter weight vector $\mathbf{w}$ such that this complex harmonic with frequency ω_0 should pass unattenuated, while any other frequency component of the input signal should be attenuated as much as possible. Assume now that $u[n]$ contains beside our desired component at ω_0 many other spectral components. We therefore minimize the variance of the filter output but make sure, that the desired component is not suppressed

$$\mathbf{w}_{\text{opt}} = \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathrm{E}\left[\mathbf{w}^{\mathrm{H}}\mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n]\mathbf{w}\right] = \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}$$

subject to the following constraint

$$\mathbf{w}^{\mathrm{H}}\mathbf{a}(\phi_0) = g^*,$$

g^* being the complex gain of our filter at ω_0. The problem, therefore, belongs to the class we have dealt with in Section 2.3 (Quadratic Optimization with Linear Constraints).

Since we have only one constraint, the corresponding Lagrangian function reads

$$\begin{aligned}
L(\mathbf{w},\lambda) &= \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} + \lambda(\mathbf{w}^{\mathrm{H}}\mathbf{a}(\phi_0) - g^*) + \lambda^*(\mathbf{a}^{\mathrm{H}}(\phi_0)\mathbf{w} - g). \\
\frac{\partial L(\mathbf{w},\lambda)}{\partial \mathbf{w}^*} &= \mathbf{R}\mathbf{w} + \lambda\mathbf{a}(\phi_0) = \mathbf{0} \rightarrow \mathbf{w}_{\text{opt}} = -\lambda\mathbf{R}^{-1}\mathbf{a}(\phi_0) \\
\frac{\partial L(\mathbf{w},\lambda)}{\partial \lambda^*} &= \mathbf{a}^{\mathrm{H}}(\phi_0)\mathbf{w} - g = \mathbf{0} \rightarrow \mathbf{a}^{\mathrm{H}}(\phi_0)\mathbf{w} = g \\
g &= -\lambda\mathbf{a}^{\mathrm{H}}(\phi_0)\mathbf{R}^{-1}\mathbf{a}(\phi_0) \rightarrow \lambda = \frac{-g}{\mathbf{a}^{\mathrm{H}}(\phi_0)\mathbf{R}^{-1}\mathbf{a}(\phi_0)}.
\end{aligned}$$

Finally we have

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

If we interpret $\phi_0 = 2\pi\frac{\Delta}{\lambda}\sin\theta_0$, then this is the so called *linearly constrained minimum variance* (LCMV) beamformer. If we set $g = 1$, then we have the *minimum variance distortionless response* (MVDR) beamformer.

The minimum variance, which is attained by the MVDR beamformer is

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

and the so called spatial power spectrum reads

$$S_{\text{MVDR}}(\phi) = \frac{1}{\mathbf{a}^{\mathrm{H}}(\phi)\mathbf{R}^{-1}\mathbf{a}(\phi)}, \qquad \phi \in [-\pi,\pi].$$

3.2.2 Generalized Sidelobe Canceller (GSC)

Let us now extend the previous problem to more than one linear constraint, i.e. we look for a solution of the following problem

$$\mathbf{w}_{\text{opt}} = \underset{\mathbf{w}}{\text{argmin}}\ \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} \qquad \text{s.t.} \qquad \mathbf{C}^{\mathrm{H}}\mathbf{w} = \mathbf{g},$$

where $\mathbf{C}^{\mathrm{H}}$ is the $K \times M$ ($K < M$) constraint matrix

$$\mathbf{C}^{\mathrm{H}} = \begin{bmatrix} \mathbf{a}^{\mathrm{H}}(\theta_0) \\ \mathbf{a}^{\mathrm{H}}(\theta_1) \\ \vdots \\ \mathbf{a}^{\mathrm{H}}(\theta_{K-1}) \end{bmatrix}$$

and $\mathbf{g}$ is the vector of antenna gain in the directions of arrival (or departure) θ_k for $k = 0,\ldots,K-1$. Using zeros and ones as entries of this vector means that signals impinging from the corresponding directions are either suppressed or preserved. Let us augment the columns of $\mathbf{C}$ with some orthogonal columns to obtain a square matrix

$$\mathbf{U} = \left[\ \overbrace{\mathbf{C}}^{K}\ \middle|\ \overbrace{\mathbf{C}_{\text{a}}}^{M-K}\ \right] \in \mathbb{C}^{M\times M}, \qquad \mathbf{C}^{\mathrm{H}}\mathbf{C}_{\text{a}} = \mathbf{0}.$$

These additional column vectors, which are collected in $\mathbf{C}_\mathrm{a}$ are a basis for the orthogonal complement of the space spanned by the columns of $\mathbf{C}$, i.e. $\mathrm{image}(\mathbf{C})$ and $\dim(\mathrm{image}(\mathbf{C})) = K$. Since $\mathbf{U}$ is full rank, the column vectors are a basis for the M-dimensional vector space and we can write

$$\mathbf{w} = \mathbf{U} \cdot \mathbf{q} \qquad \text{and} \qquad \mathbf{q} = \mathbf{U}^{-1} \cdot \mathbf{w}, \qquad \mathbf{q}, \mathbf{w} \in \mathbb{C}^M.$$

Let us partition $\mathbf{q}$ in the following way

$$\mathbf{q} = \left[\begin{array}{c} \mathbf{v} \\ \hline -\mathbf{w}_\mathrm{a} \end{array}\right] \begin{array}{l} \} \;\; K \\ \} \;\; M-K \end{array},$$

$$\mathbf{w} = \left[\begin{array}{c|c} \mathbf{C} & \mathbf{C}_\mathrm{a} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{v} \\ \hline -\mathbf{w}_\mathrm{a} \end{array}\right] = \mathbf{C}\mathbf{v} - \mathbf{C}_\mathrm{a}\mathbf{w}_\mathrm{a}.$$

$$\mathbf{C}^\mathrm{H} \cdot \mathbf{w} = \mathbf{C}^\mathrm{H}\mathbf{C}\mathbf{v} - \underbrace{\mathbf{C}^\mathrm{H}\mathbf{C}_\mathrm{a}}_{\mathbf{0}}\mathbf{w}_\mathrm{a} = \mathbf{C}^\mathrm{H}\mathbf{C}\mathbf{v} = \mathbf{g}.$$

Therefore, we have

$$\mathbf{v} = (\mathbf{C}^\mathrm{H}\mathbf{C})^{-1}\mathbf{g}$$

and

$$\mathbf{w}_\mathrm{q} = \mathbf{C}\mathbf{v} = \mathbf{C}(\mathbf{C}^\mathrm{H}\mathbf{C})^{-1}\mathbf{g}$$

is the so called quiescent weight vector and finally we have

$$\mathbf{w} = \mathbf{w}_\mathrm{q} - \mathbf{C}_\mathrm{a}\mathbf{w}_\mathrm{a},$$

which can be cast into the block diagram in Figure 3.2.

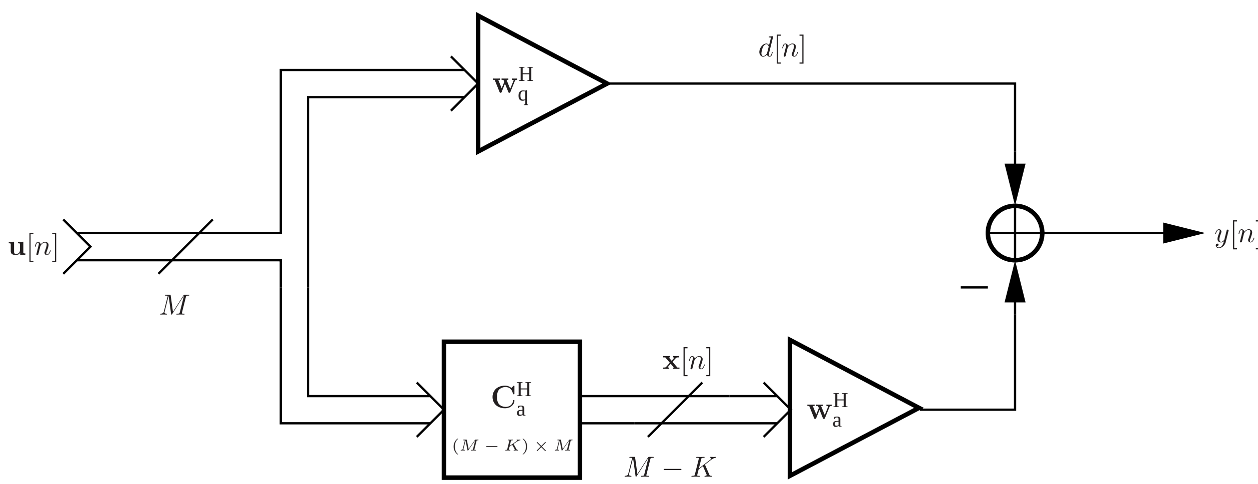


Fig. 3.2. Block Diagram of the GSC

From Figure 3.2 we have

$$y[n] = \mathbf{w}^\mathrm{H}\mathbf{u}[n] = (\mathbf{w}_\mathrm{q}^\mathrm{H} - \mathbf{w}_\mathrm{a}^\mathrm{H}\mathbf{C}_\mathrm{a}^\mathrm{H})\mathbf{u}[n] = d[n] - \mathbf{w}_\mathrm{a}^\mathrm{H}\mathbf{x}[n].$$

Let us now minimize the output variance

$$\begin{aligned}\mathbf{w}_{\text{a,opt}} &= \underset{\mathbf{w}_\text{a}}{\text{argmin}}\ \mathrm{E}\left[y[n]y^*[n]\right] \\ &= \underset{\mathbf{w}_\text{a}}{\text{argmin}}\ (\sigma_d^2 - \mathbf{p}_\mathbf{x}^\text{H}\mathbf{w}_\text{a} - \mathbf{w}_\text{a}^\text{H}\mathbf{p}_\mathbf{x} + \mathbf{w}_\text{a}^\text{H}\mathbf{R}_{\mathbf{xx}}\mathbf{w}_\text{a}),\end{aligned}$$

with $\sigma_d^2 = \mathrm{E}\left[|d[n]|^2\right]$, $\mathbf{p}_\mathbf{x} = \mathrm{E}\left[\mathbf{x}[n]d^*[n]\right]$, and $\mathbf{R}_{\mathbf{xx}} = \mathrm{E}\left[\mathbf{x}[n]\mathbf{x}^\text{H}[n]\right]$.

Note that by decomposing $\mathbf{w}$ into a quiescent component $\mathbf{w}_\text{q}$ and an adaptive component $\mathbf{w}_\text{a}$ we have transformed the original constrained optimization into an unconstrained one. The quiescent vector $\mathbf{w}_\text{q}$ and the ”blocking” matrix $\mathbf{C}_\text{a}$ take care, that the constraints are fulfilled, no matter how $\mathbf{w}_\text{a}$ is adapted. The optimum solution for $\mathbf{w}_\text{a}$ can easily be obtained by recognizing, that the problem we have formulated is identical to the original Wiener filter, which we have derived earlier.

$$\mathbf{w}_{\text{a,opt}} = \mathbf{R}_{\mathbf{xx}}^{-1}\mathbf{p}_\mathbf{x} = (\mathbf{C}_\text{a}^\text{H}\mathbf{R}\mathbf{C}_\text{a})^{-1}\mathbf{C}_\text{a}^\text{H}\mathbf{R}\mathbf{w}_\text{q}$$

and

$$\mathbf{w}_{\text{opt}} = (\mathbf{1} - \mathbf{C}_\text{a}(\mathbf{C}_\text{a}^\text{H}\mathbf{R}\mathbf{C}_\text{a})^{-1}\mathbf{C}_\text{a}^\text{H}\mathbf{R})\mathbf{C}(\mathbf{C}^\text{H}\mathbf{C})^{-1}\mathbf{g}.$$

Of course, this quadratic optimization problem with multiple linear constraints could be solved with the Lagrangian approach, which we have used in the single constraint case (LCMV). The solution reads

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^\text{H}\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{g},$$

which is identical to the aforementioned one.

Remark: Although not obvious the identity can be shown by solving an equivalent problem formulated in a transformed variable $\mathbf{z}$ with

$$\mathbf{w} = \mathbf{A}\mathbf{z}$$

and

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}},$$

where $\mathbf{V}$ and $\mathbf{\Lambda}$ can be obtained from an EVD of $\mathbf{R} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\text{H}$. The approach here with $\mathbf{w}_\text{q}$ and $\mathbf{w}_\text{a}$, although more involved, gives more insight into the structure of the solution.

3.3 Iterative Solution of the Normal Equation

We have seen that we can find the optimum linear filter by solving the so called normal equation (Wiener-Hopf equation)

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

This has been shown for temporal equalization (Fig. 3.1) and for spatial filtering by applying the Generalized Sidelobe Canceller (GSC) concept (Fig. 3.2). In the latter case we have transformed our linearly constrained quadratic optimization problem into an unconstrained minimization problem in reduced dimensions having a normal equation.

Now we will try to avoid the computationally involved direct solution and aim at an iterative algorithm instead to converge to the optimum solution $\mathbf{w}_{\text{opt}}$. Fig. 3.3 illustrates the problem for a two dimensional filter vector. We take the cost function from our previous elaborations

$$J(\mathbf{w},\mathbf{w}^*) = \sigma_d^2 - \mathbf{w}^{\mathrm{H}}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w},$$

where

$$\begin{aligned}
\sigma_d^2 &= \mathrm{E}\left[d[n]d^*[n]\right] && \text{variance of the desired signal}\\
\mathbf{p} &= \mathrm{E}\left[\mathbf{x}[n]d^*[n]\right] && \text{cross correlation vector}\\
\mathbf{R} &= \mathrm{E}\left[\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n]\right] && \text{correlation matrix.}
\end{aligned}$$

The gradient of the cost function with respect to $\mathbf{w}^*$

$$\frac{\partial J(\mathbf{w},\mathbf{w}^*)}{\partial \mathbf{w}^*} = -\mathbf{p} + \mathbf{R}\mathbf{w}$$

points into the direction of steepest ascent of the cost function. The minimum of J we have already computed previously to

$$J_{\text{min}} = J(\mathbf{w}_{\text{opt}},\mathbf{w}_{\text{opt}}^*) = \sigma_d^2 - \mathbf{p}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{p}. \tag{3.1}$$

Starting from an arbitrary initial value for filter vector $\mathbf{w}[0]$ we try to approach this minimum by incrementing $\mathbf{w}[0]$ with a step in the direction of steepest descent, i.e. in the direction of the negative gradient.

With that, we arrive at

$$\mathbf{w}[1] = \mathbf{w}[0] + \boldsymbol{\Delta}\mathbf{w}[0] = \mathbf{w}[0] + \mu(\mathbf{p} - \mathbf{R}\mathbf{w}[0]),$$

where $\mu > 0$ is the stepsize, which we will choose appropriately.

Generalizing the above equation for arbitrary iteration steps we get

$$\begin{aligned}
\mathbf{w}[n+1] &= \mathbf{w}[n] + \boldsymbol{\Delta}\mathbf{w}[n] = \mathbf{w}[n] + \mu(\mathbf{p} - \mathbf{R}\mathbf{w}[n])\\
\mathbf{w}[n+1] &= (\mathbf{1} - \mu\mathbf{R})\mathbf{w}[n] + \mu\mathbf{p}.
\end{aligned}$$

This last equation can be viewed as a linear discrete-time state-space system, where $\mathbf{w}[n]$ is the state vector with constant excitation $\mu\mathbf{p}$. Let us first transform this system to a homogeneous one by shifting the fixed point to the origin

$$\mathbf{c}[n] = \mathbf{w}[n] - \mathbf{w}_{\text{opt}} \qquad \text{and} \qquad \mathbf{c}[n+1] = \mathbf{w}[n+1] - \mathbf{w}_{\text{opt}},$$
$$\mathbf{c}[n+1] = (\mathbf{1} - \mu\mathbf{R})\mathbf{c}[n].$$

Next we will diagonalize the homogeneous discrete-time state-space system by first computing the EVD of $\mathbf{R} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{H}}$ and

$$\begin{aligned}
\mathbf{c}[n+1] &= (\mathbf{1} - \mu\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{H}})\mathbf{c}[n] \qquad |\to \mathbf{Q}^{\mathrm{H}}\\
\mathbf{Q}^{\mathrm{H}}\mathbf{c}[n+1] &= (\mathbf{Q}^{\mathrm{H}}\mathbf{Q} - \mu\mathbf{Q}^{\mathrm{H}}\mathbf{Q}\boldsymbol{\Lambda})\mathbf{Q}^{\mathrm{H}}\mathbf{c}[n].
\end{aligned}$$

With $\boldsymbol{\nu}[n] = \mathbf{Q}^{\mathrm{H}}\mathbf{c}[n]$ we have

$$\boldsymbol{\nu}[n+1] = (\mathbf{1} - \mu\boldsymbol{\Lambda})\boldsymbol{\nu}[n].$$

Since $(\mathbf{1} - \mu\boldsymbol{\Lambda})$ is diagonal, we get

$$\nu_k[n+1] = (1 - \mu\lambda_k)\nu_k[n] \quad \forall k = 1,\ldots,M.$$

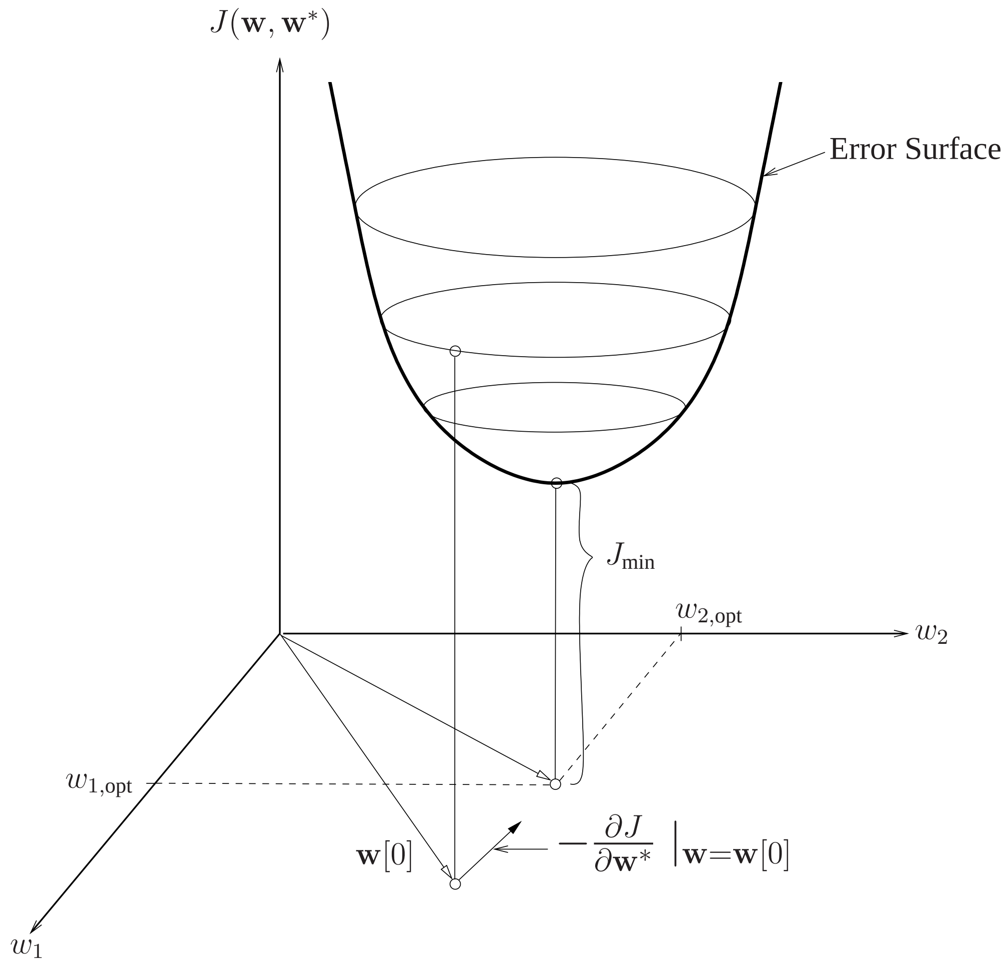


Fig. 3.3. Parabolic error surface

Thereby $\lambda_k \geq 0 \ \forall k = 1, \ldots, M$ are the eigenvalues of the positive semidefinite correlation matrix $\mathbf{R}$. To assure the fixed point at $\boldsymbol{\nu} = \mathbf{0}$ to be stable, the inequality

$$|1 - \mu\lambda_k| < 1 \Rightarrow 0 < \mu < \frac{2}{\lambda_k}$$

must hold and therefore $0 < \mu < \frac{2}{\lambda_{\text{max}}}$.

This is not a very useful criterion, because it would necessitate an EVD of $\mathbf{R}$. Reminding that the motivation for the iterative solution was reducing complexity in finding a solution for the normal equation it is not a good idea to perform an even more involved EVD instead. But we can make use of the following relation:

$$\lambda_{\text{max}} \leq \sum_{k=1}^{M} \lambda_k = \text{tr}(\mathbf{R})$$

and come up with a more stringent but simple to compute upper limit for the stepsize guaranteeing convergence

$$0 < \mu < \frac{2}{\text{tr}(\mathbf{R})} = \frac{2}{Mr(0)}.$$

Let us look at a simple two-dimensional example:

$$\begin{aligned} \mathrm{tr}(\mathbf{R}) &= 4, \\ \lambda_1 &= 3, \lambda_2 = 1, \\ \mathbf{w}^{\mathrm{T}}[0] &= [2, 8] \\ \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}} &= [2, 1] \end{aligned}$$

We have chosen to stop the iterations, if the norm of the gradient falls below 10^{-5}:

$$\|\mathbf{p} - \mathbf{R}\mathbf{w}[n]\|_2^2 = \|\mathbf{r}[n]\|_2^2 < 10^{-5}.$$

Fig. 3.4 shows the trajectory of $\mathbf{w}[n]$ for a small stepsize $\mu = 0.1$. The upper bound for μ is $\frac{2}{\lambda_{\max}} = 0.667$ and the more stringent but easier to compute upper bound is $\frac{2}{\mathrm{tr}(\mathbf{R})} = 0.5$. We see that it takes 70 steps to converge from the initial value to the optimum one within the given residual error. Let us next choose the stepsize more aggressive as $\mu = 0.6$ still below the true upper bound. Now we need only 38 steps to converge Fig. 3.5. In Fig. 3.6, we have the trajectory for $\mu = 0.5$ and we need only 13 steps.

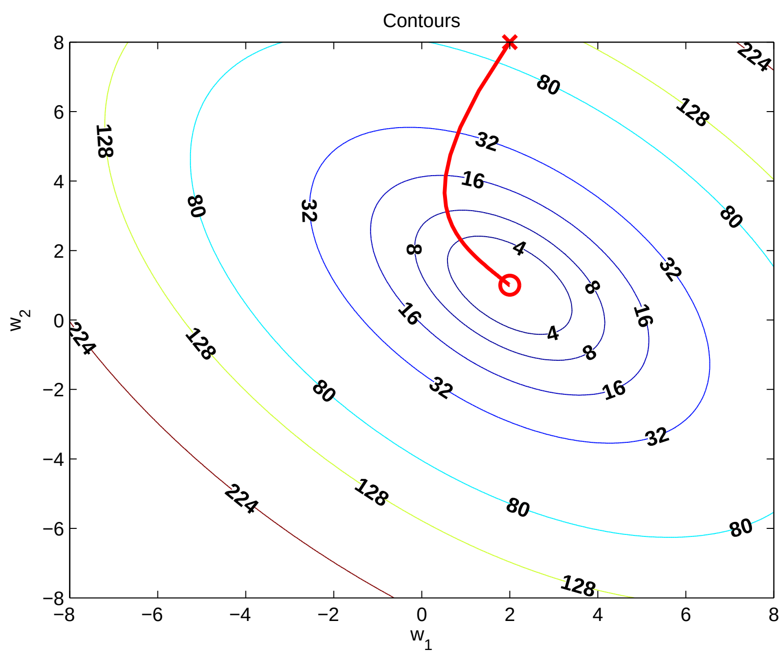


Fig. 3.4. Small Stepsize ($\mu = 0.1$): Trajectory from $\mathbf{w}[0]$ toward the solution. The number of iterations required for convergence is 70.

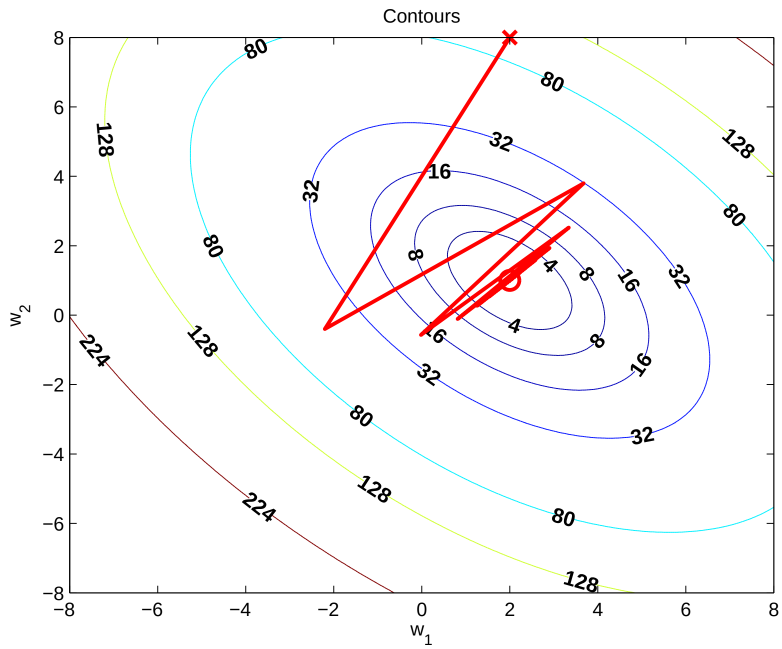


Fig. 3.5. Stepsize chosen close to the Upper Bound ($\mu = 0.6$): Trajectory from $\mathbf{w}[0]$ toward the solution. Notice the oscillations due to the choice of the stepsize. The number of iterations required for convergence is 38.

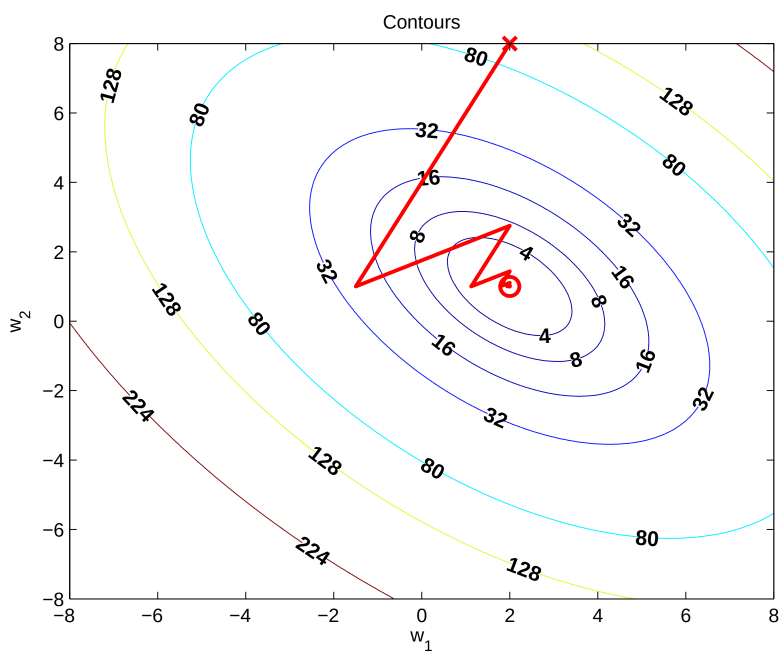


Fig. 3.6. Moderate stepsize ($\mu = 0.5$): Trajectory from $\mathbf{w}[0]$ toward the solution. The number of iterations required for convergence is 13, which is smaller than the previous two examples.

Now we try to optimize the step size for every iteration step. For this we calculate the error function at step n.

$$J(\mathbf{w}[n]) = J[n] = \sigma_d^2 - \mathbf{w}^{\mathrm{H}}[n]\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{w}[n] + \mathbf{w}^{\mathrm{H}}[n]\mathbf{R}\mathbf{w}[n]$$

and at step $n+1$:

$$\begin{aligned} J[n+1] &= J[n] - \mathbf{\Delta w}^{\mathrm{H}}[n]\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{\Delta w}[n] + \mathbf{\Delta w}^{\mathrm{H}}[n]\mathbf{R}\mathbf{\Delta w}[n] + \mathbf{w}^{\mathrm{H}}[n]\mathbf{R}\mathbf{\Delta w}[n] + \mathbf{\Delta w}^{\mathrm{H}}[n]\mathbf{R}\mathbf{w}[n] \\ &= J[n] - \mu^*(\mathbf{p}^{\mathrm{H}} - \mathbf{w}^{\mathrm{H}}[n]\mathbf{R})\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mu(\mathbf{p} - \mathbf{R}\mathbf{w}[n]) + \mu^*(\mathbf{p}^{\mathrm{H}} - \mathbf{w}^{\mathrm{H}}[n]\mathbf{R})\mathbf{R}\mu(\mathbf{p} - \mathbf{R}\mathbf{w}[n]) \\ &\quad + \mathbf{w}^{\mathrm{H}}[n]\mathbf{R}\mu(\mathbf{p} - \mathbf{R}\mathbf{w}[n]) + \mu^*(\mathbf{p}^{\mathrm{H}} - \mathbf{w}^{\mathrm{H}}[n]\mathbf{R})\mathbf{R}\mathbf{w}[n]. \end{aligned}$$

We now minimize $J[n+1]$ by choosing μ appropriately

$$\frac{\partial J[n+1]}{\partial \mu^*} = -(\mathbf{p}^{\mathrm{H}} - \mathbf{w}^{\mathrm{H}}[n]\mathbf{R})(\mathbf{p} - \mathbf{R}\mathbf{w}[n]) + \mu(\mathbf{p}^{\mathrm{H}} - \mathbf{w}^{\mathrm{H}}[n]\mathbf{R})\mathbf{R}(\mathbf{p} - \mathbf{R}\mathbf{w}[n]).$$

From this we find

$$\mu_{\text{opt}}[n] = \frac{(\mathbf{p} - \mathbf{R}\mathbf{w}[n])^{\mathrm{H}}(\mathbf{p} - \mathbf{R}\mathbf{w}[n])}{(\mathbf{p} - \mathbf{R}\mathbf{w}[n])^{\mathrm{H}}\mathbf{R}(\mathbf{p} - \mathbf{R}\mathbf{w}[n])} = \frac{\mathbf{r}^{\mathrm{H}}[n]\mathbf{r}[n]}{\mathbf{r}^{\mathrm{H}}[n]\mathbf{R}\mathbf{r}[n]}.$$

$\mathbf{r}[n]$ and $\|\mathbf{r}[n]\|_2^2 = \mathbf{r}^{\mathrm{H}}[n]\mathbf{r}[n]$ have to be computed anyway, because they are the negative gradient and the stopping criterion. The extra burden for optimizing the stepsize therefore is the quadratic form in the denominator. Fig. 3.7 shows the trajectory for this case with 8 iteration steps.

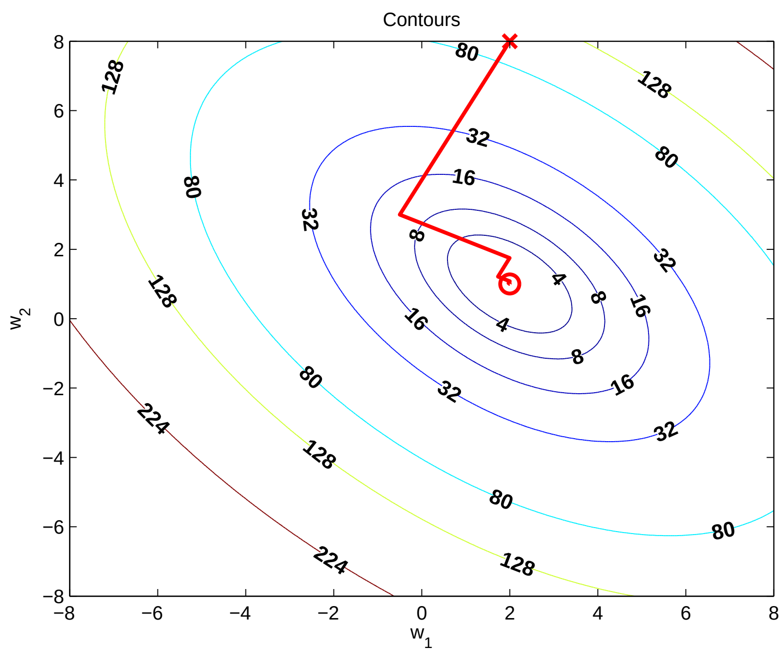


Fig. 3.7. Optimum stepsize ($\mu_{\text{opt}}[n]$): Trajectory from $\mathbf{w}[0]$ toward the solution. The number of iterations required for convergence is 8.

Finally we see in Fig. 3.8 a situation with a larger spread of eigenvalues ($\lambda_{\text{max}} = 3, \lambda_{\text{min}} = 0.1$). The constant error ellipses become slimmer and the number of iteration increases.

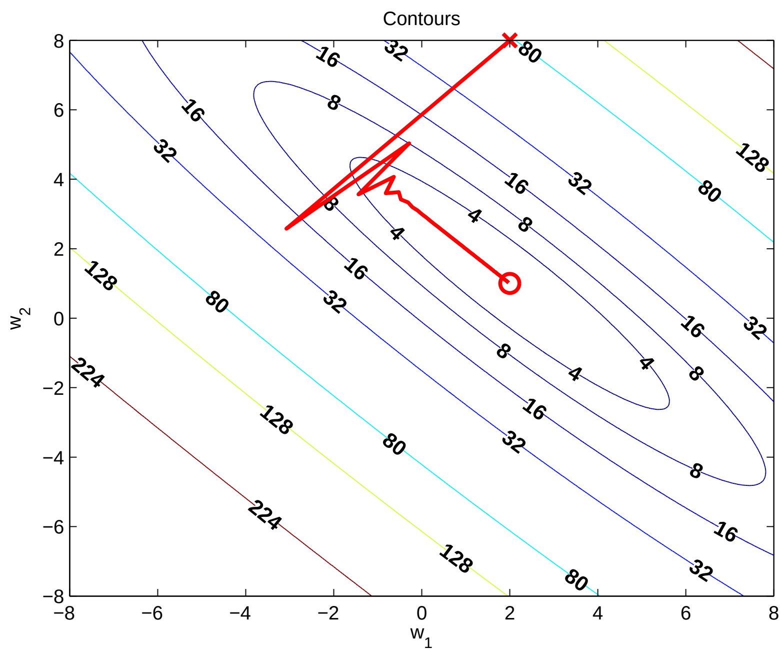


Fig. 3.8. Moderate Stepsize ($\mu = 0.5$): Trajectory from $\mathbf{w}[0]$ toward the solution with a large spread of eigenvalues ($\lambda_{\text{max}} = 3, \lambda_{\text{min}} = 0.1$). The number of iterations required for convergence is 99.

3.4 Least Mean Square Algorithm (LMS)

The LMS algorithm (1960 Widrow and Hoff) is a ”stochastic gradient algorithm” approximating the steepest descent procedure described in the previous section.

In a real-world application we do not know the true correlation matrix $\mathbf{R}$ and the true cross-correlation vector $\mathbf{p}$. Therefore, we have to estimate these expectations to have an estimate for the gradient

$$\widehat{\left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^*}\Big|_{\mathbf{w}[n]}\right)} = -\hat{\mathbf{p}}[n] + \hat{\mathbf{R}} \cdot \mathbf{w}[n].$$

We take a very simple approximation by using only one sample

$$\begin{aligned} \hat{\mathbf{R}}[n] &= \mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n] \quad \text{and} \\ \hat{\mathbf{p}}[n] &= \mathbf{u}[n]d^*[n]. \end{aligned}$$

The update equation for the weight vector now reads

$$\begin{aligned} \mathbf{w}[n+1] &= \mathbf{w}[n] + \mu(\mathbf{u}[n]d^*[n] - \mathbf{u}[n]\mathbf{u}^{\mathrm{H}}[n]\mathbf{w}[n]) \\ &= \mathbf{w}[n] + \mu\mathbf{u}[n](d^*[n] - \mathbf{u}^{\mathrm{H}}[n]\mathbf{w}[n]) \\ &= \mathbf{w}[n] + \mu\mathbf{u}[n] \cdot e^*[n]. \end{aligned}$$

This shows that we simply have to multiply the actual input vector $\mathbf{u}[n]$ with the actual complex conjugate error $e^*[n] = (d[n] - \mathbf{w}^{\mathrm{H}}[n]\mathbf{u}[n])^* = d^*[n] - y^*[n]$ and use this as the update for the next step. Fig. 3.9 shows a block diagram implementing both the filtering or equalization process computing the actual $y[n]$ and the adaptation or updating of the filter vector $\mathbf{w}[n]$.

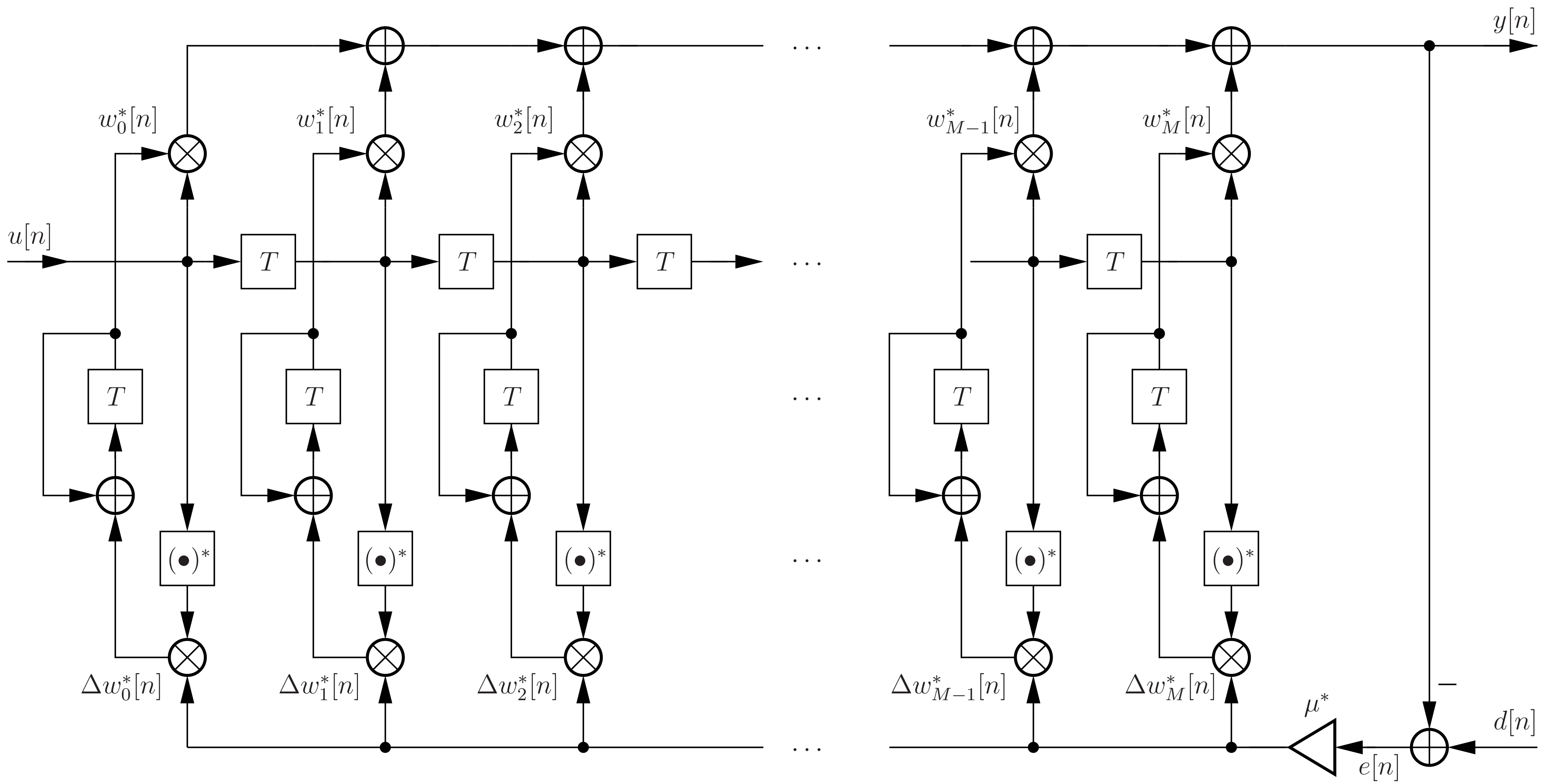


Fig. 3.9. Block Diagram of the Implementation of the LMS

4. High Resolution Direction-of-Arrival (DoA) Estimation

In Section 3.2 we were dealing with the beamforming problem, i.e. spatial filtering based on the knowledge of direction of desired as well as some undesired wavefronts. Now we will address the problem of how to get this knowledge with so called subspace based high resolution techniques. Such techniques are not limited by the aperture of the antenna array.

First we introduce two assumptions, which are important for the derivation of rather simple models and algorithms. We start with the far-field data model, which leads to planar wavefronts impinging on the array, see Fig. 4.1.

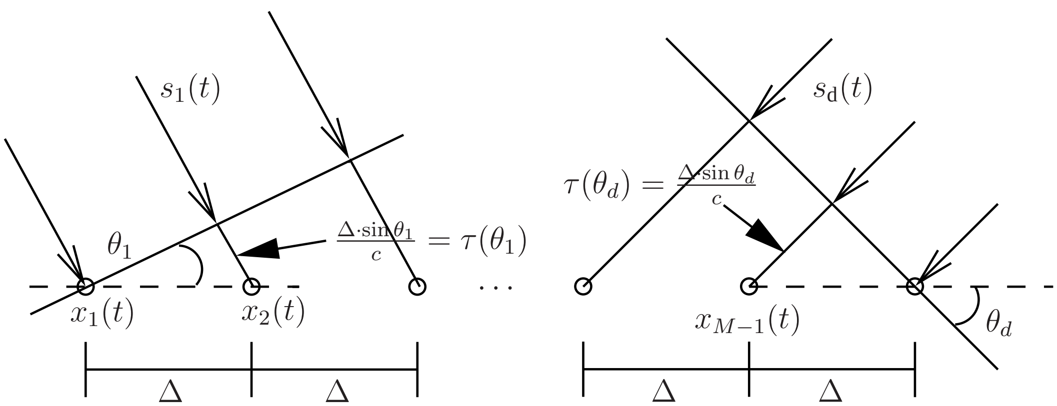


Fig. 4.1. Uniform Linear Array (ULA) with M elements with spacing Δ and d impinging planar wavefronts with angles of arrivals θ_1 up to θ_d.

The second important assumption is the narrow band data model. Here we assume, the time τ which it takes for a wavefront to propagate along the array from the first to the last sensor is very small compared to the symbol period of the data modulated onto the wavefront.

$$s_i(t-\tau) \approx s_i(t)\mathrm{e}^{-\mathrm{j}2\pi f_\mathrm{c}\tau}, \quad \forall i = 1, \ldots, d \quad \text{and} \quad \tau = (M-1)\tau(\theta_i).$$

Here f_c is the carrier frequency of the modulated radio signal and $s_i(t)$ is the complex envelope.

Therefore, we can write our receive signal vector $x(t)$ as

$$\begin{aligned}
\mathbf{x}(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_M(t) \end{bmatrix} = \sum_{i=1}^{d} \begin{bmatrix} a_1(\theta_i)\mathrm{e}^{-\mathrm{j}2\pi f_c \tau_1(\theta_i)} \\ a_2(\theta_i)\mathrm{e}^{-\mathrm{j}2\pi f_c \tau_2(\theta_i)} \\ a_3(\theta_i)\mathrm{e}^{-\mathrm{j}2\pi f_c \tau_3(\theta_i)} \\ \vdots \\ a_M(\theta_i)\mathrm{e}^{-\mathrm{j}2\pi f_c \tau_M(\theta_i)} \end{bmatrix} \cdot s_i(t) + \begin{bmatrix} n_1(t) \\ n_2(t) \\ n_3(t) \\ \vdots \\ n_M(t) \end{bmatrix} \\
&= [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \mathbf{a}(\theta_3), \ldots, \mathbf{a}(\theta_d)] \cdot \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ \vdots \\ s_d(t) \end{bmatrix} + \mathbf{n}(t) \\
&= \mathbf{A} \cdot \mathbf{s}(t) + \mathbf{n}(t).
\end{aligned}$$

Let us choose the first sensor as reference sensor. Denoting c as the speed of propagation:

$$\begin{aligned}
\tau_1(\theta_i) &= 0 \\
\tau_2(\theta_i) &= \frac{\Delta \cdot \sin\theta_i}{c} \\
&\vdots \\
\tau_M(\theta_i) &= (M-1)\frac{\Delta \cdot \sin\theta_i}{c}.
\end{aligned}$$

With that we define spatial frequencies

$$\mu_i = -\frac{2\pi f_c}{c}\Delta \sin\theta_i = \Big|_{\lambda = \frac{c}{f_c}} -\frac{2\pi}{\lambda}\Delta \sin\theta_i = \Big|_{\Delta = \frac{\lambda}{2}} -\pi \cdot \sin\theta_i$$

leading to the so called array steering vector

$$\mathbf{a}^{\mathrm{T}}(\theta_i) = \mathbf{a}^{\mathrm{T}}(\mu_i) = \left[1, \mathrm{e}^{\mathrm{j}\mu_i}, \mathrm{e}^{\mathrm{j}2\mu_i}, \ldots, \mathrm{e}^{\mathrm{j}(M-1)\mu_i}\right] \cdot a_0(\mu_i).$$

Here $a_0(\mu_i)$ is the antenna pattern of the identical sensors, which we will assume to be omnidirectional, i.e. $a_0(\mu_i) = 1$. This leads us to the array steering matrix of a ULA

$$\mathbf{A}_{\mathrm{ULA}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathrm{e}^{\mathrm{j}\mu_1} & \mathrm{e}^{\mathrm{j}\mu_2} & \cdots & \mathrm{e}^{\mathrm{j}\mu_d} \\ \mathrm{e}^{\mathrm{j}2\mu_1} & \mathrm{e}^{\mathrm{j}2\mu_2} & \cdots & \mathrm{e}^{\mathrm{j}2\mu_d} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{e}^{\mathrm{j}(M-1)\mu_1} & \mathrm{e}^{\mathrm{j}(M-1)\mu_2} & \cdots & \mathrm{e}^{\mathrm{j}(M-1)\mu_d} \end{bmatrix} \in \mathbb{C}^{M \times d}.$$

With this Vandermonde matrix $\mathbf{A}_{\mathrm{ULA}}$ we can write our data model as

$$\mathbf{x}(t) = \mathbf{A}_{\mathrm{ULA}} \cdot \mathbf{s}(t) + \mathbf{n}(t).$$

4.1 Subspace Estimation

The true signal subspace can be computed if we know the true signal correlation matrix

$$\begin{aligned}\mathbf{R_{xx}} &= \mathrm{E}\left[\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t)\right] = \mathrm{E}\left[(\mathbf{A}\mathbf{s}(t)+\mathbf{n}(t))(\mathbf{s}^{\mathrm{H}}(t)\mathbf{A}^{\mathrm{H}}+\mathbf{n}^{\mathrm{H}}(t))\right] \\ &= \mathbf{A}\mathbf{R_{ss}}\mathbf{A}^{\mathrm{H}}+\mathbf{R_{nn}},\end{aligned}$$

where

$$\mathbf{R_{ss}} = \mathrm{E}\left[\mathbf{s}(t)\mathbf{s}^{\mathrm{H}}(t)\right] \quad \text{and} \quad \mathbf{R_{nn}} = \mathrm{E}\left[\mathbf{n}(t)\mathbf{n}^{\mathrm{H}}(t)\right]$$

and noise and signal are uncorrelated as usual. If the noise at the antenna elements is i.i.d. we have $\mathbf{R_{nn}} = \sigma_n^2 \cdot \mathbf{1}_M$. In the noisefree case, $\mathbf{R_{xx}}$ will be rank deficient

$$\mathrm{rank}(\mathbf{R_{xx}})\Big|_{\sigma_n^2=0} = \mathrm{rank}(\mathbf{A}\mathbf{R_{ss}}\mathbf{A}^{\mathrm{H}}) = d < M.$$

The EVD of $\mathbf{R_{xx}}$ reads

$$\begin{aligned}\mathbf{R_{xx}} &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}} \Big|_{\sigma_n^2=0} = \begin{bmatrix} \mathbf{U}_{\mathrm{S}} & \mathbf{U}_{\mathrm{O}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Lambda}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_{\mathrm{S}}^{\mathrm{H}} \\ \mathbf{U}_{\mathrm{O}}^{\mathrm{H}} \end{bmatrix} \\ &= \mathbf{U}_{\mathrm{S}}\mathbf{\Lambda}_d\mathbf{U}_{\mathrm{S}}^{\mathrm{H}} \\ &= \sum_{k=1}^{d} \lambda_k \mathbf{u}_k \mathbf{u}_k^{\mathrm{H}},\end{aligned}$$

with $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_d > \lambda_{d+1} = \lambda_{d+2} = \cdots = \lambda_M = 0$ and $\mathbf{u}_k$ being the the eigenvectors of $\mathbf{R_{xx}}$ corresponding to the d nonzero eigenvalues.

The columnspace of the array steering matrix can be expressed as

$$\mathrm{image}\{\mathbf{A}\} = \mathrm{image}\{\mathbf{U}_{\mathrm{S}}\} = \mathcal{S},$$

because

$$\mathbf{A}\mathbf{R_{ss}}\mathbf{A}^{\mathrm{H}} = \mathbf{U}_{\mathrm{S}}\mathbf{\Lambda}_d\mathbf{U}_{\mathrm{S}}^{\mathrm{H}}.$$

The nullspace of $\mathbf{A}^{\mathrm{H}}$, which is the same as the noise subspace or left nullspace of $\mathbf{A}$, reads

$$\mathrm{kernel}\{\mathbf{A}^{\mathrm{H}}\} = \mathrm{kernel}\{\mathbf{U}_{\mathrm{S}}^{\mathrm{H}}\} = \mathcal{N} = \mathrm{image}\{\mathbf{U}_{\mathrm{O}}\}$$

Now taking the noise into account we have

$$\begin{aligned}\mathbf{R_{xx}} &= \begin{bmatrix} \mathbf{U}_{\mathrm{S}} & \mathbf{U}_{\mathrm{O}} \end{bmatrix} \cdot \left(\begin{bmatrix} \mathbf{\Lambda}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sigma_n^2 \cdot \mathbf{1}_M \right) \cdot \begin{bmatrix} \mathbf{U}_{\mathrm{S}}^{\mathrm{H}} \\ \mathbf{U}_{\mathrm{O}}^{\mathrm{H}} \end{bmatrix} \\ &= \sum_{k=1}^{M} p_k \mathbf{u}_k \mathbf{u}_k^{\mathrm{H}},\end{aligned}$$

with $p_k = \lambda_k + \sigma_n^2$.

But because we do not know the true correlation matrix we have to estimate the signal subspace from received snapshots of data

$$\begin{aligned}\mathbf{X} &= [\mathbf{x}(t_1), \mathbf{x}(t_2), \cdots, \mathbf{x}(t_N)] \in \mathbb{C}^{M\times N}, \quad N \ \text{snapshots.} \\ &= \mathbf{A}\cdot[\mathbf{s}(t_1), \mathbf{s}(t_2), \cdots, \mathbf{s}(t_N)] + [\mathbf{n}(t_1), \mathbf{n}(t_2), \cdots, \mathbf{n}(t_N)] \\ &= \mathbf{A}\mathbf{S}+\mathbf{N},\end{aligned}$$

where we have assumed that our scenario does not change during the time from t_1 to t_N and $\mathbf{A}$ therefore stays constant.

$$\hat{\mathbf{R}}_{\mathbf{xx}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}(t_n)\mathbf{x}^{\mathrm{H}}(t_n) = \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{H}}$$

is the estimate of the correlation matrix obtained from the received samples. We can either compute an EVD of $\hat{\mathbf{R}}_{\mathbf{xx}}$ or an SVD of $\mathbf{X}$

$$\begin{aligned}
\mathbf{X} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} = \begin{bmatrix} \mathbf{U}_{\mathrm{s}} & \mathbf{U}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_{\mathrm{s}}^{\mathrm{H}} \\ \mathbf{V}_{\mathrm{o}}^{\mathrm{H}} \end{bmatrix} \\
\hat{\mathbf{R}}_{\mathbf{xx}} &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}} = \begin{bmatrix} \mathbf{U}_{\mathrm{s}} & \mathbf{U}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_{\mathrm{s}}^{\mathrm{H}} \\ \mathbf{U}_{\mathrm{o}}^{\mathrm{H}} \end{bmatrix}
\end{aligned}$$

with $\mathbf{\Lambda} = \frac{1}{N}\mathbf{\Sigma}\mathbf{\Sigma}^{\mathrm{T}}$, where we have assumed $N > M$.

The link between true and estimated subspace is

$$\begin{aligned}
\mathcal{S} &= \mathrm{image}\{\mathbf{U}_{\mathrm{S}}\} = \lim_{N\to\infty}\left(\mathrm{image}\{\mathbf{U}_{\mathrm{s}}\}\right) \quad \text{and} \\
\mathcal{N} &= \mathrm{image}\{\mathbf{U}_{\mathrm{O}}\} = \lim_{N\to\infty}\left(\mathrm{image}\{\mathbf{U}_{\mathrm{o}}\}\right).
\end{aligned}$$

4.2 MUltiple SIgnal Classification (MUSIC)

MUSIC is an algorithm for DoA estimation based on the estimated noise subspace image$\{\mathbf{U}_{\mathrm{o}}\}$. We assume the array manifold to be known, i.e. we know the functional dependence of the array steering vector on the angle of arrival $\mathbf{a}(\theta)$.

Any vector $\mathbf{a}(\theta_i)$ is element of the signal subspace $\mathcal{S}$, since $\mathcal{S} = \mathrm{image}\{\mathbf{A}\}$ and, therefore, is orthogonal to any vector element of the noise subspace, or in other words, we know the array geometry:

$$\mathbf{a}^{\mathrm{H}}(\theta_i)\mathbf{U}_{\mathrm{O}} = [0, 0, \cdots, 0] \approx \mathbf{a}^{\mathrm{H}}(\theta_i)\mathbf{U}_{\mathrm{o}} \quad \forall i = 1, \ldots, d.$$

The so called MUSIC spectrum is defined as

$$\begin{aligned}
S_{\mathrm{MUSIC}}(\theta) &= \frac{\|\mathbf{a}(\theta)\|_2^2}{\|\mathbf{a}^{\mathrm{H}}\mathbf{U}_{\mathrm{o}}\|_2^2} \\
&= \frac{\mathbf{a}^{\mathrm{H}}(\theta)\mathbf{a}(\theta)}{\mathbf{a}^{\mathrm{H}}(\theta)\mathbf{U}_{\mathrm{o}}\mathbf{U}_{\mathrm{o}}^{\mathrm{H}}\mathbf{a}(\theta)} \\
&= \frac{\mathbf{a}^{\mathrm{H}}(\theta)\mathbf{a}(\theta)}{\mathbf{a}^{\mathrm{H}}(\theta)\mathbf{P}_{\mathrm{o}}\mathbf{a}(\theta)},
\end{aligned}$$

with $\mathbf{P}_{\mathrm{o}} = \mathbf{U}_{\mathrm{o}}\mathbf{U}_{\mathrm{o}}^{\mathrm{H}}$ the projector onto $\mathcal{N}$.

Performing a spectral search of $S_{\mathrm{MUSIC}}(\theta)$ we obtain estimates for the θ_i's. By varying θ, we see that θ approaching a θ_i leads to a denominator, which is zero or at least very little. Therefore, the maxima of $S_{\mathrm{MUSIC}}(\theta)$ are indicating, that the argument θ is an estimate for θ_i. Fig. 4.2 depicts such a MUSIC spectrum.

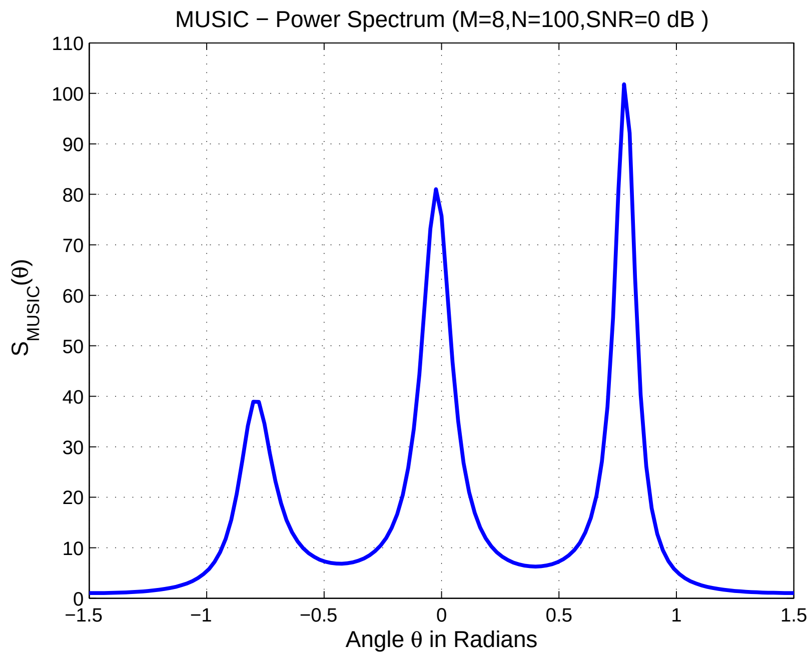


Fig. 4.2. MUSIC Spectrum with $M = 8$ antennas, $d = 3$ impinging wavefronts and $N = 100$ snapshots with SNR $= 0$ dB.

4.3 The Standard ESPRIT Algorithm

The standard ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) is also a high resolution subspace-based algorithm, but in contrast to MUSIC, it works on the signal subspace instead of the noise subspace.

The key to the application of ESPRIT is a translational shift invariance structure of the antenna array. As shown in Fig. 4.3, the array must consist of $m \geq d$ pairs of sensors, so called doublets. The two sensors of each doublet must have the same radiation pattern and there must be the same displacement Δ between the sensors of each pair. Therefore, the array can be partitioned into two subarrays, where we can map one onto the other by a simple translational shift by Δ.

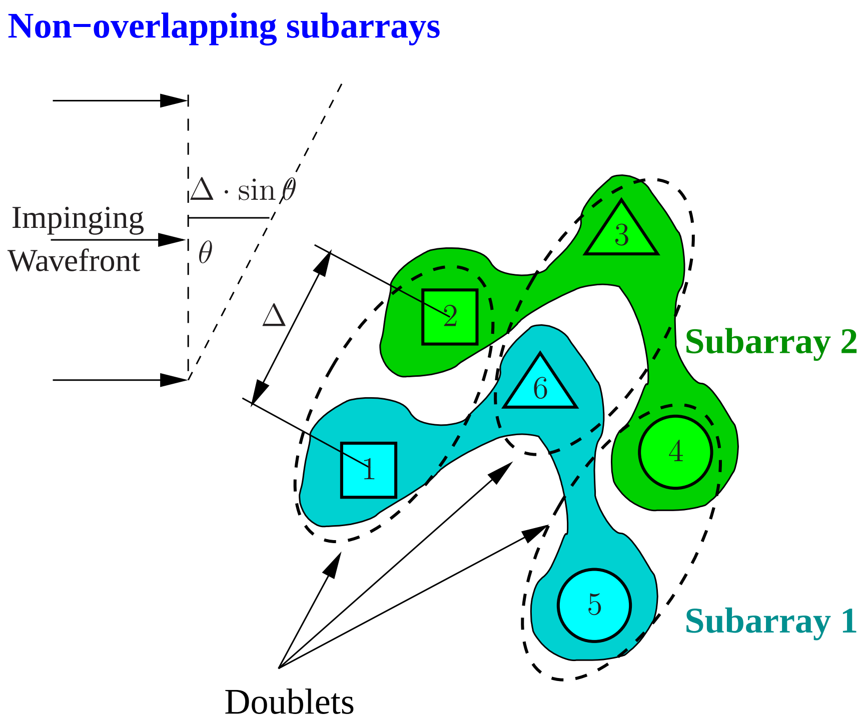


Fig. 4.3. Antenna Array with $M = 6$ antennas 3 non-overlapping doublets and a translational shift invariance structure. Different shapes for each antenna (triangles, squares and circles) represent different antenna patterns.

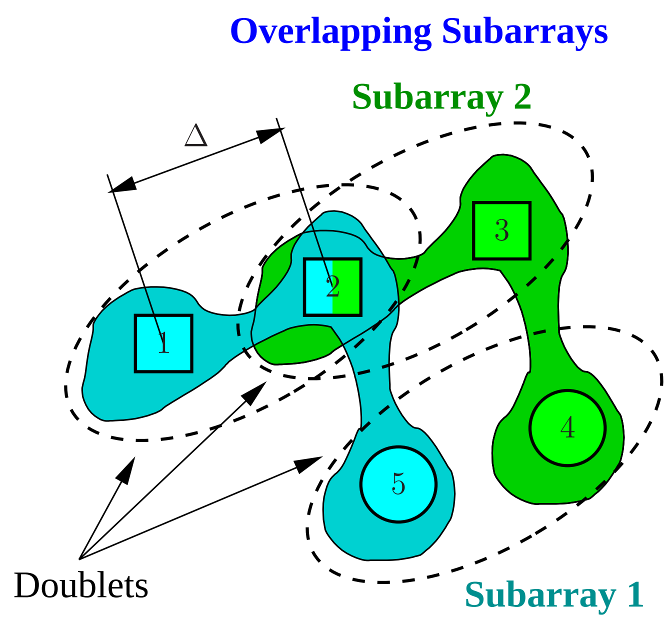


Fig. 4.4. Antenna Array with $M = 5$ antennas and 3 overlapping doublets and a translational shift invariance structure. Different shapes for each antenna (squares and circles) represent different antenna patterns.

The overall number of antenna elements is $2m = M = 6$ in the example of Fig. 4.3. Obviously, we can be more efficient by working with overlapping subarrays as it is depicted in Fig. 4.4. Furthermore, this is especially the case with a ULA as shown in Fig. 4.5.

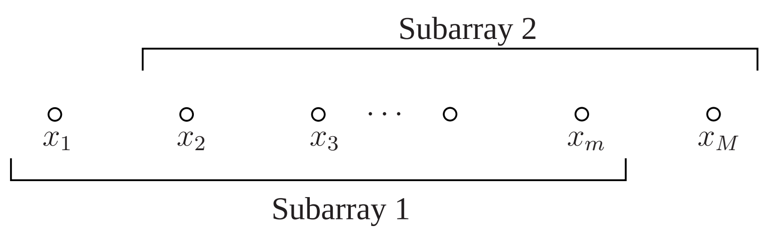


Fig. 4.5. Revealing the translational shift invariant subarray structure of a ULA with $M = m + 1$.

The receive signal vector of such an antenna array can be split into two vectors with the aid of two selection matrices

$$
\mathbf{J}_1 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = [\ \mathbf{1}_m \quad \mathbf{0}\ ],
$$

$$
\mathbf{J}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = [\ \mathbf{0} \quad \mathbf{1}_m\ ].
$$

Starting from

$$
\begin{aligned}
\mathbf{x}(t) &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \phi_1 & \phi_2 & \cdots & \phi_d \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1^{m-1} & \phi_2^{m-1} & \cdots & \phi_d^{m-1} \\ \phi_1^m & \phi_2^m & \cdots & \phi_d^m \end{bmatrix} \cdot \mathbf{s}(t) + \mathbf{n}(t), \quad \phi_i = \mathrm{e}^{\mathrm{j}\mu_i} \\
\mathbf{x}_1(t) &= \mathbf{J}_1\mathbf{x}(t) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \phi_1 & \phi_2 & \cdots & \phi_d \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1^{m-1} & \phi_2^{m-1} & \cdots & \phi_d^{m-1} \end{bmatrix} \cdot \mathbf{s}(t) + \mathbf{J}_1\mathbf{n}(t) = \mathbf{A}^{'}\mathbf{s}(t) + \mathbf{n}^{'}(t) \\
\mathbf{x}_2(t) &= \mathbf{J}_2\mathbf{x}(t) = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_d \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1^{m-1} & \phi_2^{m-1} & \cdots & \phi_d^{m-1} \\ \phi_1^m & \phi_2^m & \cdots & \phi_d^m \end{bmatrix} \cdot \mathbf{s}(t) + \mathbf{J}_2\mathbf{n}(t) = \mathbf{A}^{''}\mathbf{s}(t) + \mathbf{n}^{''}(t).
\end{aligned}
$$

We can see that $\mathbf{A}^{''} = \mathbf{A}^{'}\mathbf{\Phi}$ with $\mathbf{\Phi} = \mathrm{diag}\{\phi_i\}_{i=1}^{d}$. Furthermore,

$$\mathbf{J}_1\mathbf{A} = \mathbf{A}^{'},\ \mathbf{J}_2\mathbf{A} = \mathbf{A}^{''} \Rightarrow \mathbf{J}_1\mathbf{A}\mathbf{\Phi} = \mathbf{J}_2\mathbf{A}.$$

Since $\mathrm{image}\{\mathbf{A}\} = \mathrm{image}\{\mathbf{U}_\mathrm{S}\} = \mathcal{S}$ it follows that there exists a nonsingular $\mathbf{T}_\mathrm{A}$ such that

$$\mathbf{A} = \mathbf{U}_\mathrm{S}\mathbf{T}_\mathrm{A} : \ \mathbf{J}_1\mathbf{U}_\mathrm{S}\mathbf{T}_\mathrm{A}\mathbf{\Phi} = \mathbf{J}_2\mathbf{U}_\mathrm{S}\mathbf{T}_\mathrm{A} \quad \text{and}$$

$$\mathbf{J}_1\mathbf{U}_\mathrm{S}\mathbf{T}_\mathrm{A}\mathbf{\Phi}\mathbf{T}_\mathrm{A}^{-1} = \mathbf{J}_2\mathbf{U}_\mathrm{S} \quad \text{and with} \quad \mathbf{T}_\mathrm{A}\mathbf{\Phi}\mathbf{T}_\mathrm{A}^{-1} = \mathbf{\Psi}$$

we obtain the so called invariance equation

$$\mathbf{J}_1\mathbf{U}_\mathrm{S}\mathbf{\Psi} = \mathbf{J}_2\mathbf{U}_\mathrm{S}.$$

This is an overdetermined system of linear equations ($\mathbf{J}_1\mathbf{U}_\mathrm{S}\mathbf{\Psi} \in \mathbb{C}^{m\times d} : m \cdot d$ equations for d^2 unknowns). Since we know only the estimated signal subspace with basis $\mathbf{U}_\mathrm{s}$, the equality sign in the invariance equation is never fulfilled exactly. Therefore, we are looking for a least squares solution for

$$\mathbf{J}_1\mathbf{U}_\mathrm{s}\mathbf{\Psi} \approx \mathbf{J}_2\mathbf{U}_\mathrm{s}.$$

With the solution $\mathbf{\Psi}$ we are performing an EVD

$$\mathbf{\Psi} = \mathbf{T}_\mathrm{A}\mathbf{\Phi}\mathbf{T}_\mathrm{A}^{-1}$$

and with $\mu_i = \arg\phi_i$ and $\theta_i = \arcsin\left(-\frac{\lambda}{2\pi\Delta}\mu_i\right)$ we have estimates for the DoA's.

Let us now summarize the three steps of the Standard ESPRIT algorithm:

1) **Signal Subspace Estimation:** Compute $\mathbf{U}_\mathrm{s} \in \mathbb{C}^{M \times d}$ either via the
   - **Square root approach or direct data approach:** From the SVD of

$$\mathbf{X} = \begin{bmatrix} \mathbf{U}_\mathrm{s} & \mathbf{U}_\mathrm{o} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Sigma}_\mathrm{s} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_\mathrm{o} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_\mathrm{s}^\mathrm{H} \\ \mathbf{V}_\mathrm{o}^\mathrm{H} \end{bmatrix} \in \mathbb{C}^{M \times N}, \;\; N > M,$$

   as the d dominant left singular vectors of $\mathbf{X}$ or via the
   - **Covariance Approach:** From the EVD of

$$\mathbf{X}\mathbf{X}^\mathrm{H} = \begin{bmatrix} \mathbf{U}_\mathrm{s} & \mathbf{U}_\mathrm{o} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Lambda}_\mathrm{s} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_\mathrm{o} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_\mathrm{s}^\mathrm{H} \\ \mathbf{U}_\mathrm{o}^\mathrm{H} \end{bmatrix} \in \mathbb{C}^{M \times M},$$

   as the d dominant eigenvectors of $\mathbf{X}\mathbf{X}^\mathrm{H}$
2) **Solution of the Invariance Equation:**

$$\underbrace{\mathbf{J}_1 \mathbf{U}_\mathrm{s}}_{\mathbb{C}^{m \times d}} \mathbf{\Psi} \approx \underbrace{\mathbf{J}_2 \mathbf{U}_\mathrm{s}}_{\mathbb{C}^{m \times d}},$$

   by means of Least Squares (LS), Total Least Squares (TLS) or Structured Least Squares (SLS).
3) **Spatial Frequency (DoA) Estimation:** Compute the eigenvalues of $\mathbf{\Psi}$

$$\mathbf{\Psi} = \mathbf{T}_\mathrm{A} \mathbf{\Phi} \mathbf{T}_\mathrm{A}^{-1}, \quad \mathbf{\Phi} = \mathrm{diag}\{\phi_i\}_{i=1}^{d}$$
$$\mu_i = \arg \phi_i, \quad \theta_i = \arcsin\left(-\frac{\lambda}{2\pi\Delta}\mu_i\right).$$

Therefore, we have a closed form solution for the problem at hand without any numerical search like in spectral MUSIC.

4.4 Unitary ESPRIT: Real Valued Subspace Estimation

4.4.1 Centro-Hermitian Matrices

Definition: a matrix $\mathbf{M}$ is centro-Hermitian, if

$$\begin{aligned} \mathbf{M} \in \mathbb{C}^{p \times q} : \mathbf{\Pi}_p \cdot \mathbf{M}^* \cdot \mathbf{\Pi}_q &= \mathbf{M} \\ \mathbf{\Pi}_p \cdot \mathbf{M} \cdot \mathbf{\Pi}_q &= \mathbf{M}^*, \end{aligned}$$

holds. $\mathbf{\Pi}_p$ is a so-called exchange matrix, which is a square $p \times p$ matrix given by

$$\mathbf{\Pi}_p = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \in \{0,1\}^{p \times p},$$

since when multiplying a given matrix $\mathbf{A}$ with $\mathbf{\Pi}$ from the left we are exchanging the rows of the $\mathbf{A}$. Additionally, if we multiply $\mathbf{A}$ with $\mathbf{\Pi}$ from the right we are exchanging the columns of the matrix. Furthermore, note that $\mathbf{\Pi}_p^{-1} = \mathbf{\Pi}_p$.

4.4.2 Left Π-real Matrices

Definition: a matrix $\mathbf{Q}$ is left-Π-real, if

$$\begin{aligned}\mathbf{Q} \in \mathbb{C}^{p\times q} : \mathbf{\Pi}_p \cdot \mathbf{Q}^* &= \mathbf{Q} \\ \mathbf{\Pi}_p \cdot \mathbf{Q} &= \mathbf{Q}^*,\end{aligned}$$

holds.

Theorem 4.4.1. Let $\mathbf{Q}_p$ and $\mathbf{Q}_q$ being left $\mathbf{\Pi}$-real, square and nonsingular:

$$\phi : \mathbf{M} \to \mathbf{Q}_p^{-1}\mathbf{M}\mathbf{Q}_q = \mathbf{R} = \phi(\mathbf{M}).$$

ϕ maps bijectively any Centro-Hermitian matrix $\mathbf{M}$ onto a real matrix $\mathbf{R} = \mathbf{Q}_p^{-1}\mathbf{M}\mathbf{Q}_q$. □

Proof Let us take the conjugate of $\phi(\mathbf{M})$:

$$\begin{aligned}\to: (\phi(\mathbf{M}))^* &= \left(\mathbf{Q}_p^{-1}\mathbf{M}\mathbf{Q}_q\right)^* = \mathbf{Q}_p^{-1,*}\mathbf{M}^*\mathbf{Q}_q^* \\ &= \mathbf{Q}_p^{-1,*}\underbrace{\mathbf{\Pi}_p\mathbf{M}\mathbf{\Pi}_q}_{\mathbf{M}^*}\mathbf{Q}_q^* = \underbrace{\mathbf{Q}_p^{-1,*}\mathbf{\Pi}_p}_{\mathbf{Q}_p^{-1}}\mathbf{M}\underbrace{\mathbf{\Pi}_q\mathbf{Q}_q^*}_{\mathbf{Q}_q} \\ &= \mathbf{Q}_p^{-1}\mathbf{M}\mathbf{Q}_q \\ &= \phi(M) = \mathbf{R} \in \mathbb{R}^{p\times q}\end{aligned}$$

$$\begin{aligned}\leftarrow: \phi^{-1}(\mathbf{R}) &= \mathbf{Q}_p\mathbf{R}\mathbf{Q}_q^{-1} \\ \mathbf{\Pi}_p\left(\phi^{-1}(\mathbf{R})\right)^*\mathbf{\Pi}_q &= \underbrace{\mathbf{\Pi}_p\mathbf{Q}_p^*}_{\mathbf{Q}_p}\mathbf{R}\underbrace{\mathbf{Q}_q^{-1,*}\mathbf{\Pi}_q}_{\mathbf{Q}_q^{-1}} \\ &= \mathbf{Q}_p\mathbf{R}\mathbf{Q}_q^{-1} \\ &= \phi^{-1}(\mathbf{R}) = \mathbf{M}. \qquad \text{q.e.d.}\end{aligned}$$

Assume now, $\mathbf{Q}_p$ and $\mathbf{Q}_q$ are not only left $\mathbf{\Pi}$-real, square and non-singular, but also unitary, then a real-valued SVD results in

$$\phi(\mathbf{M}) = \mathbf{R} = \mathbf{Q}_p^{\mathrm{H}}\mathbf{M}\mathbf{Q}_q \overset{\mathrm{SVD}}{=} \mathbf{E}\mathbf{\Sigma}_\phi\mathbf{F}^{\mathrm{H}},$$

while a complex-valued SVD of $\mathbf{M}$ leads to

$$\begin{aligned}\mathbf{M} &= \phi^{-1}(\mathbf{E}\mathbf{\Sigma}_\phi\mathbf{F}^{\mathrm{H}}) \overset{\mathrm{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} = \mathbf{Q}_p\underbrace{\left(\mathbf{E}\mathbf{\Sigma}_\phi\mathbf{F}^{\mathrm{H}}\right)}_{\mathbf{R}}\mathbf{Q}_q^{\mathrm{H}} \\ &= (\mathbf{Q}_p\mathbf{E}) \cdot \mathbf{\Sigma}_\phi \cdot (\mathbf{Q}_q\mathbf{F})^{\mathrm{H}},\end{aligned}$$

and therefore, we have

$$\begin{aligned}\mathbf{U} &= \mathbf{Q}_p\mathbf{E} \\ \mathbf{\Sigma} &= \mathbf{\Sigma}_\phi \\ \mathbf{V} &= \mathbf{Q}_q\mathbf{F}.\end{aligned}$$

More specific, let us choose $\mathbf{Q}_p$ and $\mathbf{Q}_q$ as follows

- for p, q even:

$$\mathbf{Q}_{2n} = \frac{1}{\sqrt{2}} \cdot \left[\begin{array}{c|c} \mathbf{1}_n & \mathrm{j} \cdot \mathbf{1}_n \\ \hline \mathbf{\Pi}_n & -\mathrm{j} \cdot \mathbf{\Pi}_n \end{array}\right],$$

- for p, q odd:

$$\mathbf{Q}_{2n+1} = \frac{1}{\sqrt{2}} \cdot \left[\begin{array}{c|c|c} \mathbf{1}_n & \mathbf{0} & \mathrm{j} \cdot \mathbf{1}_n \\ \hline \mathbf{0}^{\mathrm{T}} & \sqrt{2} & \mathbf{0}^{\mathrm{T}} \\ \hline \mathbf{\Pi}_n & \mathbf{0} & -\mathrm{j} \cdot \mathbf{\Pi}_n \end{array}\right],$$

which both are left $\mathbf{\Pi}$-real, square, nonsingular and unitary!

Next, let us transform our array measurement matrix $\mathbf{X} \in \mathbb{C}^{M \times N}$ into a centro-Hermitian matrix $\mathbf{Z}$.

$$\mathbf{Z} = \left[\ \mathbf{X}, \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N\ \right] \in \mathbb{C}^{M \times 2N}.$$

Checking now, if it is centro-Hermitian:

$$\begin{aligned} \mathbf{\Pi}_M \mathbf{Z}^* \mathbf{\Pi}_{2N} &= \mathbf{\Pi}_M \left[\ \mathbf{X}^*, \quad \mathbf{\Pi}_M \mathbf{X} \mathbf{\Pi}_N\ \right] \begin{bmatrix} \mathbf{0} & \mathbf{\Pi}_N \\ \mathbf{\Pi}_N & \mathbf{0} \end{bmatrix} \\ &= \left[\ \mathbf{\Pi}_M \mathbf{X}^*, \quad \mathbf{X} \mathbf{\Pi}_N\ \right] \cdot \begin{bmatrix} \mathbf{0} & \mathbf{\Pi}_N \\ \mathbf{\Pi}_N & \mathbf{0} \end{bmatrix} \\ &= \left[\ \mathbf{X}, \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N\ \right] = \mathbf{Z}. \end{aligned}$$

Now, we apply our mapping ϕ onto the centro-Hermitian array measurement $\mathbf{Z}$.

$$\begin{aligned} \phi(\mathbf{Z}) &= \phi(\left[\ \mathbf{X}, \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N\ \right]) \\ &= \mathbf{Q}_M^{\mathrm{H}} \left[\ \mathbf{X}, \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N\ \right] \mathbf{Q}_{2N} \\ &= \mathcal{T}(\mathbf{X}) \\ \mathcal{T}(\mathbf{X}) &\in \mathbb{R}^{M \times 2N}. \end{aligned}$$

Assume now M an even number, i.e. $M = 2n$ and let us partition $\mathbf{X}$ into two submatrices

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} \in \mathbb{C}^{2n \times N},$$

with $\mathbf{X}_1 \in \mathbb{C}^{n \times N}$ and $\mathbf{X}_2 \in \mathbb{C}^{n \times N}$. Now use our specific choice of $\mathbf{Q}_p$ and $\mathbf{Q}_q$:

$$\begin{aligned} \mathcal{T}(\mathbf{X}) &= \mathbf{Q}_M^{\mathrm{H}} \left[\ \mathbf{X}, \quad \mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N\ \right] \mathbf{Q}_{2N} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{1}_n & \mathbf{\Pi}_n \\ -\mathrm{j}\mathbf{1}_n & \mathrm{j}\mathbf{\Pi}_n \end{bmatrix} \cdot \left[\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{\Pi}_n \\ \mathbf{\Pi}_n & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_1^* \\ \mathbf{X}_2^* \end{bmatrix} \cdot \mathbf{\Pi}_N \right] \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{1}_n & \mathrm{j}\mathbf{1}_n \\ \mathbf{\Pi}_n & -\mathrm{j}\mathbf{\Pi}_n \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \mathbf{1}_n & \mathbf{\Pi}_n \\ -\mathrm{j}\mathbf{1}_n & \mathrm{j}\mathbf{\Pi}_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_1 & \mathbf{\Pi}_n \mathbf{X}_2^* \mathbf{\Pi}_N \\ \mathbf{X}_2 & \mathbf{\Pi}_n \mathbf{X}_1^* \mathbf{\Pi}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_n & \mathrm{j}\mathbf{1}_n \\ \mathbf{\Pi}_n & -\mathrm{j}\mathbf{\Pi}_n \end{bmatrix}. \end{aligned}$$

Finally, we arrive at

$$\begin{aligned} \mathcal{T}(\mathbf{X}) &= \frac{1}{2} \begin{bmatrix} \mathbf{X}_1 + \mathbf{\Pi}_n \mathbf{X}_2, & \mathbf{\Pi}_n \mathbf{X}_2^* \mathbf{\Pi}_N + \mathbf{X}_1^* \mathbf{\Pi}_N \\ -\mathrm{j}\mathbf{X}_1 + \mathrm{j}\mathbf{\Pi}_n \mathbf{X}_2, & -\mathrm{j}\mathbf{\Pi}_n \mathbf{X}_2^* \mathbf{\Pi}_N + \mathrm{j}\mathbf{X}_1^* \mathbf{\Pi}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_n & \mathrm{j}\mathbf{1}_n \\ \mathbf{\Pi}_n & -\mathrm{j}\mathbf{\Pi}_n \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \mathbf{X}_1 + \mathbf{\Pi}_n \mathbf{X}_2 + \mathbf{\Pi}_n \mathbf{X}_2^* + \mathbf{X}_1^*, & \mathrm{j}\mathbf{X}_1 + \mathrm{j}\mathbf{\Pi}_n \mathbf{X}_2 - \mathrm{j}\mathbf{\Pi}_n \mathbf{X}_2^* - \mathrm{j}\mathbf{X}_1^* \\ -\mathrm{j}\mathbf{X}_1 + \mathrm{j}\mathbf{\Pi}_n \mathbf{X}_2 - \mathrm{j}\mathbf{\Pi}_n \mathbf{X}_2^* + \mathrm{j}\mathbf{X}_1^*, & \mathbf{X}_1 - \mathbf{\Pi}_n \mathbf{X}_2 - \mathbf{\Pi}_n \mathbf{X}_2^* + \mathbf{X}_1^* \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{Re}\left\{\mathbf{X}_1 + \mathbf{\Pi}_n \mathbf{X}_2^*\right\}, & -\mathrm{Im}\left\{\mathbf{X}_1 - \mathbf{\Pi}_n \mathbf{X}_2^*\right\} \\ \mathrm{Im}\left\{\mathbf{X}_1 + \mathbf{\Pi}_n \mathbf{X}_2^*\right\}, & \mathrm{Re}\left\{\mathbf{X}_1 - \mathbf{\Pi}_n \mathbf{X}_2^*\right\} \end{bmatrix} \in \mathbb{R}^{M \times 2N}. \end{aligned}$$

Therefore, after a lengthy derivation we have arrived at a simple and beautiful result: we have a real-valued matrix, which we obtain from the complex-valued measurement by really simple calculations (no actual multiplication needed!). Now we can perform the computationally simpler real-valued SVD of $\mathcal{T}(\mathbf{X})$ and relate it to the complex valued SVD of $\mathbf{Z}$:

$$
\begin{aligned}
\mathcal{T}(\mathbf{X}) &= \phi(\mathbf{Z}) = \mathbf{E} \cdot \mathbf{\Sigma} \cdot \mathbf{F}^{\mathrm{H}} \\
\mathbf{Z} &= \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^{\mathrm{H}} = \mathbf{Q}_M \cdot \mathbf{E} \cdot \mathbf{\Sigma} \cdot \mathbf{F}^{\mathrm{H}} \cdot \mathbf{Q}_{2N}^{\mathrm{H}} \\
&= \mathbf{Q}_M \cdot \begin{bmatrix} \mathbf{E}_{\mathrm{s}}, & \mathbf{E}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{s}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{F}_{\mathrm{s}}^{\mathrm{H}} \\ \mathbf{F}_{\mathrm{o}}^{\mathrm{H}} \end{bmatrix} \cdot \mathbf{Q}_{2N}^{\mathrm{H}} \\
&= \begin{bmatrix} \mathbf{U}_{\mathrm{s}}, & \mathbf{U}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{s}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_{\mathrm{s}}^{\mathrm{H}} \\ \mathbf{V}_{\mathrm{o}}^{\mathrm{H}} \end{bmatrix}
\end{aligned}
$$

$$
\Rightarrow \mathbf{U}_{\mathrm{s}} = \mathbf{Q}_M \mathbf{E}_{\mathrm{s}} \quad \text{or} \quad \mathbf{E}_{\mathrm{s}} = \mathbf{Q}_M^{\mathrm{H}} \mathbf{U}_{\mathrm{s}}.
$$

4.4.3 Real Valued Invariance Equation

From Standard ESPRIT we know

$$
\begin{aligned}
\mathbf{X} &= \mathbf{A}\mathbf{S} + \mathbf{N} = \mathbf{U}_{\mathrm{s}}\mathbf{\Sigma}_{\mathrm{s}}\mathbf{V}_{\mathrm{s}}^{\mathrm{H}} + \mathbf{U}_{\mathrm{o}}\mathbf{\Sigma}_{\mathrm{o}}\mathbf{V}_{\mathrm{o}}^{\mathrm{H}} \\
\mathbf{J}_1\mathbf{A}\mathbf{\Phi} &\approx \mathbf{J}_2\mathbf{A}, \\
\text{image}\{\mathbf{X}\} &= \text{image}\{\mathbf{U}_{\mathrm{s}}\} = \text{image}\{\mathbf{A}\},
\end{aligned}
$$

therefore,

$$
\mathbf{A} = \mathbf{U}_{\mathrm{s}} \cdot \mathbf{T}_{\mathrm{A}},
$$

where $\mathbf{T}_{\mathrm{A}}$ is square and nonsingular. Hence,

$$
\begin{aligned}
\mathbf{J}_1\mathbf{U}_{\mathrm{s}}\mathbf{T}_{\mathrm{A}}\mathbf{\Phi} &\approx \mathbf{J}_2\mathbf{U}_{\mathrm{s}}\mathbf{T}_{\mathrm{A}} \\
\mathbf{J}_1\mathbf{U}_{\mathrm{s}}\mathbf{\Psi} &\approx \mathbf{J}_2\mathbf{U}_{\mathrm{s}},
\end{aligned}
$$

where

$$
\mathbf{\Psi} = \mathbf{T}_{\mathrm{A}}\mathbf{\Phi}\mathbf{T}_{\mathrm{A}}^{-1}.
$$

Now let us define a matrix $\mathbf{D}$

$$
\begin{aligned}
\mathbf{D} = \mathbf{Q}_M^{\mathrm{H}}\mathbf{A} \quad &\rightarrow \quad \mathbf{A} = \mathbf{Q}_M\mathbf{D} \\
\mathbf{J}_1\mathbf{Q}_M\mathbf{D}\mathbf{\Phi} &\approx \mathbf{J}_2\mathbf{Q}_M\mathbf{D} \quad | \rightarrow \mathbf{Q}_m^{\mathrm{H}} \\
\mathbf{Q}_m^{\mathrm{H}}\mathbf{J}_1\mathbf{Q}_M \cdot \mathbf{D}\mathbf{\Phi} &\approx \mathbf{Q}_m^{\mathrm{H}}\mathbf{J}_2\mathbf{Q}_M \cdot \mathbf{D},
\end{aligned}
$$

where $\mathbf{J}_1, \mathbf{J}_2 \in \mathbb{R}^{m \times M}$

$$
\begin{aligned}
\mathbf{J}_1 &= \begin{bmatrix} \mathbf{1}_m, & \mathbf{0} \end{bmatrix} \\
\mathbf{J}_2 &= \begin{bmatrix} \mathbf{0}, & \mathbf{1}_m \end{bmatrix} \\
\Rightarrow \mathbf{J}_1 &= \mathbf{\Pi}_m\mathbf{J}_2\mathbf{\Pi}_M.
\end{aligned}
$$

In addition, it is easy to show that

$$
\begin{aligned}
\mathbf{Q}_m^{\mathrm{H}} \overbrace{\boldsymbol{\Pi}_m \boldsymbol{\Pi}_m}^{\mathbf{1}_m} \mathbf{J}_2 \overbrace{\boldsymbol{\Pi}_M \boldsymbol{\Pi}_M}^{\mathbf{1}_M} \mathbf{Q}_M &= \underbrace{\mathbf{Q}_m^{\mathrm{H}} \boldsymbol{\Pi}_m}_{\mathbf{Q}_m^{*,\mathrm{H}}} \underbrace{\boldsymbol{\Pi}_m \mathbf{J}_2 \boldsymbol{\Pi}_M}_{\mathbf{J}_1} \underbrace{\boldsymbol{\Pi}_M \mathbf{Q}_M}_{\mathbf{Q}_M^*} \\
&= \left(\mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_1 \mathbf{Q}_M\right)^* \\
&= \mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_2 \mathbf{Q}_M .
\end{aligned}
$$

Let us split $\mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_2 \mathbf{Q}_M$ into its real and imaginary part

$$
\mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_2 \mathbf{Q}_M = \frac{1}{2} \left(\mathbf{K}_1 + \mathrm{j} \mathbf{K}_2\right),
$$

then it follows that

$$
\mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_1 \mathbf{Q}_M = \frac{1}{2} \left(\mathbf{K}_1 - \mathrm{j} \mathbf{K}_2\right),
$$

with

$$
\begin{aligned}
\mathbf{K}_1 &= \mathbf{Q}_m^{\mathrm{H}} \left(\mathbf{J}_1 + \mathbf{J}_2\right) \mathbf{Q}_M = 2\mathrm{Re}\left\{\mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_2 \mathbf{Q}_M\right\} \in \mathbb{R}^{m \times M} \\
\mathbf{K}_2 &= \frac{1}{\mathrm{j}} \mathbf{Q}_m^{\mathrm{H}} \left(\mathbf{J}_2 - \mathbf{J}_1\right) \mathbf{Q}_M = 2\mathrm{Im}\left\{\mathbf{Q}_m^{\mathrm{H}} \mathbf{J}_2 \mathbf{Q}_M\right\} \in \mathbb{R}^{m \times M}.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
(\mathbf{K}_1 - \mathrm{j}\mathbf{K}_2)\mathbf{D}\boldsymbol{\Phi} &\approx (\mathbf{K}_1 + \mathrm{j}\mathbf{K}_2)\mathbf{D} \\
\mathbf{K}_1 \mathbf{D}(\boldsymbol{\Phi} - \mathbf{1}) &\approx \mathrm{j}\mathbf{K}_2 \mathbf{D}(\boldsymbol{\Phi} + \mathbf{1}) \\
\mathbf{K}_1 \mathbf{D} \underbrace{(\boldsymbol{\Phi} - \mathbf{1})(\boldsymbol{\Phi} + \mathbf{1})^{-1} \frac{1}{\mathrm{j}}}_{\boldsymbol{\Omega}} &\approx \mathbf{K}_2 \mathbf{D}.
\end{aligned}
$$

The matrix $\boldsymbol{\Omega}$ can be simplified to

$$
\begin{aligned}
\boldsymbol{\Omega} &= \frac{1}{\mathrm{j}} \underbrace{(\boldsymbol{\Phi} - \mathbf{1})}_{\text{diagonal}} \underbrace{(\boldsymbol{\Phi} + \mathbf{1})}_{\text{diagonal}}^{-1} = \mathrm{diag}\left\{\frac{1}{\mathrm{j}} \cdot \frac{\phi_i - 1}{\phi_i + 1}\right\}_{i=1}^{d} \\
&= \mathrm{diag}\left\{\frac{\mathrm{e}^{\mathrm{j}\mu_i} - 1}{\mathrm{j}(\mathrm{e}^{\mathrm{j}\mu_i} + 1)}\right\}_{i=1}^{d} = \mathrm{diag}\left\{\frac{\mathrm{e}^{\mathrm{j}\frac{\mu_i}{2}} - \mathrm{e}^{-\mathrm{j}\frac{\mu_i}{2}}}{\mathrm{j}\left(\mathrm{e}^{\mathrm{j}\frac{\mu_i}{2}} + \mathrm{e}^{-\mathrm{j}\frac{\mu_i}{2}}\right)}\right\}_{i=1}^{d} \\
&= \mathrm{diag}\left\{\frac{\sin\frac{\mu_i}{2}}{\cos\frac{\mu_i}{2}}\right\}_{i=1}^{d} = \mathrm{diag}\left\{\tan\frac{\mu_i}{2}\right\}_{i=1}^{d}.
\end{aligned}
$$

Since $\mathbf{D} = \mathbf{Q}_M^{\mathrm{H}} \mathbf{A}$ and $\mathbf{E}_{\mathrm{s}} = \mathbf{Q}_M^{\mathrm{H}} \mathbf{U}_{\mathrm{s}}$ and $\mathbf{A} = \mathbf{U}_{\mathrm{s}} \mathbf{T}_{\mathrm{A}}$ we have

$$
\begin{aligned}
\mathbf{D} &= \mathbf{Q}_M^{\mathrm{H}} \mathbf{U}_{\mathrm{s}} \mathbf{T}_{\mathrm{A}} = \mathbf{E}_{\mathrm{s}} \mathbf{T}_{\mathrm{A}} \\
\mathbf{K}_1 \mathbf{E}_{\mathrm{s}} \mathbf{T}_{\mathrm{A}} \boldsymbol{\Omega} &\approx \mathbf{K}_2 \mathbf{E}_{\mathrm{s}} \mathbf{T}_{\mathrm{A}},
\end{aligned}
$$

and therefore

$$
\begin{aligned}
\mathbf{K}_1\mathbf{E}_\mathrm{s}\underbrace{\left(\mathbf{T}_\mathrm{A}\boldsymbol{\Omega}\mathbf{T}_\mathrm{A}^{-1}\right)}_{\boldsymbol{\Upsilon}} &\approx \mathbf{K}_2\mathbf{E}_\mathrm{s} \\
\mathbf{K}_1\mathbf{E}_\mathrm{s}\boldsymbol{\Upsilon} &\approx \mathbf{K}_2\mathbf{E}_\mathrm{s},
\end{aligned}
$$

which is the *real-valued Invariance Equation*!

Look at a simple example how the new selection matrices, which can be computed off-line look like:

Example 4.4.2. Uniform Linear Array with $M = 6$ antennas, overlapping sub-arrays with $m = 5$ antennas each.

$$
\mathbf{J}_1 = \begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix}
$$

$$
\mathbf{J}_2 = \begin{bmatrix} 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix}
$$

$$
\begin{aligned}
\mathbf{K}_1 &= \mathbf{Q}_m^\mathrm{H}\cdot(\mathbf{J}_1+\mathbf{J}_2)\cdot\mathbf{Q}_M \\
&= \frac{1}{\sqrt{2}}\left[\begin{array}{cc|c|cc} 1&0&0&0&1\\ 0&1&0&1&0\\ \hline 0&0&\sqrt{2}&0&0\\ \hline -\mathrm{j}&0&0&0&\mathrm{j}\\ 0&-\mathrm{j}&0&\mathrm{j}&0 \end{array}\right]\cdot\begin{bmatrix} 1&1&0&0&0&0\\ 0&1&1&0&0&0\\ 0&0&1&1&0&0\\ 0&0&0&1&1&0\\ 0&0&0&0&1&1 \end{bmatrix}\cdot\frac{1}{\sqrt{2}}\left[\begin{array}{ccc|ccc} 1&0&0&\mathrm{j}&0&0\\ 0&1&0&0&\mathrm{j}&0\\ 0&0&1&0&0&\mathrm{j}\\ \hline 0&0&1&0&0&-\mathrm{j}\\ 0&1&0&0&-\mathrm{j}&0\\ 1&0&0&-\mathrm{j}&0&0 \end{array}\right] \\
&= \begin{bmatrix} 1&1&0&0&0&0\\ 0&1&1&0&0&0\\ 0&0&\sqrt{2}&0&0&0\\ 0&0&0&1&1&0\\ 0&0&0&0&1&1 \end{bmatrix},
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbf{K}_2 &= \frac{1}{\mathrm{j}}\mathbf{Q}_m^\mathrm{H}\cdot(\mathbf{J}_2-\mathbf{J}_1)\cdot\mathbf{Q}_M \\
&= \begin{bmatrix} 0&0&0&-1&1&0\\ 0&0&0&0&-1&1\\ 0&0&0&0&0&\sqrt{2}\\ 1&-1&0&0&0&0\\ 0&1&-1&0&0&0 \end{bmatrix}.
\end{aligned}
$$

□

There is still one question open: *is the columnspace spanned by* $\mathbf{X}$ *and the one spanned by* $\mathbf{Z}$ *the same?* Because only then our transformation of the array measurement $\mathbf{X}$ to the centro-Hermitian matrix $\mathbf{Z}$ will not change the estimated directions of arrival (DoA's).

To assure that, we have to use centro-symmetric arrays, which are characterized by the following equation:

$$\mathbf{\Pi}_M \mathbf{A}^* = \mathbf{A}\mathbf{\Delta},$$

where $\mathbf{A}$ is the steering matrix and $\mathbf{\Delta}$ being a diagonal and unitary matrix.

For a ULA this holds, as we will see:

$$\mathbf{\Pi}_M \mathbf{A}^*_{\mathrm{ULA}} = \mathbf{A}_{\mathrm{ULA}}\mathbf{\Delta}_{\mathrm{ULA}}$$

with

$$\mathbf{\Delta}_{\mathrm{ULA}} = \mathbf{\Phi}^{-(M-1)} = \mathrm{diag}\left\{\mathrm{e}^{-\mathrm{j}(M-1)\mu_i}\right\}_{i=1}^{d}.$$

Therefore, we can compute

$$\begin{aligned}
\mathbf{A}_{\mathrm{ULA}}\mathbf{\Delta}_{\mathrm{ULA}} &= \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathrm{e}^{\mathrm{j}\mu_1} & \mathrm{e}^{\mathrm{j}\mu_2} & \cdots \mathrm{e}^{\mathrm{j}\mu_d} & \\ \mathrm{e}^{\mathrm{j}2\mu_1} & \mathrm{e}^{\mathrm{j}2\mu_2} & \cdots \mathrm{e}^{\mathrm{j}2\mu_d} & \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{e}^{\mathrm{j}(M-1)\mu_1} & \mathrm{e}^{\mathrm{j}(M-1)\mu_2} & \cdots \mathrm{e}^{\mathrm{j}(M-1)\mu_d} & \end{bmatrix}}_{\mathbf{A}_{\mathrm{ULA}}} \cdot \underbrace{\begin{bmatrix} \mathrm{e}^{-\mathrm{j}(M-1)\mu_1} & 0 & \cdots & 0 \\ 0 & \mathrm{e}^{-\mathrm{j}(M-1)\mu_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathrm{e}^{-\mathrm{j}(M-1)\mu_d} \end{bmatrix}}_{\mathbf{\Delta}_{\mathrm{ULA}}} \\
&= \begin{bmatrix} \mathrm{e}^{-\mathrm{j}(M-1)\mu_1} & \mathrm{e}^{-\mathrm{j}(M-1)\mu_2} & \cdots \mathrm{e}^{-\mathrm{j}(M-1)\mu_d} & \\ \mathrm{e}^{-\mathrm{j}(M-2)\mu_1} & \mathrm{e}^{-\mathrm{j}(M-2)\mu_2} & \cdots \mathrm{e}^{-\mathrm{j}(M-2)\mu_d} & \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{e}^{\mathrm{j}\mu_1} & \mathrm{e}^{\mathrm{j}\mu_2} & \cdots \mathrm{e}^{\mathrm{j}\mu_d} & \\ 1 & 1 & \cdots & 1 \end{bmatrix} \\
&= \mathbf{\Pi}_M \mathbf{A}^*_{\mathrm{ULA}} \qquad \text{q.e.d.}
\end{aligned}$$

Therefore, ULA's are centro-symmetric and now $\mathrm{image}\{\mathbf{A}\} = \mathrm{image}\{\mathbf{A}\mathbf{\Delta}\}$ because $\mathbf{\Delta}$ is diagonal but $\mathbf{A}\mathbf{\Delta} = \mathbf{\Pi}_M \mathbf{A}^*$ and hence,

$$\mathrm{image}\{\mathbf{\Pi}_M \mathbf{A}^*\} = \mathrm{image}\{\mathbf{A}\} = \mathrm{image}\{\mathbf{U}_{\mathrm{s}}\},$$

and the augmentation of $\mathbf{X}$ by $\mathbf{\Pi}_M \mathbf{X}^* \mathbf{\Pi}_N$ has not changed the subspace!

Let us summarize, what we have achieved. We have replaced the complex-valued subspace estimation of a $M \times N$ matrix by a real-valued one of a $M \times 2N$ matrix. This reduces the number of real-valued multiplications by a factor of two. Since the subspace estimation is the most involved step of ESPRIT, this is a substantial reduction. In addition, UNITARY ESPRIT incorporates what is called *forward-backward averaging* (mapping from $\mathbf{X}$ to $\mathbf{Z}$) which increases the accuracy of the estimated DoA's.

Just as for the Standard ESPRIT, let us now summarize the three steps for Unitary ESPRIT

1) **Signal Subspace Estimation:** Compute $\mathbf{E}_{\mathrm{s}} \in \mathbb{R}^{M \times d}$ either via the
   - **Square root approach or direct data approach:** From the SVD of

$$\mathcal{T}(\mathbf{X}) = \begin{bmatrix} \mathbf{E}_{\mathrm{s}} & \mathbf{E}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{s}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{F}_{\mathrm{s}}^{\mathrm{H}} \\ \mathbf{F}_{\mathrm{o}}^{\mathrm{H}} \end{bmatrix} \in \mathbb{R}^{M \times 2N}, \ \ N > M,$$

     by taking the the d dominant left singular vectors of $\mathcal{T}(\mathbf{X})$. or via the
   - **Covariance Approach:** From the EVD of

$$\mathcal{T}(\mathbf{X})\left(\mathcal{T}(\mathbf{X})\right)^{\mathrm{H}} = \begin{bmatrix} \mathbf{E}_{\mathrm{s}} & \mathbf{E}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\Lambda}_{\mathrm{s}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{\mathrm{o}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{E}_{\mathrm{s}}^{\mathrm{H}} \\ \mathbf{E}_{\mathrm{o}}^{\mathrm{H}} \end{bmatrix} \in \mathbb{R}^{M \times M},$$

     as the d dominant eigenvectors of $\mathcal{T}(\mathbf{X})\left(\mathcal{T}(\mathbf{X})\right)^{\mathrm{H}}$
2) **Solution of the Invariance Equation:** Then solve,

$$\underbrace{\mathbf{K}_1\mathbf{E}_{\mathrm{s}}}_{\mathbb{R}^{m \times d}} \mathbf{\Upsilon} \approx \underbrace{\mathbf{K}_2\mathbf{E}_{\mathrm{s}}}_{\mathbb{R}^{m \times d}},$$

   by means of Least Squares (LS), Total Least Squares (TLS) or Structured Least Squares (SLS).
3) **Spatial Frequency (DoA) Estimation:** Compute the eigenvalues of the resulting real-valued solution $\mathbf{\Upsilon}$

$$\mathbf{\Upsilon} = \mathbf{T}_{\mathrm{A}}\mathbf{\Omega}\mathbf{T}_{\mathrm{A}}^{-1} \in \mathbb{R}^{d \times d}, \quad \text{with} \quad \mathbf{\Omega} = \mathrm{diag}\{\omega_i\}_{i=1}^{d}.$$

   Afterward
   - **Reliability Test:** If all eigenvalues ω_i are real, the estimates will be reliable. Otherwise, start again with more measurements.
   - If all eigenvalues ω_i are real then

$$\begin{aligned} \mu_i &= 2\arctan\omega_i \\ \theta_i &= \arcsin\left(-\frac{\lambda}{2\pi\Delta}\mu_i\right). \end{aligned}$$

5. Signal Reconstruction

In many cases not the DoA's but the wavefronts, i.e. their complex envelopes (signals), are of interest. Consider the case when we obtain an estimate of the signals or wavefronts by applying a linear filter $\mathbf{W}^{\mathrm{H}} \in \mathbb{C}^{d \times M}$ on the received data $\mathbf{X}$, i.e. the estimate of the signals is given by

$$\hat{\mathbf{S}} = [\hat{\mathbf{s}}(t_1), \hat{\mathbf{s}}(t_2), \cdots, \hat{\mathbf{s}}(t_N)] = \mathbf{W}^{\mathrm{H}} \cdot \mathbf{X},$$

and let us recall that the data model is

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N},$$

where $\mathbf{A}$ now is assumed to be known from the DoA estimation! To this end, let us now consider different alternatives for computing the receive filter $\mathbf{W}^{\mathrm{H}}$:

1) LS Solution (**L**east **S**quares)
2) MVDR Solution (**M**inimum **V**ariance, **D**istortionless **R**esponse),
3) MMSE Solution (**M**inimum **M**ean **S**quare **E**rror),
4) MF Solution (**M**atched **F**ilter).

5.1 LS Solution

We assume no statistical knowledge about $\mathbf{S}$ and $\mathbf{N}$ and that

$$\mathbf{X} \approx \mathbf{A}\mathbf{S},$$

hence

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmin}} \, \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2.$$

The solution to this optimization problem is obtained with a pseudo-inverse

$$\begin{aligned} \mathbf{A}^{\mathrm{H}}\mathbf{X} &= \mathbf{A}^{\mathrm{H}}\mathbf{A}\hat{\mathbf{S}} \\ (\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{X} &= (\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{A}\hat{\mathbf{S}} \\ \hat{\mathbf{S}} &= (\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{X} = \mathbf{A}^{+}\mathbf{X} \\ &= \mathbf{W}^{\mathrm{H}}\mathbf{X}. \end{aligned}$$

Therefore, the linear reconstruction filter optimal in the aforementioned sense is with

$$\begin{aligned} \mathbf{W}_{\mathrm{LS}}^{\mathrm{H}} &= \mathbf{A}^{+}, \\ \mathbf{W}_{\mathrm{LS}}^{\mathrm{H}}\mathbf{A} &= \mathbf{1}_d. \end{aligned}$$

The estimate of the signals is then

$$\begin{aligned}\hat{\mathbf{S}} &= \mathbf{W}^{\mathrm{H}}(\mathbf{AS}+\mathbf{N}) \\ &= \mathbf{S}+\mathbf{W}^{\mathrm{H}}\mathbf{N},\end{aligned}$$

which is an unbiased estimate of $\mathbf{S}$.

5.2 MVDR Solution

Again let us constrain that

$$\mathbf{W}_{\mathrm{MVDR}}^{\mathrm{H}}\mathbf{A} = \mathbf{1}_d,$$

i.e. we desire a *distortionless response* leading to an unbiased estimate, but additionally we would also like to minimize the total output power since that also *minimizes* the noise power (*variance*), hence the name *minimum variance distortionless response*. We consider the d individual outputs of our reconstruction filter by partitioning the filter matrix into beamforming vectors

$$\mathbf{W}_{\mathrm{MVDR}}^{\mathrm{H}} = \begin{bmatrix} \mathbf{w}_1^{\mathrm{H}} \\ \mathbf{w}_2^{\mathrm{H}} \\ \vdots \\ \mathbf{w}_d^{\mathrm{H}} \end{bmatrix} \in \mathbb{C}^{d\times M}.$$

Let us set up our optimization problem

$$\mathbf{w}_{i,\mathrm{MVDR}} = \operatorname*{argmin}_{\mathbf{w}_i} \mathrm{E}\left[|\hat{s}_i[n]|^2\right], \qquad \text{s.t.} \quad \mathbf{w}_i^{\mathrm{H}}\cdot\mathbf{A} = \mathbf{e}_i^T,$$

where $\mathbf{e}_i \in \{0,1\}^d$ is an all-zero vector with a 1 only at the i-th entry.

Thus, we have

$$\begin{aligned}\hat{s}_i[n] &= \mathbf{w}_i^{\mathrm{H}}\cdot\mathbf{x}[n] \\ &= \mathbf{w}_i^{\mathrm{H}}\left(\mathbf{As}[n]+\mathbf{n}[n]\right) \\ &= \mathbf{e}_i^T\mathbf{s}[n]+\mathbf{w}_i^{\mathrm{H}}\mathbf{n}[n] \\ &= s_i[n]+\mathbf{w}_i^{\mathrm{H}}\mathbf{n}[n].\end{aligned}$$

Therefore, we can find the MVDR solution from either one of two optimization problems

$$\begin{aligned}(*)\quad \mathbf{w}_{i,\mathrm{MVDR}} &= \operatorname*{argmin}_{\mathbf{w}_i} \mathrm{E}\left[|\mathbf{w}_i^{\mathrm{H}}\mathbf{n}[n]|^2\right] = \operatorname*{argmin}_{\mathbf{w}_i} \mathbf{w}_i^{\mathrm{H}}\mathbf{R}_{\mathbf{nn}}\mathbf{w}_i \\ (**)\quad \mathbf{w}_{i,\mathrm{MVDR}} &= \operatorname*{argmin}_{\mathbf{w}_i} \mathrm{E}[|\mathbf{w}_i^{\mathrm{H}}\underbrace{\left(\mathbf{As}[n]+\mathbf{n}[n]\right)}_{\mathbf{x}[n]}|^2] = \operatorname*{argmin}_{\mathbf{w}_i} \mathbf{w}_i^{\mathrm{H}}\mathbf{R}_{\mathbf{xx}}\mathbf{w}_i.\end{aligned}$$

where

$$\mathbf{R}_{\mathbf{xx}} = \mathbf{A}\mathbf{R}_{\mathbf{ss}}\mathbf{A}^{\mathrm{H}}+\mathbf{R}_{\mathbf{nn}}.$$

From $(**)$, we obtain with the Lagrangian function

$$\begin{aligned}
L(\mathbf{w}_i, \boldsymbol{\lambda}_i) &= \mathbf{w}_i^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}} \mathbf{w}_i + \boldsymbol{\lambda}_i^{\mathrm{H}} \left(\mathbf{A}^{\mathrm{H}} \mathbf{w}_i - \mathbf{e}_i\right) + \left(\mathbf{w}_i^{\mathrm{H}} \mathbf{A} - \mathbf{e}_i^T\right) \boldsymbol{\lambda}_i \\
\frac{\partial L}{\partial \mathbf{w}_i^*} &= \mathbf{R}_{\mathbf{xx}} \mathbf{w}_i + \mathbf{A} \boldsymbol{\lambda}_i \stackrel{!}{=} \mathbf{0} \\
\Rightarrow \mathbf{w}_i &= -\mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A} \boldsymbol{\lambda}_i \\
\frac{\partial L}{\partial \boldsymbol{\lambda}_i^*} &= \mathbf{A}^{\mathrm{H}} \mathbf{w}_i - \mathbf{e}_i \stackrel{!}{=} \mathbf{0} \\
\Rightarrow -\mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A} \boldsymbol{\lambda}_i &= \mathbf{e}_i \\
\Rightarrow \boldsymbol{\lambda}_i &= \left(\mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A}\right)^{-1} \mathbf{e}_i \\
\mathbf{w}_{i,\mathrm{MVDR}} &= \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A} \left(\mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A}\right)^{-1} \mathbf{e}_i \\
\mathbf{w}_{i,\mathrm{MVDR}}^{\mathrm{H}} &= \mathbf{e}_i^{\mathrm{T}} \left(\mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A}\right)^{-1} \mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1}.
\end{aligned}$$

Hence from (**) we have

$$\mathbf{W}_{\mathrm{MVDR}}^{\mathrm{H}} = \left(\mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{A}\right)^{-1} \mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}}^{-1}.$$

Similarly, working with (*) leads to

$$\mathbf{W}_{\mathrm{MVDR}}^{\mathrm{H}} = \left(\mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{nn}}^{-1} \mathbf{A}\right)^{-1} \mathbf{A}^{\mathrm{H}} \mathbf{R}_{\mathbf{nn}}^{-1}.$$

MVDR is a *zero forcing* (ZF) solution, completely suppressing interference between impinging wavefronts and minimizing the noise.

Example 5.2.1. Special case (i.i.d. noise):

$$\mathbf{R}_{\mathbf{nn}} = \sigma_n^2 \cdot \mathbf{1}_M$$

Then the MVDR solution is

$$\begin{aligned}
\mathbf{W}_{\mathrm{MVDR}}^{\mathrm{H}} &= \left(\mathbf{A}^{\mathrm{H}} \frac{1}{\sigma_n^2} \mathbf{1}_M \mathbf{A}\right)^{-1} \mathbf{A}^{\mathrm{H}} \frac{1}{\sigma_n^2} \mathbf{1}_M \\
&= \left(\mathbf{A}^{\mathrm{H}} \mathbf{A}\right)^{-1} \mathbf{A}^{\mathrm{H}} \\
&= \mathbf{A}^{+} \\
&= \mathbf{W}_{\mathrm{LS}}^{\mathrm{H}}.
\end{aligned}$$

Hence if the noise is i.i.d. $\Rightarrow$ MVDR solution and LS solution are identical! □

5.3 MMSE Solution

For $1 \leq i \leq d$, we get for the mean square error (MSE)

$$\begin{aligned}
e_i[n] &= s_i[n] - \hat{s}_i[n] = s_i[n] - \mathbf{w}_i^{\mathrm{H}} \mathbf{x}[n] \\
\mathrm{E}\left[|e_i[n]|^2\right] &= \mathrm{E}\left[\left(s_i[n] - \mathbf{w}_i^{\mathrm{H}} \mathbf{x}[n]\right)\left(s_i^*[n] - \mathbf{x}^{\mathrm{H}}[n] \mathbf{w}_i\right)\right] \\
&= \mathrm{E}\left[|s_i[n]|^2 - \mathbf{w}_i^{\mathrm{H}} \mathbf{x}[n] s_i^*[n] - s_i[n] \mathbf{x}^{\mathrm{H}}[n] \mathbf{w}_i - \mathbf{w}_i^{\mathrm{H}} \mathbf{x}[n] \mathbf{x}^{\mathrm{H}}[n] \mathbf{w}_i\right] \\
&= \sigma_s^2 - \mathbf{w}_i^{\mathrm{H}} \mathbf{p}_i - \mathbf{p}_i^{\mathrm{H}} \mathbf{w}_i + \mathbf{w}_i^{\mathrm{H}} \mathbf{R}_{\mathbf{xx}} \mathbf{w}_i \\
&= J(\mathbf{w}_i).
\end{aligned}$$

Deriving $J(\mathbf{w}_i)$ with respect to $\mathbf{w}_i^*$

$$\begin{aligned}\frac{\partial J(\mathbf{w}_i)}{\partial \mathbf{w}_i^*} &= -\mathbf{p}_i + \mathbf{R_{xx}}\mathbf{w}_i = \mathbf{0}\\ \mathbf{R_{xx}}\mathbf{w}_i &= \mathbf{p}_i,\end{aligned}$$

for each $i = 1, \cdots, d$. For all $i = 1, \cdots, d$, we can collect each of the previous expressions to obtain

$$\mathbf{R_{xx}}\mathbf{W} = \mathbf{R_{xs}},$$

where

$$\begin{aligned}\mathbf{R_{xs}} &= [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_d] \in \mathbb{C}^{M\times d},\\ \mathbf{W} &= [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d] \in \mathbb{C}^{M\times d}.\end{aligned}$$

Hence,

$$\mathbf{W}_{\text{MMSE}} = \mathbf{R_{xx}^{-1}}\mathbf{R_{xs}}.$$

Note that

$$\begin{aligned}\mathbf{R_{xs}} &= \mathrm{E}\left[\mathbf{x}[n]\mathbf{s}^{\mathrm{H}}[n]\right] = \mathrm{E}\left[(\mathbf{A}\mathbf{s}[n] + \mathbf{n}[n]) \cdot \mathbf{s}^{\mathrm{H}}[n]\right]\\ &= \mathbf{A}\mathrm{E}\left[\mathbf{s}[n]\mathbf{s}^{\mathrm{H}}[n]\right] = \mathbf{A}\mathbf{R_{ss}}.\end{aligned}$$

Therefore, we can express the MMSE solution as

$$\mathbf{W}_{\text{MMSE}}^{\mathrm{H}} = \mathbf{R_{ss}}\mathbf{A}^{\mathrm{H}}\left(\mathbf{A}\mathbf{R_{ss}}\mathbf{A}^{\mathrm{H}} + \mathbf{R_{nn}}\right)^{-1}.$$

Applying the matrix inversion lemma, we can obtain

$$\mathbf{W}_{\text{MMSE}}^{\mathrm{H}} = \left(\mathbf{A}^{\mathrm{H}}\mathbf{R_{nn}^{-1}}\mathbf{A} + \mathbf{R_{ss}^{-1}}\right)^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{R_{nn}^{-1}}.$$

In the high SNR regime, the MMSE solution converges to the MVDR solution.

5.4 MF Solution

$$\begin{aligned}\hat{s}_i &= \mathbf{w}_i^{\mathrm{H}}\mathbf{x}_i,\\ \mathbf{x}_i &= \mathbf{a}(\mu_i)s_i + \mathbf{n}\\ \hat{s}_i &= \mathbf{w}_i^{\mathrm{H}}\mathbf{a}(\mu_i)s_i + \mathbf{w}_i^{\mathrm{H}}\mathbf{n},\\ \mathrm{E}\left[|\hat{s}_i|^2\right] &= \mathrm{E}\left[\left(\mathbf{w}_i^{\mathrm{H}}\mathbf{a}(\mu_i)s_i + \mathbf{w}_i^{\mathrm{H}}\mathbf{n}\right)\left(\mathbf{w}_i^{\mathrm{H}}\mathbf{a}(\mu_i)s_i + \mathbf{w}_i^{\mathrm{H}}\mathbf{n}\right)^*\right]\\ &= \mathbf{w}_i^{\mathrm{H}}\mathbf{a}(\mu_i)\mathbf{a}^{\mathrm{H}}(\mu_i)\mathbf{w}_i\mathrm{E}\left[|s_i|^2\right] + \mathbf{w}_i^{\mathrm{H}}\mathbf{R_{nn}}\mathbf{w}_i\\ \mathrm{SNR}_i &= \frac{\mathbf{w}_i^{\mathrm{H}}\mathbf{a}(\mu_i)\mathbf{a}^{\mathrm{H}}(\mu_i)\mathbf{w}_i\mathrm{E}\left[|s_i|^2\right]}{\mathbf{w}_i^{\mathrm{H}}\mathbf{R_{nn}}\mathbf{w}_i}.\end{aligned}$$

If

$$\begin{aligned}\mathbf{R_{nn}} &= \sigma_n^2 \cdot \mathbf{I}\\ \mathrm{E}\left[|s_i|^2\right] &= \sigma_s^2,\end{aligned}$$

then we have that

$$\mathrm{SNR}_i = \frac{\sigma_s^2}{\sigma_n^2} \frac{\mathbf{w}_i^{\mathrm{H}} \mathbf{a}(\mu_i) \mathbf{a}^{\mathrm{H}}(\mu_i) \mathbf{w}_i}{\mathbf{w}_i^{\mathrm{H}} \mathbf{w}_i}.$$

The matched filter solution is computed from

$$\mathbf{w}_{i,\mathrm{MF}} = \underset{\mathbf{w}_i}{\operatorname{argmax}} \ \mathrm{SNR}_i.$$

Hence,

$$\mathbf{w}_{i,\mathrm{MF}} = \mathbf{a}(\mu_i),$$

so

$$\mathbf{W}^{\mathrm{H}} = \mathbf{A}^{\mathrm{H}}.$$

Again, we see that in the low SNR regime (i.e. $\sigma_s^2 \ll \sigma_n^2$) the Wiener solution (MMSE) converges to the matched filter solution, while in the high SNR regime it converges to the MVDR (ZF) solution.

6. Downlink Beamforming

Consider the downlink of a single cell with M transmit antennas at the base station and with K single-antenna users as depicted in Fig. 6.1. We denote $s_k(t)$ as the signal of user k at time instance t. Additionally, we assume that there are Q_k number of paths from the base station to user k. The spatial frequency, the gain and the delay of path q_k of user k are denoted by $\mu_{k,q}$, $b_{k,q}$ and $\tau_{k,q}$.

Let us assume that the base station knows the angles of arrivals of the users after performing angle of arrival estimation in the uplink, through MUSIC or ESPRIT for instance. Based on reciprocity we can assume that the angles of departure θ from the base station to the users in the downlink are the same as the angles of arrival. Through downlink beamforming, the beamforming vector $\mathbf{w}_k \in \mathbb{C}^M$ is employed to transmit to user k. In the following we employ $\mathbf{a}_{k,q} = \mathbf{a}(\mu_{k,q}) = \mathbf{a}(\theta_{k,q})$ for the array steering vector of the angle of departure $\theta_{k,q}$.

From Fig. 6.1, we have that the received signal $x_k[n]$ of user k at time slot n, is given by

$$\begin{aligned} x_k[n] &= \sum_{q=1}^{Q_k} \left(\sum_{l=1}^{K} \mathbf{w}_l^{\mathrm{H}} s_l(nT - \tau_{k,q}) \cdot b_{k,q} \cdot \mathbf{a}_{k,q} \right) + n_k[n] + i_k[n] \\ &= \sum_{l=1}^{K} \mathbf{w}_l^{\mathrm{H}} \sum_{q=1}^{Q_k} s_l(nT - \tau_{k,q}) \cdot b_{k,q} \cdot \mathbf{a}_{k,q} + n_k[n] + i_k[n], \end{aligned}$$

where $n_k[n]$ is the noise at user k at time slot n and $i_k[n]$ is the intercell interference at user k at time slot n. We assume that $s_k[n]$, $n_k[n]$, and $i_k[n]$ are mutually uncorrelated. Additionally, we denote the power of the noise and the interference at user k as N_k and I_k, respectively, i.e.

$$\begin{aligned} \mathrm{E}\left[|n_k[n]|^2\right] &= N_k \\ \mathrm{E}\left[|i_k[n]|^2\right] &= I_k[n]. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathrm{E}\left[|x_k[n]|^2\right] = \mathrm{E}&\left[\left(\sum_{l=1}^{K} \mathbf{w}_l^{\mathrm{H}} \sum_{q_1=1}^{Q_k} s_l(nT - \tau_{k,q}) b_{k,q_1} \mathbf{a}_{k,q_1} \right) \left(\sum_{h=1}^{K} \mathbf{w}_h^{\mathrm{H}} \sum_{q_2=1}^{Q_k} s_h(nT - \tau_{k,q_2}) b_{k,q_2} \mathbf{a}_{k,q_2} \right)^{\mathrm{H}} \right] \\ &+ \; N_k + I_k[n]. \end{aligned}$$

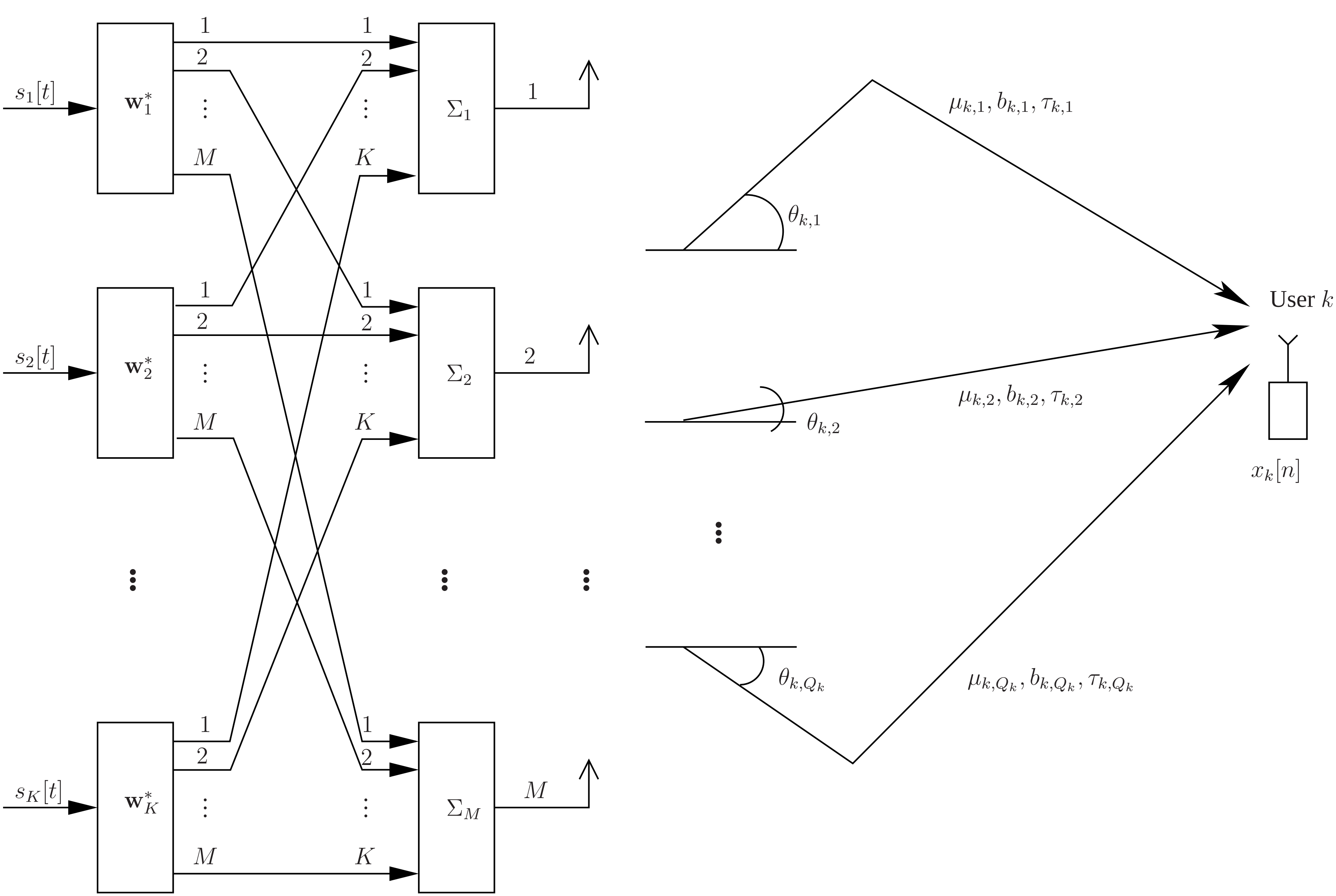


Fig. 6.1. Downlink Beamforming

In the double sums $\sum_{l=1}^{K}\sum_{q_1=1}^{Q_k}$ and $\sum_{h=1}^{K}\sum_{q_2=1}^{Q_k}$, only the terms with $l=h$ contribute to the expectation, because s_l and s_h $l \neq h$ (signals bound for different users) are uncorrelated:

$$\begin{aligned}
\mathrm{E}\left[|x_k[n]|^2\right] &= \sum_{l=1}^{K}\mathbf{w}_l^{\mathrm{H}}\mathrm{E}\left[\sum_{q_1=1}^{Q_k}\sum_{q_2=1}^{Q_k} s_l(nT-\tau_{k,q_1})s_l^*(nT-\tau_{k,q_2})b_{k,q_1}b_{k,q_2}^*\mathbf{a}_{k,q_1}\mathbf{a}_{k,q_2}^{\mathrm{H}}\right]\mathbf{w}_l + N_k + I_k[n] \\
&= \sum_{l=1}^{K}\mathbf{w}_l^{\mathrm{H}}\mathbf{R}_{kl}\mathbf{w}_l + N_k + I_k[n],
\end{aligned}$$

with

$$\mathbf{R}_{kl} = \sum_{q_1=1}^{Q_k}\sum_{q_2=1}^{Q_k}\mathbf{a}_{k,q_2}^{\mathrm{H}}\mathbf{a}_{k,q_1}\cdot \mathrm{E}\left[s_l(nT-\tau_{k,q_1})s_l^*(nT-\tau_{k,q_2})b_{k,q_1}b_{k,q_2}^*\right].$$

Note that,

$$\mathrm{E}\left[s_l(nT-\tau_{k,q_1})s_l^*(nT-\tau_{k,q_2})b_{k,q_1}b_{k,q_2}^*\right] = \begin{cases} 0 & \text{for} \quad q_1 \neq q_2 \\ \sigma_l^2\cdot \mathrm{E}\left[|b_{k,q}|^2\right] & \text{for} \quad q_1 = q_2 = q \end{cases},$$

since distinct path gains for a given user are uncorrelated. We assume that $\sigma_l^2 = 1\ \forall\ l = 1,\ldots,K$. Then,

$$\mathbf{R}_{kl} = \mathbf{R}_k = \sum_{q=1}^{Q_k}\mathbf{a}_{k,q}\mathbf{a}_{k,q}^{\mathrm{H}}\cdot \mathrm{E}\left[|b_{k,q}|^2\right],$$

i.e., it is independent of l.

The SINR for user k reads as

$$\mathrm{SINR}_k = \frac{\mathbf{w}_k^{\mathrm{H}}\mathbf{R}_k\mathbf{w}_k}{\underbrace{\sum_{l=1,l\neq k}^{K}\mathbf{w}_l^{\mathrm{H}}\mathbf{R}_k\mathbf{w}_l}_{\text{Intracell-Interference}} + \underbrace{I_k}_{\text{Intercell-}} + \underbrace{N_k}_{\text{Noise}}}.$$

Note that the transmit power assigned to user k is given by $P_k = \|\mathbf{w}_k\|_2^2$, since we have assumed $\sigma_k^2 = 1$.

The problem that we will consider in order to compute the beamforming vectors $\mathbf{w}_l$ for $l = 1,\ldots,K$ would be to minimize the total transmit power $\mathrm{P_T} = \sum_{l=1}^{K} P_k$ subject to guaranteeing a specified SINR_k for every user k.

Therefore, we consider

$$\min_{\mathbf{w}_k}\sum_{k=1}^{K}P_k = \min_{\mathbf{w}_k}\sum_{k=1}^{K}\|\mathbf{w}_k\|_2^2 \quad \text{s.t.} \quad \mathrm{SINR}_k = \frac{\mathbf{w}_k^{\mathrm{H}}\mathbf{R}_k\mathbf{w}_k}{\sum_{l=1,l\neq k}^{K}\mathbf{w}_l^{\mathrm{H}}\mathbf{R}_k\mathbf{w}_l + I_k + N_k} \quad k = 1,\ldots,K,$$

i.e. we have K quadratic constraints

$$\mathrm{SINR}_k\cdot\left(\sum_{l=1,l\neq k}^{K}\mathbf{w}_l^{\mathrm{H}}\mathbf{R}_k\mathbf{w}_l + I_k + N_k\right) = \mathbf{w}_k^{\mathrm{H}}\mathbf{R}_k\mathbf{w}_k \quad k = 1,\ldots,K.$$

This is a tough optimization problem: *quadratic optimization with quadratic constraints!*

We propose a linearized version (*Linearized Power Minimizer* (LPM)) by

1) First, choose beamforming vectors with unit length such, that the power bound for user k is optimized for every user $k = 1, \dots, K$ and
2) Second, that the norm of the beamforming vectors is adjusted such, that every user gets the SINR he needs (if possible).

6.1 First Step

Writing the first step mathematically, we have

$$\max_{\mathbf{w}_k} \mathbf{w}_k^{\mathrm{H}} \mathbf{R}_k \mathbf{w}_k \qquad \forall\, k = 1, \dots, K.$$

The solution of this problem is obvious[1]: *the eigenvector of* $\mathbf{R}_k$, *which corresponds to the largest eigenvalue of* $\mathbf{R}_k$ *will maximize the given quadratic form.*

Let us designate this eigenvector by $\mathbf{u}_k$, with $\|\mathbf{u}_k\|_2^2 = 1$.

6.2 Second Step

We obtain the beamforming vector $\mathbf{w}_k$ by scaling the eigenvector obtained in the first step by the power assigned to user k

$$\mathbf{w}_k = \sqrt{P_k} \cdot \mathbf{u}_k.$$

The K constraint equations

$$\mathrm{SINR}_k \cdot \left(\sum_{l=1, l\neq k}^{K} \mathbf{w}_l^{\mathrm{H}} \mathbf{R}_k \mathbf{w}_l + I_k + N_k \right) = \mathbf{w}_k^{\mathrm{H}} \mathbf{R}_k \mathbf{w}_k \qquad k = 1, \dots, K,$$

can be rearranged in the following way:

$$\frac{P_k \mathbf{u}_k^{\mathrm{H}} \mathbf{R}_k \mathbf{u}_k}{\mathrm{SINR}_k (N_k + I_k)} - \frac{\sum\limits_{l=1, l\neq k}^{K} P_l \mathbf{u}_l^{\mathrm{H}} \mathbf{R}_k \mathbf{u}_l}{N_k + I_k} = 1, \qquad k = 1, \dots, K.$$

Hence we can set up the K constraint equations in the following matrix-form:

$$\underbrace{\begin{bmatrix} \frac{\mathbf{u}_1^{\mathrm{H}} \mathbf{R}_1 \mathbf{u}_1}{\mathrm{SINR}_1 (N_1 + I_1)} & -\frac{\mathbf{u}_2^{\mathrm{H}} \mathbf{R}_1 \mathbf{u}_2}{N_1 + I_1} & \cdots & -\frac{\mathbf{u}_K^{\mathrm{H}} \mathbf{R}_1 \mathbf{u}_K}{N_1 + I_1} \\ -\frac{\mathbf{u}_1^{\mathrm{H}} \mathbf{R}_2 \mathbf{u}_1}{N_2 + I_2} & \frac{\mathbf{u}_2^{\mathrm{H}} \mathbf{R}_2 \mathbf{u}_2}{\mathrm{SINR}_2 (N_2 + I_2)} & \cdots & -\frac{\mathbf{u}_K^{\mathrm{H}} \mathbf{R}_2 \mathbf{u}_K}{N_2 + I_2} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\mathbf{u}_1^{\mathrm{H}} \mathbf{R}_K \mathbf{u}_1}{N_K + I_K} & -\frac{\mathbf{u}_2^{\mathrm{H}} \mathbf{R}_K \mathbf{u}_2}{N_K + I_K} & \cdots & \frac{\mathbf{u}_K^{\mathrm{H}} \mathbf{R}_K \mathbf{u}_K}{\mathrm{SINR}_K (N_K + I_K)} \end{bmatrix}}_{\mathbf{\Psi} \in \mathbb{R}^{K \times K}} \cdot \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_K \end{bmatrix}}_{\mathbf{P} \in \mathbb{R}_+^K} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathbf{1} \in \{1\}^K} .$$

[1]We have already solved a similar problem in Section 2.6.

Hence we have now a problem with K linear equations and K unknows, which we can solve as follows:

$$\begin{aligned} \mathbf{\Psi} \cdot \mathbf{P} &= \mathbf{1} \\ \mathbf{P} &= \mathbf{\Psi}^{-1} \cdot \mathbf{1} \in \mathbb{R}_{+}^{K}. \end{aligned}$$

However, note that only if all components of $\mathbf{P}$ are positive, this is a valid solution! In addition, if P_{Tmax} is the maximum available transmit power, only if

$$\sum_{l=1}^{K} P_l = \|\mathbf{P}\|_1 \leq P_{\text{Tmax}},$$

this solution can be implemented! If one of the two conditions is not fulfilled, then at least one of the K users has to be taken out. If the reduced problem then has a valid solution and $\sum_{l=1}^{K-1} P_l < P_{\text{Tmax}}$, then the scheduling algorithm can try to another user, if there are some waiting for service.

Appendix

A1 Subspaces of a Matrix

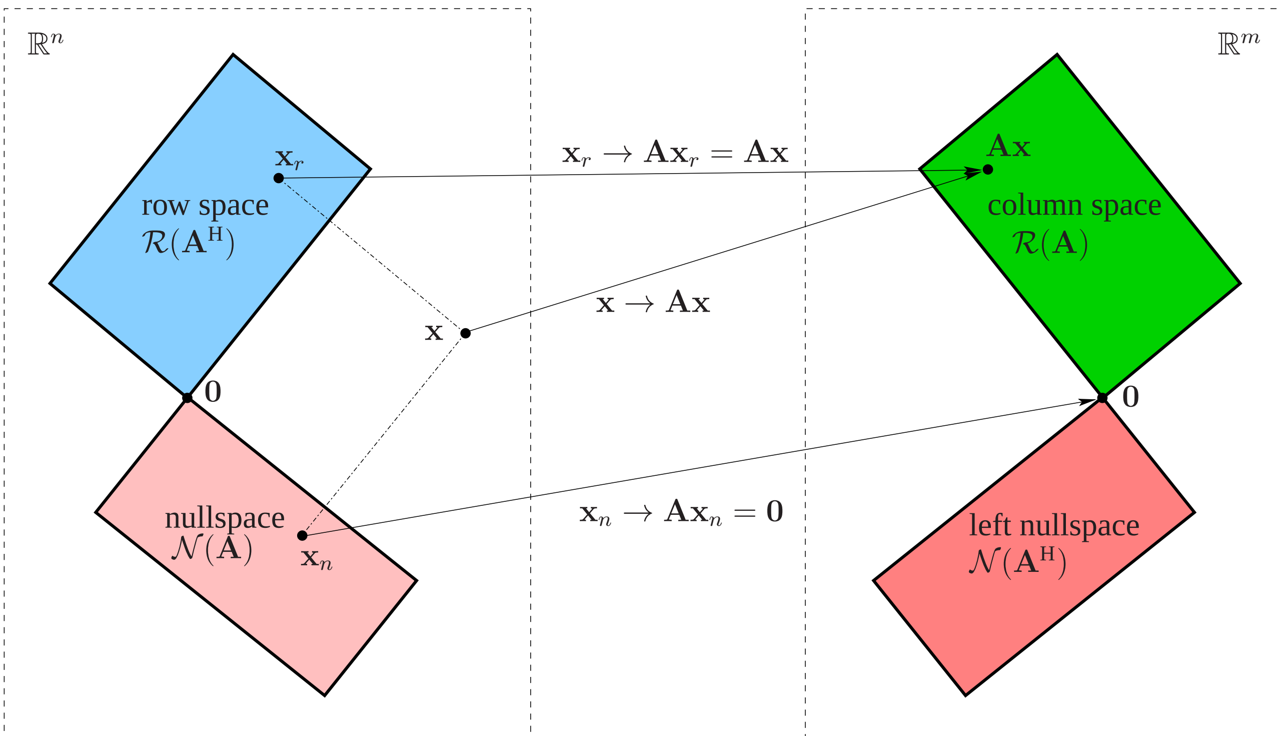


Fig. A1. The action of matrix $\mathbf{A}$. Figure taken from [2].

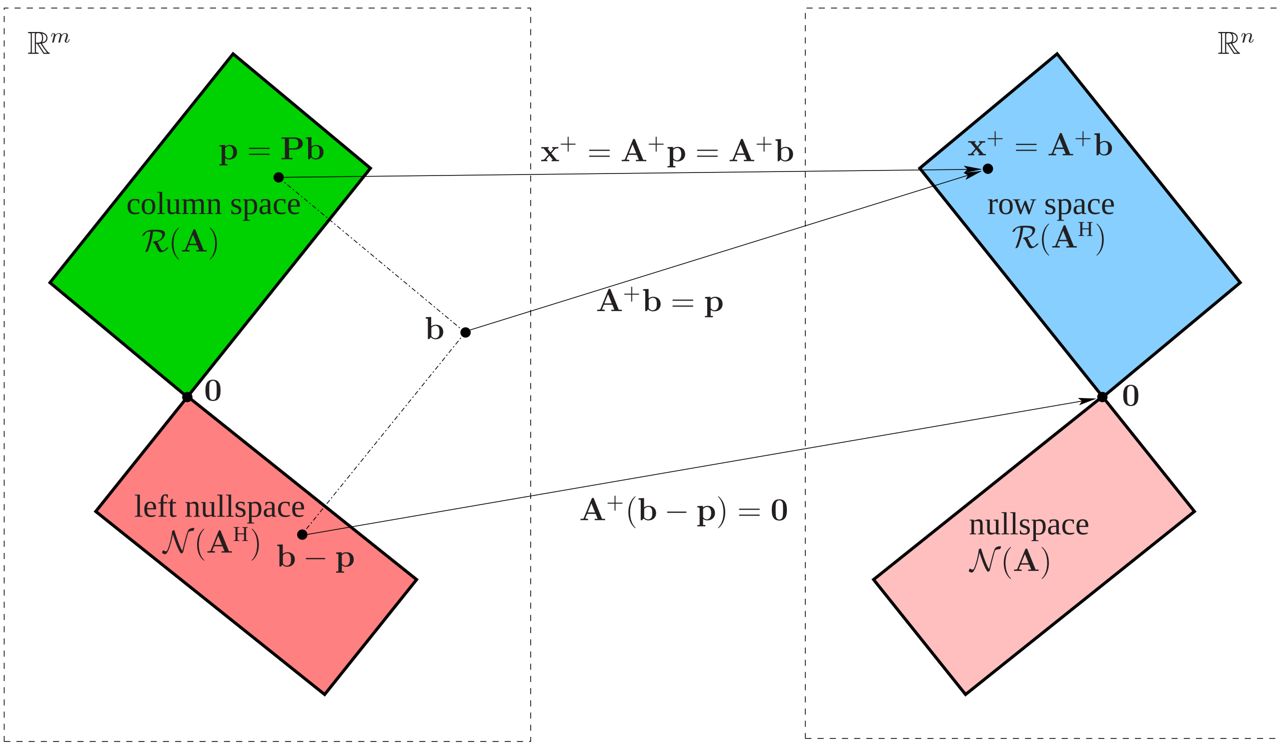


Fig. A2. The action of the pseudoinverse matrix $\mathbf{A}^+$. Figure taken from [2].

Bibliography

[1] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, 1996.
[2] Gilbert Strang. *Linear Algebra and its Applications*. Harcourt Publishers Ltd., 1988.