Signal processing antennas 4ED024

Sven Nordebo

March 1, 2011
## Contents

1 Optimum beamforming .......................... 3
   1.1 The array response function ................. 3
      1.1.1 Beamforming and spatial filtering ........ 3
      1.1.2 The array factor ....................... 5
      1.1.3 The linear array ....................... 6
      1.1.4 Matrix representation .................. 7
   1.2 The adaptive beamforming problem .......... 8
      1.2.1 Optimization using normal equations .... 8
      1.2.2 Optimization by differentiation ......... 9
   1.3 Linearly Constrained Minimum Variance Beamformer .......... 10
   1.4 Optimum beamforming ...................... 11
   1.5 Matlab programming tasks .................. 12
      1.5.1 The array response function ............ 12
      1.5.2 Linear phase FIR filter design by the window method ........ 12
      1.5.3 Optimum beamformer .................... 13

2 Adaptive beamforming ......................... 15
   2.1 The complex LMS algorithm ................. 15
   2.2 The complex RLS algorithm .................. 17
   2.3 Matlab programming tasks .................. 19
      2.3.1 The LMS and RLS algorithm ............. 19

3 Estimation of Direction of Arrival .......... 21
   3.1 Classical Estimation of Arrival ............. 21
      3.1.1 The Periodogram method for ULA .......... 22
   3.2 The Capon Minimum Variance Estimator ........ 23
   3.3 The MUSIC method for DOA estimation ........ 24
   3.4 Matlab Programming Tasks .................. 27

4 The Singular Value Decomposition in Array Processing ........ 28
   4.1 The Singular Value Decomposition .......... 28
   4.2 The Pseudoinverse ......................... 30
   4.3 Antenna Array MIMO systems ............... 31
      4.3.1 Optimum beamforming for SISO array systems .......... 32
      4.3.2 The SVD and multiple beamforming .......... 33
      4.3.3 A far–field multiscattering channel model .......... 34
      4.3.4 Information capacity .................... 35
      4.3.5 The Alamouti space–time block code .......... 36
   4.4 Matlab programming tasks .................. 38
      4.4.1 A Far–field multiscattering channel model .......... 38
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2</td>
<td>Capacity of the Rayleigh fading MIMO channel</td>
<td>38</td>
</tr>
<tr>
<td>4.4.3</td>
<td>The Alamouti space–time block code</td>
<td>39</td>
</tr>
<tr>
<td>5</td>
<td>The MIMO-OFDM system</td>
<td>40</td>
</tr>
</tbody>
</table>
Chapter 1

Optimum beamforming

1.1 The array response function

1.1.1 Beamforming and spatial filtering

Consider the receiving antenna array depicted in Fig. 1.1. The array consists of \( M \) omni-directional sensors, and each sensor \( m \) is receiving a complex bandpass signal \( \tilde{x}_m(t) \) which is downconverted to a baseband signal \( x_m(t) \). The baseband signals are then multiplied with complex weights \( w_m^* \) and summed together to yield a baseband output signal \( y(t) \).

\[
\tilde{s}(t) = s(t)e^{j\omega_c t}
\]

where \( \hat{k} \) is the unit wave vector having the direction of the wave propagation, \( c \) is the speed of wave propagation, \( \mathbf{r} \) is the spatial position vector, and

\[
\tilde{s} = s(t)e^{j\omega_c t}
\]

is the transmitted complex bandpass signal, \( s(t) \) the corresponding baseband signal and \( \omega_c \) the carrier frequency. It is often convenient to describe the cartesian components of the unit wave

![Figure 1.1: Receiving antenna array.](image-url)
vector \( \mathbf{k} \) in a spherical coordinate system as follows

\[
\hat{k} = \hat{x} \sin \theta \cos \phi + \hat{y} \sin \theta \sin \phi + \hat{z} \cos \theta \tag{1.3}
\]

where \( \theta \) and \( \phi \) are the spherical coordinate angles, respectively.

Suppose that sensor \( m \) is sampling the wave at spatial position \( \mathbf{r}_m \). Hence, the received signal at sensor \( m \) is given by

\[
\tilde{x}_m(t) = \tilde{s}(\mathbf{r}_m, t) = \tilde{s}(t - \frac{\mathbf{k} \cdot \mathbf{r}_m}{c}) = s(t - \frac{\mathbf{k} \cdot \mathbf{r}_m}{c})e^{j\omega_c(t - \frac{\mathbf{k} \cdot \mathbf{r}_m}{c})}. \tag{1.4}
\]

The received signal \( \tilde{x}_m(t) \) is downconverted using the factor \( e^{-j\omega_c t} \) yielding

\[
x_m(t) = s(t - \frac{\mathbf{k} \cdot \mathbf{r}_m}{c})e^{-j\omega_c \frac{\mathbf{k} \cdot \mathbf{r}_m}{c}}. \tag{1.5}
\]

Now, given that the signal \( s(t) \) can be considered to be sufficiently narrowbanded, the received baseband signal \( x_m(t) \) can be approximated by

\[
x_m(t) = s(t)e^{-j\omega_c \frac{\mathbf{k} \cdot \mathbf{r}_m}{c}}. \tag{1.6}
\]

The assumption is here that the baseband signal \( s(t) \) can be considered to be constant, or slowly varying across the aperture of the antenna array, and it is only the phase of the carrier that is changing. This assumption can also be formalized as

\[
\frac{L}{c} \ll \frac{1}{B} \tag{1.7}
\]

where \( L \) is the size of the antenna array and \( B \) is the bandwidth of the signal.

We introduce now the \( M \times 1 \) array response vector \( \mathbf{a}(\hat{k}) \) given by

\[
\mathbf{a}(\hat{k}) = \begin{pmatrix} e^{-j\omega_c \frac{\mathbf{k} \cdot \mathbf{r}_0}{c}} \\ e^{-j\omega_c \frac{\mathbf{k} \cdot \mathbf{r}_1}{c}} \\ \vdots \\ e^{-j\omega_c \frac{\mathbf{k} \cdot \mathbf{r}_{M-1}}{c}} \end{pmatrix} \tag{1.8}
\]

and the \( M \times 1 \) array input vector \( \mathbf{x}(t) \)

\[
\mathbf{x}(t) = \begin{pmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{M-1}(t) \end{pmatrix}. \tag{1.9}
\]

Hence, from (1.6), the array input vector can also be written

\[
\mathbf{x}(t) = s(t)\mathbf{a}(\hat{k}). \tag{1.10}
\]

Now, the array output signal is given by

\[
y(t) = \sum_{m=0}^{M-1} w_m^* x_m(t) = \mathbf{w}^H \mathbf{x}(t) \tag{1.11}
\]
where $w$ is the $M \times 1$ vector of complex weights

$$w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix}.$$  

(1.12)

From (1.10), we see that the array output signal can also be written $y(t) = w^H x(t) = s(t) w^H a(\mathbf{k})$, or

$$y(t) = s(t) H(\mathbf{k})$$  

(1.13)

where

$$H(\mathbf{k}) = w^H a(\mathbf{k}) = \sum_{m=0}^{M-1} w_m^* a_m(\mathbf{k}) = \sum_{m=0}^{M-1} w_m^* e^{-j\omega_c \mathbf{k} \cdot r_m c}$$  

(1.14)

is the array response function.

If we regard the incoming baseband signal $s(t)$ as the input signal coming from direction $(\theta, \phi)$, then from (1.13) it is clear that we may regard the array response function $H(\mathbf{k}) = H(\theta, \phi)$ as a spatial filter whose response depends on the spatial coordinates $\theta$ and $\phi$. The array response function will have characteristics which depend on the chosen set of complex weights $w_n$. In particular, with these weights properly chosen, the array will have an ability to emphasize certain directions and to cancel out or reject others.

Assuming that there is no noise present at the array, the mean output power is then

$$E = E\{y^2(t)\} = E\{|s(t)|^2|H(\mathbf{k})|^2\} = \sigma_s^2 |H(\mathbf{k})|^2$$  

(1.15)

where $\sigma_s^2$ is the variance of the signal. We will refer to the quantity $|H(\mathbf{k})|^2$ as the array gain.

Suppose that we put a constraint on the magnitude of the weight vector $w$ so that $\|w\|^2 = w^H w = 1$, and ask what the “best” possible set of weights are that maximizes the array gain $|H(\mathbf{k})|^2$ for a certain direction $\mathbf{k}$. From Cauchy–Schwarz inequality we have

$$|H(\mathbf{k})|^2 = |w^H a(\mathbf{k})|^2 \leq w^H w \cdot a^H(\mathbf{k}) a(\mathbf{k}) = a^H(\mathbf{k}) a(\mathbf{k})$$  

(1.16)

with equality iff $w = C \cdot a(\mathbf{k})$ where $C$ is a constant. Hence, the optimum weight vector $w$ that maximizes the array gain $|H(\mathbf{k})|^2$ is proportional to the array response vector $a(\mathbf{k})$. The array response vector $a(\mathbf{k})$ is therefore also often referred to as the steering vector. With the choice $w = C \cdot a(\mathbf{k})$ the array is steered towards the direction $\mathbf{k}$.

### 1.1.2 The array factor

The function $H(\mathbf{k}) = w^H a(\mathbf{k})$ in (1.14) is also called the array factor and is of fundamental importance for an antenna array.

If each individual sensor has a complex spatial (antenna) gain function $G_m(\mathbf{k})$, then the received signal at sensor $m$ is given by a simple modification of (1.6) as

$$x_m(t) = s(t) e^{-j\omega_c \mathbf{k} \cdot r_m c} G_m(\mathbf{k}) = s(t) a_m(\mathbf{k}) G_m(\mathbf{k}).$$  

(1.17)

The output signal is now given by

$$y(t) = w^H x(t) = s(t) \sum_{m=0}^{M-1} w_m^* a_m(\mathbf{k}) G_m(\mathbf{k})$$  

(1.18)
and we can write $y(t) = s(t)H(\mathbf{k})$ where $H(\mathbf{k})$ is the array response function given by

$$H(\mathbf{k}) = \sum_{m=0}^{M-1} w_m^* a_m(\mathbf{k}) G_m(\mathbf{k}). \tag{1.19}$$

If all the sensor (antenna) gain functions are equal $G_m(\mathbf{k}) = G(\mathbf{k})$, the array response function becomes

$$H(\mathbf{k}) = G(\mathbf{k}) \sum_{m=0}^{M-1} w_m^* a_m(\mathbf{k}) = G(\mathbf{k})w^H a(\mathbf{k}). \tag{1.20}$$

Hence, we see that the array factor $w^H a(\mathbf{k})$ is of fundamental importance for the antenna array. If the individual sensors (or antennas) have equal spatial response $G(\mathbf{k})$, then the array response function is simply given by $H(\mathbf{k}) = G(\mathbf{k})w^H a(\mathbf{k})$.

### 1.1.3 The linear array

Consider a linear array consisting of $M$ sensors uniformly distributed along the $y$–axis at spatial positions

$$r_m = \hat{y} \cdot md, \quad m = 0, 1, \ldots, M - 1 \tag{1.21}$$

where $d$ is the distance between elements. Suppose further that the incident wave is propagating in the $x – y$ plane with unit wave vector ($\theta = \pi/2$)

$$\mathbf{k} = \hat{x}\cos \phi + \hat{y}\sin \phi \tag{1.22}$$

The array response vector in (1.8) becomes

$$a(\phi) = \begin{pmatrix} 1 \\ e^{-j\omega_c \frac{d \sin \phi}{c}} \\ \vdots \\ e^{-j\omega_c \frac{d \sin \phi}{c}(M-1)} \end{pmatrix} \tag{1.23}$$

with elements $a_m(\phi) = e^{-j\omega_c \frac{\mathbf{k} \cdot r_m}{c}} = e^{-j\omega_c \frac{d \sin \phi}{c} m}$ for $m = 0, 1, \ldots, M - 1$.

The array response function (1.14) is thus given by

$$H(\phi) = w^H a(\phi) = \sum_{m=0}^{M-1} w_m^* a_m(\phi) = \sum_{m=0}^{M-1} w_m^* e^{-j\omega_c \frac{d \sin \phi}{c} m}. \tag{1.24}$$

We note the close resemblance of the expression (1.24) with the response of a FIR digital filter [1–3]. Using the substitution with the spatial frequency variable

$$\Omega = \omega_c \frac{d \sin \phi}{c} \tag{1.25}$$

and the digital filter response

$$H_d(\Omega) = \sum_{m=0}^{M-1} w_m^* e^{-j\Omega m} \tag{1.26}$$

we get the relationship

$$H(\phi) = H_d(\Omega) \bigg|_{\Omega = \omega_c \frac{d \sin \phi}{c}} \tag{1.27}$$
In order to avoid spatial aliasing we must require that
\[ |\omega c \frac{d \sin \phi}{c}| \leq \pi, \forall \phi \] (1.28)
due to the periodicity of the digital filter response \( H_d(\Omega) \). It is easily shown that the condition (1.28) is also equivalent with the requirement that
\[ d \leq \frac{\lambda}{2} \] (1.29)
where \( \lambda \) is the wave length corresponding to the frequency \( \omega_c \).

With \( d = \lambda/2 \), or \( \omega_c d/c = \pi \), we get the simplified expression
\[ H(\phi) = H_d(\pi \sin \phi) = \sum_{m=0}^{M-1} w_m^* e^{-j(\pi \sin \phi)m}. \] (1.30)

Suppose now that \( h(m) \) is the real impulse response of a given lowpass FIR digital filter with frequency function
\[ H_{LP}(\Omega) = \sum_{m=0}^{M-1} h(m)e^{-j\Omega m}. \] (1.31)

Let us now define the complex weights \( w_m \) by weighting the steering vector \( a(\phi_0) \) as follows
\[ w_m = h(m)e^{-j(\pi \sin \phi_0)m} \] (1.32)
where \( \phi_0 \) is some desired direction of interest. Note that with \( h(m) = 1 \) (rectangular window) this corresponds to \( w = a(\phi_0) \). With a general weighting \( h(m) \) as in (1.32), we perform a tapering of the steering vector.

The resulting array response function becomes
\[ H(\phi) = \sum_{m=0}^{M-1} w_m^* e^{-j(\pi \sin \phi)m} = \sum_{m=0}^{M-1} h(m)e^{-j\pi(\sin \phi - \sin \phi_0)m} = H_{LP}(\pi(\sin \phi - \sin \phi_0)). \] (1.33)

### 1.1.4 Matrix representation

For computer calculations and simulations it is convenient to represent the array response function in (1.14) or (1.24) on a discrete domain. Thus, suppose that we have \( I \) discrete spatial directions \( \hat{k}_i \), \( i = 1, \ldots, I \) (or \( I \) discrete spatial angles \( \phi_i \), \( i = 1, \ldots, I \)). The discrete array response function is then given by
\[ H(\hat{k}_i) = w^H a(\hat{k}_i) \] (1.34)
or, in matrix notation
\[ H = w^H A \] (1.35)
where \( H \) is an \( 1 \times I \) row vector of discrete responses
\[ H = \begin{bmatrix} H(\hat{k}_1), \ldots, H(\hat{k}_I) \end{bmatrix} \] (1.36)
and \( A \) is an \( M \times I \) response matrix given by
\[ A = \begin{bmatrix} a(\hat{k}_1), \ldots, a(\hat{k}_I) \end{bmatrix}. \] (1.37)
1.2 The adaptive beamforming problem

We consider the optimum solution to a basic adaptive beamforming problem as described below. For simplicity of notation, we consider a problem with one-dimensional spatial directions $\phi$.

Consider the following signal scenario

$$x(t) = s(t)a(\phi_0) + \sum_{i=1}^{L} s_i(t)a(\phi_i) + n(t)$$

where $s(t)$ is the desired signal with variance $\sigma_s^2$, coming from direction $\phi_0$. Further, there are $L$ undesired interfering signals $s_i(t)$ with variances $\sigma_i^2$, coming from directions $\phi_i$, and $n(t)$ is an $M \times 1$ vector of uncorrelated sensor noise $[4, 5]$ with equal variances $\sigma_n^2$.

The array output signal is given by

$$y(t) = w^H x(t).$$

All signal and noise sources are assumed to be mutually uncorrelated and have zero mean. The covariance matrix $[5]$ for the input signal $x(t)$ is thus given by

$$R = E\{x(t)x^H(t)\} = \sigma_s^2 a(\phi_0)a^H(\phi_0) + \sum_{i=1}^{L} \sigma_i^2 a(\phi_i)a^H(\phi_i) + \sigma_n^2 I = \sigma_s^2 a(\phi_0)a^H(\phi_0) + R_{NI}$$

where $R_{NI}$ is the noise and interferer part.

$$R_{NI} = \sum_{i=1}^{L} \sigma_i^2 a(\phi_i)a^H(\phi_i) + \sigma_n^2 I.$$ (1.41)

1.2.1 Optimization using normal equations

The optimum weight vector $w_o$ is the vector $w$ that minimizes the real cost function

$$f(w) = E\{|s(t) - y(t)|^2\} = E\{|s(t) - w^H x(t)|^2\}. \quad (1.42)$$

The solution is readily obtained directly from the normal equations

$$E\{x(t)(s^*(t) - y^*(t))\} = E\{x(t)(s^*(t) - x^H(t)w)\} = 0$$ \quad (1.43)

yielding

$$R w_o = p$$ \quad (1.44)

where $R = E\{x(t)x^H(t)\}$ and

$$p = E\{x(t)s^*(t)\}.$$

Note that $p$ is readily obtained from (1.38) since $s(t)$ is uncorrelated with all interferers and noise

$$p = \sigma_s^2 a(\phi_0).$$ \quad (1.46)

The final, optimum solution is hence given by

$$w_o = R^{-1} p$$ \quad (1.47)

where $R$ and $p$ are given by (1.40) and (1.46).
1.2.2 Optimization by differentiation

The cost function in (1.42) can be expanded as a quadratic form

\[ f(w) = E \{ |s(t) - w^H x(t)|^2 \} = w^H R w - p^H w - w^H p + \sigma_s^2 \] (1.48)

where \( R = E\{x(t)x^H(t)\} \), \( p = E\{x(t)s^*(t)\} \) and \( \sigma_s^2 = E\{|s(t)|^2\} \). Note that the covariance matrix \( R \) is positively definite so that \( f(w) \) is strictly convex and a unique minimum of \( f(w) \) exist.

Now, we wish to optimize the real cost function \( f(w) \) by differentiating with respect to a complex vector \( w \). In principle, the complex vector implies no complication since everything could be done using real variables only. The complex vector \( w = w_x + jw_y \) consists of a real part \( w_x \) and an imaginary part \( w_y \), and we wish to differentiate the real cost function \( f(w) = f(w_x, w_y) \) with respect to both \( w_x \) and \( w_y \). However, the real notation becomes cumbersome and it is convenient to represent the derivatives on a compact complex form.

We define the complex derivative with respect to \( w^* \) by

\[ \frac{\partial f}{\partial w^*} = \frac{1}{2} \left( \frac{\partial f}{\partial w_x} + j \frac{\partial f}{\partial w_y} \right) \] (1.49)

and note that the condition \( \frac{\partial f}{\partial w^*} = 0 \) is equivalent to the condition \( \frac{\partial f}{\partial w_x} = \frac{\partial f}{\partial w_y} = 0 \), see also [6].

Now, using the definition (1.49) we find the following simple relations

\[ \frac{\partial}{\partial w^*} w^H P = P \] (1.50)
\[ \frac{\partial}{\partial w^*} P^H w = 0 \] (1.51)
\[ \frac{\partial}{\partial w^*} w^H R w = R w \] (1.52)

The relation (1.50) can be shown as follows

\[
\frac{1}{2} \left( \frac{\partial}{\partial w_x} + j \frac{\partial}{\partial w_y} \right) \left( w_x^T P_x + w_y^T P_y - j w_y^T P_x + j w_x^T P_y \right) = \frac{1}{2} \left( P_x + j P_y + P_x + j P_y \right) = P_x + j P_y = P.
\] (1.53)

The relation (1.51) can be shown as follows

\[
\frac{1}{2} \left( \frac{\partial}{\partial w_x} + j \frac{\partial}{\partial w_y} \right) \left( w_x^T P_x + w_y^T P_y + j w_y^T P_x - j w_x^T P_y \right) = \frac{1}{2} \left( P_x + j P_y - P_x - j P_y \right) = 0.
\] (1.54)

The relation (1.52) can be shown as follows

\[
\frac{\partial}{\partial w^*} w^H R w = \frac{\partial}{\partial w^*} w^H [R w] + \frac{\partial}{\partial w^*} [w^H R] w = R w + 0 = R w
\] (1.55)

where the chain rule has been used and the vectors inside brackets \([·]\) are considered constant.

Now, by using the relations (1.50) through (1.52) we find the derivative of \( f(w) \) in (1.48) as

\[ \frac{\partial f}{\partial w^*} = R w - p \] (1.56)

and by setting \( \frac{\partial f}{\sigma_w} = R w - p = 0 \) the solution is identical with (1.47).
Completing the square

Of course, we can also do the optimization by completing the square in (1.48)

\[ f(w) = w^H Rw - p^H w - w^H p + c = (w - R^{-1}p)^H R (w - R^{-1}p) - p^H R^{-1}p + c \]  \hspace{1cm} (1.57)

and since \( R \) is positively definite, it is verified that \( w_o = R^{-1}p \) is the optimum solution and 
\( c - p^H R^{-1}p \) the optimum value of \( f(w) \).

1.3 Linearly Constrained Minimum Variance Beamformer

Consider again the same signal scenario as in (1.38) and where the covariance matrix is given by (1.40). The power of the output signal \( y(t) = w^H x(t) \) can be decomposed as

\[ E \{ |y(t)|^2 \} = w^H Rw = \sigma_s^2 |w^H a(\phi_0)|^2 + w^H R_{NI} w \]  \hspace{1cm} (1.58)

where the first term is the power of the desired signal and the second term the power of the noise and interferers.

The Linearly Constrained Minimum Variance (LCMV) beamformer [5, 6] minimizes the power of the noise and interferers \( w^H R_{NI} w \) with a linear constraint on the array response \( H(\phi_0) = w^H a(\phi_0) = g \) for the desired direction \( \phi_0 \). Hence, the LCMV optimization criterion is given by

\[ \left\{ \begin{array}{l}
\min_w w^H R_{NI} w, \text{ subject to } \\
H(\phi_0) = w^H a(\phi_0) = g 
\end{array} \right. \]  \hspace{1cm} (1.59)

The solution to this convex quadratic optimization problem is readily obtained using the Lagrange multiplier method [6]. Hence, let the Lagrange function be given by

\[ L(w, \lambda) = w^H R_{NI} w + \Re \{ \lambda^* (w^H a(\phi_0) - g) \} \]  \hspace{1cm} (1.60)

or

\[ L(w, \lambda) = w^H R_{NI} w + \lambda_x \Re \{ w^H a(\phi_0) - g \} + \lambda_y \Im \{ w^H a(\phi_0) - g \} \]  \hspace{1cm} (1.61)

where \( \lambda = \lambda_x + j\lambda_y \).

Now, since

\[ \frac{\partial}{\partial w^*} \Re \{ w^H a \} = \frac{\partial}{\partial w^*} \frac{w^H a + a^H w}{2} = \frac{1}{2} a \]  \hspace{1cm} (1.62)

we get

\[ \frac{\partial}{\partial w^*} L(w, \lambda) = R_{NI} w + \lambda^* \frac{1}{2} a(\phi_0) = 0 \]  \hspace{1cm} (1.63)

and

\[ w = -\lambda^* \frac{1}{2} R_{NI}^{-1} a(\phi_0). \]  \hspace{1cm} (1.64)

The multiplicator \( \lambda \) is found from the constraint requirement \( w^H a(\phi_0) = g \). Hence,

\[ -\lambda^* \frac{1}{2} a^H(\phi_0) R_{NI}^{-1} a(\phi_0) = g \]  \hspace{1cm} (1.65)

or

\[ \lambda = -\frac{2g}{a^H(\phi_0) R_{NI}^{-1} a(\phi_0)} \]  \hspace{1cm} (1.66)

By inserting (1.66) in (1.64) we get finally the optimal solution

\[ w = \frac{g^*}{a^H(\phi_0) R_{NI}^{-1} a(\phi_0)} R_{NI}^{-1} a(\phi_0). \]  \hspace{1cm} (1.67)
The corresponding minimum value of the cost function $w^H R_{NI} w$ is readily obtained as

$$w^H R_{NI} w = \frac{|y|^2}{a^H(\phi_0) R_{NI}^{-1} a(\phi_0)}.$$  \hspace{1cm} (1.68)

**Comparison with the adaptive beamforming problem**

It is interesting to note that the solution to the adaptive beamforming problem in (1.47) is the same as in (1.67). This can be seen by using the following matrix inversion lemma [6]:

**Lemma 1.3.1** Matrix Inversion Lemma. If the matrices $A$, $B$, $C$ and $D$ satisfy the equation

$$B^{-1} = A^{-1} + C^H D^{-1} C$$  \hspace{1cm} (1.69)

where all matrix inverses are assumed to exist, then

$$B = A - A C^H (C A C^H + D)^{-1} C A.$$  \hspace{1cm} (1.70)

**Proof:** Multiply $B$ by $B^{-1}$ above to show that $B B^{-1} = I$.

Using the matrix inversion lemma above, with $B^{-1} = R$, $A^{-1} = R_{NI}$, $C^H = a(\phi_0)$ and $D^{-1} = \sigma_s^2$, it is easily shown that the optimum solution in (1.47) is given by

$$w_o = (\sigma_s^2 a(\phi_0) a^H(\phi_0) + R_{NI})^{-1} \sigma_s^2 a(\phi_0) = R_{NI}^{-1} \sigma_s^2 a(\phi_0) \left( \frac{1 - \sigma_s^2 a^H(\phi_0) R_{NI}^{-1} a(\phi_0)}{1 + \sigma_s^2 a^H(\phi_0) R_{NI}^{-1} a(\phi_0)} \right)$$  \hspace{1cm} (1.71)

which is the same as (1.67) except for a scalar factor.

### 1.4 Optimum beamforming

Consider again the same signal scenario as in (1.38) and where the covariance matrix is given by (1.40). The power of the output signal $y(t) = w^H x(t)$ can be decomposed as

$$E \{ |y(t)|^2 \} = w^H R w = \sigma_s^2 |w^H a(\phi_0)|^2 + w^H R_{NI} w$$  \hspace{1cm} (1.72)

where the first term is the power of the desired signal and the second term the power of the noise and interferers.

The optimum beamformer can be defined as the weight vector $w$ that maximizes the signal-to-noise-and-interference ratio $SNIR$

$$SNIR = \sigma_s^2 |w^H a(\phi_0)|^2 = \sigma_s^2 \frac{w^H a(\phi_0) a(\phi_0)^H w}{w^H R_{NI} w}. \hspace{1cm} (1.73)$$

By introducing the variable substitution $y = R_{NI}^{1/2} w$ and $w = R_{NI}^{-1/2} y$ the SNIR can be written as the following Rayleigh quotient

$$SNIR = \sigma_s^2 \frac{y^H R_{NI}^{-1/2} a(\phi_0) a^H(\phi_0) R_{NI}^{-1/2} y}{y^H y}$$  \hspace{1cm} (1.74)

and we see that the optimum $y$ is the maximum eigenvector of $R_{NI}^{-1/2} a(\phi_0) a^H(\phi_0) R_{NI}^{-1/2}$.

Since

$$R_{NI}^{-1/2} a(\phi_0) a^H(\phi_0) R_{NI}^{-1/2} = a^H(\phi_0) R_{NI}^{-1} a(\phi_0) \cdot R_{NI}^{-1/2} a(\phi_0)$$  \hspace{1cm} (1.75)
it is concluded that \( y = R^{-\frac{1}{2}}_{N_1} a(\phi_0) \) is the optimum solution, or in terms of the original weight vector

\[
w = R^{-\frac{1}{2}}_{N_1} y = R^{-1}_{N_1} a(\phi_0).
\]

The optimum solution (1.76) is a generalized eigenvector and is thus scalable by a constant. Again, we see that (1.76) is consistent with the adaptive beamforming problem in (1.71) as well as the linearly constrained minimum variance (LCMV) beamformer solution in (1.67).

1.5 Matlab programming tasks

1.5.1 The array response function

This programming task is to design the weights \( \mathbf{w} \) of an equispaced linear array with \( M = 50 \) (or less), \( d = \lambda/2 \) and array response function given by (1.30). The spatial filter specification is given by

\[
\begin{align*}
-3 &\leq 20 \log_{10} |H(\phi)| \leq 0 \quad [\text{dB}] \\
30 &\leq \phi \leq 50 \quad [\text{degrees}] \\
-90 &\leq \phi \leq 20 \quad [\text{degrees}]
\end{align*}
\]

The passband is \( 30 \leq \phi \leq 50 \) [degrees] and the stopbands are \( -90 \leq \phi \leq 20 \) [degrees] and \( 60 \leq \phi \leq 90 \) [degrees].

The response matrix is given by

\[
\mathbf{A} = [a(\phi_1), \ldots, a(\phi_I)]
\]

(1.77)

where the \( M \times 1 \) column vectors \( a(\phi_i) \) have elements \( a_m(\phi_i) = e^{-j(\pi \sin \phi_i)m} \) for \( m = 0, 1, \ldots, M-1 \) and \( \phi_i \) are angles between -90 and 90 degrees.

**Programming Task:** Design a suitable lowpass FIR filter \( h(m) \) using \( \mathbf{h} = \text{fir1}(\cdot) \) in Matlab. Let \( \phi_0 = 40\pi/180 \) (40 degrees), and let \( w_m = h(m)e^{-j(\pi \sin \phi_0)m} \). This is conveniently done in Matlab using

\[
\mathbf{w} = \mathbf{h} \ast \exp(-j \ast \pi \ast \sin(\phi_0) \ast ((0 : M - 1)'))
\]

so that \( \mathbf{w} \) becomes an \( M \times 1 \) vector. Remember that Matlab calculates with radians. The \( M \times I \) response matrix \( \mathbf{A} \) can be formed by using

\[
\text{phi} = (-90 : 0.1 : 90) \ast \pi/180
\]

(1 \times I row vector with angles \( \phi_i \) between -90 and 90 degrees) and

\[
\mathbf{A} = \exp(-j \ast \pi \ast (0 : M - 1)' \ast \sin(\text{phi})).
\]

Finally, plot the solution using

\[
\text{plot}(\text{phi} \ast 180/\pi, 20 \ast \log_{10}(\text{abs}(\mathbf{w}' \ast \mathbf{A}))).
\]

1.5.2 Linear phase FIR filter design by the window method

As an aid in the programming task above, a linear phase FIR filter design by the window method is briefly outline below, see also [1–3]. Assume that the digital filter response is defined by

\[
H_{LP}(\Omega) = \sum_{m=0}^{M-1} h(m)e^{-j\Omega m}.
\]
A “desired” impulse response \( h^{(d)}(m) \) corresponding to a linear phase low-pass IIR digital filter with normalized frequency bandwidth \( \nu_B \) is obtained from the inverse Fourier transform as

\[
h^{(d)}(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H^{(d)}_\text{LP}(\Omega) e^{j\Omega m} d\Omega = \frac{1}{2\pi} \int_{-2\pi\nu_B}^{2\pi\nu_B} e^{-j\Omega \left(\frac{M-1}{2}\right)} e^{j\Omega m} d\Omega
\]

\[
= \frac{1}{2\pi} \int_{-2\pi\nu_B}^{2\pi\nu_B} e^{j\Omega \left(m-\frac{(M-1)}{2}\right)} d\Omega = 2\nu_B \frac{\sin(2\pi\nu_B\left(m-\frac{(M-1)}{2}\right))}{2\nu_B\left(m-\frac{(M-1)}{2}\right)}.
\]

Suppose e.g., that a Hanning window will be applied, with window function

\[
w(m) = 0.5 \cdot (1 - \cos(\frac{2\pi m}{M-1}))
\]

where \( 0 \leq m \leq M - 1 \).

The impulse response \( h(m) \) of the FIR digital filter is then finally obtained by multiplying the desired impulse response \( h^{(d)}(m) \) with the window function \( w(m) \) to yield

\[
h(m) = h^{(d)}(m) \cdot w(m).
\]

In Matlab-code, this can be generated as for example:

\[
\begin{align*}
M &= 50; \\
\nu_B &= 0.08; \\
m &= (0:M-1)'; \\
hd &= 2*\nu_B*(\sin(2*pi*\nu_B*(m-(M-1)/2+eps)))./(2*pi*\nu_B*(m-(M-1)/2+eps)); \\
w &= 0.5*(1-cos(2*pi*m/(M-1))); \\
h &= hd.*w;
\end{align*}
\]

### 1.5.3 Optimum beamformer

Consider a linear array with \( d = \lambda/2 \) and array response vector given by

\[
\mathbf{a}(\phi) = \begin{pmatrix}
1 \\
e^{-j\pi \sin(\phi)} \\
\vdots \\
e^{-j\pi \sin(\phi)(M-1)}
\end{pmatrix}
\]

with elements \( a_m(\phi) = e^{-j\pi \sin(\phi)m} \) for \( m = 0, 1, \ldots, M - 1 \).

Consider the signal scenario as in (1.38). The optimum beamformer weights \( \mathbf{w}_o \) are given by (1.47) where \( \mathbf{R} \) and \( \mathbf{p} \) are given by (1.40) and (1.46).

Let the desired direction be \( \phi_0 = 40\pi/180 \) (40 degrees) with signal variance \( \sigma_s^2 = 1 \). Let the variance of the \( L \) interferers be \( \sigma_i^2 = 1 \), and let the variance of the noise be \( \sigma_n^2 = 10^{-3} \) (SNR=30 dB).

**Programming Task:**

- Let there be \( L = 3 \) interferers with \( \phi_1 = 10\pi/180 \) (10 degrees), \( \phi_2 = 60\pi/180 \) (60 degrees) and \( \phi_3 = 70\pi/180 \) (70 degrees). Plot the optimal response \( H(\phi) = \mathbf{w}_o^H \mathbf{a}(\phi) \) (in dB) for \(-90 \leq \phi \leq 90 \) (degrees), in four different plots with \( M = 3, 4, 5 \) and 10.

- Let there be \( L = 1 \) interferer. Let \( M = 5 \) and plot the optimal response \( H(\phi) = \mathbf{w}_o^H \mathbf{a}(\phi) \) (in dB) for \(-90 \leq \phi \leq 90 \) (degrees), in three different plots with \( \phi_1 = 10, 60 \) and 70 degrees.
Matlab Tips: The last task with $\phi_1 = 70$ degrees above can be solved using the following Matlab code:

```matlab
M=5;
phi0=40*pi/180;
phi1=70*pi/180;
sigman2=10^(-30/10);
a0=exp(-j*pi*(0:M-1)'*sin(phi0));
a1=exp(-j*pi*(0:M-1)'*sin(phi1));
R=a0*a0'+a1*a1'+sigman2*eye(M);
p=a0;
w0=inv(R)*p;
phi=(-90:0.1:90)*pi/180;
A=exp(-j*pi*(0:M-1)'*sin(phi));
figure(3)
plot(phi*180/pi,20*log10(abs(w0'*A)))
axis([-90 90 -120 20])
grid on
zoom on
```

OBS! Matrix hermite transpose $A^H$ is programmed as $A'$ in Matlab. Matrix transpose $A^T$ is programmed as $A.^*$ in Matlab.
Chapter 2

Adaptive beamforming

2.1 The complex LMS algorithm

Consider the adaptive beamforming problem described in section 1.2. The $M \times 1$ array input vector is denoted by $x(t)$ and the output signal is $y(t) = w^H x(t)$ where $w$ is the $M \times 1$ complex weight vector. Further, $s(t)$ denotes the signal of interest.

The steepest descent algorithm

We begin by deriving the steepest descent, or gradient algorithm. The cost function is given by

$$ f(w) = E \{|s(t) - w^H x(t)|^2\} = w^H R w - p^H w - w^H p + \sigma_s^2 $$

(2.1)

where $R = E\{x(t)x^H(t)\}$, $p = E\{x(t)s^*(t)\}$ and $\sigma_s^2 = E\{|s(t)|^2\}$.

The complex gradient of (2.1) is given by

$$ \nabla f(w) = \frac{\partial f}{\partial w_x} + j \frac{\partial f}{\partial w_y} = 2 \frac{\partial}{\partial w^*} f(w) = 2(Rw - p) $$

(2.2)

and by equating (2.2) to zero, we obtain the optimum beamformer

$$ w_o = R^{-1}p. $$

(2.3)

A steepest descent, or gradient algorithm to minimize (2.1) is given by the following recursion

$$ w_n = w_{n-1} - \mu \nabla f(w_{n-1}) = w_{n-1} - 2\mu (Rw_{n-1} - p) $$

(2.4)

where $\mu$ is a real and positive step size parameter. By introducing

$$ \hat{w}_n = w_n - w_o $$

(2.5)

we can write (2.4) as

$$ \hat{w}_n = \hat{w}_{n-1} - 2\mu R\hat{w}_{n-1}. $$

(2.6)

The spectral theorem for Hermitian matrices [7] can be stated as

$$ R = Q \Lambda Q^H $$

(2.7)

where $\Lambda$ is the diagonal matrix containing the eigenvalues of $R$, and $Q$ is the unitary matrix containing the corresponding eigenvectors. The matrix $R$ is Hermitian and positively semidefinite so that all eigenvalues are real and non-negative. We can now introduce the vector

$$ v_n = Q^H \hat{w}_n = Q^H (w_n - w_o) $$

(2.8)
which is the coordinate representation of the vector $\tilde{w}_n = w_n - w_o$ in the coordinate system defined by the eigenvectors of $R$ (the columns of $Q$). The recursion in (2.6) become

$$v_n = v_{n-1} - 2\mu \Delta v_{n-1}. \tag{2.9}$$

Due to the diagonal form of (2.9), this recursion can also be written in the following component form

$$v_n(m) = v_{n-1}(m) - 2\mu \lambda_m v_{n-1}(m) = (1 - 2\mu \lambda_m)v_{n-1}(m) \tag{2.10}$$

where $v_n(m)$ denotes the components of the vector $v_n$ for $m = 1, \ldots, M$. The solution to the recursion (2.10) can be written as

$$v_n(m) = (1 - 2\mu \lambda_m)^n v_0(m). \tag{2.11}$$

Hence, convergence of the algorithm in (2.4) is guaranteed if

$$|1 - 2\mu \lambda_m| < 1, \text{ for all } m \tag{2.12}$$

or equivalently, if

$$\mu < \frac{1}{\lambda_m} \text{ for all } m. \tag{2.13}$$

The convergence criterion of (2.13) can also be written as

$$\mu < \frac{1}{\lambda_{\text{max}}} \tag{2.14}$$

where $\lambda_{\text{max}}$ is the largest eigenvalue.

A good strategy to satisfy the convergence criterion of (2.14) is to employ the following inequality

$$\mu < \frac{1}{\text{trace}(R)} \leq \frac{1}{\lambda_{\text{max}}}. \tag{2.15}$$

A good rule of thumb satisfying the convergence criteria (2.14) and (2.15) is to choose

$$\mu = \frac{0.1}{\text{trace}(R)}. \tag{2.16}$$

The complex LMS algorithm

Next we derive the stochastic gradient algorithm, also commonly known as the Least Mean Squares or LMS algorithm [6] in a complex formulation. We will use discrete time so that $x(n)$ denotes a sample (snapshot) vector of the array input vector $x(t)$ for $t = nT$, etc.

The stochastic gradient algorithm is a practical and very useful approximation of the steepest descent algorithm described above. Instead of using the true gradient in (2.2) we employ the approximate, stochastic gradient estimate

$$\hat{\nabla} f(w) = 2 \left( x(n)x^H(n)w - x(n)s^*(n) \right) \tag{2.17}$$

which is obtained directly from (2.2) by dropping the expectation $E\{\cdot\}$ so that $R = E\{x(n)x^H(n)\} \approx x(n)x^H(n)$ and $p = E\{x(n)s^*(n)\} \approx x(n)s^*(n)$. Note that (2.17) is an unbiased estimate of the true gradient, i.e.

$$E\{\hat{\nabla} f(w)\} = 2 (Rw - p) = \nabla f(w). \tag{2.18}$$

We see that (2.17) can be further simplified

$$\hat{\nabla} f(w) = 2x(n)(x^H(n)w - s^*(n)) = -2x(n)e^*(n) \tag{2.19}$$
where $\varepsilon(n) = s(n) - w^H x(n)$.

The complex Least Mean Squares (LMS) algorithm can now be summarized as follows

given $w_0$,
repeat for $n = 1, 2, \ldots$

$$
\varepsilon(n) = s(n) - w^H_{n-1} x(n)
$$

$$
w_n = w_{n-1} - \mu \nabla f(w_{n-1}) = w_{n-1} + 2\mu x(n)\varepsilon^*(n)
$$

(2.20)

where $\mu$ is a positive step size parameter.

Despite its very simple form, the convergence behaviour of the LMS algorithm in (2.20) is in general very complicated to analyze. Since the gradient estimate is data dependent and stochastic in its nature, the LMS algorithm is a stochastic algorithm. Strictly speaking, the sequence $w_n$ will not converge. However, the mean path of $w_n$ will approximate the corresponding path of the true gradient algorithm which converges to $w_o$. Therefore, the qualitative results of the steepest descent algorithm are still (approximately) valid for the LMS algorithm, and the convergence criteria (2.14) through (2.15) can still be used to “guarantee” convergence. Of course, convergence is not guaranteed in a strict mathematical sense, but (2.16) is a good rule of thumb to obtain a practical useful convergence criterion for $\mu$.

Here, a “converged” solution means that the mean of $w_n$ has converged. But even if the mean has converged there will always be a remaining excess mean square error [6] associated with the “converged” solution. It should be noted that a larger $\mu$ give a faster convergence, but a poor quality (large excess mean square error) of the “converged” solution, whereas a smaller $\mu$ give a slower convergence but a better quality (low excess mean square error) of the “converged” solution. Again, as a good rule of thumb, (2.16) gives a good starting guess for the trade off between convergence speed and quality of the solution.

2.2 The complex RLS algorithm

A complex formulation for the Recursive Least Squares (RLS) algorithm is derived, see also [6]. We consider the following problem formulation. The discrete time $M \times 1$ array input vector is denoted by $x(n)$ and the output signal is $y(n) = w^H x(n)$ where $w$ is the $M \times 1$ complex weight vector. Further, $s(n)$ denotes the signal of interest.

At time instant $n$, the weight vector $w_n$ minimizes the least squares cost function $f_n(w)$ given by

$$
f_n(w) = \sum_{k=1}^{n} \lambda^{n-k} |w^H x(k) - s(k)|^2 + (w - w_0)^H \lambda^n R_0 \lambda^n (w - w_0), \ n = 1, 2, \ldots
$$

(2.21)

where $\lambda$ is a real and positive constant $0 < \lambda < 1$ and $w_0$ and $R_0$ are initial parameters that can be chosen freely. We define also $f_0(w) = (w - w_0)^H R_0 (w - w_0)$ and $p_0 = R_0 w_0$.

By introducing

$$
A = \begin{bmatrix}
\lambda^{n-1} x^H(1) \\
\vdots \\
x^H(n)
\end{bmatrix}, \quad b = \begin{bmatrix}
\lambda^{n-1} s^*(1) \\
\vdots \\
s^*(n)
\end{bmatrix}
$$

(2.22)

we may write the cost function in (2.21) as

$$
f_n(w) = (Aw - b)^H (Aw - b) + (w - w_0)^H \lambda^n R_0 (w - w_0).
$$

(2.23)

Hence, when $n$ is large ($n > M$ and $\lambda^n \to 0$), the first term in (2.23) corresponds to the least squares error of the over–determined linear system of equations $Aw = b$, and $w_n$ denotes
the corresponding solution. The purpose of the second term is to make the solution well defined (\(R_0\) is invertible) also when the system of linear equations are under-determined \((n < M)\). We will now derive a recursive solution for solving (2.21) which is numerically efficient and does not require any matrix inversion operations. To this end, the purpose of the second term is also to give the initial solution \(w_0\) and \(R_0\) a well defined meaning. Note that the purpose of the “forgetting factor” \(\lambda^n\) is to emphasize the initial solution \(w_0\) for small \(n\) and to emphasize the first term in (2.23) for larger \(n\). The purpose of the “forgetting factor” \(\lambda^{n-k}\) is also to “forget” the older data in (2.22). Finally, observe that the theory and algorithm will work also when neglecting \(\lambda\) altogether, i.e. by setting \(\lambda = 1\).

By differentiating (2.23) we find that the solution \(w_n\) is given by

\[
w_n = R_n^{-1}p_n
\]  

(2.24)

where

\[
R_n = A^H A + \lambda^n R_0 = \sum_{k=1}^{n} \lambda^{n-k} x(k)x^H(k) + \lambda^n R_0
\]  

(2.25)

\[
p_n = A^H b + \lambda^n p_0 = \sum_{k=1}^{n} \lambda^{n-k} x(k) s^*(k) + \lambda^n p_0.
\]  

(2.26)

Now, we can find recursive relations for \(R_n\) and \(p_n\) as follows. We have

\[
R_n = \sum_{k=1}^{n-1} \lambda^{n-k} x(k)x^H(k) + x(n)x^H(n) + \lambda^n R_0 = \lambda \left( \sum_{k=1}^{n-1} \lambda^{n-1-k} x(k)x^H(k) + \lambda^{n-1} R_0 \right) + x(n)x^H(n) = \lambda R_{n-1} + x(n)x^H(n)
\]  

(2.27)

and

\[
p_n = \sum_{k=1}^{n-1} \lambda^{n-k} x(k) s^*(k) + x(n)s^*(n) + \lambda^n p_0 = \lambda \left( \sum_{k=1}^{n-1} \lambda^{n-1-k} x(k) s^*(k) + \lambda^{n-1} p_0 \right) + x(n)s^*(n) = \lambda p_{n-1} + x(n)s^*(n)
\]  

(2.28)

which are valid for \(n \geq 1\).

Now, we rewrite the recursion for \(R_n\) as

\[
\lambda^{-1} R_n = R_{n-1} + x(n) \lambda^{-1} x^H(n)
\]  

(2.29)

and use the matrix inversion lemma (Lemma 1.3.1) with \(B^{-1} = \lambda^{-1} R_0, A^{-1} = R_{n-1}, C^H = x(n)\) and \(D^{-1} = \lambda^{-1}\). The inverse of \(\lambda^{-1} R_n\) in (2.29) is hence given by

\[
\lambda R_n^{-1} = R_{n-1}^{-1} - R_{n-1}^{-1} x(n) \frac{1}{x^H(n) R_{n-1}^{-1} x(n) + \lambda} x^H(n) R_{n-1}^{-1}
\]  

(2.30)

or

\[
R_n^{-1} = \lambda^{-1} R_{n-1}^{-1} - \frac{\lambda^{-1} R_{n-1}^{-1} x(n) x^H(n) R_{n-1}^{-1}}{x^H(n) R_{n-1}^{-1} x(n) + \lambda}.
\]  

(2.31)

The following relation will be useful, and is readily established by writing with common denominator

\[
R_n^{-1} x(n) = \lambda^{-1} R_{n-1}^{-1} x(n) \frac{\lambda^{-1} R_{n-1}^{-1} x(n) x^H(n) R_{n-1}^{-1} x(n)}{x^H(n) R_{n-1}^{-1} x(n) + \lambda} = \frac{R_{n-1}^{-1} x(n)}{x^H(n) R_{n-1}^{-1} x(n) + \lambda}.
\]  

(2.32)
We are now ready to derive the following recursive relation for the least squares solution

\[ w_n = R_n^{-1}p_n = R_n^{-1}(\lambda p_{n-1} + x(n)s^*(n)) = R_n^{-1}\lambda p_{n-1} + R_n^{-1}x(n)s^*(n) = \\
= \left( \lambda^{-1} R_n^{-1} - \frac{\lambda^{-1} R_n^{-1} x(n) x^H(n) R_n^{-1}}{x^H(n)x(n) + \lambda} \right) \lambda p_{n-1} + R_n^{-1}x(n)s^*(n) = \\
= R_n^{-1}p_{n-1} - \frac{R_n^{-1}x(n)x^H(n)R_n^{-1}p_{n-1}}{x^H(n)x(n) + \lambda} + R_n^{-1}x(n)s^*(n) = \\
= w_{n-1} - R_n^{-1}x(n)x^H(n)w_{n-1} + R_n^{-1}x(n)s^*(n) = \\
= w_{n-1} + R_n^{-1}x(n) (s^*(n) - x^H(n)w_{n-1}) = \\
= w_{n-1} + R_n^{-1}x(n)\varepsilon^*(n). \tag{2.33} \]

where \( \varepsilon(n) = s(n) - w_{n-1}^H x(n) \).

The complex Recursive Least Squares (RLS) algorithm can now be summarized as follows

given \( w_0, R_0, R_0^{-1} \)
repeat for \( n = 1, 2, \ldots \)
\( \varepsilon(n) = s(n) - w_{n-1}^H x(n) \)
\( R_n^{-1} = \lambda^{-1} R_{n-1}^{-1} - \frac{\lambda^{-1} R_{n-1}^{-1} x(n)x^H(n) R_{n-1}^{-1}}{x^H(n)x(n) + \lambda} \)
\( w_n = w_{n-1} + R_n^{-1}x(n)\varepsilon^*(n). \tag{2.34} \)

2.3 Matlab programming tasks

2.3.1 The LMS and RLS algorithm

Consider a linear array with \( d = \lambda/2 \) and array response vector given by

\[ a(\phi) = \begin{pmatrix}
1 \\
e^{-j\pi\sin(\phi)} \\
\vdots \\
e^{-j\pi\sin(\phi)(M-1)}
\end{pmatrix}. \tag{2.35} \]

The \( M \times 1 \) array input signal is given by

\[ x(k) = s(k)a(\phi) + s_1(k)a(\phi_1) + n(k) \tag{2.36} \]

where \( s(k) \) is the desired signal with variance \( \sigma_s^2 = 1 \), coming from direction \( \phi \approx 10\pi/180 \). Further, there is \( L = 1 \) undesired interfering signal \( s_1(k) \) with variance \( \sigma_1^2 = 1 \), coming from direction \( \phi_1 \approx 60\pi/180 \), and \( n(t) \) is an \( M \times 1 \) vector of uncorrelated sensor noise with equal variances \( \sigma_n^2 = 10^{-3} \) (30 dB SNR).

Suppose that the array input signal \( x(k) \) and the desired signal \( s(k) \) are available. The sequence of adaptive beamformer weights \( w_n \) to cancel the interferer can be obtained either by using the LMS algorithm (2.20) or the RLS algorithm (2.34).

**Programming Task:** The array input signal \( x(k) \) and desired signal \( s(k) \) for an \( M = 4 \) sensor array are available at the course home-page with file name: LMSRLSdata.mat. Here, the signal \( x(k) \) is represented by an \( 4 \times 200 \) matrix \( X = [x(1) \cdots x(200)] \) and an \( 200 \times 1 \) vector \( s = [s(1) \cdots s(200)]^T \) containing the data.

Write a Matlab program that implements both the LMS and the RLS algorithms as described in (2.20) and (2.34). Study the array response function \( w_n^H a(\phi) \) during adaptation (for \( -90 \leq \phi \leq 90 \) degrees). Compare and plot the error signal \( 20\log_{10}|\varepsilon(n)| \) (in dB) for both algorithms. Use \( \mu = 0.01 \) and \( w_0 = 0 \) for the LMS algorithm. Use \( \lambda = 0.99 \), \( w_0 = 0 \) and \( R_0 = I \) for the RLS algorithm.
Matlab Tips: A polar diagram for the optimum array response (Optimum solution according to section 1.2) is given by:

M=4; % Number of antennas
phi=(-90:0.1:90)*pi/180; % angular domain
A=exp(-j*pi*(0:M-1)'*sin(phi)); % response matrix for angular domain
phi0=12*pi/180; % desired direction
phi1=61*pi/180; % interferer direction
sigman2=10^(-30/10); % noise variance
a0=exp(-j*pi*(0:M-1)'*sin(phi0)); % response vector for desired signal
a1=exp(-j*pi*(0:M-1)'*sin(phi1)); % response vector for interferer
R=a0*a0'+a1*a1'+sigman2*eye(M); % signal covariance matrix
p=a0; % signal correlation vector
w0=inv(R)*p; % optimum solution
figure(4)
polar(phi,max(0,60+20*log10(abs(w0'*A))),'-') % polar plot of response function in dB (with 60 dB dynamics)
Chapter 3

Estimation of Direction of Arrival

3.1 Classical Estimation of Arrival

We consider the problem of estimating the direction of arrival (DOA) of a single signal $s(t)$ coming from direction $\phi_1$, see also [8]. The signal model for the $M \times 1$ array input vector $x(t)$ is given by

$$x(t) = s(t)a(\phi_1) + v(t)$$ (3.1)

where $a(\phi)$ is the steering vector (or array response vector) and $v(t)$ is uncorrelated measurement noise. The covariance matrix for the input signal $x(t)$ is given by

$$R = E\{x(t)x^H(t)\} = \sigma_s^2 a(\phi_1)a^H(\phi_1) + \sigma_v^2 I$$ (3.2)

where $\sigma_s^2$ is the variance of the signal and $\sigma_v^2$ is the variance of the noise.

Assume that an array output signal is obtained by applying an $M \times 1$ weight vector $w$ to the input signal $x(t)$. Hence,

$$y(t) = w^H x(t) = s(t)w^H a(\phi_1) + w^H v(t)$$ (3.3)

where the first term on the right hand side is the signal term and the second term is noise. The signal–to–noise–ratio SNR for the output signal $y(t)$ is given by

$$SNR = \frac{E\{|s(t)w^H a(\phi_1)|^2\}}{E\{|w^H v(t)|^2\}} = \frac{\sigma_s^2 w^H a(\phi_1)a^H(\phi_1)w}{\sigma_v^2 w^H w}$$ (3.4)

From Cauchy–Schwarz inequality we have

$$|w^H a(\phi_1)|^2 \leq w^H w \cdot a^H(\phi_1)a(\phi)$$ (3.5)

with equality iff $w = ca(\phi_1)$ where $c$ is an arbitrary constant. Hence, we see that the SNR in (3.4) is maximized by the vector $w = ca(\phi_1)$ where $c$ is an arbitrary constant. This is actually the reason why $a(\phi)$ is called a steering vector.

Consider now the average output power given by

$$E\{|y(t)|^2\} = w^H R w = \sigma_s^2 w^H a(\phi_1)a^H(\phi_1)w + \sigma_v^2 w^H w$$ (3.6)

where $R = E\{x(t)x^H(t)\}$. By choosing $w = a(\phi)$ (and thus $c = 1$ above), we define now the classical spectrum by

$$P(\phi) = a^H(\phi)Ra(\phi) = \sigma_s^2 a^H(\phi)a(\phi_1)a^H(\phi_1)a(\phi) + \sigma_v^2 a^H(\phi)a(\phi).$$ (3.7)
By (1.8) it is noted that $\mathbf{a}^H(\phi)\mathbf{a}(\phi) = M$ and the classical spectrum becomes
\[
P(\phi) = \sigma_s^2|\mathbf{a}^H(\phi)\mathbf{a}(\phi_1)|^2 + \sigma_v^2 M.
\] (3.8)

Again, we use the Cauchy–Schwarz inequality,
\[
|\mathbf{a}^H(\phi)\mathbf{a}(\phi_1)|^2 \leq \mathbf{a}^H(\phi)\mathbf{a}(\phi) \cdot \mathbf{a}^H(\phi_1)\mathbf{a}(\phi_1) = M^2
\] (3.9)
with equality iff $\mathbf{a}(\phi) = \mathbf{a}(\phi_1)$, or equivalently if $\phi = \phi_1$. Hence, we find that $P(\phi)$ is maximized by the incident direction $\phi_1$ and we can therefore write
\[
\phi_1 = \arg \max_{\phi} P(\phi) = \arg \max_{\phi} \mathbf{a}^H(\phi)\mathbf{R}\mathbf{a}(\phi).
\] (3.10)

Eventhough (3.10) is an exactly correct expression for the incident direction $\phi_1$ only when the single signal scenario in (3.1) is valid and the covariance matrix $\mathbf{R}$ is exactly known, the same idea may be used as an estimation method for more general cases. Hence, the classical spectrum estimate of the direction of arrival is given by
\[
\hat{\phi} = \arg \max_{\phi} P(\phi)
\] (3.11)
where
\[
P(\phi) = \mathbf{a}^H(\phi)\hat{\mathbf{R}}\mathbf{a}(\phi)
\] (3.12)
and $\hat{\mathbf{R}}$ is an estimate of the covariance matrix given by
\[
\hat{\mathbf{R}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}(t_i)\mathbf{x}^H(t_i)
\] (3.13)
where $t_i$ are the sample points and $N$ is the number of samples. Observe that the method requires that the steering vector $\mathbf{a}(\phi)$ is known for all values of $\phi$.

### 3.1.1 The Periodogram method for ULA

We investigate the classical spectrum estimate given in (3.11) for the case with a uniform linear array (ULA). The array response vector for an ULA with $d = \lambda/2$ is given by
\[
\mathbf{a}(\phi) = \begin{bmatrix} 1 \\ e^{-j\pi \sin(\phi)} \\ \vdots \\ e^{-j\pi \sin(\phi)(M-1)} \end{bmatrix},
\] (3.14)
see also (1.23) in section 1.1.3. Hence, the elements of the steering vector are given by
\[
a_n(\phi) = e^{-j\Omega n}, \quad n = 0, 1, \ldots, M - 1
\] (3.15)
where
\[
\Omega = \pi \sin(\phi)
\] (3.16)
can be interpreted as the spatial frequency. Now, the signal model for a single incident signal $s(t)$ coming from direction $\phi_1$ is given by (3.1), or in component form
\[
x_n(t) = s(t)e^{-j\Omega_1 n} + v_n(t), \quad n = 0, 1, \ldots, M - 1.
\] (3.17)
Hence, for fixed $t$ the signal in (3.17) is a sinusoidal signal over the spatial dimension $n$, and the spatial frequency is $\Omega_1 = \pi \sin(\phi_1)$. 

22
The spectrum function \( P(\phi) = a^H(\phi)R a(\phi) \) in (3.11) can now be expressed as
\[
P(\phi) = a^H(\phi) \frac{1}{N} \sum_{i=1}^{N} x(t_i) x^H(t_i) a(\phi) = \frac{1}{N} \sum_{i=1}^{N} |a^H(\phi) x(t_i)|^2 = \frac{1}{N} \sum_{i=1}^{N} \sum_{n=0}^{M-1} |x_n(t_i) e^{j\Omega n}|^2 \quad (3.18)
\]
which is just the same as a Fourier based periodogram of the signal \( x_n(t_i) \), at spatial frequency \( \Omega = \pi \sin(\phi) \), and averaged over the time instances \( t_i \) for \( i = 1, \ldots, N \).

3.2 The Capon Minimum Variance Estimator

We consider a general signal scenario with \( L \) signals \( s_i(t) \) impinging on the array coming from \( L \) different directions \( \phi_i \). The signal model for the \( M \times 1 \) array input vector \( x(t) \) is given by
\[
x(t) = \sum_{i=1}^{L} s_i(t) a(\phi_i) + n(t) \quad (3.19)
\]
where \( n(t) \) is an \( M \times 1 \) vector of uncorrelated sensor noise with equal variances \( \sigma_n^2 \).

All signal and noise sources are assumed to be mutually uncorrelated and have zero mean. The covariance matrix for the input signal \( x(t) \) is thus given by
\[
R = E\{x(t)x^H(t)\} = \sum_{i=1}^{L} \sigma_i^2 a(\phi_i) a^H(\phi_i) + \sigma_n^2 I \quad (3.20)
\]
where \( \sigma_i^2 \) are the variances of the signals \( s_i(t) \).

The array output signal is given by
\[
y(t) = w^H x(t) \quad (3.21)
\]
where \( w \) is the \( M \times 1 \) array weight vector. The average output power is given by
\[
E\{|y(t)|^2\} = w^H R w = \sum_{i=1}^{L} \sigma_i^2 |w^H a(\phi_i)|^2 + \sigma_n^2 w^H w. \quad (3.22)
\]

The Capon minimum variance estimator is now derived as follows, see also [9, 10]. Given a test direction \( \phi \), an optimum array weight vector \( w_o \) is chosen such that the average output power \( E\{|y(t)|^2\} \) is minimized subjected to the linear constraint \( w^H a(\phi) = 1 \). This is equivalent to saying that the average output power is minimized subjected to the constraint that the array response \( H(\phi) = w^H a(\phi) \) is unity in the test direction \( \phi \). As an angular “spectrum” estimate we chose the resulting minimum output power \( P(\phi) = E\{|y(t)|^2\} \). The effect will be as follows: If the test direction \( \phi \) is not close to any of the present signal directions \( \phi_i \), the beamformer will be able to efficiently cancel all the present signals \( s_i(t) \) without violating the linear constraint. The average output power will then be low. If on the other hand, the test direction \( \phi \) is equal to, or very close to any of the present signal directions \( \phi_i \), the beamformer will not be able to efficiently cancel the corresponding signal \( s_i(t) \) without violating the linear constraint. The average output power will then be high.

Now, we explain the method more explicitly. The optimization criterion is given by
\[
\begin{cases}
\min_w E\{|y(t)|^2\}, \text{ subject to} \\
H(\phi) = 1
\end{cases} \quad (3.23)
\]
or, equivalently

\[
\begin{align*}
\min_w w^H Rw, \quad & \text{subject to} \\
 w^H a(\phi) = 1.
\end{align*}
\] (3.24)

The solution to this optimization problem was derived in section (1.3) and is given by

\[
w_0 = \frac{1}{a^H(\phi) R^{-1} a(\phi)} R^{-1} a(\phi).
\] (3.25)

The corresponding minimum value of the cost function \( P(\phi) = E\{|y(t)|^2\} = w_0^H R w_0 \) is readily obtained as

\[
P(\phi) = w_0^H R w_0 = \frac{1}{a^H(\phi) R^{-1} a(\phi)}.
\] (3.26)

The function \( P(\phi) \) in (3.26) is an angular “spectrum” in the \( \phi \)-variable and the corresponding peaks are the estimated directions. This estimator has significantly better resolution properties than the classical periodogram estimator \( P(\phi) = a^H(\phi) R a(\phi) \). Note however the close resemblance with both estimation procedures. Both methods require that the covariance matrix \( R \) is known, or alternatively, that the covariance matrix estimate \( \hat{R} \) in (3.13) is known. Furthermore, both methods require that the steering vector \( a(\phi) \) is known for all values of \( \phi \).

The final Capon minimum variance estimator which is based on the estimated covariance matrix \( \hat{R} \) is given by

\[
P(\phi) = \frac{1}{a^H(\phi) \hat{R}^{-1} a(\phi)}
\] (3.27)
and the corresponding peaks are the angular estimates.

### 3.3 The MUSIC method for DOA estimation

We will now derive a “modern” method for DOA estimation commonly referred to as MUltiple SIgnal Classification (MUSIC), see e.g., [8, 11]. The MUSIC method is an eigenvector based method for high–resolution frequency estimation. Here, the method is described for the estimation of direction of arrival (DOA), but the technique can also be used for “usual” frequency estimation with sinusoids in additive white noise. The strength of the method, as compared to the classical methods, is the ability to resolve closely spaced directions (or frequencies) based on very few data.

We consider the following general signal scenario with \( L \) signals \( s_i(t) \) impinging on the array coming from \( L \) different directions \( \phi_i \). The signal model for the \( M \times 1 \) array input vector \( x(t) \) is given by

\[
x(t) = \sum_{i=1}^{L} s_i(t) a(\phi_i) + n(t) = A s(t) + n(t)
\] (3.28)

where \( n(t) \) is an \( M \times 1 \) vector of uncorrelated sensor noise with equal variances \( \sigma_n^2 \) and covariance matrix \( \sigma_n^2 I \). Here the response matrix \( A \) and signal vector \( s(t) \) are defined by

\[
A = \begin{pmatrix} a(\phi_1) & \ldots & a(\phi_L) \end{pmatrix}
\] (3.29)

and

\[
s(t) = \begin{pmatrix} s_1(t) \\ \vdots \\ s_L(t) \end{pmatrix}
\] (3.30)

It is assumed that \( L < M \) and that the columns of the matrix \( A \) are linearly independent.

We now make the following formal definitions.
• The signal subspace is the range space of the matrix $\mathbf{A}$, i.e. $\mathcal{R}\{\mathbf{A}\}$.
  The dimension of this space is $\text{dim}\{\mathcal{R}\{\mathbf{A}\}\} = L$.

• The noise subspace is the left null space of the matrix $\mathbf{A}$, i.e. $\mathcal{N}\{\mathbf{A}^H\}$.
  The dimension of this space is $K = \text{dim}\{\mathcal{N}\{\mathbf{A}^H\}\} = M - L$.

The two spaces are orthogonal complements, $\mathcal{N}\{\mathbf{A}^H\} = \mathcal{R}\{\mathbf{A}\}^\perp$, and together they span the complex vector space $C^M$.

The covariance matrix for the array input signal $\mathbf{x}(t)$, assuming that $\mathbf{s}(t)$ and $\mathbf{n}(t)$ are zero mean and uncorrelated, is given by

$$\mathbf{R} = E\{\mathbf{x}(t)\mathbf{x}^H(t)\} = \mathbf{A}\mathbf{S}\mathbf{A}^H + \sigma_n^2 \mathbf{I}$$

(3.31)

where

$$\mathbf{S} = E\{\mathbf{s}(t)\mathbf{s}^H(t)\}$$

(3.32)

is the covariance matrix for the impinging signals $\mathbf{s}_i(t)$. If all the incident signals $\mathbf{s}_i(t)$ are mutually uncorrelated the covariance matrix $\mathbf{S}$ becomes diagonal

$$\mathbf{S} = \begin{pmatrix} 
\sigma_1^2 \\
\vdots \\
\sigma_L^2 
\end{pmatrix}$$

(3.33)

where $\sigma_1^2, \ldots, \sigma_L^2$ are the variances of the signals $\mathbf{s}_1(t), \ldots, \mathbf{s}_L(t)$, respectively. In general, we may assume that the matrix $\mathbf{S}$ is positively definite, i.e. $\mathbf{S} > 0$.

The MUSIC method is based on the eigenvector decomposition (spectral decomposition) of the covariance matrix $\mathbf{R}$.

$$\mathbf{R} = \mathbf{Q}\mathbf{A}\mathbf{Q}^H$$

(3.34)

where $\mathbf{A}$ is a diagonal matrix with eigenvalues $\lambda_i$ and $\mathbf{Q}$ is an orthogonal matrix of eigenvectors $\mathbf{q}_i$. We have the eigenvector relationship

$$\mathbf{R}\mathbf{q}_i = \lambda_i\mathbf{q}_i, \ i = 1, \ldots, M$$

(3.35)

and we chose to organize the eigenvalues in increasing order, that is

$$\lambda_1 \leq \ldots \leq \lambda_M.$$ 

(3.36)

Now, we make the following observations about the eigenvalues and eigenvectors of $\mathbf{R}$. First, any vector $\mathbf{w}$ that belongs to the left null space $\mathcal{N}\{\mathbf{A}^H\}$, or noise subspace, is also an eigenvector of $\mathbf{R}$ with eigenvalue $\sigma_n^2$. This is readily seen since $\mathbf{A}^H\mathbf{w} = \mathbf{0}$ implies that

$$\mathbf{R}\mathbf{w} = (\mathbf{A}\mathbf{A}^H + \sigma_n^2 \mathbf{I})\mathbf{w} = \sigma_n^2 \mathbf{w}.$$ 

(3.37)

In general, the eigenvalues $\lambda_i$ of $\mathbf{R}$ are given by the following quadratic form

$$\lambda_i = \mathbf{q}_i^H\mathbf{R}\mathbf{q}_i = \mathbf{q}_i^H(\mathbf{A}\mathbf{A}^H + \sigma_n^2 \mathbf{I})\mathbf{q}_i = \mathbf{q}_i^H\mathbf{A}\mathbf{A}^H\mathbf{q}_i + \sigma_n^2 \mathbf{q}_i^H\mathbf{q}_i = \mathbf{q}_i^H\mathbf{A}\mathbf{A}^H\mathbf{q}_i + \sigma_n^2.$$ 

(3.38)

Since $\mathbf{S} > 0$ we can therefore establish the following inequality

$$\lambda_i = \mathbf{q}_i^H\mathbf{R}\mathbf{q}_i = (\mathbf{A}\mathbf{q}_i)^H\mathbf{S}(\mathbf{A}\mathbf{q}_i) + \sigma_n^2 \geq \sigma_n^2$$

(3.39)

with equality if and only if $\mathbf{A}\mathbf{q}_i = \mathbf{0}$. In words, all eigenvalues of $\mathbf{R}$ are greater than or equal to the noise variance $\sigma_n^2$, and the equality is valid if and only if the eigenvector $\mathbf{q}_i$ is in the noise subspace $\mathcal{N}\{\mathbf{A}^H\}$.

From (3.39) it is also concluded that an eigenvalue is strictly greater than the noise variance, i.e. $\lambda_i > \sigma_n^2$, if and only if $\mathbf{A}\mathbf{q}_i \neq \mathbf{0}$ and hence $\mathbf{q}_i \notin \mathcal{N}\{\mathbf{A}^H\}$.

We summarize the properties above in a slightly more formal way:
• $\lambda_i = \sigma_n^2 \Leftrightarrow A^H q_i = 0 \Leftrightarrow q_i \in N\{A^H\}$, i.e. the eigenvector $q_i$ belongs to the noise subspace. The dimension of this space is $K = M - L$ and $\lambda_1 = \cdots = \lambda_K = \sigma_n^2$.

• $\lambda_i > \sigma_n^2 \Leftrightarrow A^H q_i \neq 0 \Leftrightarrow q_i \notin N\{A^H\} \Leftrightarrow q_i \in R\{A\}$, i.e. the eigenvector $q_i$ belongs to the signal subspace. The dimension of this space is $L$ and $\sigma_n^2 < \lambda_{K+1} \leq \cdots \leq \lambda_M$ ($L$ eigenvalues).

We proceed by making the following conclusions and further definitions

• The noise subspace $N\{A^H\}$ is spanned by the eigenvectors $q_1, \ldots, q_K$ corresponding to the eigenvalues $\lambda_1 = \cdots = \lambda_K = \sigma_n^2$. We denote the corresponding $M \times K$ noise eigenvector matrix $Q_n = [q_1 \cdots q_K]$.

• The signal subspace $R\{A\}$ is spanned by the eigenvectors $q_{K+1}, \ldots, q_M$ corresponding to the eigenvalues $\sigma_n^2 < \lambda_{K+1} \leq \cdots \leq \lambda_M$. We denote the corresponding $M \times L$ signal eigenvector matrix $Q_s = [q_{K+1} \cdots q_M]$.

Note that the definitions made above means that the eigenvector matrix $Q = [q_1 \cdots q_M]$ is decomposed as $Q = [Q_n, Q_s]$ where $Q_n$ are the noise eigenvectors (eigenvalues $\lambda_i = \sigma_n^2$) and $Q_s$ are the signal eigenvectors (eigenvalues $\lambda_i > \sigma_n^2$).

The idea of the MUSIC method is to search for response vectors $a(\phi)$ which belong to the signal subspace $R\{A\}$, or equivalently, to search for response vectors $a(\phi)$ which are orthogonal to the noise subspace $N\{A^H\}$. Since $Q_n$ is a basis for $N\{A^H\}$, this search can be formalized as finding those directions $\phi$ for which

$$Q_n^H a(\phi) = 0,$$

or equivalently,

$$\|Q_n^H a(\phi)\|^2 = a^H(\phi)Q_nQ_n^H a(\phi) = 0. \quad (3.41)$$

 Obviously, if $\phi$ is equal to any one of the present signal directions $\phi_1, \ldots, \phi_L$, then $a(\phi)$ is equal to one of the columns of $A = [a(\phi_1) \cdots a(\phi_L)]$ and $a(\phi) \in R\{A\}$, implying that $Q_n^H a(\phi) = 0$.

The question then arises if there can be other directions of $\phi \neq \phi_1, \ldots, \phi_L$ so that $a(\phi) \in R\{A\}$ and thus $Q_n^H a(\phi) = 0$? That is to say, can there be other response vectors $a(\phi) \neq a(\phi_1) \cdots a(\phi_L)$ so that $a(\phi) \in R\{A\}$? The answer is in general yes, if the response vector $a(\phi)$ is linearly dependent of the columns of $A = [a(\phi_1) \cdots a(\phi_L)]$ then $a(\phi) \in R\{A\}$ and consequently $Q_n^H a(\phi) = 0$.

However, we will assume here that a strong condition for linear independence hold which is known as the Haar condition. The Haar condition implies that if the signal directions $\phi_1, \ldots, \phi_L$ are distinct, i.e. if $\phi_1 \neq \cdots \neq \phi_L$ and $L < M$, then the columns of the matrix $A = [a(\phi_1) \cdots a(\phi_L)]$ are linearly independent. The Haar condition also implies that if $\phi \neq \phi_1, \ldots, \phi_L$ and $L < M$, then the response vector $a(\phi)$ is linearly independent of the columns of the matrix $A$. Hence, $a(\phi) \notin R\{A\}$ and consequently $Q_n^H a(\phi) \neq 0$. Thus, if the Haar condition is satisfied and $L < M$ we see that $Q_n^H a(\phi) = 0$ if and only if $\phi$ is one of the signal directions $\phi_1, \ldots, \phi_L$. It can be shown that the response vectors for a uniform linear array (ULA) satisfies the Haar condition.

In practice, the covariance matrix $R$ is not known and we employ instead the covariance estimate

$$\hat{R} = \frac{1}{N} \sum_{i=1}^{N} x(t_i)x^H(t_i) \quad (3.42)$$
where $N$ is the number of snapshots and $t_i$ are the sample instances. The MUSIC spectrum is then defined by

$$P(\phi) = \frac{1}{a^H(\phi) \hat{Q}_n \hat{Q}_n^H a(\phi)} \quad (3.43)$$

where

$$\hat{R} = \hat{Q} \hat{\Lambda} \hat{Q}^H \quad (3.44)$$

is the eigendecomposition of $\hat{R}$ and the eigenvector matrix $\hat{Q} = [q_1 \cdots q_M]$ is decomposed as $\hat{Q} = [\hat{Q}_n \hat{Q}_s]$ where $\hat{Q}_n$ are the noise eigenvectors (eigenvalues $\hat{\lambda}_i \approx \sigma_n^2$) and $\hat{Q}_s$ are the signal eigenvectors (eigenvalues $\hat{\lambda}_i > \sigma_n^2$).

The estimation process can be summarized as follows. First the covariance matrix $\hat{R}$ is obtained by (3.42) and the spectral decomposition in (3.44) is calculated. Then, the number $K$ of smallest eigenvalues $\hat{\lambda}_i \approx \sigma_n^2$ determines an estimate of the dimension of the noise subspace. An estimate of the number of incident signals is then $L = M - K$. The noise eigenvectors $\hat{Q}_n$ of $\hat{R}$ correspond to the $K$ smallest eigenvalues and the MUSIC spectrum in (3.43) can be calculated.

### 3.4 Matlab Programming Tasks

Consider a linear array with $d = \lambda/2$ and array response vector given by

$$a(\phi) = \begin{pmatrix} 1 \\ e^{-j\pi \sin(\phi)} \\ \vdots \\ e^{-j\pi \sin(\phi)(M-1)} \end{pmatrix} \quad (3.45)$$

The $M \times 1$ array input signal is given by

$$x(k) = s_1(k)a(\phi_1) + s_2(k)a(\phi_2) + n(k) \quad (3.46)$$

where $s_1(k)$ and $s_2(k)$ are complex stationary random signals with variances $\sigma_1^2 = \sigma_2^2 = 1$, coming from direction $\phi_1$ and $\phi_2$, respectively. Further, $n(t)$ is an $M \times 1$ vector of uncorrelated sensor noise with equal variances $\sigma_n^2 = 10^{-3}$ (30 dB SNR).

Suppose that the array input signal $x(k)$ is available. The three: Classical, Capon and MUSIC DOA spectrum estimates in (3.12), (3.27) and (3.43) can now be computed.

**Programming Task:** The array input signal $x(k)$ for an $M = 4$ sensor array is available at the course home-page with file name: **LMSRLSdata.mat**. Here, the signal $x(k)$ is represented by an $4 \times 200$ matrix $X = [x(1) \cdots x(200)]$.

Compute the covariance estimate $\hat{R}$ using (3.42). Then, calculate and plot the three: Classical, Capon and MUSIC DOA spectrum estimates in (3.12), (3.27) and (3.43), respectively.

**Matlab tips:** The eigendecomposition can be computed using Matlabs $[Q,L]=\text{eig}(R)$ command.
Chapter 4

The Singular Value Decomposition in Array Processing

4.1 The Singular Value Decomposition

In this section we derive the Singular Value Decomposition (SVD) [7]. The SVD is a matrix factorization which is valid for any matrix and it has become a powerful tool in theoretical as well as in applied linear algebra. The SVD is a relatively modern result in matrix theory and it has attracted much attention due to its many applications in areas such as numerical analysis, control theory and signal processing, only to name a few. Here, we will of course focus on applications in array signal processing.

The factorization using the SVD is formalized in the following theorem:

Theorem 4.1.1 (The Singular Value Decomposition) Any $m \times n$ matrix $A$ with rank $r$ can be factorized as

$$A = U \Sigma V^H$$

(4.1)

where $U$ is an $m \times m$ orthogonal matrix, $V$ is an $n \times n$ orthogonal matrix, and $\Sigma$ is an $m \times n$ matrix

$$\Sigma = \begin{pmatrix}
\sigma_1 & 0 \\
0 & \ddots \\
0 & 0 & \sigma_r
\end{pmatrix}$$

(4.2)

with $r$ singular values $\sigma_i$, $i = 1, \ldots, r$ on the main diagonal. The singular values are strictly positive $\sigma_i > 0$ and are usually organized in decreasing order, that is

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$ 

(4.3)

Corollary 4.1.1 (Properties of the SVD) From theorem 4.1.1 is noted that

$$AA^H = U (\Sigma \Sigma^H) U^H$$

(4.4)

$$A^H A = V (\Sigma^H \Sigma) V^H$$

(4.5)

and we see that the column vectors of $U$ are eigenvectors of $AA^H$ with eigenvalues $\sigma_i^2$ (or zero), and the column vectors of $V$ are eigenvectors of $A^H A$ with eigenvalues $\sigma_i^2$ (or zero).
Furthermore, it may be shown that if $U = [u_1 \cdots u_m]$ then
\begin{align*}
\mathcal{R}\{A\} &= \mathcal{R}\{[u_1 \cdots u_r]\} \quad \text{(4.6)} \\
\mathcal{N}\{A^H\} &= \mathcal{R}\{[u_{r+1} \cdots u_m]\} \quad \text{(4.7)}
\end{align*}
and if $V = [v_1 \cdots v_n]$ then
\begin{align*}
\mathcal{R}\{A^H\} &= \mathcal{R}\{[v_1 \cdots v_r]\} \quad \text{(4.8)} \\
\mathcal{N}\{A\} &= \mathcal{R}\{[v_{r+1} \cdots v_n]\}. \quad \text{(4.9)}
\end{align*}

In words, the column space $\mathcal{R}\{A\}$ is spanned by the first $r$ columns of $U$, and the remaining columns constitute a basis for the left null space $\mathcal{N}\{A^H\}$. Also, the row space $\mathcal{R}\{A^H\}$ is spanned by the first $r$ columns of $V$, and the remaining columns constitute a basis for the null space $\mathcal{N}\{A\}$.

**Proof:** We prove theorem 4.1.1 and corollary 4.1.1 by constructing the SVD. From the spectral theorem for Hermitian matrices [7] we know that the Hermitian matrix $AA^H$ has $m$ orthogonal eigenvectors. We denote these eigenvectors $u_i$ and the corresponding nonzero eigenvalues by $\sigma_i^2$ which are organized in decreasing order. We have thus
\begin{align*}
AA^H u_i &= \sigma_i^2 u_i, \quad i = 1, \ldots, r \quad \text{(4.10)} \\
AA^H u_i &= 0, \quad i = r + 1, \ldots, m. \quad \text{(4.11)}
\end{align*}

In establishing (4.10) and (4.11), it is noted that $\mathcal{N}\{AA^H\} = \mathcal{N}\{A^H\}$ and consequently $\mathcal{R}\{AA^H\} = \mathcal{R}\{A\}$. Note also that the dimensions of these spaces are $m - r$ and $r$, respectively. Thus, the $m - r$ eigenvectors in (4.11) span the left null space $\mathcal{N}\{A^H\}$, and the $r$ eigenvectors in (4.10) span the column space $\mathcal{R}\{A\}$.

Now, the first $r$ eigenvectors $v_i$ are constructed by
\begin{align*}
v_i &= \frac{1}{\sigma_i} A^H u_i, \quad i = 1, \ldots, r \quad \text{(4.12)}
\end{align*}
where it is noted that $u_i \not\in \mathcal{N}\{A^H\}$ for $i = 1, \ldots, r$ and hence $A^H u_i \neq 0$. It is readily verified that the vectors $v_i$ are eigenvectors of $A^H A$ with eigenvalues $\sigma_i^2$
\begin{align*}
A^H A v_i &= A^H \frac{1}{\sigma_i} A^H u_i = A^H \frac{1}{\sigma_i} AA^H u_i = A^H \frac{1}{\sigma_i} \sigma_i^2 u_i = \sigma_i^2 v_i \quad \text{(4.13)}
\end{align*}
for $i = 1, \ldots, r$, and that they are mutually orthonormal
\begin{align*}
v_i^H v_j &= \frac{1}{\sigma_i} u_i^H A \frac{1}{\sigma_j} A^H u_j = \frac{1}{\sigma_i \sigma_j} u_i^H \sigma_j^2 u_j = \frac{\sigma_j}{\sigma_i} \delta_{ij} = \delta_{ij} \quad \text{(4.14)}
\end{align*}
for $i, j = 1, \ldots, r$.

The remaining $n - r$ orthogonal eigenvectors $v_i$ for $i = r + 1, \ldots, n$ can now be obtained by the Gram–Schmidt process of orthogonalization [7]. We have finally
\begin{align*}
A^H A v_i &= \sigma_i^2 v_i, \quad i = 1, \ldots, r \quad \text{(4.15)} \\
A^H A v_i &= 0, \quad i = r + 1, \ldots, n. \quad \text{(4.16)}
\end{align*}

In establishing (4.15) and (4.16), it is noted that $\mathcal{N}\{A^H A\} = \mathcal{N}\{A\}$ and consequently $\mathcal{R}\{A^H A\} = \mathcal{R}\{A^H\}$. Note also that the dimensions of these spaces are $n - r$ and $r$, respectively.
Thus, the \( n - r \) eigenvectors in (4.16) span the null space \( \mathcal{N}\{A\} \), and the \( r \) eigenvectors in (4.15) span the row space \( \mathcal{R}\{A^H\} \).

Now, since \( A^H u_i = \sigma_i v_i \) for \( i = 1, \ldots, r \), we have

\[
u_i^H A v_j = \sigma_i \delta_{ij}, \quad i, j = 1, \ldots, r.\] (4.17)

Furthermore, for \( i, j > r \) we have \( u_i \in \mathcal{N}\{A^H\} \) and \( v_j \in \mathcal{N}\{A\} \) and thus

\[
u_i^H A v_j = 0, \quad i, j > r.\] (4.18)

Finally, equations (4.17) and (4.18) can be written in matrix notation as

\[
U^H A V = \Sigma
\] (4.19)

which proves (4.1).

### 4.2 The Pseudoinverse

Consider the following linear system of equations

\[
Ax = b
\] (4.20)

where \( A \) is any \( m \times n \) matrix, \( x \) is the \( n \times 1 \) variable vector and \( b \) is a given \( m \times 1 \) vector.

The pseudoinverse solution \( x^+ = A^+ b \) as defined below gives the most general solution to (4.20) in the sense that \( x^+ \) is an exact solution if an exact solution does exist, it is a least squares solution when an exact solution does not exist, and it is a minimum norm solution when the solution is nonunique. When a unique solution does exist, the pseudoinverse coincides with the usual inverse \( A^{-1} \).

**Theorem 4.2.1 (The Pseudoinverse)** Let the SVD of the \( m \times n \) matrix \( A \) of rank \( r \) be given by \( A = U \Sigma V^H \) according to theorem 4.1.1. The \( n \times m \) pseudoinverse \( A^+ \) is then given by

\[
A^+ = V \Sigma^+ U^H
\] (4.21)

where the \( n \times m \) matrix \( \Sigma^+ \) is given by

\[
\Sigma^+ = \begin{pmatrix}
\frac{1}{\sigma_1} & 0 & \cdots & 0 \\
0 & \ddots & \ddots & 0 \\
0 & \cdots & \frac{1}{\sigma_r} & 0 \\
0 & 0 & \cdots & 0
\end{pmatrix}.
\] (4.22)

The pseudoinverse solution \( x^+ = A^+ b \) is a solution to (4.20) with the following properties

1. The solution \( x^+ = A^+ b \) is a least squares solution in the sense that \( \|Ax^+ - b\| \) is minimum.
2. The solution \( x^+ = A^+ b \) is a minimum norm solution in the sense that \( \|x^+\| \) is minimum.
**Proof:** Property 1 above is proved by showing that the error vector \( Ax^+ - \mathbf{b} \) is orthogonal to the range space \( \mathcal{R}\{A\} \). Since \( \{\mathbf{u}_1, \ldots, \mathbf{u}_r\} \) is a basis for \( \mathcal{R}\{A\} \) we consider the following scalar products for \( j = 1, \ldots, r \)

\[
(Ax^+ - \mathbf{b})^H \mathbf{u}_j = (A\mathbf{A}^+ \mathbf{b} - \mathbf{b})^H \mathbf{u}_j = (U\Sigma V^H \Sigma^+ U^H \mathbf{b} - \mathbf{b})^H \mathbf{u}_j = (U\Sigma^+ U^H \mathbf{b} - \mathbf{b})^H \mathbf{u}_j = (\sum_{i=1}^{r} \mathbf{u}_i^H \mathbf{b} - \mathbf{b})^H \mathbf{u}_j = \sum_{i=1}^{r} \mathbf{b}^H \mathbf{u}_i^H \mathbf{u}_j - \mathbf{b}^H \mathbf{u}_j = \mathbf{b}^H \mathbf{u}_j - \mathbf{b}^H \mathbf{u}_j = 0.
\]

Property 2 is proved by showing that \( x^+ \) belongs to the row space \( \mathcal{R}\{A^H\} \). We have

\[
x^+ = A^+ \mathbf{b} = V \Sigma^+ U^H \mathbf{b} = \sum_{i=1}^{r} \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^H \mathbf{b}
\]

and since \( \{\mathbf{v}_1, \ldots, \mathbf{v}_r\} \) is a basis for \( \mathcal{R}\{A^H\} \) we see that \( x^+ \in \mathcal{R}\{A^H\} \). Now, let \( \hat{\mathbf{b}} = A x^+ \) be the unique projection of \( \mathbf{b} \) onto the column space \( \mathcal{R}\{A\} \) according to property 1. Suppose that \( \mathbf{x} \) is any other solution such that \( A \mathbf{x} = \hat{\mathbf{b}} \). This \( \mathbf{x} \) can then be uniquely written as \( \mathbf{x} = x^+ + \mathbf{x}_n \) where \( x^+ \in \mathcal{R}\{A^H\} \) and \( \mathbf{x}_n \in \mathcal{N}\{A\} \). Since \( \|\mathbf{x}\|^2 = \|x^+\|^2 + \|\mathbf{x}_n\|^2 \geq \|x^+\|^2 \) it is concluded that \( x^+ \) is the minimum norm solution.

\[
\square
\]

### 4.3 Antenna Array MIMO systems

We consider a Multiple–In–Multiple Out (MIMO) antenna array system given by the following model

\[
\begin{pmatrix}
\tilde{y}_1(t) \\
\vdots \\
\tilde{y}_m(t)
\end{pmatrix} =
\begin{pmatrix}
h_{11}(t) * \tilde{x}_1(t) + \cdots + h_{1n}(t) * \tilde{x}_n(t) \\
\vdots \\
h_{m1}(t) * \tilde{x}_1(t) + \cdots + h_{mn}(t) * \tilde{x}_n(t)
\end{pmatrix} +
\begin{pmatrix}
\tilde{n}_1(t) \\
\vdots \\
\tilde{n}_m(t)
\end{pmatrix}
\]

(4.25)

consisting of \( n \) transmitting antenna input signals \( \tilde{x}_j(t) \), \( m \) receiving antenna output signals \( \tilde{y}_i(t) \), \( mn \) impulse responses \( h_{ij}(t) \) and \( m \) noise sources \( \tilde{n}_i(t) \), see also [12–16].

Assuming that the input signals are narrowband, we may write \( \tilde{x}_j(t) = x_j(t) e^{j\omega_0 t} \) where \( x_j(t) \) is the baseband signal and \( \omega_0 \) is the carrier frequency. Now, we have the following convolution relationship

\[
h_{ij}(t) * \tilde{x}_j(t) = \int_{-\infty}^{\infty} h_{ij}(\tau) \tilde{x}_j(t-\tau) d\tau = \int_{-\infty}^{\infty} h_{ij}(\tau) x_j(t-\tau) e^{j\omega_0(t-\tau)} d\tau.
\]

(4.26)

Assuming that the input signal \( x_j(t) \) is slowly varying (contains low frequencies) as compared to the impulse response \( h_{ij}(t) \), and by using the approximation \( x_j(t-\tau) \approx x_j(t) \) which is valid over the impulse response support \( [0, \tau_h] \), we get

\[
h_{ij}(t) * \tilde{x}_j(t) \approx x_j(t) \int_{0}^{\tau_h} h_{ij}(\tau) e^{-j\omega_0 \tau} d\tau e^{j\omega_0 t}.
\]

(4.27)

Hence, the corresponding baseband signal is given by

\[
x_j(t) H_{ij}(\omega_0)
\]

(4.28)

where \( H_{ij}(\omega_0) \) is the Fourier transform of \( h_{ij}(t) \) at the carrier frequency \( \omega_0 \).
Using the narrowband assumptions above, the baseband signal model corresponding to (4.25) is given by

\[
\begin{pmatrix}
y_1(t) \\
y_2(t) \\
\vdots \\
y_m(t)
\end{pmatrix} = \begin{pmatrix}
H_{11}(\omega_0) & \cdots & H_{1n}(\omega_0) \\
\vdots & \ddots & \vdots \\
H_{m1}(\omega_0) & \cdots & H_{mn}(\omega_0)
\end{pmatrix} \begin{pmatrix}
x_1(t) \\
x_2(t) \\
\vdots \\
x_n(t)
\end{pmatrix} + \begin{pmatrix}
n_1(t) \\
n_2(t) \\
\vdots \\
n_m(t)
\end{pmatrix}
\]  
(4.29)

or in matrix notation

\[
y(t) = Hx(t) + n(t)
\]
(4.30)

where \(x(t)\) is the \(n \times 1\) (transmitting antenna array) input vector, \(y(t)\) is the \(m \times 1\) (receiving antenna array) output vector, \(H\) is the \(m \times n\) channel matrix and \(n(t)\) is the \(m \times 1\) noise vector.

Note! In the derivation above we have assumed that the delay \(T\) between the transmitting and receiving antenna arrays is incorporated in the input signal \(x(t)\). Hence, strictly speaking, the transmitted signal at time \(t\) is \(x(t + T)\).

### 4.3.1 Optimum beamforming for SISO array systems

We consider an optimum beamforming problem for a Single–In–Single–Out (SISO) array system. We suppose that there is one single input signal of interest \(s(t)\) and that the input vector is given by

\[
x(t) = s(t)w_x
\]
(4.31)

where \(w_x\) is the transmitting antenna array beamforming vector.

The receiving antenna array output vector is given by

\[
y(t) = Hx(t) + n(t)
\]
(4.32)

where \(H\) is the channel matrix and \(n(t)\) is the noise. The noise is assumed to be uncorrelated with covariance matrix \(E\{n(t)n^H(t)\} = \sigma_n^2I\). The SVD of the channel matrix is given by \(H = U\Sigma V^H\).

The output signal of the system is obtained by applying the receiving antenna array beamforming vector \(w_y\) to the received output vector \(y(t)\)

\[
z(t) = w_y^H y(t) = w_y^H Hx(t) + w_y^H n(t)
\]
(4.33)

and the total output power is

\[
E\{|z(t)|^2\} = \sigma_s^2 |w_y^H Hw_x|^2 + \sigma_n^2 w_y^H w_y
\]
(4.34)

where \(\sigma_s^2 = E\{|s(t)|^2\}\) is the variance of the signal and \(\sigma_n^2 = E\{|n_i(t)|^2\}\) is the variance of the noise.

We restrict the norm of the transmitting vector \(w_x\) such that \(w_y^H w_x = 1\). Hence, the total transmitted power is \(E\{x^H(t)x(t)\} = E\{|s(t)|^2 w_y^H w_x\} = E\{|s(t)|^2\} = \sigma_s^2\). We can now establish the following inequality for the output signal to noise ratio

\[
SNR = \frac{\sigma_s^2 |w_y^H Hw_x|^2}{\sigma_n^2 w_y^H w_y} \leq \frac{\sigma_s^2 w_y^H w_y \cdot w_y^H H H w_x}{\sigma_n^2 w_y^H w_y} = \frac{\sigma_s^2 \cdot w_y^H H H w_x}{\sigma_n^2 w_x^H w_x} \leq \frac{\sigma_s^2}{\sigma_n^2}
\]
(4.35)

where the first inequality is the Cauchy–Schwarz inequality with equality iff \(w_y \sim Hw_x\). The second inequality is a Rayleigh quotient which is maximized when \(w_x \sim v_1\) which is the eigenvector corresponding the largest eigenvalue \(\sigma_1\) of the matrix \(H^H H\). The receiving beamforming vector is hence given by \(w_y = \alpha H w_x = \alpha H v_1 = \beta u_1\) where \(\beta\) is a constant.
To summarize, the output signal to noise ratio (4.35) is maximized by the transmitting beamforming vector

\[ w_x = v_1 \]  

and the receiving beamforming vector

\[ w_y = \beta u_1 \]  

where \( \beta \) is a constant.

### 4.3.2 The SVD and multiple beamforming

We will now investigate the use of the SVD to accomplish multiple independent parallel communication channels in a general Multiple–In–Multiple–Out (MIMO) communication system, see also [12–16]. Consider the MIMO communication model given in (4.29) and (4.30), and depicted in Fig. 4.1 below.

\[ x(t) \xrightarrow{H} y(t) = Hx(t) + n(t) \]

Figure 4.1: General MIMO communication model.

By introducing the SVD of the channel matrix \( H = U \Sigma V^H \) in (4.30) we have

\[ y(t) = U \Sigma V^H x(t) + n(t) \]  

and by multiplying with the orthogonal matrix \( U^H \) from the left we get

\[ U^H y(t) = \Sigma V^H x(t) + U^H n(t). \]  

Now, we introduce the substitutions

\[ \tilde{x}(t) = V^H x(t) \]
\[ \tilde{y}(t) = U^H y(t) \]
\[ \tilde{n}(t) = U^H n(t) \]

which means that (4.39) can be written as

\[ \tilde{y}(t) = \Sigma \tilde{x}(t) + \tilde{n}(t). \]  

The corresponding modified communication systems are depicted in Fig. 4.2 below where \( \tilde{x}(t) \) is input signal, \( \tilde{n}(t) \) noise and \( \tilde{y}(t) \) output signal. This modified communication system requires a preprocessor \( x(t) = V \tilde{x}(t) \) and a postprocessor \( \tilde{y}(t) = U^H y(t) \). Note that if the noise \( n(t) \) is uncorrelated, i.e if \( E\{n(t)n^H(t)\} = \sigma_n^2 I \), then the modified noise is also uncorrelated, i.e. \( E\{\tilde{n}(t)\tilde{n}^H(t)\} = E\{U^H n(t)n^H(t)U\} = U^H E\{n(t)n^H(t)\}U = \sigma_n^2 I \).

The system in (4.43) consists of a diagonal matrix \( \Sigma \) and can thus be interpreted as \( r \) parallel uncoupled systems

\[ \tilde{y}_1(t) = \sigma_1 \tilde{x}_1(t) + \tilde{n}_1(t) \]
\[ \vdots \]
\[ \tilde{y}_r(t) = \sigma_r \tilde{x}_r(t) + \tilde{n}_r(t) \]
where $r$ is the rank of the channel matrix. The system in (4.43) or (4.44) can thus be interpreted as a communication system with $r$ parallel input signals $\tilde{x}_i(t)$, noise $\tilde{n}_i(t)$, channel gain $\sigma_i$ and output signals $\tilde{y}_i(t)$ for $i = 1, \ldots, r$. If the noise is Gaussian, and since the noise is uncorrelated with $E\{\tilde{n}_i(t)\tilde{n}_j^*(t)\} = \sigma_n^2 \delta_{ij}$, these parallel, uncoupled channels are also statistically independent.

![Figure 4.2: Equivalent MIMO communication systems.](image)

**4.3.3 A far–field multiscattering channel model**

We consider a simple far–field multiscattering channel model as depicted in Fig. 4.3. It is assumed that the antenna arrays are uniformly distributed linear arrays (ULA’s) with distance $d$ between elements. The array response vectors for both the transmitting and receiving arrays are thus given by

$$a(\phi) = \begin{pmatrix} 1 \\ e^{-j\frac{\pi d}{\lambda} \sin(\phi)} \\ \vdots \\ e^{-j\frac{\pi d}{\lambda} \sin(\phi)(M-1)} \end{pmatrix}$$  \hspace{1cm} (4.45)

with elements $a_l(\phi) = e^{-j\frac{\pi d}{\lambda} \sin(\phi) l}$ for $l = 0, 1, \ldots, M - 1$. Here $M = n$ for the transmitting array and $M = m$ for the receiving array.

In Fig. 4.3 is illustrated the signal path from transmit antenna output signal $x_l(t)$ to the receive array signal $y(t)$ via one single scatterer with transmit angle $\phi_t$ and receive angle $\phi_r$. With $n$ transmitting antennas the receive array signal $y(t)$ becomes

$$y(t) = \sum_{l=1}^{n} x_l(t)A e^{-j\frac{\pi d}{\lambda} \sin(\phi_t) a(\phi_r) = A \cdot a(\phi_r)^T a(\phi_t) x(t)}$$  \hspace{1cm} (4.46)

where $A$ is a complex gain factor associated with the particular path.

If there are two scattering signal paths we obtain the channel model

$$y(t) = [A_1 \cdot a(\phi_{t1})^T a(\phi_{r1}) + A_2 \cdot a(\phi_{t2})^T a(\phi_{r2})] x(t)$$  \hspace{1cm} (4.47)

where $A_1$ and $A_2$ are complex gains and $\phi_{r1}, \phi_{r2}, \phi_{t1}, \phi_{t2}$ are the receive and transmit angles for the two signal paths, respectively. The channel model for any number of scatterers is obtained similarly.
4.3.4 Information capacity

The information capacity in \( \text{bits/sec/Hz} \) (or in \( \text{bits/channel use} \)) for the MIMO signal model in (4.30) is given by

\[
C = \log_2 \det(I + \frac{1}{\sigma_n^2} \mathbf{R}_x \mathbf{H} \mathbf{H}^H)
\]  

(4.48)

where \( \mathbf{R}_x \) is the covariance matrix of the input array signal \( \mathbf{x}(t) \), i.e. \( \mathbf{R}_x = E\{\mathbf{x}(t)\mathbf{x}^H(t)\} \) and \( \sigma_n^2 \) the variance of the uncorrelated and Gaussian noise, see e.g., \([12–15, 17]\). Assuming that the input signal \( \mathbf{x}(t) \) is uncorrelated, the capacity of the equivalent parallel uncoupled system in (4.43) or (4.44) is given by

\[
C = \log_2 \det(I + \frac{1}{\sigma_n^2} \mathbf{\Sigma}_R \mathbf{\Sigma}^H) = \sum_{i=1}^{r} \log_2(1 + \frac{1}{\sigma_n^2} \sigma_i^2 P_i)
\]  

(4.49)

where \( \sigma_i \) are the singular values of the channel and \( P_i \) is the power of each uncorrelated input signal \( \tilde{x}_i(t) \), i.e. \( P_i = E\{|\tilde{x}_i(t)|^2\} \). Hence, the covariance matrix of \( \mathbf{x}(t) \) is given by

\[
\mathbf{R}_{\tilde{x}} = E\{\tilde{x}(t)\tilde{x}^H(t)\} = \mathbf{V}^H \mathbf{R}_x \mathbf{V} = \begin{pmatrix} P_1 & \cdots \\ \vdots & \ddots \\ & & P_n \end{pmatrix}.
\]  

(4.50)

Given that the total transmitted output power is constrained, i.e.

\[
P_{\text{tot}} = \text{trace}\{\mathbf{R}_x\} = \text{trace}\{\mathbf{R}_{\tilde{x}}\} = \sum_{i=1}^{n} P_i,
\]  

(4.51)

the optimum power strategy \( P_i \) maximizing \( C \) in (4.49) and satisfying (4.51) is obtained by the so called waterfilling strategy [14]. According to the waterfilling strategy (which can be proved using convex optimization theory), there is a number \( \gamma \) such that

\[
P_i = \begin{cases} 
\gamma - \frac{\sigma_i^2}{\sigma_n^2} & \text{if } \gamma - \frac{\sigma_i^2}{\sigma_n^2} > 0 \\
0 & \text{if } \gamma - \frac{\sigma_i^2}{\sigma_n^2} \leq 0
\end{cases}
\]  

(4.52)
\[ \sum_{i=1}^{n} P_i = \sum_{i=1}^{N_{\text{act}}} \left( \gamma - \frac{\sigma_i^2}{\sigma_n^2} \right) = P_{\text{tot}} \]  

(4.53)

where \( N_{\text{act}} \) is the number of active channels for which \( P_i > 0 \). The solution to (4.52) and (4.53) is the optimum waterfilling solution which maximizes the capacity \( C \) under a total transmitted power constraint \( P_{\text{tot}} \). Observe that it is assumed that the singular values are organized in decreasing order \( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 \).

The following algorithm can be used to calculate the optimum waterfilling solution [14] and the optimum channel capacity of a given channel \( H \), noise variance \( \sigma_n^2 \) and total power \( P_{\text{tot}} \).

**Example 4.3.1** The Waterfilling algorithm.

1. Given the channel \( H \), the noise variance \( \sigma_n^2 \) and the total power \( P_{\text{tot}} \). Calculate the SVD of \( H \) to get \( H = U \Sigma V^H \), the rank \( r \) and the singular values \( \sigma_1, \ldots, \sigma_r \) in decreasing order.

2. Use the following algorithm to calculate the optimum waterfilling solution \((P_i, N_{\text{act}})\)

   - Initialize with \( N_{\text{act}} = r \)
   - Calculate:
     \[ \gamma = \frac{\sigma_n^2 \sum_{i=1}^{N_{\text{act}}} \frac{1}{\sigma_i^2} + P_{\text{tot}}}{N_{\text{act}}} \]
     \[ P_i = \gamma - \frac{\sigma_n^2}{\sigma_i^2} \]
     for \( i = 1, \ldots, N_{\text{act}} \)  
     - If \( P_i > 0 \), for \( i = 1, \ldots, N_{\text{act}} \)
       then stop
     - else
       \[ N_{\text{act}} = N_{\text{act}} - 1 \]
       and return to calculate \( \gamma \)

3. Calculate the optimum channel capacity as

   \[ C = \sum_{i=1}^{N_{\text{act}}} \log_2(1 + \frac{1}{\sigma_n^2 \sigma_i^2 P_i}) \]  

(4.55)

**4.3.5 The Alamouti space–time block code**

Below we describe a simple space–time block coding strategy called the Alamouti code [18] which is useful for a \( 2 \times 2 \) channel \( H \). The number of transmit and receive antennas is hence \( M = m = n = 2 \). The assumption here is that the transmitter does not have channel knowledge, but the receiver has full knowledge about the channel \( H \). Hence, the coding strategy is independent of the channel and does not employ any information about \( H \). The receiver on the other hand, has full knowledge about the channel and is employing the explicit values of \( H \) in the decoder.

Suppose that we wish to send a message \( m = 0, 1, 2, 3 \) (2 bits) using QPSK modulation. The modulated baseband signal at each antenna at time \( n \) is then given by

\[ s(n) = A e^{j \pi m} e^{j m n} \]  

(4.56)

where \( A \) is the transmitted amplitude, and \( A^2 \) the transmitted power. The totally transmitted power by the two antennas is then

\[ P_{\text{tot}} = 2A^2. \]  

(4.57)
We consider the following channel model for two consecutive channel uses at time \( n \) and \( n+1 \)
\[
\begin{align*}
\mathbf{y}(n) &= \mathbf{Hc}(n) + \mathbf{v}(n) \\
\mathbf{y}(n+1) &= \mathbf{Hc}(n+1) + \mathbf{v}(n+1)
\end{align*}
\] (4.58)
where \( \mathbf{c}(n) \) are code words, and \( \mathbf{v}(n) \) are zero mean uncorrelated additive noise with variance \( \sigma_v^2 \). The signal to noise ratio is defined by
\[
\text{SNR} = 10 \log_{10} \left( \frac{P_{\text{tot}}}{\sigma_v^2} \right) = 10 \log_{10} \left( \frac{2A^2}{\sigma_v^2} \right). \tag{4.59}
\]
The two code words \( \mathbf{c}(n) \) and \( \mathbf{c}(n+1) \) in the Alamouti code are given by
\[
\mathbf{c}(n) = \begin{pmatrix} s(n) \\ s(n+1) \end{pmatrix}, \quad \mathbf{c}(n+1) = \begin{pmatrix} -s^*(n+1) \\ s^*(n) \end{pmatrix}. \tag{4.60}
\]
By inserting the code words (4.60) in the signal model (4.58) we get the following signal model in explicit component form
\[
\begin{align*}
y_1(n) &= H_{11}s(n) + H_{12}s(n+1) + v_1(n) \\
y_2(n) &= H_{21}s(n) + H_{22}s(n+1) + v_2(n) \\
y_1(n+1) &= -H_{11}s^*(n+1) + H_{12}s^*(n) + v_1(n+1) \\
y_2(n+1) &= -H_{21}s^*(n+1) + H_{22}s^*(n) + v_2(n+1).
\end{align*}
\] (4.61)
Now, the Alamouti decoding is given by
\[
\begin{align*}
\hat{s}(n) &= H_{11}^*y_1(n) + H_{12}^*y_1^*(n+1) + H_{21}^*y_2(n) + H_{22}^*y_2^*(n+1) \\
\hat{s}(n+1) &= H_{12}^*y_1(n) - H_{11}^*y_1^*(n+1) + H_{22}^*y_2(n) - H_{21}^*y_2^*(n+1)
\end{align*}
\] (4.62)
By inserting the signal model (4.61) into the decoding equations (4.62) we obtain finally
\[
\begin{align*}
\hat{s}(n) &= (|H_{11}|^2 + |H_{12}|^2 + |H_{21}|^2 + |H_{22}|^2)s(n) + v'(n) \\
\hat{s}(n+1) &= (|H_{11}|^2 + |H_{12}|^2 + |H_{21}|^2 + |H_{22}|^2)s(n+1) + v'(n+1)
\end{align*}
\] (4.63)
or
\[
\begin{align*}
\hat{s}(n) &= \text{tr} \{ \mathbf{H}^H \mathbf{y} \} s(n) + v'(n) \\
\hat{s}(n+1) &= \text{tr} \{ \mathbf{H}^H \mathbf{y} \} s(n+1) + v'(n+1)
\end{align*}
\] (4.64)
where \( v'(n) \) and \( v'(n+1) \) are noise terms which are linear combinations of the original noise terms \( v_1(n), v_2(n), v_1(n+1) \) and \( v_2(n+1) \).

Equations (4.63) and (4.64) show the action of the Alamouti coder/decoder. Note that the two equations are decoupled, i.e. the estimate \( \hat{s}(n) \) is proportional to \( s(n) \) (except for the noise) and is independent of the other message \( s(n+1) \), etc. The gain factor for each decoded symbol is \( \text{tr} \{ \mathbf{H}^H \mathbf{H} \} = |H_{11}|^2 + |H_{12}|^2 + |H_{21}|^2 + |H_{22}|^2 \) which displays the spatial diversity obtained. Eventhough there is strong fading, it is very unlikely that all channel components should fade at the same time.

The channel capacity in this case when the receiver knows the channel but the transmitter does not is given by (4.48) and employing \( \mathbf{R}_x = \frac{P_{\text{tot}}}{M} \mathbf{I} \) (the power is distributed equally on the transmitting antennas). Hence,
\[
C = \log_2 \det(\mathbf{I} + \frac{P_{\text{tot}}}{2\sigma_v^2} \mathbf{H}^H \mathbf{H}). \tag{4.65}
\]
Observe that waterfilling is not necessary in this case (4.65).
4.4 Matlab programming tasks

4.4.1 A Far–field multiscattering channel model

Consider the far–field multiscattering channel model as described in section 4.3.3 and as depicted in Fig. 4.3. It is assumed that the antenna arrays are uniformly distributed linear arrays (ULA’s) with distance \(d = \lambda/2\) between elements. The array response vectors for both the transmitting and receiving arrays are thus given by

\[
a(\phi) = \begin{pmatrix} 1 \\ e^{-j\pi \sin(\phi)} \\ \vdots \\ e^{-j\pi \sin(\phi)(M-1)} \end{pmatrix}
\]

(4.66)

with elements \(a_l(\phi) = e^{-j\pi \sin(\phi)l}\) for \(l = 0, 1, \ldots, M - 1\). Here \(M = m = n\) for the transmitting as well as the receiving array.

Suppose that there are two scattering signal paths and we obtain the channel model

\[
H = A_1 \cdot a(\phi_1) a^T(\phi_1) + A_2 \cdot a(\phi_2) a^T(\phi_2)
\]

(4.67)

where \(A_1\) and \(A_2\) are complex gains and \(\phi_1, \phi_2, \phi_{t1}, \phi_{t2}\) are the receive and transmit angles for the two signal paths, respectively. Here

\[
\begin{align*}
A_1 &= 10 \\
A_2 &= 1 \\
\phi_{r1} &= 40\pi/180 \\
\phi_{r2} &= 60\pi/180 \\
\phi_{t1} &= 20\pi/180 \\
\phi_{t2} &= 30\pi/180 \\
M &= 10.
\end{align*}
\]

(4.68)

Hence, the signal path no. 1 has 10 times the magnitude compared to signal path no. 2.

Matlab task: Perform a SVD of the channel \(H\) described above, i.e. \(H = U\Sigma V^H\) and plot the response of the two transmit beamformers

\[
\begin{align*}
v_1^T a(\phi) \\
v_2^T a(\phi)
\end{align*}
\]

(4.69)

(4.70)

as well as the two receive beamformers

\[
\begin{align*}
u_1^H a(\phi) \\
u_2^H a(\phi)
\end{align*}
\]

(4.71)

(4.72)

where \(u_1\) and \(u_2\) are the left singular vectors and \(v_1\) and \(v_2\) are the right singular vectors corresponding to the two largest singular values \(\sigma_1\) and \(\sigma_2\), respectively.

4.4.2 Capacity of the Rayleigh fading MIMO channel

Consider a Rayleigh fading \(m \times n\) channel model \(H\) where the entries of the channel matrix are uncorrelated and complex Gaussian distributed with variance 2. Let \(\sigma_n^2 = 1\) and define the signal to noise ratio by

\[
SNR = 10\log_{10}(\frac{P_{tot}}{\sigma_n^2}) = 10\log_{10}(P_{tot}).
\]

(4.73)

Let the number of antennas be \(M = m = n = 5\).

Matlab task: Develop a Matlab code as follows

38
1. Suppose that the SNR is given, calculate $P_{\text{tot}}$. Let $M = 5$ and simulate one single channel realization using $H = \text{randn}(M,M) + j \cdot \text{randn}(M,M)$. Perform the SVD using $[U, S, V] = \text{svd}(H)$. The rank $r$ can be calculated using $r = \text{rank}(H)$. Use $\text{diag}(S)$ to extract the singular values. Write a Matlab code that calculates the optimum channel capacity $C$ in (4.55) according to the waterfilling algorithm as described in Example 4.3.1. Observe that here $\sigma_n^2 = 1$. Calculate also the channel capacity $C_1$ when only one single channel is employed

$$C_1 = \log_2(1 + \sigma_1^2 P_{\text{tot}})$$

(4.74)

where $\sigma_1$ is the largest singular value.

2. Modify the code so that $C$ and $C_1$ are averaged over 1000 realizations of $H$.

3. Modify the code, and calculate and plot the average $C$ and $C_1$ for values of SNR ranging from -20 to 40 dB.

### 4.4.3 The Alamouti space–time block code

Write a Matlab code that simulates a Rayleigh fading channel and implements the Alamouti coder and decoder described in section 4.3.5.

Use $M = 2$ antennas at both the receiver and the transmitter. One realization of the Rayleigh fading channel can then be simulated by using $H = \text{randn}(2,2) + j \cdot \text{randn}(2,2)$.

Let the noise variance be $\sigma_v^2 = 1$. Suppose that the SNR is given (in dB). The constant amplitude $A$ and total power $P_{\text{tot}}$ can then be calculated using (4.57) and (4.59).

Consider one single channel realization of $H$. Two independent messages $(m_1, m_2 \in \{0, 1, 2, 3\})$ can now be generated as $m_1 = \text{floor}(4 \cdot \text{rand})$ and $m_2 = \text{floor}(4 \cdot \text{rand})$. These two discrete messages can then be QPSK modulated using (4.56) yielding the two complex symbols $s(n)$ and $s(n+1)$.

The complex QPSK symbols $s(n)$ and $s(n+1)$ are then coded using (4.60) and the received signals $y(n)$ and $y(n+1)$ are generated according to the signal model in (4.58) where the random noise $v(n)$ and $v(n+1)$ is obtained as e.g. $v_1 = (\text{randn}(2,1) + j \cdot \text{randn}(2,1))/\sqrt{2}$ and $v_2 = (\text{randn}(2,1) + j \cdot \text{randn}(2,1))/\sqrt{2}$ (zero mean complex Gaussian with variance 1).

The decoded signals $\hat{s}(n)$ and $\hat{s}(n+1)$ are then obtained by using (4.62). Finally, estimates $\hat{m}_1$ and $\hat{m}_2$ of the transmitted messages $m_1$ and $m_2$ are obtained by checking in which quadrant the complex symbols $\hat{s}(n)$ and $\hat{s}(n+1)$ are.

**Matlab task:** Use 1000 realizations of the Rayleigh fading channel $H$ to estimate the probability of an error (wrong decision) $P_{\text{error}}$ using the Alamouti coding/decoding strategy as described above. Calculate and plot $P_{\text{error}}$ as a function of SNR for values of SNR ranging from -20 to 40 dB. Also, calculate and plot the optimum capacity $C$ in (4.65) (averaged over 1000 realizations) as a function of the same values of SNR.
Chapter 5

The MIMO-OFDM system

The basic discrete–time complex baseband MIMO–OFDM system is depicted in Fig. 2. We assume a multiple antenna system with \( N \) transmitting and \( M \) receiving antennas. Here \( s_n(t) \) and \( r_m(t) \) denote the transmitted and received antenna signals, respectively, and \( h_{mn}(t) \) denotes the impulse response connecting the transmitting antenna \( n \) to the receiving antenna \( m \). Let the total delay spread be denoted \( \tau_d \).

\[
\begin{align*}
\text{DFT} & \quad \text{add CP} & \quad \text{rem. CP} & \quad \text{DFT} \\
x_1(k) & \quad \mathbf{V}(k) & \quad s_1(k) & \quad r_1(k) & \quad y_1(k) \\
x_r(k) & \quad s_N(k) & \quad r_M(k) & \quad y_r(k) \\
& \quad s_n(t) & \quad h_{mn}(t) & \quad r_m(t) & \quad r_m(k)
\end{align*}
\]

Figure 5.1: The baseband MIMO–OFDM system.

Given that the cyclic prefix \( CP > \tau_d \cdot f_s \) where \( f_s = 1/T_s \) and \( T_s \) is the symbol time, linear convolution becomes identical with circular convolution over the entire DFT–frame, and we have

\[
r_m(t) = \sum_{n=1}^{N} h_{mn}(t) \otimes s_n(t) + v_m(t), \quad (5.1)
\]

for \( t \in [0, \frac{N_{DFT} - 1}{f_s}] \) where \( N_{DFT} \) is the size of the DFT and \( m = 1, \ldots, M \). Here \( v_m(t) \) is assumed to be uncorrelated Gaussian noise.

Taking the DFT of (5.1), we obtain

\[
r_m(k) = \sum_{n=1}^{N} h_{mn}(k)s_n(k) + v_m(k), \quad (5.2)
\]

for \( k \in [0, N_{DFT} - 1] \), which can be written in matrix form as

\[
r(k) = \mathbf{H}(k)s(k) + \mathbf{v}(k), \quad (5.3)
\]

where \( \mathbf{r}(k) = (r_m(k)) \) is \( M \times 1 \), \( \mathbf{s}(k) = (s_n(k)) \) is \( N \times 1 \), \( \mathbf{v}(k) = (v_m(k)) \) is \( M \times 1 \), and \( \mathbf{H}(k) = (h_{mn}(k)) \) is \( M \times N \). The correlation matrix of the noise vector \( \mathbf{v}(k) \) is given by

\[
E\{\mathbf{v}(k)\mathbf{v}^H(k)\} = \sigma_n^2 \cdot \mathbf{I}_M \text{ where } \sigma_n^2 \text{ is the variance of the noise and } \mathbf{I}_M \text{ is the } M \times M \text{ identity matrix.}
\]
The Singular Value Decomposition (SVD) of the matrix $\mathbf{H}(k)$ is given by $\mathbf{H}(k) = \mathbf{U}(k)\mathbf{\Sigma}(k)\mathbf{V}^H(k)$ where $\mathbf{U}(k)$ is $M \times M$, $\mathbf{V}(k)$ is $N \times N$, $\mathbf{\Sigma}(k)$ is $M \times N$ containing $r$ singular values greater than zero $\sigma_1(k) \geq \cdots \geq \sigma_r(k) > 0$ and $r$ is the rank of the matrix $\mathbf{H}(k)$.

By defining $\mathbf{y}(k) = \mathbf{U}^H(k)r(k)$, $\mathbf{s}(k) = \mathbf{V}(k)\mathbf{x}(k)$ and $\mathbf{n}(k) = \mathbf{U}^H(k)v(k)$ we obtain the diagonalized system $\mathbf{y}(k) = \mathbf{\Sigma}(k)\mathbf{x}(k) + \mathbf{n}(k)$ or

$$
\begin{cases}
  y_1(k) = \sigma_1(k)x_1(k) + n_1(k) \\
  \vdots \\
  y_r(k) = \sigma_r(k)x_r(k) + n_r(k).
\end{cases}
$$

(5.4)

The correlation matrix of the noise vector $\mathbf{n}(k)$ is given by $E\{\mathbf{n}(k)\mathbf{n}^H(k)\} = \sigma_n^2 \mathbf{I}_M$ since $\mathbf{U}(k)$ is orthonormal.

Since the communication channels in (5.4) are completely decoupled and the noise terms $n_m(k)$ are uncorrelated over space as well as over frequency, the maximum information capacity can be calculated as

$$
C = \sum_{k=0}^{N_{DFT} - 1} \sum_{m=1}^{r} \log_2(1 + \frac{\sigma^2_{x_m}(k)}{\sigma_n^2})
$$

(5.5)

where $\sigma^2_{x_m}(k)$ denotes the variance of the uncorrelated input signals $x_m(k)$. The capacity is constrained by the total transmitted power

$$
E = \sum_{k=0}^{N_{DFT} - 1} \sum_{m=1}^{r} \sigma^2_{x_m}(k).
$$

(5.6)

The optimum water–filling solution is easily obtained as follows: Given a testing level $B$. For $k = 0, \ldots, N_{DFT} - 1$ and $m = 1, \ldots, r$, choose

- $\sigma^2_{x_m}(k) = B - \sigma_n^2/\sigma_m^2(k)$ if $B - \sigma_n^2/\sigma_m^2(k) > 0$
- $\sigma^2_{x_m}(k) = 0$ if $B - \sigma_n^2/\sigma_m^2(k) \leq 0$.

Then calculate the corresponding transmitted power level $E$ according to (5.6). The capacity given by (5.5) is now the optimum water–filling solution for this particular power level $E$.

In our numerical examples we plot the capacity versus the average signal–to–noise ratio at the receiver, $\text{SNR}_av$, defined as

$$
\text{SNR}_av = \frac{1}{N_{act}} \sum_{k=0}^{N_{DFT} - 1} \sum_{m=1}^{r} \frac{\sigma^2_{x_m}(k)}{\sigma_n^2} \cdot \sigma_m^2(k)
$$

(5.7)

where $N_{act}$ is the number of active channels for which $\sigma^2_{x_m}(k) \neq 0$.

For large signal–to–noise ratios we see that the capacity in (5.5) can be approximated by

$$
C \sim N_{act} \log_2(\text{SNR}_av).
$$

(5.8)
Bibliography


