ARRAY SIGNAL PROCESSING

An algebraic approach

TU Delft
Faculty of Electrical Engineering, Mathematics, and Computer Science
Section Circuits and Systems

ARRAY SIGNAL PROCESSING

An algebraic approach

EE 4715
Spring 2022

Alle-Jan van der Veen

Contents

PREFACE

This reader contains the course material for the MSc level course on array signal processing at TU Delft, ET4 147. Over the past 20 years, this course was presented as "signal processing for communications"; in 2022 it was combined with another course on speech and audio processing into one that is more general and focuses on multisensor array processing.

Sensor arrays are present in many applications:

- In wireless communications, multiple antennas at the transmitter and/or the receiver allows to increase data rates and suppress unwanted interference.
- In radio astronomy, collections of the hallmark telescope dishes have been the workhorse for many years. The array is called an interferometer. Over the past decade, the dishes have been upgraded with antenna arrays in the focal plane, or been replaced with massive arrays of "simple" (non-steerable) antennas, typically arranged in some hierarchy. Using the observed data of one night (or even many nights), the aim is to create images of the sky, as function of frequency.
- In a medical setting, ultrasound transducers are used to create images of organs in the human body. Such a transducer can consist of a line array of piezo-electric elements, or of a 2D array.
- Still in a medical setting, electrode arrays are used to capture electrical signals from the skull, i.e. electro-encephalogram (EEG) signals. These are then processed to obtain a crude 3D-localized image of functional regions in the brain.
- Microphone arrays are being used to filter out unwanted interference in noise-cancelling headphones. In hearing aids, they are used to focus on an intended speaker while suppressing background noise.
- Other applications are phased array radar, sonar, and seismic exploration.

While these applications are very diverse, the underlying signal processing data models and mathematical techniques are in fact very similar. The course will focus on these data models, introduce the appropriate mathematical technique, then derive generic signal processing algo-

rithms, and relate to one of the applications as an example. While you probably have seen already many models and mathematical techniques in other courses, such as Detection and Estimation, or Machine Learning, or Convex Optimization, the present course will angle towards matrix models and advanced techniques in linear algebra, such as the singular value decomposition, factor analysis, generalized eigenvalues, and some tensor techniques. This is in line with the origins of array processing.

However, we will hardly discuss adaptive techniques related to array signal processing, such as LMS or CMA. An in-depth discussion would need an entire course of its own. Thus, the course is mostly focused on algebraic technques for array processing.

It is assumed that the participants already have a fair background in linear algebra, although one lecture is spent to refresh this knowledge.

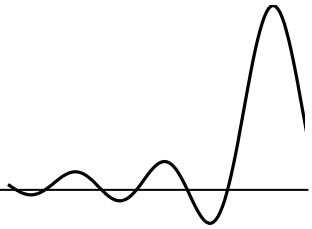
Acknowledgements

The course material is derived in part from previously published papers, stemming from joint work with many colleagues and (former) PhD students over the course of 30 years. In particular I would like to acknowledge the collaboration with Amir Leshem, Stefan Wijnholds, Millad Sardarabadi: text from (overview) papers we wrote together has been used, but edited to fit the course.

Alle-Jan van der Veen
Spring 2022

Chapter **1**

INTRODUCTION



Contents

Signal processing is the theory and engineering art of converting acquired sensor measurements into "information" (or "useful data"). It starts by deriving data models, or concise abstractions of the physics behind the observations. Next, methods are developed and algorithms are proposed to extract the "information". Strongly depending on the application, this could consist of signal parameters, propagation parameters, reconstructed time domain signals, images, etc. Finally, part of signal processing is concerned with efficient implementations on computational platforms.

Array signal processing is the branch of signal processing that considers multiple sensors, or an array of sensors. This could occur in many applications, e.g., an array of antennas in wireless communication, or a microphone array inside hearing aids or teleconferencing equipment.

The received data from the multiple sensors are stacked into vectors, and simple data models express each sensor signal as a linear combination of a stack of transmitted signals $\mathbf{s}(t)$ to which noise $\mathbf{n}(t)$ is added, i.e.,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)\,. \tag{1.1}$$

The tools relevant to analyze and process this data are then found in linear algebra: in this course we will be looking at matrix multiplications and inversion, subspace estimation, eigenvalue decompositions, and more. Since noise is added and needs to be taken into account, we will also need tools from statistics, as seen e.g., in a course on Estimation and Detection. These tools are then generalized to the matrix-vector case.

Thirty years ago, an overview article was publised with the title "Two decades of array signal processing research" [1]. We can thus consider that the area of array signal processing is about 50 years old, although its origins are of course much older (going back, e.g., to optics interferometry).

1.1 APPLICATIONS OF ARRAY PROCESSING

Sensor arrays can be used for many things. This section lists some basic applications.

1.1.1 Diversity

A straightforward application of having multiple sensors is SNR improvement. Suppose we have M sensors each receiving a copy of the desired signal, but with independent additive noise contributions:

$$x_m(t) = s(t) + n_m(t)\,, \qquad m = 1, \cdots, M\,.$$

If the desired signal has power σ_s^2 and the noise has power σ_n^2, then each sensor has SNR

$$\mathrm{SNR}_m = \frac{\sigma_s^2}{\sigma_n^2}\,.$$

If we average the M received signals,

$$x(t) = \frac{1}{M}\sum_{m=1}^{M} x_m(t) = s(t) + \frac{1}{M}\sum_{m=1}^{M} n_m(t)\,,$$

then $s(t)$ is unaffected, but the noised is averaged out: the power of the noise present in $x(t)$ is σ_n^2/M, and the SNR after averaging is

$$\mathrm{SNR}_{\mathrm{out}} = M\frac{\sigma_s^2}{\sigma_n^2}\,.$$

We say that we have an array gain of M. This is easily generalized to the model (1.1), which for a single signal in noise is

$$\mathbf{x}(t) = \mathbf{a}s(t) + \mathbf{n}(t)$$

where in the previous example we had a unit-weight vector $\mathbf{a} = \mathbf{1}$, with $\mathbf{1} = [1, 1, \cdots, 1]^{\mathrm{T}}$. The signal $s(t)$ is recovered by computing the weighted average[1]

$$\hat{s}(t) = \mathbf{w}^{\mathrm{H}}\mathbf{x}(t)\,.$$

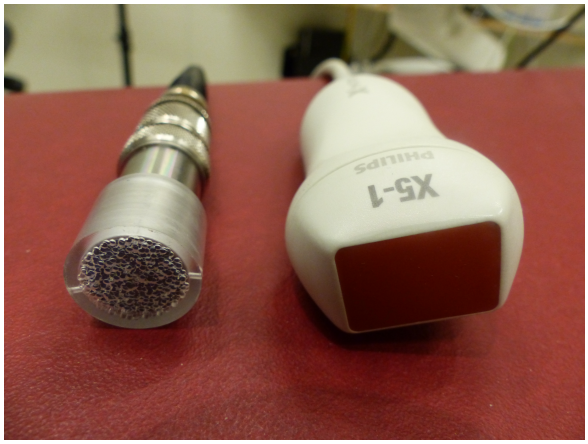
We will see later that the optimal weights are $\mathbf{w} = \mathbf{a}/\|\mathbf{a}\|^2$. The vector $\mathbf{w}$ is known as a *beamformer*.

This is applied in wireless communication, where multiple antennas are used to provide diversity. In the presence of multipath reflections, it may happen that a reflection cancels the desired signal at the location of one antenna. If we have a second antenna at a slightly different location that therefore captures a different linear combination of these signals, we can still receive the signal.

[1] Superscript $^{\mathrm{T}}$ denotes a transpose, and superscript $^{\mathrm{H}}$ a complex conjugate transpose.

(a)

(b)



(c)

Figure 1.1. (a) Example of a radio telescope: The Very Large Array, New Mexico; (b) a single element ultrasound transducer next to a 3D ultrasound array; (c) MiG-35 phased array radar.

1.1.2 Wavefield sampling

An array of sensors is used to sample signals in space. This is useful if the signals have spatial properties: we consider *wavefields*, where signals propagate in space. Much of the early research (1950–1990) is concerned with modeling and estimating the propagation conditions, e.g., directions of arrival, propagation delays, propagation velocities. If we represent directions of arrival in two dimensions, then we obtain images, and direction finding is called image formation. Prime application areas are radar, radio astronomy, ultrasound imaging, underwater acoustics, and seismic exploration. Fig. 1.1 shows examples of sensor arrays in these applications. In relation to (1.1), we would say that these applications are interested in estimating parameters of $\mathbf{A}$: a model for the propagation.

1.1.3 MIMO communication

In other applications, we are interested in the transmitted signals $\mathbf{s}(t)$. E.g., if the matrix $\mathbf{A}$ in (1.1) is invertible, we can compute the estimate $\hat{\mathbf{s}}(t) = \mathbf{A}^{-1}\mathbf{x}(t)$. In this case, the multiple antennas are combined by $\mathbf{A}^{-1}$ such that interfering signals are cancelled and the desired signal is found. A common application is MIMO wireless communication ("multiple input multiple output", i.e., multiple antennas at the transmitter and at the receiver), where we increase the total capacity of the system by spatially separating overlapping signals. Using M antennas, we can expect to separate M overlapping signals and thus to increase our capacity by a factor of M. Fig. 1.2 shows an example of a MIMO antenna array that is used for this. In *Massive MIMO* designs, we have $M > 100$, leading to huge capacity gains but also hardware complexities: not every antenna can be equipped with a transmitter or receiver.

Similarly, in microphone array processing, we are interested in the audio signal (e.g., hearing aids which nowadays employ multiple microphones to enable noise cancellation).

1.2 APPROACH

Signal processing starts with modeling. Given an application, we first construct a forward data model which shows how the received sensor signals depend on the sources of interest and the propagation medium. This can take the simple form of (1.1), but very often, more detail is needed depending on the situation at hand. E.g., antenna gains may be direction dependent, multipath may be present such that delayed signals $s(t-\tau)$ also enter into the model, etc.

In wireless communication, source signals are often known up to the unknown symbols in the message that we try to receive: the signals are deterministic with a number of unknown parameters. In other cases, such as radio astronomy, the source signals are quite random (e.g., described by temporally white Gaussian processes) and it may be more appropriate to define a stochastic data model. We will frequently look at second order correlation models of the form

$$\mathbf{R}_\mathbf{x} = \mathbf{A}\mathbf{R}_\mathbf{s}\mathbf{A}^{\mathrm{H}} + \mathbf{R}_\mathbf{n} \tag{1.2}$$

Figure 1.2. 5G MIMO communication array

where $\mathbf{R_x} = \mathrm{E}[\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t)]$ is the correlation matrix of the received signals, and similarly for $\mathbf{R_s}$ and $\mathbf{R_n}$.

Several assumptions were already made to arrive at this model, e.g., stationarity, and independence of the signals and the noise. In the modeling phase, it is important to specify the assumptions that were made to arrive at the model. Classical array processing textbooks often provide a lot of details on translating wave propagation into models [2]. We will cover some of this in Chap. 2.

Once we have a model for either $\mathbf{x}(t)$ or $\mathbf{R_x}$, we can start to look at methods to estimate the parameters we are interested in. These could be $\mathbf{A}$, or parameters on which $\mathbf{A}$ depends, or the source signals or parameters related to them. Not surprisingly, the methods we consider are based on linear algebra, and various methods target various structures that may be present in the model. E.g., in future chapters we will consider the structure that arises if, in (1.2), $\mathbf{R_s}$ is diagonal, if $\mathbf{R_n}$ is diagonal or equal to $\mathbf{R_n} = \sigma_n^2\mathbf{I}$. This will then result in eigenvalue decomposition problems or, more general, in factor analysis.

Linear algebra was the main workhorse for array signal processing in the period 1990–2010. Since then, the attention has shifted to methods arising from compressed sensing, resulting in formulations of problems as constrained optimization problems. These are then solved using generic optimization techniques. Nonetheless, the focus of the book is on tools from linear algebra.

1.3 NOTES

"Classical" array processing textbooks are the books by Johnson and Dudgeon [2], and Van Trees [3]. The state-of-the-art in 1995 is also quite nicely summarized by the Signal Processing Magazine article of Krim and Viberg [1], Since then, blind beamforming techniques have given a major impetus to the field. A few books giving an overview are found under the headings of blind source separation and independent component analysis [4, 5], although this material is probably better studied by consulting some of the original overview papers [].

A nice overview of applications is found in Haykin [6] which has extensive chapters on geophysics exploration, sonar, radar, radio astronomy, and medical tomographic imaging (e.g., MRI and CT scans). An early introduction to phased array radar is presented in Skolnik [7].

Linear algebra is used throughout the book, and a standard reference to this is Golub and Van Loan [8].

Bibliography

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[2] D.H. Johnson and D.E. Dudgeon, *Array signal processing: concepts and techniques*. Prentice Hall, 1993.

[3] H.L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. Wiley, 2004.

[4] J.V. Stone, *Independent component analysis: a tutorial introduction*. MIT press, 2004.

[5] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[6] S. Haykin, ed., *Array signal processing*. Prentice Hall, 1985.

[7] M.I. Skolnik, *Introduction to radar systems*. McGraw-Hill, 1980.

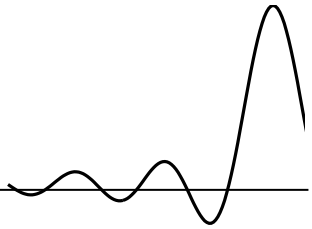
[8] G.H. Golub and C.F. Van Loan, *Matrix computations*. Johns Hopkins University Press, 1996.

Part I

DATA MODELS

Chapter 2

WAVE PROPAGATION



Contents

In signal processing, data models are used as an abstraction of the physics in an application. The model should be based on reality but not be overly detailed. Often, a variety of data models are suitable, with different assumptions leading to different algorithms.

2.1 THE WAVE EQUATION

In free space, RF signals propagate following the Maxwell equations. These describe the relations between the vector electric and magnetic field intensities. If we specialize them to scalar components (most sensors will not measure vector fields), we arrive at the *wave equation*:

$$\nabla^2 s(\mathbf{x}, t) = \frac{1}{c^2} \frac{\partial^2 s(\mathbf{x}, t)}{\partial t^2} \tag{2.1}$$

where ∇^2 is the Laplace operator,

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \,.$$

Here, $\mathbf{x}$ is a position in space; assuming 3D space, $\mathbf{x} = [x, y, z]^{\mathrm{T}}$. The scalar field $s(\mathbf{x}, t)$ is a function of both space $\mathbf{x}$ and time t, and we will call it a signal. The coefficient c in (2.1) will

later be interpreted as the speed of propagation. It depends on the properties of the medium (specifically, the dielectric permittivity and the magnetic permeability).

In acoustics, a similar equation holds for the acoustic pressure of a sound wave in gas or in a fluid, and for the longitudinal and transverse waves in solids. In this case, c represents the speed of sound. In a gas (air), it depends on pressure and temperature.

Different media (or materials) will have different propagation speeds. Interesting effects occur at the interface between materials, or objects in space, such as reflection and diffraction. It is also possible for c to vary continuously in space, e.g., due to gradients in salinity or temperature in ocean water, or due to varying electron densities in the ionosphere.

2.1.1 Plane waves

A typical solution ("eigenfunction") of this equation has the form[1]

$$s(\mathbf{x}, t) = e^{j(\omega t - \mathbf{k}\cdot\mathbf{x})} \,. \tag{2.2}$$

If we insert this function into the wave equation, we find the constraint

$$k = \frac{\omega}{c} \,, \tag{2.3}$$

where $k = \|\mathbf{k}\|$ is called the wavenumber (or spatial frequency, in analogy to its role next to ω in (2.2)), with unit radians per meter. The function $s(\mathbf{x}, t)$ represents a monochromatic plane wave. Indeed, ω represents the radial frequency (in rad/s), and the vector $\mathbf{k}$ is called the wavenumber vector.

To interprete this signal, pick a constant C and look at the function argument,

$$\omega t - \mathbf{k} \cdot \mathbf{x} = C \,.$$

Clearly, for each time t, this describes a plane in 3D space where the function is constant, and this defines a wavefront. The vector $\mathbf{k}$ is the normal to the plane, and indicates the direction of propagation: in directions $\mathbf{x}$ parallel to $\mathbf{k}$, the function argument changes fastest.

For the monochromatic wave, the time period of a cycle is

$$T = \frac{2\pi}{\omega} \,.$$

Over this time, the wavefront moves over a distance

$$\lambda = \frac{2\pi}{k}$$

[1] To remain consistent with the literature, we use here a dot, $\mathbf{k} \cdot \mathbf{x}$, to represent the inner product between two vectors. Other notations are $\langle \mathbf{k}, \mathbf{x} \rangle$ and $\mathbf{k}^{\mathrm{T}}\mathbf{x}$.

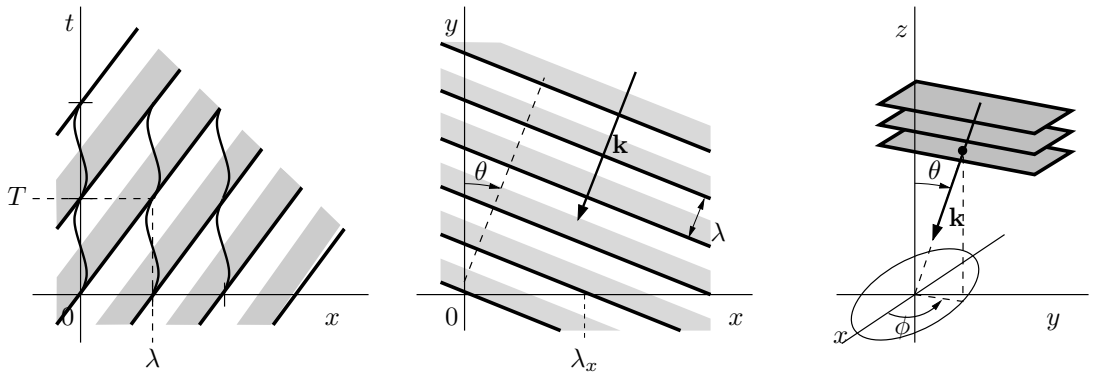


Figure 2.1. Propagation of a monochromatic wave in 1, 2 and 3 dimensions.

meters, in the direction of $\mathbf{k}$. Combining these expressions with (2.3), we see that the distance λ covered over time T has the ratio

$$\frac{\lambda}{T} = \frac{\omega}{k} = c\,,$$

showing that, indeed, c in (2.1) defines the propagation speed. We can interprete λ as the wavelength in meters, and $k/(2\pi) = \lambda^{-1}$ as the number of wave cycles that fit into 1 meter.

Let $\boldsymbol{\zeta}$ be a unit-norm vector in the direction of $\mathbf{k}$ so that $\mathbf{k} = k\boldsymbol{\zeta}$, and write

$$\omega t - \mathbf{k}\cdot\mathbf{x} = \omega\left(t - \frac{1}{c}\boldsymbol{\zeta}\cdot\mathbf{x}\right).$$

This takes care of the constraint (2.3). The factor $1/c$ represents a delay in the propagation direction: the time it takes the wave to cover 1 meter.

Fig. 2.1 shows the propagation in 1, 2 and 3 dimensions. Propagation in 1 dimension, e.g., on a rope, is less relevant for the book, but sometimes provides a nice example as (x,t) can be visualized in a simple plot. The figure shows $s(x,t) = \cos(\omega t - kx)$ where the positive part of the wave is shaded, and the plot shows both the period T and the wavelength λ.

For propagation in 2 dimensions, a 2D plot can show (x,y) but not t, so the meaning of the plot is in fact quite different. We can parametrize the propagation direction $\boldsymbol{\zeta}$ with a single angle θ, the angle of incidence of the wave:

$$\boldsymbol{\zeta} = -\begin{bmatrix}\sin(\theta)\\ \cos(\theta)\end{bmatrix}. \tag{2.4}$$

The minus sign in this expression comes from the choice to let the wave propagate towards the origin. For fixed ω and c, this allows to parametrize $\mathbf{k}$ with a single parameter θ,

$$\mathbf{k} = \begin{bmatrix}k_x\\ k_y\end{bmatrix} = -\frac{\omega}{c}\begin{bmatrix}\sin(\theta)\\ \cos(\theta)\end{bmatrix} = -\frac{2\pi}{\lambda}\begin{bmatrix}\sin(\theta)\\ \cos(\theta)\end{bmatrix}. \tag{2.5}$$

Fig. 2.1(b) shows that the wavefronts are orthogonal to $\mathbf{k}$, and that λ is the (shortest) distance between wavefronts. If we observe the wavefronts only on the x-axis, the apparent wavelength λ_x is longer; similarly, the $\mathbf{k}$-vector projected on the x-axis is shorter. As a result, the apparent propagation velocity is larger and depends on the direction θ. This provides a means to recover the direction θ even from 1D observations, for cases where we know the true propagation velocity c. This plays a role later, when we place our sensors on the x-axis.

Likewise, in 3 dimensions, we will need 2 parameters ϕ and θ, the azimuth and elevation, respectively:

$$\boldsymbol{\zeta} = -\begin{bmatrix}\sin(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi)\\ \cos(\theta)\end{bmatrix}. \tag{2.6}$$

This leads to

$$\mathbf{k} = \begin{bmatrix}k_x\\ k_y\\ k_z\end{bmatrix} = -\frac{\omega}{c}\begin{bmatrix}\sin(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi)\\ \cos(\theta)\end{bmatrix} = -\frac{2\pi}{\lambda}\begin{bmatrix}\sin(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi)\\ \cos(\theta)\end{bmatrix}. \tag{2.7}$$

In the plot, the wavefronts are shown as planes orthogonal to $\mathbf{k}$.

More generally, the wave equation supports the addition of multiple monochromatic solutions, e.g., solutions at different frequencies ω. We can also scale these solutions by some $S(\omega)$, and in the limit we find

$$s\left(t - \frac{1}{c}\boldsymbol{\zeta}\cdot\mathbf{x}\right) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)\, e^{j\omega(t-\frac{1}{c}\boldsymbol{\zeta}\cdot\mathbf{x})}\mathrm{d}\omega\,, \tag{2.8}$$

provided this inverse Fourier transform integral converges. Thus, functions of the form $s(t - (1/c)\boldsymbol{\zeta}\cdot\mathbf{x})$ are also plane waves and a solution of the wave equation. Note that the corresponding time-domain signal $s(t)$ could be anything; it could be a sinusoid, but also a short pulse traveling through space. The reason this pulse does not get distorted on its way is that all frequency components receive the same delay as function of position, as represented by $(1/c)\boldsymbol{\zeta}\cdot\mathbf{x}$. The frequency components remain coherent because in the present formulation c is not a function of frequency. More in general, c does depend on frequency and this leads to *dispersion*: a distortion of the pulse as different frequency components experience different delays (or phase shifts) during propagation.

The additivity of the wave equation also supports the superposition of signals coming from different directions. These can be monochromatic signals at the same frequency, at different frequencies, or general signals of the form (2.8). E.g., the original signal could have been reflected in an object, resulting in a copy of the signal traveling in a different direction (multipath), or we can have multiple sources transmitting from different locations.

2.1.2 Spherical waves

The wave equation (2.1) also admits other solutions than plane waves. If we switch from Cartesian coordinates to spherical coordinates (r, ϕ, θ) centered around the origin, and assume spher-

ically symmetric solutions $s(r,t)$, then these will satisfy the spherical wave equation [1]

$$\nabla^2(rs) = \frac{1}{c^2}\frac{\partial^2(rs)}{\partial t^2}\,.$$

This is the same equation as before, but now in terms of $rs(t)$ instead of $s(t)$. We thus obtain similar solutions as before, but now as a function of radius r, and scaled by $1/r$. E.g., the monochromatic spherical wave, propagating from the origin, has the form

$$s(r,t) = \frac{1}{r}e^{j(\omega t - kr)}\,,$$

and a more general solution has the form $s(r,t) = \frac{1}{r}s(t - \frac{1}{c}r)$. The scaling by $1/r$ is related to the Friis freespace transmission equation used in telecommunication and radar.

Far away from the origin, in the so-called far field, the solution can be approximated by a plane wave in case we only study a limited part of space.

2.1.3 Dispersive and diffractive effects

The wave equation (2.1) describes propagation in a lossless, homogeneous medium with constant propagation velocity c. However, the situation may be more general:

- If the medium is not lossless, waves get attenuated as they propagate.
- If the medium is non-homogeneous, it has different propagation speeds in different areas, and refraction occurs at the interfaces: part of the wave could be reflected, and part could be transmitted but at a different angle.

  Refraction leads to multipath: a propagating signal arrives at a receiver via several paths, each with its own direction, attenuation, and propagation delay.
- Diffraction is the effect of waves bending around objects. In RF, this happens around sharp edges, e.g., rooftops in mobile communication, but scattering also occurs on small objects such as raindrops. The physics of this are complicated!

Other interesting effects occur if the propagation velocity is frequency dependent: in this case we do not have the linear relation $\omega = ck$, and dispersion occurs. An example is a prism, where, at the interface with air, different colors of light are bent in slightly different angles. Other examples are the propagation of RF signals through the atmosphere or ionosphere, or the acoustic propagation in the ocean, where propagation speed depends on salinity and temperature.

The effect of dispersion is that of a linear filter $H(r,\omega)$, which introduces range-dependent, frequency-dependent effects in the transmitted signal. Sensors placed in the far field (r large) will measure the same waveforms, but they are not equal to the transmitted waveform because of the frequency-dependent filtering: pulse distortion has occured.

2.2 SPATIAL FOURIER TRANSFORMS

The Fourier transform has shown to be an essential tool in the analysis of linear time-invariant systems: such systems are characterized by an impulse response, an input signal is convolved by this impulse response, and in the Fourier domain, this convolution becomes a frequency-wise multiplication with the transfer function of the system. The transfer function is the Fourier transform of the impulse response.

Viewed in another way, the inverse Fourier transform shows that a signal can be represented as a sum of sinusoids, $e^{j\omega t}$.

2.2.1 Space-time Fourier transform

For space-time signals $s(\mathbf{x}, t)$, we can generalize by defining

$$S(\mathbf{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k}\cdot\mathbf{x})} \mathrm{d}\mathbf{x}\mathrm{d}t \,,$$

where, for convenience, the three-dimensional integral over space is represented by a single integral sign. $S(\mathbf{k}, \omega)$ is called the wavenumber-frequency representation, and correponding plots are called F-K plots (with frequency in Hz). Such plots are used in seismic exploration (geophysics) and underwater acoustics [2].

The corresponding inverse Fourier transform is

$$s(\mathbf{x}, t) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega)\, e^{j(\omega t - \mathbf{k}\cdot\mathbf{x})} \mathrm{d}\mathbf{k}\mathrm{d}\omega \,. \tag{2.9}$$

Equation (2.2) showed that a monochromatic plane wave is given by $e^{j(\omega t - \mathbf{k}\cdot\mathbf{x})}$. Thus, (2.9) shows that any space-time signal can be represented by a weighted sum of monochromatic plane waves.

If $s(\mathbf{x}, t)$ is a single monochromatic plane wave with frequency ω_0 and wavenumber vector $\mathbf{k}_0$,

$$s(\mathbf{x}, t) = e^{j(\omega_0 t - \mathbf{k}_0\cdot\mathbf{x})} \,, \tag{2.10}$$

then its spectrum is

$$S(\mathbf{k}, \omega) = (2\pi)^4 \delta(\omega - \omega_0)\delta(\mathbf{k} - \mathbf{k}_0) \,, \tag{2.11}$$

which represents a single point in wavenumber-frequency space. This expression is verified by inserting (2.11) into (2.9).

Let us extend this to a wideband source with spectrum $S(\omega)$. The velocity of propagation was previously shown to be $c = \omega/k$ with $k = \|\mathbf{k}\|$. Since the velocity of propagation is given by the medium, $\mathbf{k}$ and ω are not completely independent. We can pick the direction of propagation, $\boldsymbol{\zeta}_0$ (a unit-norm vector), and then $\mathbf{k} = \frac{\omega}{c}\boldsymbol{\zeta}_0$. A single wideband plane wave thus traces a line in wavenumber-frequency space (i.e., in an F-K plot), and

$$S(\mathbf{k}, \omega) = (2\pi)^3 S(\omega)\delta\left(\mathbf{k} - \mathbf{k}_0\right) \,; \qquad \mathbf{k}_0 = \frac{\omega}{c}\boldsymbol{\zeta}_0 \,. \tag{2.12}$$

This generalizes (2.11) to wideband sources. Since they come from a single direction, such sources will be called point sources (as opposed to spatially extended sources).

Next, we could look at filters. Working by analogy, a space-time filter can be defined by a frequency response $H(\mathbf{k}, \omega)$, and the output of the filter is

$$Y(\mathbf{k}, \omega) = H(\mathbf{k}, \omega) S(\mathbf{k}, \omega) \,.$$

The corresponding time domain signal is given by a convolution

$$y(\mathbf{x}, t) = h(\mathbf{x}, t) * s(\mathbf{x}, t) = \int \int h(\mathbf{x} - \mathbf{p}, t - \tau)\, s(\mathbf{p}, \tau)\, \mathrm{d}\mathbf{p}\mathrm{d}\tau$$

Practically speaking, it is not clear how such filters can be realized, as they act over all of space.

2.2.2 Apertures

In the next section we will look at sampling. Obviously, we will not be able to place sensors anywhere in space: normally they will be placed on a line or within a limited spatial region. In analogy to optics, the area over which we will sample space is called the aperture, and it acts as a spatial window $w(\mathbf{x})$:

$$y(\mathbf{x}, t) = w(\mathbf{x}) s(\mathbf{x}, t) \,. \tag{2.13}$$

E.g., for $\mathbf{x} = [x, y, z]^{\mathrm{T}}$, a linear aperture on the x-axis of size D (a "slit") is defined by

$$h(x) = \begin{cases} 1, & |x| < D/2 \\ 0, & \text{otherwise} \end{cases} \quad \Leftrightarrow \quad \mathbf{w}(\mathbf{x}) = h(x)\delta(y)\delta(z) \,, \tag{2.14}$$

and a circular aperture on the (x, y) plane in 3D with diameter D is

$$h(x, y) = \begin{cases} 1, & x^2 + y^2 < D/2 \\ 0, & \text{otherwise} \end{cases} \quad \Leftrightarrow \quad \mathbf{w}(\mathbf{x}) = h(x, y)\delta(z) \,. \tag{2.15}$$

For time-domain signals, we know that a product in time domain becomes a convolution in frequency domain. Thus, by analogy, applying the space-time Fourier transform to (2.13) yields

$$Y(\mathbf{k}, \omega) = \frac{1}{(2\pi)^3} W(\mathbf{k}) * S(\mathbf{k}, \omega) = \frac{1}{(2\pi)^3} \int W(\mathbf{k} - \mathbf{p}) S(\mathbf{p}, \omega) \mathrm{d}\mathbf{p} \tag{2.16}$$

where the aperture smoothing function is

$$W(\mathbf{k}) = \int w(\mathbf{x})\, e^{j\,\mathbf{k}\cdot\mathbf{x}} \mathrm{d}\mathbf{x} \,. \tag{2.17}$$

For the linear aperture (2.14) and with $\mathbf{k} = [k_x, k_y, k_z]^{\mathrm{T}}$, we obtain

$$W(\mathbf{k}) = \frac{\sin(k_x D/2)}{k_x/2} \,, \tag{2.18}$$

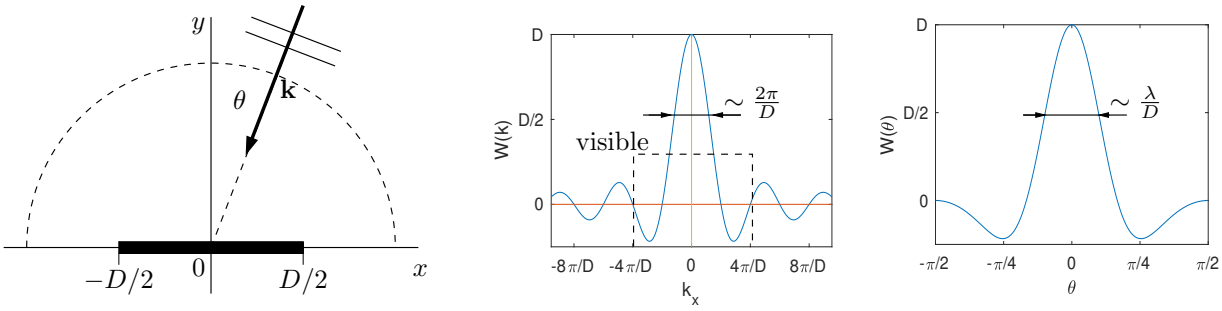


Figure 2.2. Aperture function in a 2D scenario. (a) a linear aperture (slit); (b) the corresponding $W(\mathbf{k})$, only the k_x component is shown; (c) $W(\theta)$, for $D = 2\lambda$.

which is a sinc function in k_x, and constant in k_y, k_z. For the plane wave signal $s(\mathbf{x}, t)$ with wavenumber-frequency transform (2.12), the resulting spectrum is

$$Y(\mathbf{k}, \omega) = W(\mathbf{k}) * S(\omega)\delta(\mathbf{k} - \mathbf{k}_0) = S(\omega)W(\mathbf{k} - \mathbf{k}_0)\,; \qquad \mathbf{k}_0 = \frac{\omega}{c}\boldsymbol{\zeta}_0\,. \tag{2.19}$$

Thus, the effect of the aperture (window) in spatial domain is a convolution of the signal in wavenumber-frequency domain, which smears out (smooths) the spatial spectrum. The resulting signal does not come from a single direction $\boldsymbol{\zeta}_0$ anymore, but appears to come from a range of directions around $\boldsymbol{\zeta}_0$. Thus, the effect of the aperture is a limitation on resolution.

For the linear aperture, (2.18) shows that we get some dilution in the k_x component, while k_y and k_z are completely dropped: the y and z components of the field are not measured. Thus, by looking through the slit, only a 1D propagation scenario is visible. Signals with the same k_x but different k_y, k_z are indistinguishable.

Fig. 2.2 shows a linear aperture in a 2D scenario, and the corresponding function $W(\mathbf{k})$. Since it only depends on k_x, only this component is shown. It is seen from (2.18) that the peak of the sinc function has magnitude D. The first zero crossing occurs at $k_x = 2\pi/D$, hence the main lobe width is said to be approximately $2\pi/D$ (the exact value depends on the definition of width). Thus, as $D \to \infty$, the sinc function converges to a delta spike, as expected. Consider now the parametrization of $\mathbf{k}$ as in (2.5). Then

$$k_x = -\frac{2\pi}{\lambda}\sin(\theta)\,.$$

Clearly, as θ varies from $-\pi/2$ to $\pi/2$, then k_x ranges between $\pm\frac{2\pi}{\lambda}$, and this is the part of the plot of $W(k_x)$ that is "visible" for fixed λ and varying direction of arrival θ. In this parametrization, we find (with some abuse of notation)[2]

$$W(\theta) = D\,\frac{\sin(\frac{D}{\lambda}\pi\sin(\theta))}{\frac{D}{\lambda}\pi\sin(\theta)}\,.$$

[2]Correct notation would define $W_{\mathbf{k}}(\mathbf{k})$ and $W_\theta(\theta)$, but we would like to avoid such adorned notation.

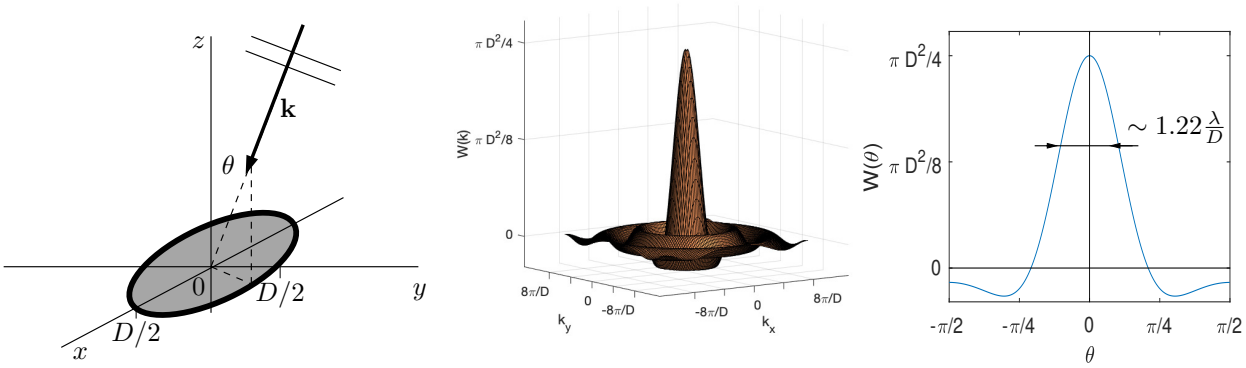


Figure 2.3. Aperture function in a 3D scenario. (a) a circular aperture; (b) the corresponding $W(\mathbf{k})$, only the (k_x, k_y) components are shown; (c) $W(\theta)$, for $D = 2\lambda$.

This function is plotted in the right panel, for $D = 2\lambda$. The first zero crossing occurs for $(D/\lambda)\pi \sin(\theta) = \pi$, i.e., $\sin(\theta) = \lambda/D$. Thus, we see that the main lobe width in the θ-plot is approximately λ/D. This will later be interpreted as the angular resolution of this aperture. Since the maximum k_x that can be obtained is $(D/\lambda)\,\pi$, we also see that only part of the plot of $W(k_x)$ is visible, as indicated by the dashed box. For a given D, the visible part depends on λ, and the ratio D/λ determines the number of sidelobes of $W(\mathbf{k})$ that are visible in $W(\theta)$.

Note that we defined $W(\theta)$, but to compute the response for a source from direction θ_0, we cannot work with $W(\theta - \theta_0)$. Instead, starting from (2.19), we we can write $Y(\theta, \omega) = S(\omega)W(\theta; \theta_0)$, where

$$W(\theta; \theta_0) = D\,\frac{\sin(\frac{D}{\lambda}\pi[\sin(\theta) - \sin(\theta_0)])}{\frac{D}{\lambda}\pi[\sin(\theta) - \sin(\theta_0)]}\,.$$

For θ_0 close to $\frac{1}{2}\pi$, the beamshape will not only center around θ_0, but be distinctively different, with a much broader main lobe.

In 3D, if we take a square aperture around the origin in the (x, y)-plane, then $\mathbf{w}(\mathbf{x}) = h(x)h(y)\delta(z)$, and

$$W(\mathbf{k}) = \frac{\sin(k_x D/2)}{k_x/2}\,\frac{\sin(k_y D/2)}{k_y/2}\,. \tag{2.20}$$

Ideally, we design apertures such that $W(\mathbf{k})$ is as close to a delta spike as possible: the width of the main lobe determines the spatial resolution in applications such as direction finding. On the other hand, we don't necessarily need narrow main lobes in all three dimensions of $\mathbf{k}$: for direction finding, we are interested in the direction vector $\boldsymbol{\zeta}$, and if c is known, there are only 2 independent dimensions to specify $\boldsymbol{\zeta}$.

Circulair aperture In 3D, for a circular aperture with diameter $D = 2R$, one shows that [1]

$$W(\mathbf{k}) = \frac{2\pi R}{k_x y} J_1(k_{xy} R)\,, \qquad k_{xy} = \sqrt{k_x^2 + k_y^2}\,,$$

where $J_1(\cdot)$ is the first-order Bessel function of the first kind. This smooting function is known in optics as the *Airy disk*. It describes the pattern (bright spot and rings around it) that is visible on a screen placed behind a small uniformly illuminated aperture. Fig. 2.3 shows the aperture, $W(\mathbf{k})$ and $W(\theta)$. The Bessel function is quite similar to a two-dimensional sinc function, but note that it is circularly symmetric (unlike (2.20))..

The first zero crossing of $J_1(x)$ occurs for $x = 3.8317\ldots$. Using (2.7), we find

$$\frac{2\pi}{\lambda} \sin(\theta) R = 3.8317 \qquad \Leftrightarrow \qquad \sin(\theta) = 1.22\,\frac{\lambda}{D} \tag{2.21}$$

where $D = 2R$ is the diameter of the array. Again, the beamwidth of this aperture is determined by λ/D.

2.3 SPATIAL SAMPLING

To measure a field, practically we can observe it only over some finite area in space: the aperture. We could use, e.g., a parabolic dish with diameter D, and then the aperture is the size of the dish. The dish casts the incoming energy onto (usually) a single sensor, and because of its directionality, we will have to scan it to cover all directions.

If D is large, then this is not practical. Instead, we can place a number of sensors inside the area covered by the aperture. This *sensor array* will spatially sample the wavefield. The sensors could be simple antennas, or they could be small dishes or arrays themselves, leading to a hierarchy of arrays. For the moment, we will assume that the sensors are ideal, omnidirectional antennas, i.e., they simply capture $s(\mathbf{x}, t)$ at a specific position $\mathbf{x}$.

At first, the theory of spatial sampling can be presented as a direct extension of the usual temporal sampling: sampling creates periodicities in the spectrum, leads to aliasing, and bandlimited signals can be perfectly reconstructed from their samples. Aliasing in this context means that sources from two different directions will result in the same sampled signal and thus cannot be distinguished.

Fig. 2.4 shows some of the notation we will be using in this section. We will first sample the wavefield, and subsequently apply the aperture (= select a finite number of spatial samples) and also apply weights to the selected samples. We use $X(\mathbf{k}, t)$ and $Y(\mathbf{k}, t)$ to keep track of the related spatial spectra.

2.3.1 Infinite number of sensors

To study spatial sampling, let us start with a 1D scenario, where a field $s(x, t)$ is sampled at regular locations $x_m = m\,d$, where d in meters is the distance between the sensors. The samples

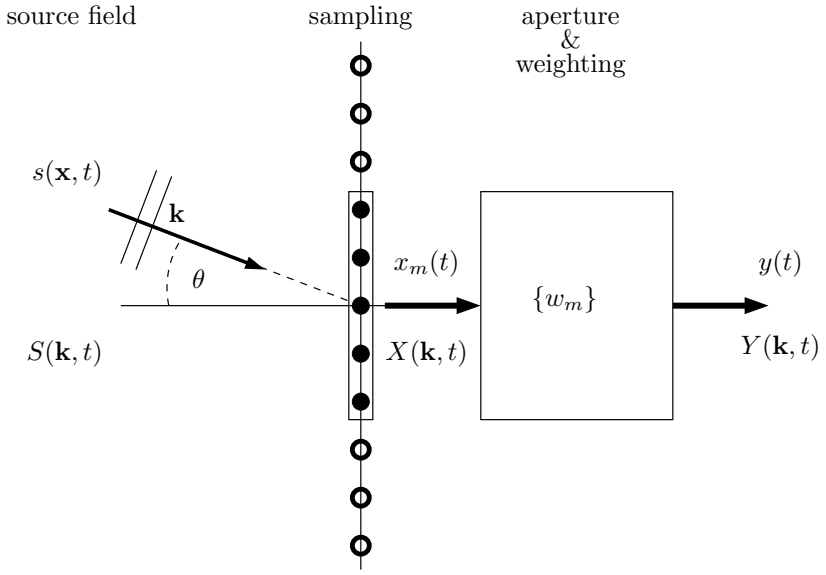


Figure 2.4. Notation related to spatial sampling and beamforming.

are

$$x_m(t) = s(md, t)\,, \qquad m = \cdots, -1, 0, 1, \cdots\,.$$

An infinite number of sensors is needed, but this will be managed later. The original "continuous" signal $s(x,t)$ has space-time spectrum

$$S(k,\omega) = \int\int s(x,t)e^{-j(\omega t - kx)}\mathrm{d}x\mathrm{d}\omega = \int \left[\int s(x,t)e^{jkx}\mathrm{d}x\right] e^{-j\omega t}\mathrm{d}\omega\,,$$

Note that the Fourier transform over space and time decouples. For simplicity of notation, we will instead consider here only the spatial spectrum (omitting the transformation in time): let

$$S(k,t) = \int s(x,t)e^{jkx}\mathrm{d}x\,.$$

This looks like the usual Fourier transform, except for the minus sign in the exponent, but that is of no consequence. Inversely,

$$s(x,t) = \frac{1}{2\pi}\int S(k,t)e^{-jkx}\mathrm{d}k\,.$$

In analogy to time-domain sampling, define the spatial sampling frequency as

$$k_s = \frac{2\pi}{d}\,.$$

Next, we split the integration over k into a fundamental interval $(-k_s/2, k_s/2]$, plus shifts nk_s of this interval, for $n = \cdots, -1, 0, 1, \cdots$. This leads to

$$\begin{aligned} s(x,t) &= \frac{1}{2\pi} \sum_n \int_{nk_s - k_s/2}^{nk_s + k_s/2} S(k,t) e^{-jkx} \mathrm{d}k \\ &= \frac{1}{2\pi} \sum_n \int_{-k_s/2}^{k_s/2} S(k - nk_s, t) e^{-jkx} e^{-jnk_s x} \mathrm{d}k \,. \end{aligned}$$

Sampling x (but not t, yet), and using $nk_s md = 2\pi nm$,

$$\begin{aligned} x_m(t) = s(md,t) &= \frac{1}{2\pi} \sum_n \int_{-k_s/2}^{k_s/2} S(k - nk_s, t) e^{-jkdm} e^{-j2\pi nm} \mathrm{d}k \\ &= \frac{1}{2\pi} \int_{-k_s/2}^{k_s/2} \left[\sum_n S(k - nk_s, t) \right] e^{-jkdm} \mathrm{d}k \,. \end{aligned} \tag{2.22}$$

Let us compare this to the spectrum that we can define for the sampled signal, in analogy to the DTFT. Various definitions are possible, and we opt for

$$X(k,t) = \sum_m x_m(t) e^{jkdm} \quad \Leftrightarrow \quad x_m(t) = \frac{d}{2\pi} \int_{-k_s/2}^{k_s/2} X(k,t) e^{-jkdm} \mathrm{d}k \,. \tag{2.23}$$

This spectrum is defined on the fundamental interval $-k_s/2 \le k \le k_s/2$, and for larger k it is periodic (since $k_s = 2\pi/d$). The usual factor $1/2\pi$ in the inverse transform is replaced here by $d/(2\pi) = 1/k_s$ because we have defined the spectrum using k, instead of a normalized frequency variable kd which would range from $-\pi$ to π.

Comparing to (2.22), we see that the spectrum of the sampled signal is related to that of the unsampled signal via

$$X(k,t) = \frac{1}{d} \sum_n S(k - nk_s, t) \,, \qquad -\frac{1}{2} k_s \le k \le \frac{1}{2} k_s \,,$$

and periodic elsewhere.

The summation in this expression represents aliasing: the original spectrum is shifted by multiples of the sampling frequency k_s, and these copies are all added. If the original spectrum is zero outside the interval $[-k_s/2, k_s/2]$, then these copies do not overlap, and

$$X(k,t) = \frac{1}{d} S(k,t) \,, \qquad -\frac{1}{2} k_s \le k \le \frac{1}{2} k_s \,.$$

This is the equivalent of the Nyquist sampling rate condition, corresponding to a spatially bandlimited signal. For constant c, the relation $k = \omega/c$ allows to immediately translate this to a condition for a temporal-frequency bandlimited signal,

$$|k| < \frac{k_s}{2} \quad \Leftrightarrow \quad |\omega| < \frac{c}{d}\pi \quad \Leftrightarrow \quad |f| < \frac{c}{2d}$$

where $\omega = 2\pi f$, with f in Hz. Alternatively, the distance between the sensors has to satisfy

$$d < \frac{c}{2B}\,,$$

where $B = f_{\max}$ is the bandwidth of the signal in Hz. Or, if $\lambda_{\min} = c/B$ is the smallest wavelength in the signal,

$$d < \tfrac{1}{2}\lambda_{\min}\,.$$

In words: the distance between sensors has to be less than half of the shortest wavelength in the signal. If this Nyquist condition holds, then Shannon's sampling theorem states that the continuous signal can be recovered perfectly by lowpass filtering the periodic spectrum of the sampled signal, which amounts to sinc interpolation. Consequently, no information is lost if the sensors are spaced closer than $\frac{1}{2}\lambda_{\min}$. Otherwise, aliasing will occur, which will be problematic for direction finding (or imaging) applications.

These results extend to higher dimensions. For a wavefield in 2D, we use uniform sampling in 2 dimensions, etc. Nonetheless, this theory is not entirely satisfying yet, as we would like (*i*) to sample using a finite number of sensors, (*ii*) to sample 3D space using only a 2D array, (*iii*) to consider using a random (non-uniformly spaced) array.

2.3.2 Finite number of sensors

A finite number of sensors is effectively obtained if we multiply the infinite samples by an aperture function that selects the range of sensors to be used. As we saw in (2.16), the effect of this is a convolution in the spatial spectrum with the aperture smoothing function $W(\mathbf{k})$, which results in smearing of the spectrum.

To study this, consider in the 1D case a uniform linear array with M sensors at locations $x_m = md,\ m = 0, \cdots, (M-1)d$. To select this range, define the aperture weighting function

$$w_m = \begin{cases} 1, & m = 0, \cdots, M-1 \\ 0, & \text{elsewhere}\,. \end{cases} \tag{2.24}$$

The spatial spectrum of the sampled signal using M sensors is

$$Y(k,t) = \sum_{m=0}^{M-1} x_m(t) e^{jk\cdot x_m} = \sum_{m=-\infty}^{\infty} w_m s(md,t) e^{jkmd}\,.$$

Let $X(k,t)$ be the (periodic) spectrum of the sampled signal using an infinite number of sensors, $m = -\infty, \cdots, \infty$. Using (2.23) gives

$$\begin{aligned} Y(k,t) &= \frac{d}{2\pi} \sum_{m=-\infty}^{\infty} w_m e^{jkmd} \int_{-k_s/2}^{k_s/2} X(p,t) e^{-jpmd} \mathrm{d}p \\ &= \frac{d}{2\pi} \int_{-k_s/2}^{k_s/2} \left[\sum_{m=-\infty}^{\infty} w_m e^{j(k-p)md} \right] X(p,t) \mathrm{d}p\,. \end{aligned} \tag{2.25}$$

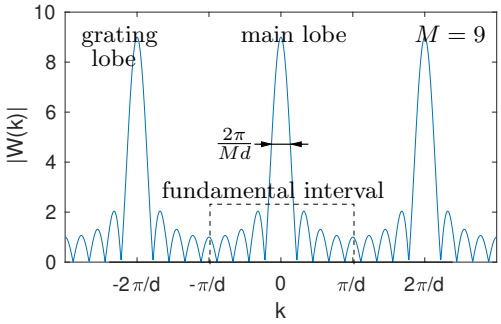


Figure 2.5. Amplitude of the discrete aperture function $W(k)$ for $M = 9$ sensors. The plot is periodic with period $k_s = \frac{2\pi}{d}$.

Now define the discrete aperture function

$$W(k) = \sum_{m=-\infty}^{\infty} w_m e^{jkmd} = \sum_{m=0}^{M-1} e^{jkmd}, \tag{2.26}$$

which is periodic with period $k_s = d/(2\pi)$. Then (2.25) can be written as

$$Y(k,t) = \frac{d}{2\pi} \int_{-k_s/2}^{k_s/2} W(k-p)X(p,t)\mathrm{d}p\,. \tag{2.27}$$

This is recognized as a (circular) convolution of $W(k)$ with the discrete spatial spectrum $X(p,t)$, over one period of the spectrum. This convolution will smooth the spectrum and limit its resolution. It will also introduce sidelobes, as we will now see. From (2.26), we obtain

$$W(k) = \sum_{m=0}^{M-1} e^{jkmd} = \frac{1-e^{jkMd}}{1-e^{jkd}} = \frac{\sin(kMd/2)}{\sin(kd/2)} e^{jk(M-1)d/2}\,. \tag{2.28}$$

The factor $e^{jk(M-1)d/2}$ is simply a phase factor that determines the phase center of the array, and can be ignored here. The real part (sin over sin) can be viewed as a "periodic sinc-function" (it is known as the Dirichlet kernel and occurs in convergence proofs for Fourier series). The amplitude $|W(k)|$ is periodic with period $k_s = \frac{2\pi}{d}$, as determined by the denominator. It has zero crossings for $k = \frac{2\pi}{Md} = k_s/M$.

A plot that shows this beamshape is shown in Fig. 2.5, for $M = 9$. The periodicity with k_s is clearly visible. The peak of $|W(k)|$ is equal to M, called the array gain. The width of the main lobe is determined by the first zero crossing, i.e., $\frac{2\pi}{Md} = k_s/M$. Ideally, for $M \to \infty$, $W(k)$ converges to a delta spike train, such that the convolution (2.27) does not change the spectrum: $Y(k,t) = X(k,t)$. For finite M, the convolution with the main lobe will smear out the spectrum, and reduce its resolution to $\frac{2\pi}{Md}$. Indeed, suppose the spectrum $S(k,t)$ contains two point sources (i.e., delta spikes at specific values for k). The convolution with $W(k)$ will

replace the delta spikes by the main lobe of $W(k)$. If the delta spikes are closer to each other than approximately $\frac{2\pi}{Md}$, then the main lobes will highly overlap, and appear in the spectrum as a single point source. This is similar to the discussion in Sec. 2.2.2 where we looked at the effect of an aperture. Note that $D = Md$ can be interpreted as the spatial coverage (aperture) of the array.

We also note that, next to the main lobe, there are in total $M - 1$ side lobes within the fundamental interval, in between the zero crossings of the sinus function in the nominator of (2.28). The sidelobes in Fig. 2.5 will cause *confusion*: if in the spectrum $Y(k, t)$ we observe a small peak, we will not know if it is a weak point source, or if it is a side lobe of another (strong) source. Thus, the sidelobes limit the sensitivity of the array.[3]

Due to the periodicity, the main lobe is repeated outside the fundamental interval; these lobes are called *grating lobes*. They might appear in the spectrum if the visible region is larger than the fundamental interval.

2.3.3 More general weights

Until now, we selected in (2.24) aperture weights that were either 0 (outside the aperture) or 1 (for the M sensors inside the aperture). However, we are not bound to take the non-zero weights equal to 1: we can select other weights. Doing so will allow us to design other smoothing functions than the Dirichlet kernel which, after all, has quite high sidelobes. Thus we define, generalizing (2.26),

$$W(k) = \sum_{m=0}^{M-1} w_m e^{jkmd}. \tag{2.29}$$

The nonzero weights are called a shading, or tapering, of the array; in general they could be complex numbers. $W(k)$ can be interpreted as a (discrete-space) Fourier transform of the sequence $[w_m]$, similar to the DTFT. Thus, similar design techniques as used for digital filters can be applied here to design weights that result in a desired "transfer function" $|W(k)|$ with minimal sidelobe heights or other desired features. For example, instead of the rectangular window (2.24), we can use a triangular window, a Hann or Hamming window, etc., or apply other window/filter design techniques such as Parks/McClellan. An extensive overview is given in [3, Ch. 3].

2.3.4 1D array in a 2D propagation scenario

In the above, we analyzed the 1D case. An extension to higher dimensions is straightforward, at least if we consider uniform rectangular arrays in 2D, or uniform cubic arrays in 3D.

Let us look now at a 1D array in a 2D propagation scenario, i.e., consider a uniform linear array

[3] We will later discuss deconvolution techniques that attempt to undo the convolution by $W(k)$.

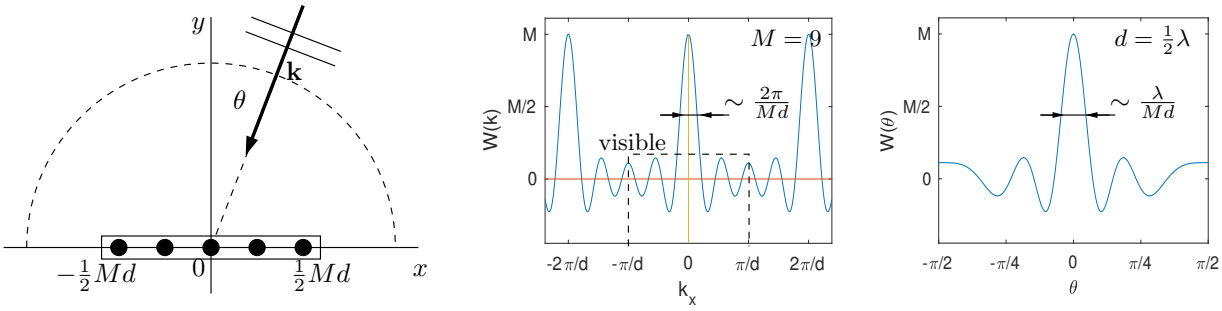


Figure 2.6. Uniform linear array with $M = 9$ sensors in a 2D scenario. (a) Configuration; (b) the corresponding $W(k_x)$; (c) $W(\theta)$, for $d = \frac{1}{2}\lambda$.

of M sensors in 2D. Thus, we observe only k_x, and drop k_y. Then, using (2.28),

$$W(\mathbf{k}) = W(k_x) = \frac{\sin(k_x Md/2)}{\sin(k_x d/2)}\,,$$

(the phase offset due to the non-zero phase center of the array was dropped for simplicity: the array is centered around the origin), and with $k_x = -\frac{2\pi}{\lambda}\sin(\theta)$ we obtain

$$W(\theta) = \frac{\sin(\frac{Md}{\lambda}\pi\sin(\theta))}{\sin(\frac{d}{\lambda}\pi\sin(\theta))}\,.$$

We saw in Sec. 2.2.2 that the relation between D and λ determined the part of $W(k)$ that is visible in case we fix λ and scan θ: the visible part is the interval $[-\frac{2\pi}{\lambda}, \frac{2\pi}{\lambda}]$. Comparing to Fig. 2.5, we see that if $d < \frac{1}{2}\lambda$, then the visible part is within one period of $W(k_x)$. Using similar arguments as before, we estimate the angular resolution as $\lambda/(Md)$. There are $M-1$ zero crossings, which corresponds to the number of sidelobes.

Fig. 2.6 shows $W(k_x)$ and $W(\theta)$, for $M = 9$ and $d = \frac{1}{2}\lambda$. For this choice of d, exactly one period of $W(k_x)$ is visible. Only the visible part is in $W(\theta)$, where we see that the horizontal axis is stretched at the edges (around $\pm\frac{\pi}{2}$) compared to $W(k_x)$. If we take $d > \frac{1}{2}\lambda$, then we do not satisfy the Nyquist criterion, and the resulting aliasing may result in visible grating lobes: secondary main lobes of sources may appear, especially for sources with angles close to $\pm\frac{1}{2}\pi$.

2.3.5 Irregular sampling

More in general, we can consider an irregular array, where M sensors are placed "randomly", at locations $\mathbf{x}_m$ in 3D. If $S(\mathbf{k}, t)$ is the wavefield, then at a location $\mathbf{x}$,

$$s(\mathbf{x}, t) = \int S(\mathbf{k}, t) e^{j(\omega t - \mathbf{k}\cdot\mathbf{x}_m)} \mathrm{d}\mathbf{k}\,.$$

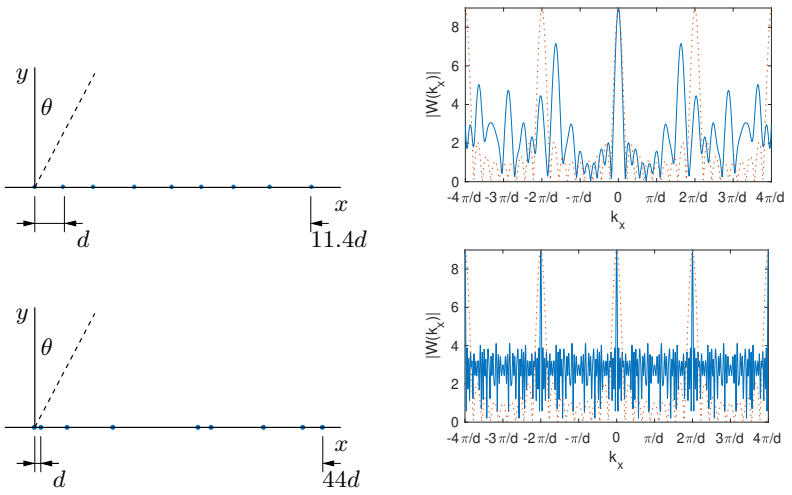


Figure 2.7. Amplitude of the discrete aperture function $W(k_x)$ for $M = 9$ non-uniformly spaced sensors. The smallest spacing is d. Two designs are shown: a random one with an aperture slightly larger than the uniform design, and a sparse nonredundant array with a much larger aperture. The dotted lines correspond to a uniform array with M sensors spaced at d.

Taking a finite number of samples at locations $\mathbf{x}_m$ and weighting them by a taper w_m gives

$$y(\mathbf{x}, t) = w(\mathbf{x}) \cdot s(\mathbf{x}, t)\,,$$

with

$$w(\mathbf{x}) = \sum_{m=0}^{M-1} w_m \delta(\mathbf{x} - \mathbf{x}_m)\,.$$

Let the (continuous) spatial Fourier transform of $y(\mathbf{x}, t)$ be $Y(\mathbf{k}, t)$. Using a similar derivation as before, we obtain

$$Y(\mathbf{k}, t) = \frac{1}{(2\pi)^3} \int W(\mathbf{k} - \mathbf{p}) S(\mathbf{p}, t) \mathrm{d}\mathbf{p} = \frac{1}{(2\pi)^3} W(\mathbf{k}) * S(\mathbf{k}, t)\,,$$

where, inserting $w(\mathbf{x})$ in (2.17),

$$W(\mathbf{k}) = \sum_{m=0}^{M-1} w_m e^{j\mathbf{k}\cdot\mathbf{x}_m}$$

is the spatial Fourier transform of $w(\mathbf{x})$. This generalizes the previous definition (2.29) of $W(k)$ to the non-uniform case. Again, the sampled spectrum $Y(\mathbf{k}, t)$ is the convolution of the original spectrum with a smoothing function $W(\mathbf{k})$. However, if the sensor locations are not uniformly spaced, then $W(\mathbf{k})$ and $Y(\mathbf{k}, t)$ will not be periodic, and it will be hard to analyze theoretically.

Two examples are shown in Fig. 2.7. The first array is mildly irregular, with a sensor at $x = 0$, one at $x = d$, and subsequent ones uniformly selected randomly between d and $1.5\,d$ away from the previous one. In total $M = 9$ sensors are used, all with equal weights. Since the array is irregular, it is seen that the plot of $|W(k_x)|$ is non-periodic. The aperture of this array is not much larger than Md, and the main lobe width is not much narrower than that for a uniform linear array of M sensors spaced at d (the red dotted line). Grating lobes are present, but not at the full height of the main lobe, and a bit closer than would be expected for a minimum spacing of d. Moreover, these plots greatly vary if another, similar, random design is selected. It is clear that without a design process such random arrays will not have desired properties.

The second array in Fig. 2.7 is based on a uniform distance d. It can be viewed as an $M = 45$ uniform array that is subsequently thinned to $M = 9$; this is called a sparse linear array. The spacings between the sensors are [1]

$$[1,\ 4,\ 7,\ 13,\ 2,\ 8,\ 6,\ 3] \cdot d\,.$$

Because of the underlying uniformity (all sensor spacings are a multiple of d), the spectrum $W(\mathbf{k})$ is periodic with period $k_s = 2\pi/d$. The main lobe is much narrower than before, as the aperture is $D = 45d$. As a penalty, the side lobes are now much stronger and appear noise-like. It could be argued that the effective array gain is only a factor 3 rather than 9.

To analyze array designs, let

$$c(\mathbf{x}) = w(\mathbf{x}) * w(-\mathbf{x}) \tag{2.30}$$

be the deterministic "autocorrelation" of $w(\mathbf{x})$. It is a delta sequence with spikes at every baseline between two sensor locations. (A baseline is the distance vector between a pair of sensors.) If multiple baselines are the same (the array is *redundant*), then the scale of the corresponding spike counts the multiplicities. For lag $\mathbf{x} = \mathbf{0}$, we have $c(\mathbf{0}) = M$. The locations where $c(\mathbf{x})$ has spikes forms the *co-array*. The motivation for (2.30) comes from this: since $|W(\mathbf{k})|^2 = W(\mathbf{k})W^*(\mathbf{k})$, then

$$|W(\mathbf{k})|^2 = \int c(\mathbf{x}) e^{j\mathbf{k}\cdot\mathbf{x}} \mathrm{d}\mathbf{x}$$

is the Fourier transform of $c(\mathbf{x})$. Thus, the co-array determines the magnitude of the spectrum smoothing function.

Filter design techniques can be used to determine, starting from a desired $|W(\mathbf{k})|^2$, the corresponding $c(\mathbf{x})$, and subsequently a set of positions and sensor weights that approximate this $c(\mathbf{x})$. Generally, however, array design comes with many constraints and is not an easy art.

2.4 CORRELATION PROCESSING

In telecommunication, we use array processing to spatially separate sources and receive one signal of interest. In many other applications, we are not so much interested in the source signal

$s(t)$, but more in the propagation parameters, i.e., $\mathbf{k}$ or the unit-norm direction vector $\boldsymbol{\zeta}$. In the 2D case, $\boldsymbol{\zeta}$ is specified by the direction of arrival θ.

In these cases, we can do away with the temporal dimension and work with correlation models. Generally, we look at second-order correlations between the sensor signals. For non-Gaussian sources, we might also consider higher-order statistics, cf. Chap. 11.

Thus, in this section we will consider signals $s(t)$ as random processes. If we limit ourselves to descriptions by second-order statistics, we look at the mean and the variance,

$$\mathrm{E}[s(t)]\,, \qquad \mathrm{E}\left[|s(t)-\mathrm{E}[s(t)]|^2\right]$$

and more in general the autocorrelation function, $r_s(t,t') = \mathrm{E}[s(t)s^*(t')]$. (For generality, complex signals are assumed, and the superscript $*$ denotes the complex conjugate.) Usually we immediately make several simplifying assumptions: we consider that the signal is wide sense stationary, so that the mean is constant over time, the autocorrelation function only depends on the time difference $\tau = t' - t$, and the variance is finite. We can then write

$$r_s(\tau) = \mathrm{E}[s(t+\tau)s^*(t)]\,.$$

Moreover, we usually consider the mean to be zero, $\mathrm{E}[s(t)] = 0$, so that $r_s(\tau)$ equals the variance. In any case, $r_s(0)$ represents the power of the random signal.

Recall from a course on random processes that the power spectral density is defined as the Fourier transform of the autocorrelation function:

$$R_s(\omega) = \int r_s(\tau) e^{-j\omega\tau} \mathrm{d}\tau\,.$$

This can be related to the spectrum $S(\omega)$ of $s(t)$, but we have to be careful as for a random process, the energy in $s(t)$ is infinite: we have to look at the energy per unit time. Therefore, let $s_T(t)$ be equal to $s(t)$ on the interval $(-\frac{1}{2}T, \frac{1}{2}T]$ and zero otherwise, and let $S_T(\omega)$ be the Fourier transform of $s_T(t)$, then one can show that

$$R_s(\omega) = \lim_{T\to\infty} \frac{1}{T} \mathrm{E}\left[|S_T(\omega)|^2\right]\,.$$

For white noise, $r_s(\tau)$ is a delta spike, and $R_s(\omega)$ is a constant. (Its power, $r_s(0)$, is actually infinite, so truly white noise does not exist.)

2.4.1 Monochromatic plane wave

These concepts can be carried over to the spatial domain. Space-time stochastic processes are called *random fields*. To start from basics, consider a scenario with a single plane wave, where the propagating source $s(t)$ is a monochromatic plane wave:

$$s(\mathbf{x},t) = \alpha\, e^{j(\omega_0 t - \mathbf{k}_0\cdot\mathbf{x})}$$

where α is a random (complex) amplitude,

$$\mathrm{E}[\alpha] = 0\,, \qquad \mathrm{E}[\,|\alpha|^2] = P\,.$$

At a position $\mathbf{x}$, we can look at the temporal autocorrelation function

$$r_s(\tau) = \mathrm{E}[s(\mathbf{x}, t+\tau)s^*(\mathbf{x}, t)] = Pe^{j\omega_0\tau}$$

(the source is stationary in time and the result does not depend on the position $\mathbf{x}$). Now, in analogy, consider a sensor at location $\mathbf{x}_0$ and one at location $\mathbf{x}_1$. The spatial cross-correlation function is

$$r_s(\mathbf{x}_0, \mathbf{x}_1, \tau) = \mathrm{E}[s(\mathbf{x}_1, t+\tau)s^*(\mathbf{x}_0, t)] = \mathrm{E}\left[|\alpha|^2 e^{j(\omega_0\tau - \mathbf{k}_0\cdot(\mathbf{x}_1-\mathbf{x}_0))}\right]\,.$$

Note that this depends only on the *baseline* $\mathbf{b} = \mathbf{x}_1 - \mathbf{x}_0$, i.e., the vector pointing from $\mathbf{x}_0$ to $\mathbf{x}_1$. Such a random field is called *homogeneous*, and we can write

$$r_s(\mathbf{b}, \tau) = \mathrm{E}[s(\mathbf{x}_0 + \mathbf{b}, t+\tau)s^*(\mathbf{x}_0, t)]\,.$$

For the monochromatic plane wave, we have

$$r_s(\mathbf{b}, \tau) = Pe^{j\omega_0\tau}e^{-j\mathbf{k}_0\cdot\mathbf{b}}\,.$$

Let $\boldsymbol{\zeta}$ be the unit-norm vector pointing in the direction of $\mathbf{k}_0$, and recall that $\mathbf{k}_0 = \frac{\omega_0}{c}\boldsymbol{\zeta}$, then

$$r_s(\mathbf{b}, \tau) = Pe^{j\omega_0\tau}e^{-j\omega_0\tau_g} \tag{2.31}$$

where

$$\tau_g = \frac{\boldsymbol{\zeta}\cdot\mathbf{b}}{c}\,.$$

See Fig. 2.8. The figure shows that τ_g is the *geometric delay*, the delay of the wavefront in propagating from $\mathbf{x}_0$ to $\mathbf{x}_1$. In the figure, the signal arrives at $\mathbf{x}_1$ first, and therefore the delay is actually an advance, and therefore negative. If $d = |\mathbf{b}|$ and θ is the angle between the source direction and the direction orthogonal to the baseline (*broadside*), then

$$\tau_g = -\frac{d}{c}\sin(\theta)\,.$$

(The minus sign is due to the orientation of $\boldsymbol{\zeta}$, and indeed, for positive θ the delay is negative.) Thus, τ_g is related to the direction of arrival (DOA). Often, DOA estimation algorithms estimate τ_g, or the phase delay $e^{-j\omega_0\tau_g}$, and determine the DOA from this.

Taking the temporal Fourier transform of (2.31) gives the cross power spectral density,

$$R_s(\mathbf{b}, \omega) = 2\pi\, Pe^{-j\omega_0\tau_g}\delta(\omega - \omega_0)\,. \tag{2.32}$$

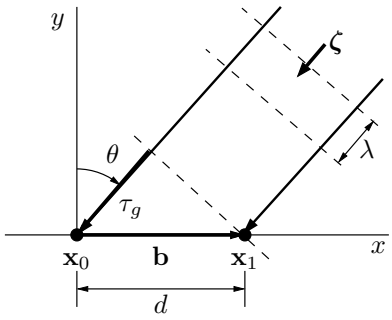


Figure 2.8. Propagating monochromatic source. The projection of $\mathbf{b}$ on the propagation direction determines the geometric delay τ_g.

2.4.2 Wideband plane wave

If we now generalize from a monochromatic source to a wideband source with power spectral density $R_s(\omega)$, we can similarly define

$$r_s(\mathbf{b},\tau) = \frac{1}{2\pi}\int R_s(\omega)e^{j\omega(\tau-\tau_g)}\mathrm{d}\omega\,, \qquad \tau_g = \frac{\boldsymbol{\zeta}\cdot\mathbf{b}}{c}\,. \tag{2.33}$$

Applying the temporal Fourier transform gives the cross power spectral density,

$$R_s(\mathbf{b},\omega) = R_s(\omega)e^{-j\omega\tau_g}\,, \qquad \tau_g = \frac{\boldsymbol{\zeta}\cdot\mathbf{b}}{c}\,. \tag{2.34}$$

which generalizes (2.32). The expression shows that we can write (2.33) as

$$r_s(\mathbf{b},\tau) = r_s(\tau) * \delta(\tau-\tau_g) \tag{2.35}$$

i.e., the crosscorrelation between two sensors (spaced by $\mathbf{b}$) is the autocorrelation of the source, convolved with a delay. Of course, this result could have been obtained also directly!

2.4.3 Random fields

TBD

More in general, we will receive a superposition of multiple sources simultaneously. These can be point sources (with a specific direction $\mathbf{k}_0$), or a random field with a certain source distribution as function of $\mathbf{k}$.

The generalization of (2.35) is

$$r_s(\mathbf{b},\tau) = \frac{1}{(2\pi)^4}\int\int R_s(\mathbf{k},\omega)e^{j(\omega\tau-\mathbf{k}\cdot\mathbf{b})}\mathrm{d}\mathbf{k}\mathrm{d}\omega\,.$$

$$R_s(\mathbf{k},\omega) = \int\int r_s(\mathbf{b},\tau)e^{-j(\omega\tau-\mathbf{k}\cdot\mathbf{b})}\mathrm{d}\mathbf{b}\mathrm{d}\tau$$

ERROR notation clash with (2.34).

White noise random field: $r_s(\mathbf{b}, \tau) = \delta(\mathbf{b}, \tau)$: zero for any nonzero lags in position or time.

Isotropic noise field: random waves propagating in all possible directions with equal probability.

$$S(\mathbf{k}, \omega) = S(\omega)\delta(k - \frac{\omega}{c}), \qquad k = \|\mathbf{k}\|.$$

$S(\omega)$ is the coloring of the noise in the temporal frequency domain. The argument of the delta function selects all $\mathbf{k}$ on a sphere that satisfies the linear dispersion condition.

2.5 APPLICATION: RADIO ASTRONOMY

Astronomical instruments measure cosmic particles or electromagnetic waves impinging on the Earth. Astronomers use the data generated by these instruments to study physical phenomena outside the Earth's atmosphere. In recent years, astronomy has transformed into a multi-modal science in which observations at multiple wavelengths are combined. Fig. 2.9 provides a nice example showing the lobed structure of the famous radio source Cygnus A as observed at 240 MHz with the Low Frequency Array (LOFAR) overlaid by an X-Ray image observed by the Chandra satellite, which shows a much more compact source.

Such images are only possible if the instruments used to observe different parts of the electromagnetic spectrum provide similar resolution. Since the resolution is determined by the ratio of observed wavelength and aperture diameter, the aperture of a radio telescope has to be 5 to 6 orders of magnitude larger than that of an optical telescope to provide the same resolution. This implies that the aperture of a radio telescope should have a diameter of several hundreds of kilometers. Most current and future radio telescopes therefore exploit interferometry to synthesize a large aperture from a number of relatively small receiving elements.

2.5.1 Interferometry

An interferometer measures the correlation of the signals received by two antennas spaced at a certain distance. After a number of successful experiments in the 1950s and 1960s, two arrays of 25-m dishes were built in the 1970s: the 3 km Westerbork Synthesis Radio Telescope (WSRT, 14 dishes, see Fig. 2.10) in Westerbork, The Netherlands and the 36 km Very Large Array (VLA, 27 movable dishes) in Socorro, New Mexico, USA (Fig. 1.1). These telescopes use Earth rotation to obtain a sequence of correlations for varying antenna baselines, resulting in high-resolution images via *synthesis mapping*. A more extensive historical overview is presented in [5].

The radio astronomy community has recently commissioned a new generation of radio telescopes for low frequency observations, including the Murchison Widefield Array (MWA) [6] in Western Australia and the Low Frequency Array (LOFAR) [7] in Europe. These telescopes exploit phased array technology to form a large collecting area with ∼1000 to ∼50,000 receiving elements. The community is also making detailed plans for the Square Kilometre Array (SKA), a future radio

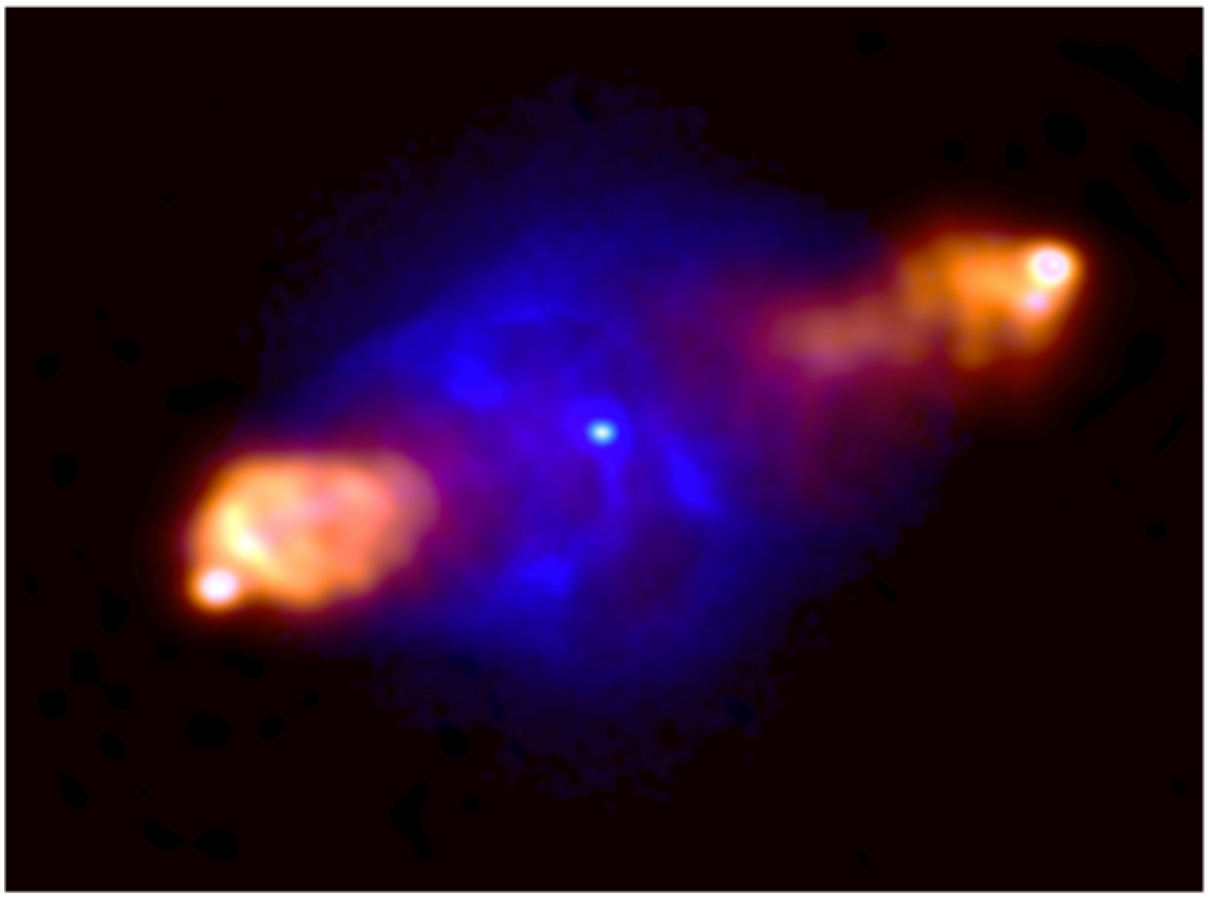


Figure 2.9. Radio image of Cygnus A observed at 240 MHz with the Low Frequency Array (showing mostly the lobes left and right), overlaid over an X-Ray image of the same source observed by the Chandra satellite (the fainter central cloud). *(Courtesy of Michael Wise and John McKean.)*

Figure 2.10. Westerbork Synthesis Radio Telescope (14 dishes)

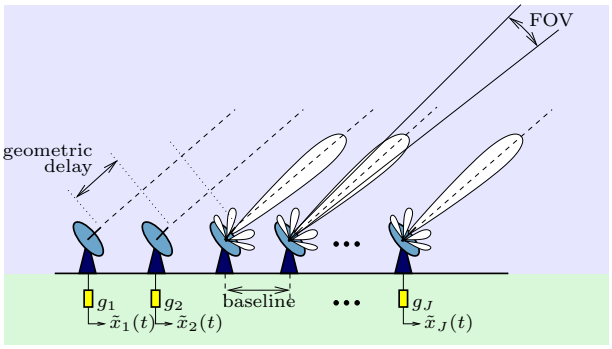


Figure 2.11. Schematic overview of a radio interferometer.

telescope that should be one to two orders of magnitude more sensitive than any radio telescope built to date [8]. This will require millions of elements to provide the desired collecting area of order one square kilometer.

The concept of interferometry is illustrated in Fig. 2.11. An interferometer measures the spatial coherency of the incoming electromagnetic field. This is done by correlating the signals from the individual receivers with each other. The correlation of each pair of receiver outputs provides the amplitude and phase of the spatial coherence function for the baseline defined by the vector pointing from the first to the second receiver in a pair. In radio astronomy, these correlations are called the *visibilities*.

Obviously, Fig. 2.11 is directly tied to Fig. 2.8. For a wideband plane wave source s propagating in the direction $\boldsymbol{\zeta}$, the power spectral density $R_s(\omega)$ of this source is called the *brightness*, and denoted by $I(\omega, \boldsymbol{\zeta})$. (Actually, $-\boldsymbol{\zeta}$ is used: this is the unit vector pointing towards the source.)

We also defined the observed cross power spectral density due to this source in (2.35) as $R_s(\mathbf{b}, \omega)$, and this is the *visibility* $V(\omega, \mathbf{b})$. Thus, for a single source,

$$V(\omega, \mathbf{b}) = I(\omega, \boldsymbol{\zeta}) e^{-j\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}}$$

For a superposition of sources, we can generalize this to

$$V(\omega, \mathbf{b}) = \int I(\omega, \boldsymbol{\zeta}) e^{-j\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}} \mathrm{d}\boldsymbol{\zeta} \tag{2.36}$$

This relation is called the Van Cittert-Zernike theorem [5, 9]. For each ω, $I(\omega, \boldsymbol{\zeta})$ is viewed as an image (called the *map*), by parametrizing $\boldsymbol{\zeta}$ in two coordinates or two angles, and the objective in radio astronomy is to obtain this map.

2.5.2 Dirty map

If we could measure $V(\omega, \mathbf{b})$ for all possible baselines $\mathbf{b}$, then we can reconstruct $I(\omega, \boldsymbol{\zeta})$ from (2.36) by an inverse spatial Fourier transform,

$$I(\omega, \boldsymbol{\zeta}) = \frac{1}{(2\pi)^3} \int V(\omega, \mathbf{b}) e^{j\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}} \mathrm{d}\mathbf{b} \tag{2.37}$$

which is computed for each ω separately. However, in practice, we can estimate $V(\omega, \mathbf{b})$ for only a discrete set of baselines $\{\mathbf{b}_k\}$: every telescope pair provides one baseline, and as the earth rotates, this baseline rotates and traces an arc in 3D space. We can obtain samples along this arc.

Thus, we cannot directly implement (2.37). Instead, we can compute an estimate

$$I_D(\omega, \boldsymbol{\zeta}) = \frac{1}{(2\pi)^3} \sum_k V(\omega, \mathbf{b}_k) e^{j\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}_k} \tag{2.38}$$

which is called the *dirty map*. It is not equal to the desired map $I(\omega, \boldsymbol{\zeta})$. Indeed, by substituting (2.36), we find

$$I_D(\omega, \boldsymbol{\zeta}) = \frac{1}{(2\pi)^3} \int I(\omega, \mathbf{n}) \sum_k e^{j\frac{\omega}{c}(\mathbf{n}-\boldsymbol{\zeta})\cdot\mathbf{b}_k} \mathrm{d}\mathbf{n} \tag{2.39}$$

This follows a similar derivation as we saw in Sec. 2.3.5, but now using baselines instead of direct location samples. We can write (2.39) as

$$I_D(\omega, \boldsymbol{\zeta}) = \frac{1}{(2\pi)^3} \int W(\mathbf{n} - \boldsymbol{\zeta}) I(\omega, \mathbf{n}) \mathrm{d}\mathbf{n} = \frac{1}{(2\pi)^3} W(\boldsymbol{\zeta}) * I(\omega, \boldsymbol{\zeta}) \tag{2.40}$$

where

$$W(\boldsymbol{\zeta}) = \sum_k e^{j\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}_k} \,. \tag{2.41}$$

Thus, the obtained dirty map is a convolution of the desired "true" map with a smoothing function $W(\boldsymbol{\zeta})$. This $W(\boldsymbol{\zeta})$ is called the dirty beam. Since, generally, the baseline sampling is quite irregular, the dirty beam also looks quite random.

An example of a set of antenna coordinates and the corresponding dirty beam is shown in Fig. 2.12. This is for a single low-band LOFAR station and a single 10 second integration interval and frequency bin. The dirty beam has heavy sidelobes, as high as -10 dB. To make this plot, the unit-norm direction vector $\boldsymbol{\zeta}$ is parametrized as $\boldsymbol{\zeta} = [\ell, m, n]^{\mathrm{T}}$, where $[\ell, m]$ are plotted and $n = \sqrt{1 - \ell^2 - m^2}$ is not shown.

A resulting dirty image is shown in Fig. 2.13. The image shows the complete sky, in (ℓ, m) coordinates, where the reference direction is pointing towards zenith. The strong visible sources are Cassiopeia A and Cygnus A, also visible is the milky way, ending in the north polar spur (NPS) and, weaker, Virgo A. In the South, the Sun is visible as well. The image was obtained by averaging 25 integration intervals, each consisting of 10 s data in 25 frequency channels of 156

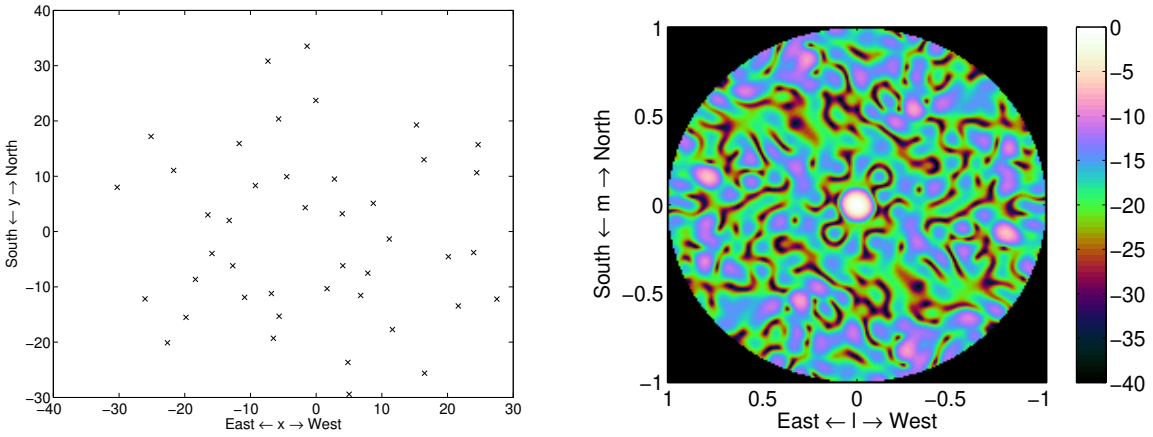


Figure 2.12. (a) Coordinates of the antennas in a LOFAR station, which defines the spatial sampling function, and (b) the resulting *dirty beam*, plotted in dB.

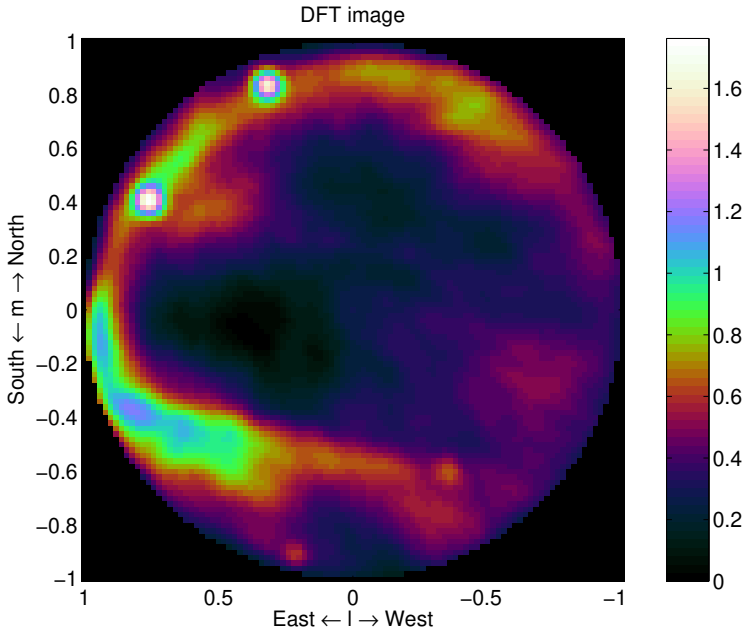


Figure 2.13. Dirty image following (2.40), using LOFAR station data.

kHz wide taken from the band 45–67 MHz, avoiding the locally present radio interference. As this shows data from a single LOFAR station, with a relatively small maximal baseline (65 m), the resolution is limited and certainly not representative of the capabilities of the full LOFAR array.

The dirty beam is essentially a non-ideal point spread function due to finite and non-uniform spatial sampling: we only have a limited set of baselines. The dirty beam has a main lobe centered at $\boldsymbol{\zeta} = \mathbf{0}$, and many side lobes. If we would have a large number of telescopes positioned in a uniform rectangular grid, the dirty beam would be a 2D sinc-function. The resulting beam size is inversely proportional to the aperture (diameter) of the array. This determines the *resolution* in the dirty image. The sidelobes of the beam give rise to confusion between sources: it is unclear whether a small peak in the image is caused by the main lobe of a weak source, or the sidelobe of a strong source. Therefore, attempts are made to design the array such that the sidelobes are low. As mentioned in Sec. 2.3.3, it is also possible to introduce weighting coefficients ("tapers") in (2.38) to obtain an acceptable beamshape.

As mentioned, an antenna array generates a set of baselines, and as the Earth rotates, these baselines also rotate and generate many of such sets. In the definition of $W(\boldsymbol{\zeta})$, the effect of summing over these sets is that the sidelobes tend to get averaged out, to some extent. Many images are also formed by averaging over a small number of frequency bins (assuming the source powers $R_s(\omega)$ are constant over these frequency bins), which enters into the equations in exactly the same way.

Since $W(\boldsymbol{\zeta})$ is data-independent, it can be nearly perfectly predicted after careful calibration of the instrument. Thus, we can try to estimate $I(\omega, \boldsymbol{\zeta})$ from $I_D(\omega, \boldsymbol{\zeta})$ using deconvolution techniques.

There are many issues that we ignored in the discussion, such as coordinate systems, approximation of the 3D integral in (2.40) by a 2D integral (on the assumption that either the field of interest is small, or that the antenna array sits on a flat plane), and how the $V(\omega, \mathbf{b}_k)$ can be estimated from received telescope signals. Also, there are directional disturbances due to non-isotropic antennas, unequal antenna gains, and disturbances due to atmospheric effects. Some of these questions are covered in future chapters.

2.6 NOTES

The discussion in Sec. 2.1 summarizes the presentation in [1]. Much more can be said about these topics, in particular for applications where the propagation speed c is position dependent (as for geophysics exploration or underwater acoustics). A range of applications and related wave models is found in [2].

A more extensive introduction to wavefield propagation and its role in image formation is offered in the course EE4595 Wavefield imaging.

The text on radio astronomy signal processing in Sec. 2.5 is based on Van der Veen e.a. [4]. This

paper gives a short introduction. A classical introduction from 1985 is in [2, Ch. 5]. Well-known reference textbooks are [5, 9].

Bibliography

[1] D.H. Johnson and D.E. Dudgeon, *Array signal processing: concepts and techniques*. Prentice Hall, 1993.

[2] S. Haykin, ed., *Array signal processing*. Prentice Hall, 1985.

[3] H.L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. Wiley, 2004.

[4] A.J. van der Veen, S.J. Wijnholds, and A.M. Sardarabadi, "Signal processing for radio astronomy," in *Handbook of Signal Processing Systems, 3rd ed.*, Springer, November 2018. ISBN 978-3-319-91734-4.

[5] A.R. Thompson, J.M. Moran, and G.W. Swenson, *Interferometry and Synthesis in Radio Astronomy*. New York: Wiley, 2nd ed., 2001.

[6] C. Lonsdale *et al.*, "The Murchison Widefield Array: Design overview," *Proceedings of the IEEE*, vol. 97, pp. 1497–1506, Aug. 2009.

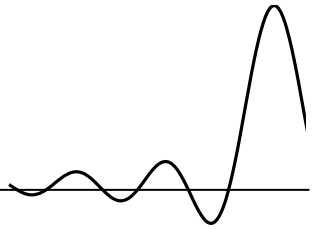
[7] M. de Vos, A.W. Gunst, and R. Nijboer, "The LOFAR telescope: System architecture and signal processing," *Proceedings of the IEEE*, vol. 97, pp. 1431–1437, Aug. 2009.

[8] P.E. Dewdney, P.J. Hall, R.T. Schilizzi, and T.J. Lazio, "The square kilometre array," *Proceedings of the IEEE*, vol. 97, pp. 1482–1496, Aug. 2009.

[9] R.A. Perley, F.R. Schwab, and A.H. Bridle, *Synthesis Imaging in Radio Astronomy*, vol. 6 of *Astronomical Society of the Pacific Conference Series*. BookCrafters Inc., 1994.

Chapter **3**

NARROWBAND DATA MODELS



Contents

In the previous chapter, we have seen the basics of wave propagation in relation to array processing. Our goal in this book is to give an overview of the basic signal processing algorithms that are used in this area, and that form the basis of more complicated algorithms in real applications. An important first step in the derivation of any algorithm should be a good description of the application scenario and a statement of the basic assumptions that can be made, such that a clear data model that captures the scenario can be stated. The model will then determine the type of algorithm that is appropriate. Different assumptions lead to different models and to different algorithms.

The model should be based on reality but not be overly detailed: if we want to estimate model parameters, their number should not be too large! The purpose of this chapter and the next is to present models for a number of prototype scenarios. Depending on the assumptions that are made, simple models with few parameters result, or more accurate models with more parameters. It ultimately depends on the requirements of the application which model is preferred.

In this chapter, we start the modeling by focusing on the reception of signals on an antenna array, under narrowband conditions. If the narrowband condition is satisfied, a delay can be translated to a phase shift: convolutions simplify to scalar products. This greatly simplifies the modeling (and subsequent processing), and allows us to develop spatial beamforming without caring much about time domain aspects.

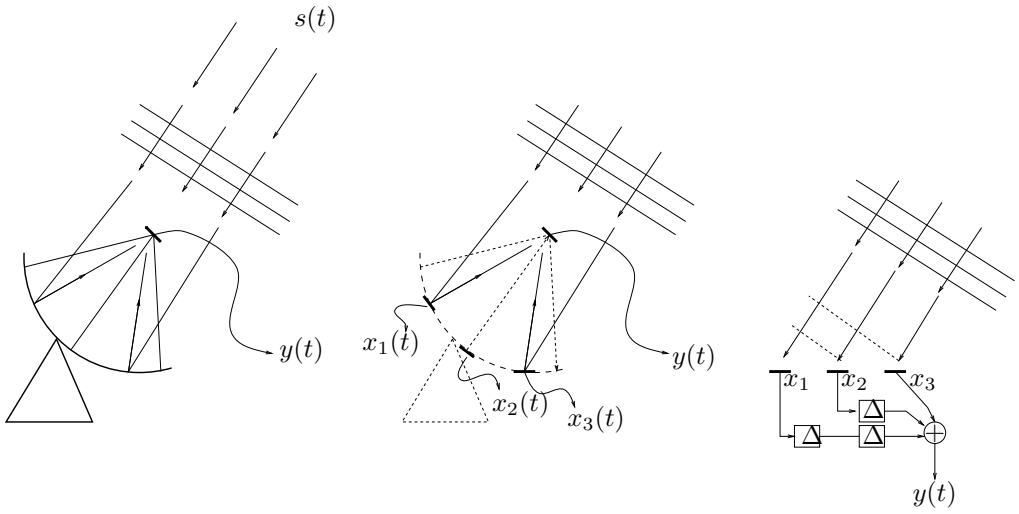


Figure 3.1. Coherent adding of signals. A parabolic dish physically ensures the correct delays for coherently adding signals that come from the same look direction. A phased array has to electronically insert the correct delays.

3.1 ANTENNA ARRAY RECEIVER MODEL

We start by considering the reception of a signal at an antenna array consisting of multiple antennas. We only consider linear receivers at this point.

3.1.1 Introduction

Beamforming An antenna array may be employed for several reasons. A traditional one is *signal enhancement*. If the same signal is received at multiple antennas and can be coherently added, then incoherent additive noise is averaged out. For example, suppose we have a signal $s(t)$ received at M antennas $x_0, \cdots, x_{M-1}$,

$$x_m(t) \;=\; s(t) + n_m(t)\,, \qquad m = 0, \cdots, M-1$$

where $s(t)$ is the desired signal and $n_m(t)$ is noise. Let us suppose that the noise variance is $\mathrm{E}[|n_m(t)|^2] = \sigma^2$. If the noise is uncorrelated from each antenna to the others, then by averaging we obtain

$$y(t) \;=\; \frac{1}{M}\sum_{m=0}^{M-1} x_m(t) \;=\; s(t) + \frac{1}{M}\sum_{m=0}^{M-1} n_m(t)\,.$$

The noise variance on $y(t)$ is given by $\mathrm{E}[|y|^2] = \frac{1}{M}\sigma^2$. We thus see that there is an *array gain* equal to a factor M, the number of antennas.

The reason that we could simply average, or add up the received signals $x_m(t)$, is that the desired signal entered coherently, with the same delay at each antenna. More in general, the

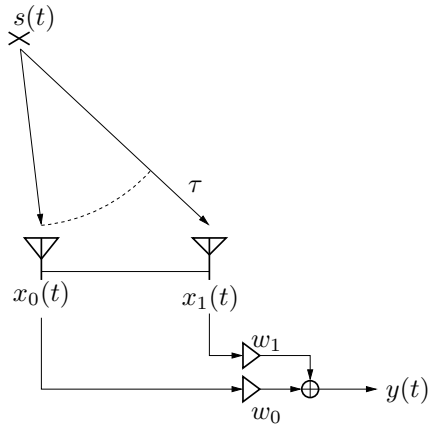


Figure 3.2. Nulling out a signal.

desired signal is received at unequal delays and we have to introduce compensating delays to be able to coherently add them. This requires knowledge on these delays, or the direction at which the signal was received. The operation of delay-and-sum is known as *beamforming*, since it can be regarded as forming a beam into the direction of the source. The delay-and-sum beamformer acts like an equivalent of a parabolic dish, which physically inserts the correct delays to look in the desired direction. See Fig. 3.1.

Spatial filtering A second reason to use an antenna array is to introduce a form of *spatial filtering*. Filtering can be done in the frequency domain —very familiar—, but similarly in the spatial domain. Spatial filtering can just be regarded as taking (often linear) combinations of the antenna outputs, and perhaps delays of them, to reach a desired spatial response.

A prime application of spatial filtering is *null steering*: the linear combinations are chosen such that a signal (interferer) is completely cancelled out. Suppose a signal $s(t)$ is received at the first antenna directly, but at the second antenna with a delay τ, see Fig. 3.2. It is easy to see how the received signals can be combined to produce a zero output, by inserting a proper delay and taking the difference. However, even without a delay we can do something. By weighting and adding the antenna outputs, we obtain a signal $y(t)$ at the output of the beamformer,

$$y(t) = w_0 s(t) + w_1 s(t - \tau)$$

In the frequency domain, this is

$$Y(\omega) = S(\omega)(w_0 + w_1 e^{-j\omega\tau})$$

Thus, we can make sure that the signal is cancelled, $Y(\omega) = 0$, at a certain frequency ω_0, if we select the weights such that

$$w_1 = -w_0 e^{j\omega_0\tau}$$

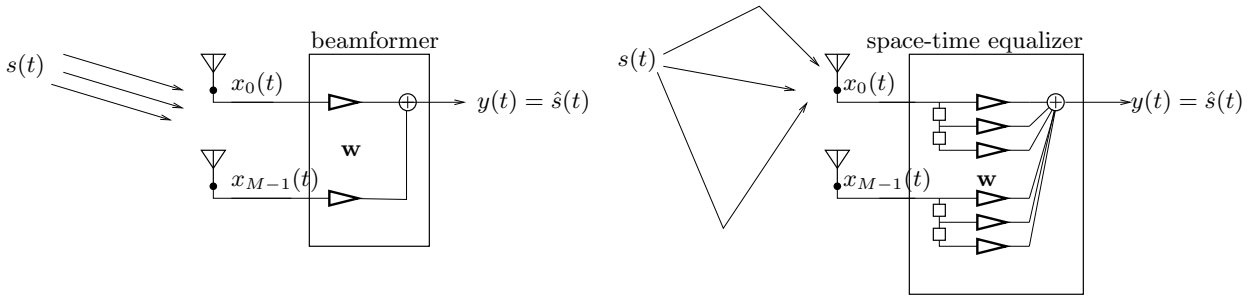


Figure 3.3. (a) Narrowband beamformer (spatial filter); (b) broadband beamformer (spatial/temporal filter).

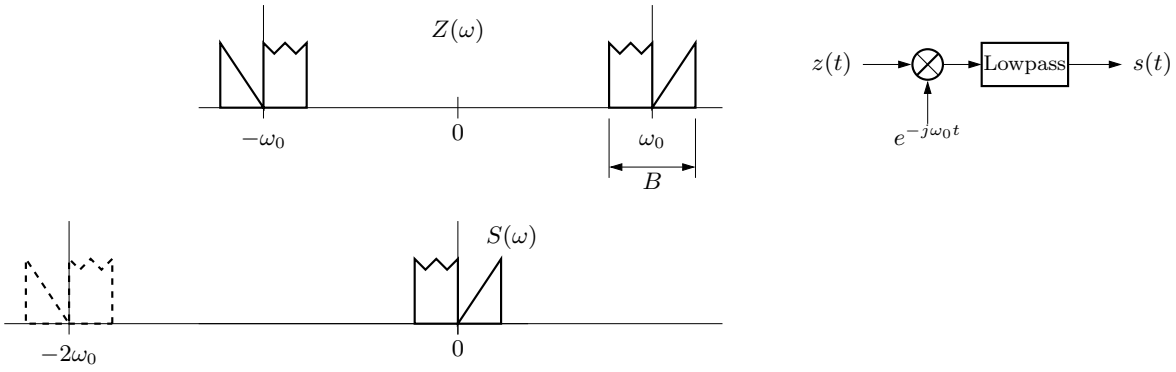


Figure 3.4. Transmitted real signal $z(t)$ and complex baseband signal $s(t)$.

Note that (i) if we do not delay the antenna outputs but only scale them before adding, then we need complex weights; (ii) with an implementation using weights, we can cancel the signal only at a specific frequency, but not at all frequencies.

Thus, for signals that consist of a single frequency, or a narrow band around a carrier frequency, we can do null steering by means of a *phased array* (i.e., summing after multiplications by complex weights). In more general situations, with broadband signals, we need a beamformer structure consisting of weights *and* delays. How narrow is narrow-band depends on the maximal delay across the antenna array, as is discussed next.

3.1.2 The narrowband assumption

Let us recall the following facts. In signal processing, signals are usually represented by their lowpass equivalents, see e.g., [1]. This is a suitable representation for narrowband signals in a digital communication system. A real valued bandpass signal with center frequency ω_0 may be written as

$$z(t) = \text{real}\{s(t)e^{j\omega_0 t}\} = x(t)\cos(\omega_0 t) - y(t)\sin(\omega_0 t) \tag{3.1}$$

where $s(t) = x(t) + jy(t)$ is the *complex envelope* of the signal $z(t)$, also called the *baseband signal*. The real and imaginary parts, $x(t)$ and $y(t)$, are called the in-phase and quadrature components of the signal $z(t)$. In practice, they are generated by multiplying the received signal with $\cos(\omega_0 t)$ and $\sin(\omega_0 t)$ followed by low-pass filtering. (An alternative is to apply a Hilbert transformation.)

Suppose that the bandpass signal $z(t)$ is delayed by a time τ. This can be written as

$$z_\tau(t) := z(t-\tau) = \text{real}\{s(t-\tau)e^{j\omega_0(t-\tau)}\} = \text{real}\{s(t-\tau)e^{-j\omega_0\tau}e^{j\omega_0 t}\}\,.$$

The complex envelope of the delayed signal is thus $s_\tau(t) = s(t-\tau)e^{-j\omega_0\tau}$. Let B be the bandwidth of the complex envelope (the baseband signal) and let $S(\omega)$ be its Fourier transform. We then have

$$s(t-\tau) = \frac{1}{2\pi}\int_{-B/2}^{B/2} S(\omega)e^{-j\omega\tau}e^{j\omega t}\mathrm{d}\omega\,.$$

If $|\omega\tau| \ll 2\pi$ for all frequencies $|\omega| \le \frac{B}{2}$ we can approximate $e^{-j\omega\tau} \approx 1$ for ω within the band, and get

$$s(t-\tau) \approx \frac{1}{2\pi}\int_{-B/2}^{B/2} S(\omega)e^{j\omega t}\mathrm{d}\omega = s(t)\,.$$

Thus, we have for the complex envelope $s_\tau(t)$ of the delayed bandpass signal $z_\tau(t)$ that

$$s_\tau(t) \approx s(t)e^{-j\omega_0\tau} \qquad \text{for } B\tau \ll 2\pi\,.$$

$B\tau \ll 2\pi$ is called the *narrowband condition*. The conclusion is that, for narrowband signals, time delays smaller than the inverse bandwidth may be represented as phase shifts of the complex envelope. This is fundamental in direction estimation using phased antenna arrays.

For propagation across an antenna array, the maximal delay depends on the maximal distance across the antenna array: the aperture. Let us work with frequencies $f = \omega/(2\pi)$ in Hz, and corresponding bandwidths $W = B/(2\pi)$ Hz. If the wavelength is $\lambda = c/f_0$ and the aperture is Δ wavelengths, then the maximal delay is $\tau = \Delta\lambda/c = \Delta/f_0$. In this context, narrowband means

$$B\tau \ll 2\pi \quad \Leftrightarrow \quad W\frac{\Delta}{f_0} \ll 1 \quad \Leftrightarrow \quad W \ll \frac{f_0}{\Delta}\,. \tag{3.2}$$

For mobile communications, the wavelength around $f_0 = 1$ GHz is about 30 cm. For practical purposes, Δ is small, say $\Delta < 5$ wavelengths, and then narrowband means $W \ll 30$ MHz. This condition is satisfied for most communication systems around 1 GHz. Bluetooth operates with channels that have a bandwidth of 1 MHz at 2.4 GHz, and the narrowband assumption is satisfied. Ultrawideband (UWB) systems in the IEEE 802.15.4 standard operate in 500 MHz bands at 3.1 GHz to 10.6 GHz, and the narrowband assumption does not hold. In low-frequency radio astronomy, we could have a center frequency at 100 MHz (wavelength 3 m), and a telescope array with a diameter of 100 km (33,000 wavelengths), so that the maximal bandwidth is in the order of 3 kHz. This is implemented by splitting the received signals into narrow subbands.

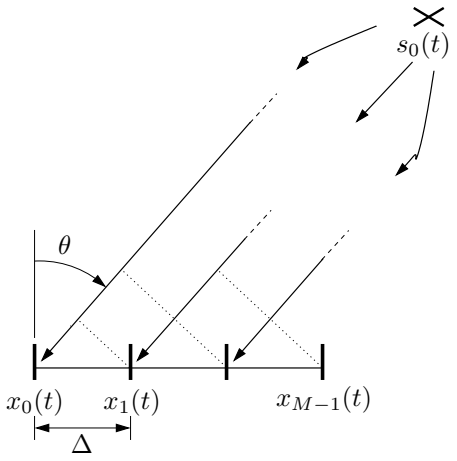


Figure 3.5. A uniform linear array receiving a far field point source.

The above considerations would be for a single plane wave traveling across the array. For outdoor propagation, the situation may be different. If there is a reflection on a distant object, there may be path length differences of a few km, or $\Delta = \mathcal{O}(1000)$ wavelengths at 1 GHz. In this context, narrow band means $W \ll 1$ MHz, and for many communication signals (e.g., GSM, UMTS), this is not really satisfied. In this case, delays of the signal cannot be represented by mere phase shifts, and we need to do broadband beamforming, i.e., space-time processing.

Usually we sample a signal at Nyquist, i.e., $f_s = W$ (assuming complex baseband samples), or $T_s = 1/W$. The narrowband condition translates to $\tau_{\max} \ll T_s$: the maximal delay for propagation across the array is less than the sampling period. This will give rise to an *instantaneous* data model, discussed later in Sec. 4.3.1.

3.1.3 Antenna array response

Consider an array consisting of M identical antenna elements placed along a line in space, and assume a point source is present in the far field. See Fig. 3.5. Let $s_0(t)$ be the transmitted baseband signal, which is subsequently modulated at ω_0. If the distance between the array and the source is large enough in comparison to the extent of the array (i.e., the source is in the far field), the wave incident on the array is approximately planar. The angle θ to the normal is called the direction of arrival (DOA) of the plane wave. Let $a_m(t, \theta)$ be the response of the ith antenna element. The signal received at the ith antenna (after demodulation by ω_0) is

$$x_m(t) = a_m(t, \theta) * s_0(t - T_m)e^{-j\omega_0 T_m} \tag{3.3}$$

where $*$ denotes convolution and T_i is the time it takes the signal to travel from the source to the mth antenna.

A uniform array has identical elements, i.e., all antennas have the same response $a(t, \theta)$. It is

reasonable to assume separability into $a(t,\theta) = a_0(\theta)g(t)$, where $a_0(\theta)$ is the antenna gain pattern in the direction θ, and $g(t)$ is its temporal response. If the antennas are onmidirectional and the frequency response is flat over the band of interest, as is often assumed, we have $a_0(\theta) = a_0$ and $g(t) = \delta(t)$.

Define by

$$s(t) = g(t) * s_0(t - T_0)e^{-j\omega_0 T_0}$$

the signal received by the first antenna element, save for the array gain, and let $\tau_m = T_m - T_0$ be the time difference of arrivals (the geometric delays). If the τ_m are small compared to the inverse bandwidth of $s(t)$, we may set $s_m(t) = s(t)e^{-j\omega_0\tau_m}$, which is the signal received at time t at the mth element of the array.

Collecting the signals received by the individual elements into a vector $\mathbf{x}(t)$, we obtain from (3.3)

$$\mathbf{x}(t) = a_0(\theta)\begin{bmatrix} s(t) \\ s_{\tau_1}(t) \\ \vdots \\ s_{\tau_{M-1}}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j\omega_0\tau_1} \\ \vdots \\ e^{-j\omega_0\tau_{M-1}} \end{bmatrix} a_0(\theta)s(t)$$

For a uniform linear array, we have the same distance d between the antenna elements, so that all delays between two consecutive array elements are the same: $\tau_m = m\tau$. We can also relate the time difference (or phase shift) to the angle of arrival θ:

$$\omega_0\tau = -\omega_0\frac{d\sin(\theta)}{c} = -\frac{2\pi}{\lambda}d\sin(\theta) = -2\pi\Delta\sin(\theta)$$

where $\Delta = d/\lambda$ is the spacing between antenna elements measured in wavelengths (corresponding to the center frequency ω_0) so that

$$\mathbf{x}(t) = \begin{bmatrix} 1 \\ e^{j2\pi\Delta\sin(\theta)} \\ \vdots \\ e^{j2\pi(M-1)\Delta\sin(\theta)} \end{bmatrix} a_0(\theta)s(t) =: \mathbf{a}(\theta)s(t)\,, \tag{3.4}$$

where the *array response vector* $\mathbf{a}(\theta)$ is the response of the array to a plane wave with DOA θ. The *array manifold* $\mathcal{A}$ is the curve that $\mathbf{a}(\theta)$ describes in the M-dimensional complex vector space $\mathbb{C}^M$ when θ is varied over the domain of interest:

$$\mathcal{A} = \{\mathbf{a}(\theta) : 0 \le \theta < 2\pi\}\,.$$

In (3.4), the array response vector $\mathbf{a}(\theta)$ has a very regular form, due to the uniform linear array structure. More in general, assume that in a 2D scenario we have an irregular array with

elements at positions $\mathbf{x}_m$. Further assume $\mathbf{x}_0 = \mathbf{0}$: this sets the phase reference at element 0. Then, following Chap. 2, we find

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ e^{j\phi_1} \\ \vdots \\ e^{j\phi_{M-1}} \end{bmatrix} a_0(\theta) \tag{3.5}$$

where the phase factors are

$$\phi_m = -\omega_0 \tau_m = -\frac{\omega_0}{c} \boldsymbol{\zeta} \cdot \mathbf{x}_m = -\frac{2\pi}{\lambda} \boldsymbol{\zeta} \cdot \mathbf{x}_m \,.$$

Here, $\boldsymbol{\zeta}$ denotes the direction vector of the incoming wave. In 2D, we have, viz. (2.4),

$$\boldsymbol{\zeta} = -\begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix}$$

so that the phase factors are

$$\phi_m = 2\pi \frac{x_m}{\lambda} \sin(\theta) + 2\pi \frac{y_m}{\lambda} \cos(\theta) \,.$$

This generalizes (3.4). A similar expression can be derived for propagation in 3D, leading to an array response vector $\mathbf{a}(\theta, \phi)$ parametrized by two angle parameters.

In many algorithms, the common factor $a_0(\theta)$ does not play a role and is often omitted even in the data model: the array is assumed to have equal response in all directions (it is "omnidirectional") although this assumption is typically not valid and not always necessary. Otherwise, this factor is a direction-dependent gain. The time response $g(t)$ is usually omitted as well, or lumped in the receiver filter description.

3.1.4 Array manifold and parametric direction finding

Data models of the form

$$\mathbf{x}(t) = \mathbf{a}(\theta) s(t)$$

play an important role throughout this book. Note that for varying source samples $s(t)$, the data vector $\mathbf{x}(t)$ is only scaled in length, but its direction $\mathbf{a}(\theta)$ is constant. Thus, $\mathbf{x}(t)$ is confined to a line in M-dimensional space. If we know the array manifold, i.e., the function $\mathbf{a}(\theta)$, then we can determine θ by intersecting the line with the curve traced by $\mathbf{a}(\theta)$ for varying θ, or "fitting" the best $\mathbf{a}(\theta)$ to the direction of the $\mathbf{x}(t)$.

For two sources, the data model becomes a superposition,

$$\mathbf{x}(t) = \mathbf{a}(\theta_1) s_1(t) \; + \; \mathbf{a}(\theta_2) s_2(t) = \begin{bmatrix} \mathbf{a}(\theta_1) & \mathbf{a}(\theta_2) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}$$

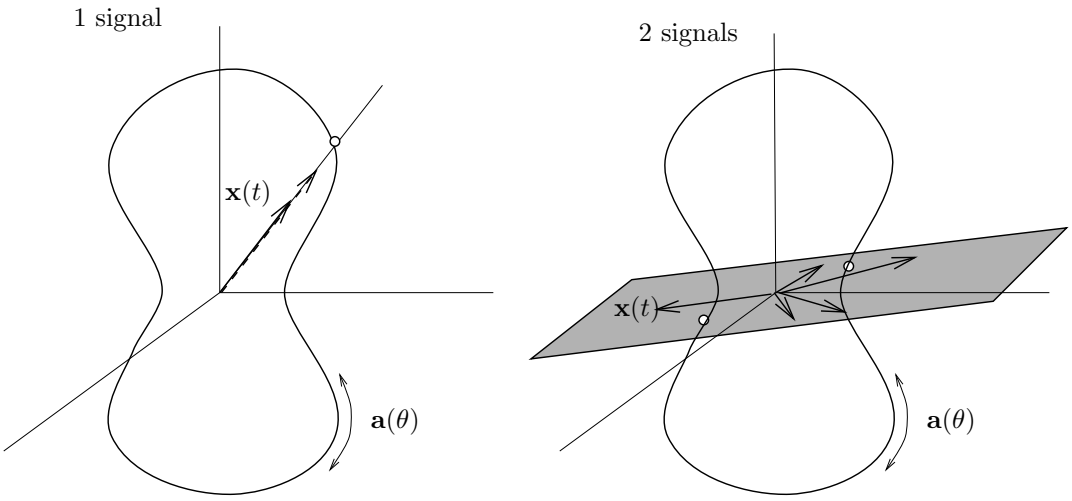


Figure 3.6. Direction finding means intersecting the array manifold with the line or plane spanned by the antenna output vectors.

or

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)\,, \qquad \mathbf{A} = \mathbf{A}(\theta_1, \theta_2) = [\mathbf{a}(\theta_1) \quad \mathbf{a}(\theta_2)]\,, \qquad \mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}\,.$$

When $s_1(t)$ and $s_2(t)$ both vary with t, $\mathbf{x}(t)$ is confined to a plane. Direction finding now amounts to intersecting this plane with the array manifold, see Fig. 3.6.

With multipath, we obtain a linear combination of the same source via two different paths. If the relative delay between the two paths is small compared to the inverse bandwidth, it can be represented by a phase shift. Thus, the data model is

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{a}(\theta_1)s(t) + \mathbf{a}(\theta_2)\beta s(t) \\ &= \{\mathbf{a}(\theta_1) + \beta\mathbf{a}(\theta_2)\}\, s(t) \quad = \mathbf{a}\, s(t)\,. \end{aligned}$$

In this case, the combined vector $\mathbf{a}$ is not on the array manifold and direction finding is more complicated. At any rate, $\mathbf{x}(t)$ contains an *instantaneous multiple* $\mathbf{a}$ of $s(t)$. In many applications β is fluctuating relatively quickly, so that $\mathbf{a}$ is time varying on short time scales (the *coherence time*).

3.1.5 Beamforming and source separation

With two narrowband sources and multipath, we receive a linear mixture of these sources,

$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \mathbf{a}_2 s_2(t) = \mathbf{A}\mathbf{s}(t)\,.$$

The objective of *source separation* is to estimate beamformers $\mathbf{w}_1$ and $\mathbf{w}_2$ to separate and recover the individual sources:

$$y_1(t) = \mathbf{w}_1^{\mathrm{H}}\mathbf{x}(t) = \hat{s}_1(t)\,, \qquad y_2(t) = \mathbf{w}_2^{\mathrm{H}}\mathbf{x}(t) = \hat{s}_2(t)\,,$$

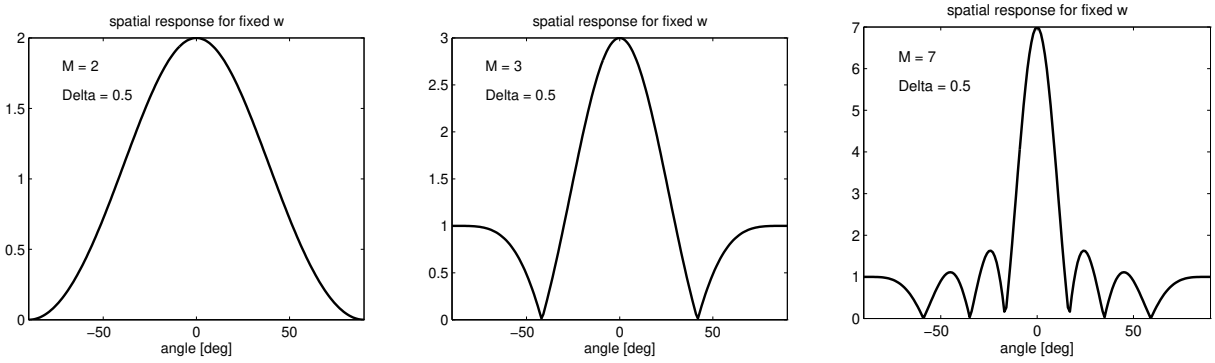


Figure 3.7. Spatial responses to a beamformer $\mathbf{w} = [1, \cdots, 1]^{\mathrm{T}}$ as a function of the incoming source direction θ.

or in matrix form, with $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2]$,

$$\mathbf{W}^{\mathrm{H}}\mathbf{x}(t) = \mathbf{s}(t) \qquad \Leftrightarrow \qquad \mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I} \qquad \Leftrightarrow \qquad \mathbf{W} = \mathbf{A}(\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1}\,.$$

Thus, we have to obtain an estimate of the mixing matrix $\mathbf{A}$ and find a left inverse to separate the sources. We assumed that $\mathbf{A}$ is such that $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is invertible; this requires $\mathbf{A}$ to be tall: at least as many antennas as sources.

There are several ways to estimate $\mathbf{A}$. One we have seen before: if there is no multipath, then $\mathbf{A} = [\mathbf{a}(\theta_1) \;\; \mathbf{a}(\theta_2)]$. By estimating the directions of the sources, we find estimates of θ_1 and θ_2, and hence $\mathbf{A}$ becomes known and can be inverted.

In other situations, in wireless communications, we may know the values of $s_1(t)$ and $s_2(t)$ for a short time interval $t = [0, T]$: the data contains a "training period". We thus have a data model

$$\mathbf{X} = \mathbf{A}\mathbf{S}\,, \qquad \mathbf{X} = [\mathbf{x}(0)\,, \cdots, \mathbf{x}(T)]\,, \qquad \mathbf{S} = [\mathbf{s}(0)\,, \cdots, \mathbf{s}(T)]\,.$$

This allows to set up a least squares problem

$$\min_{\mathbf{A}} \; \| \mathbf{X} - \mathbf{A}\mathbf{S} \|_{\mathrm{F}}^2$$

with $\mathbf{X}$ and $\mathbf{S}$ known. The solution is given by

$$\mathbf{A} = \mathbf{X}\mathbf{S}^{\mathrm{H}}(\mathbf{S}\mathbf{S}^{\mathrm{H}})^{-1}$$

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

3.1.6 Spatial response graphs

Let us know consider in some more detail the properties of the array response vector $\mathbf{a}(\theta)$. For simplicity, we will look at uniform linear arrays. Suppose we have an antenna spacing of $\lambda/2$,

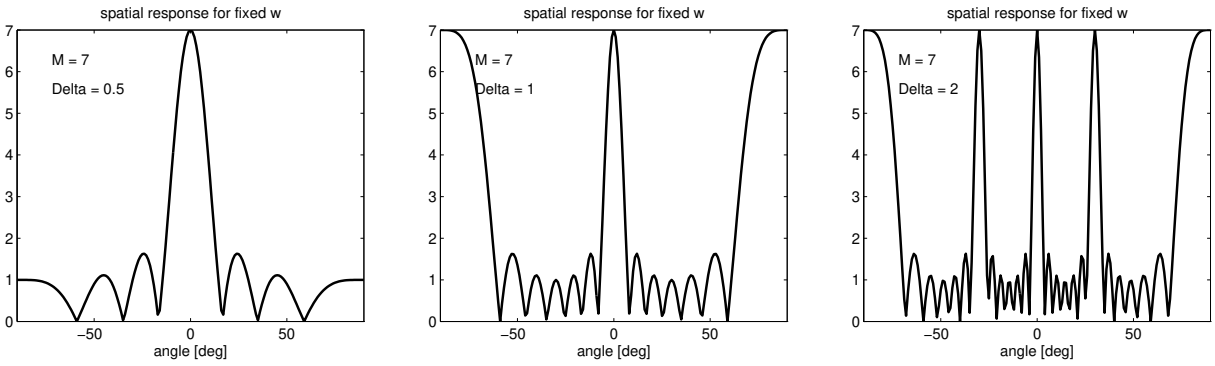


Figure 3.8. Grating lobes.

and select a simple beamformer of the form

$$\mathbf{w} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

i.e., we simply sum the outputs of the antennas. The response of the array to a unit-amplitude signal from direction θ is characterized by

$$|y(t)| = |\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)|\,.$$

Graphs of this response for $M = 2,\ 3,\ 7$ antennas are shown in Fig. 3.7, as a function of θ. Note that the response is maximal for a signal from the direction $0°$, or broadside from the array. This is natural since a signal from this direction is summed coherently, as we have seen in the begining of the section. The gain in this direction is equal to M, the array gain. From all other directions, the signal is not summed coherently. For some directions, the response is even zero, where the delayed signals add destructively. We saw in Chap. 2 that the number of zeros is equal to $M - 1$. In between the zeros, sidelobes occur. The width of the main beam is also related to the number of antennas, and in Chapter 2 we estimated it at about $180°/M$. With more antennas, the beamwidth gets smaller.

Ambiguity and grating lobes Let us now consider what happens if the antenna spacing increases beyond $d = \lambda/2$. As before, let $\Delta = d/\lambda$. We have an array response vector

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ e^{j\phi} \\ \vdots \\ e^{j(M-1)\phi} \end{bmatrix}, \qquad \phi = 2\pi\Delta\sin(\theta)\,.$$

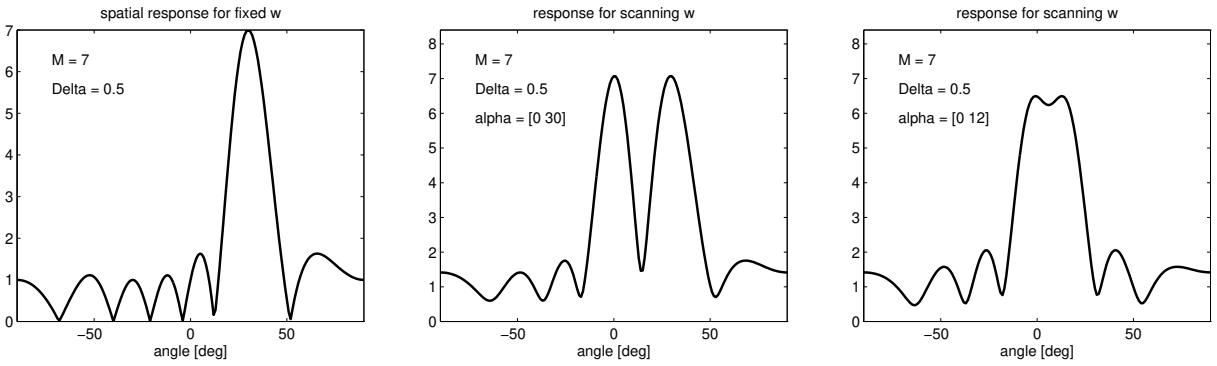


Figure 3.9. Beam steering. (a) response to $\mathbf{w} = \mathbf{a}(30^\circ)$; ($b$) response for scanning $\mathbf{w} = \mathbf{a}(\theta)$, in a scenario with two sources, well separated, and (c) separated less than a beam width.

Since $\sin(\theta) \in [-1, 1]$, we have that $2\pi\Delta \sin(\theta) \in [-2\pi\Delta, 2\pi\Delta]$. If $\Delta > 0.5$, then this interval extends beyond $[-\pi, \pi]$. In that case, there are several values of θ that give rise to the same argument of the exponent, or to the same ϕ. The effect is two-fold:

- *Spatial aliasing* occurs: we cannot recover θ from knowledge of ϕ, and
- In the array reponse graph, *grating lobes* occur, see Fig. 3.8. This is because coherent addition is now possible for several values of θ.

Grating lobes prevent a unique estimation of θ. However, we can still estimate $\mathbf{A}$ and it does not prevent the possibility of null steering or source separation. Sometimes, grating lobes can be suppressed by using directional antennas rather than omnidirectional ones (e.g., parabolic dishes): the spatial response is then multiplied with the directional response of the antenna $a_0(\theta)$ as in (3.4), and if it is sufficiently narrow, only a single lobe is left.

Beam steering Finally, let us consider what happens when we change the beamforming vector $\mathbf{w}$. Although we are free to choose anything, let us choose a structured vector, e.g., $\mathbf{w} = \mathbf{a}(30^\circ)$. Fig. 3.9 shows the response to this beamformer. Note that now the main peak shifts to 30°, signals from this direction are coherently added. By scanning $\mathbf{w} = \mathbf{a}(\theta)$, we can place the peak at any desired θ. This is called *classical beamforming*.

This also provides a simple way to do direction estimation. Suppose we have a single unit-norm source, arriving from broadside (0°),

$$\mathbf{x}(t) = \mathbf{a}(0)s(t) = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} s(t) = \mathbf{1}s(t)\,.$$

If we compute $y(t) = \mathbf{w}^{\mathrm{H}}\mathbf{x}(t)$ and scan $\mathbf{w}(\theta) = \mathbf{a}(\theta)$ over all values of θ and monitor the output power of the beamformer,

$$P_y(\theta) = \mathrm{E}[\,|y(t)|^2] = \mathrm{E}[\,|\mathbf{w}(\theta)^{\mathrm{H}}\mathbf{x}(t)|^2] = |\mathbf{a}(\theta)^{\mathrm{H}}\mathbf{1}|^2\, P_s\,, \qquad -\pi \le \theta \le \pi \tag{3.6}$$

where P_s is the source power, then (except for the square) we obtain essentially the same array graph as in Fig. 3.7 before (it is the same functional). Thus, there will be a main peak at 0°, the direction of arrival, and the beam width is related to the number of antennas. In general, if the source is coming from direction θ_0, then the graph will have a peak at θ_0.

With two sources, $\mathbf{x}(t) = \mathbf{a}(\theta_1)s_1(t) + \mathbf{a}(\theta_2)s_2(t)$, the array graph will show two peaks, at θ_1 and θ_2, at least if the two sources are well separated. If the sources are close, then the two peaks will shift, then merge and at some point we will not recognize that there are in fact two sources.

The choice $\mathbf{w}(\theta) = \mathbf{a}(\theta)$ is one of the simplest forms of beamforming.[1] It is data independent, and optimal only for a single source in white noise. One can show that for more than 1 source, the parameter estimates for the directions θ_i will be biased: the peaks have a tendency to move a little bit to each other. Unbiased estimates are obtained only for a single source.

There are other ways of beamforming, in which the beamformer is selected depending on the data, with higher resolution (sharper peaks) and better statistical properties in the presence of noise. Alternatively, we may follow a parametric approach in which we pose the model $\mathbf{x}(t) = \mathbf{a}(\theta_1)s_1(t) + \mathbf{a}(\theta_2)s_2(t)$ and try to compute the parameters θ_1 and θ_2 that best fit the observed data, as we discussed in Section 3.1.4.

For more general arrays, the array response vector $\mathbf{a}(\theta)$ is given by (3.5), and we would still pick $\mathbf{w}(\theta) = \mathbf{a}(\theta)$ to steer towards θ, i.e., set

$$\mathbf{w}(\theta) = \begin{bmatrix} 1 \\ e^{j\phi_1} \\ \vdots \\ e^{j\phi_{M-1}} \end{bmatrix}. \tag{3.7}$$

Note that when we compute $\mathbf{w}(\theta)^{\mathrm{H}}\mathbf{x}(t)$, we apply complex conjugates to the entries of $\mathbf{w}$, so that ϕ_m becomes $-\phi_m$. It is recognized that the resulting phases $-\phi_m$ are precisely those that are needed to compensate the phase of the incoming signal at sensor m, so that we sum coherently for signals coming from direction θ.

Beam shaping In (3.7), the amplitudes of each entry of $\mathbf{w}$ were all equal to 1. As a result, all beams in Fig. 3.9 looked like Dirichlet functions. In particular, they all have quite high sidelobes.

[1] It is a spatial matched filter, and known as Maximum Ratio Combining in communications.

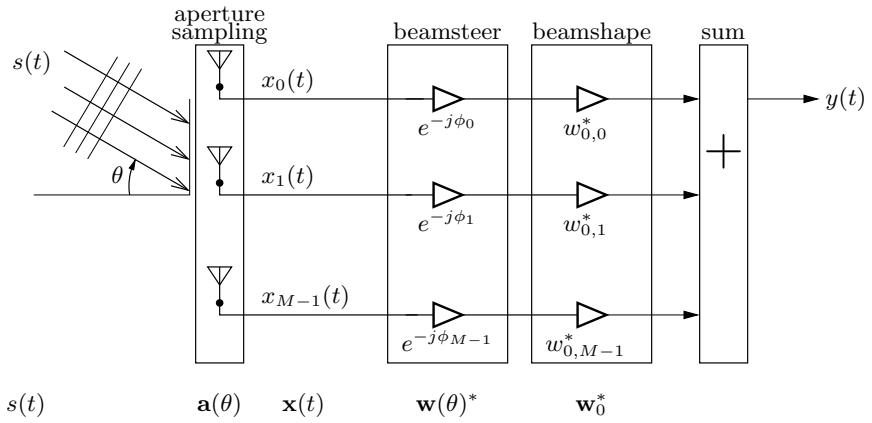


Figure 3.10. General phased array beamformer (narrowband signals).

We can address this by tapering, i.e., scale each entry $e^{j\phi_m}$ of $\mathbf{w}(\theta)$ by a weight $w_{0,m}$. The resulting beamformer can be written as

$$\mathbf{w} = \mathbf{w}_0 \odot \mathbf{w}(\theta) = \begin{bmatrix} w_{0,0} \\ w_{0,1}e^{j\phi_1} \\ \vdots \\ w_{0,M-1}e^{j\phi_{M-1}} \end{bmatrix} = \begin{bmatrix} w_{0,0} & & & \\ & w_{0,1} & & \\ & & \ddots & \\ & & & w_{0,M-1} \end{bmatrix} \mathbf{a}(\theta)\,. \tag{3.8}$$

The latter way of writing (as a matrix multiplying $\mathbf{a}(\theta)$) will be recognized in a later chapter when we consider more general beamformers.

The design of $\mathbf{w}_0$ to arrive at a desired beam shape follows the same discussion as in Sec. 2.3.3, where we discussed weighted spatial Fourier transforms.

Relation to spatial spectra In Chapter 2, we discussed the analysis of propagating waves using the spatial Fourier transform. In the present section, we derived beamforming quite independently, as a way to coherently sum signals coming from a direction θ. The connection between the two concepts is as follows.

Consider Fig. 3.10, and compare to Fig. 2.4. In the notation of this chapter, we have

$$\mathbf{x}(t) = \mathbf{a}(\theta)s(t)\,, \qquad a_m = e^{j\phi_m}, \qquad \phi_m = -\mathbf{k}\cdot\mathbf{x}_m\,,$$

where $s(t)$ is a baseband signal. In the notation of Fig. 2.4, we have

$$x_m(t) = s(\mathbf{x},t) = s(t)e^{j\phi_m}\,, \qquad \phi_m = -\mathbf{k}\cdot\mathbf{x}_m$$

where $s(t)$ is a narrowband passband signal with center frequency ω_0. However, the modulation $e^{j\omega_0 t}$ can be dropped from both $s(t)$ and $x_m(t)$. This makes the expressions for $x_m(t)$ equivalent.

Next, we considered beamforming,

$$y(t) = \mathbf{w}^{\mathrm{H}}\mathbf{x}(t)\,, \qquad \mathbf{w} = \mathbf{w}_0 \odot \mathbf{w}(\theta)$$

with $\mathbf{w}$ given in (3.8), so that the beamforming entries are $w_m = w_{0,m}e^{j\phi_m}$. In Chap. 2, we defined the discrete-space Fourier transform of the $x_m(t)$ including aperture selection and weighting as

$$Y(\mathbf{k}, t) = \sum_{m=0}^{M-1} w_m x_m(t) e^{j\mathbf{k}\cdot\mathbf{x}_m} \tag{3.9}$$

This is, in fact, equal to the beamformer output $y(t)$, if w_m in (3.9) is replaced by $w_{0,m}$ (note that the complex conjugate in $\mathbf{w}^{\mathrm{H}}$ takes care of the minus sign in front of ϕ_m). Thus, the beamformer in Fig. 3.10 is recognized as computing the (weighted) spatial Fourier transform of the selected samples of $s(\mathbf{x}, t)$.

This is entirely equivalent to the interpretation of the periodogram in time-domain spectrum estimation as the output of a subband filter in a filter bank, viz. [2, Ch. 8.2.1].

3.2 NARROWBAND CORRELATION MODELS

3.2.1 Data models

Let us revisit the narrowband data model, after sampling:

$$\mathbf{x}_n = \mathbf{A}\mathbf{s}_n\,, \qquad n = 0, \cdots, N-1\,.$$

Instead of a deterministic model for the sources, we can consider a stochastic model. We will restrict ourselves to zero-mean, wide sense stationary sources, and let

$$\mathrm{E}[\mathbf{s}_n] = \mathbf{0}\,, \qquad \mathbf{R}_{\mathbf{s}} = \mathrm{E}[\mathbf{s}_n\mathbf{s}_n^{\mathrm{H}}]\,.$$

Then the data satisfies

$$\mathrm{E}[\mathbf{x}_n] = \mathbf{0}\,, \qquad \mathbf{R}_{\mathbf{x}} = \mathrm{E}[\mathbf{x}_n\mathbf{x}_n^{\mathrm{H}}] = \mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^{\mathrm{H}}\,.$$

We will sometimes find it useful to vectorize this model, and work with $\mathbf{r}_{\mathbf{x}} = \mathrm{vec}(\mathbf{R}_x)$. Using properties of Kronecker products (see Sec. 5.1.6), we obtain

$$\mathbf{r}_{\mathbf{x}} = (\bar{\mathbf{A}} \otimes \mathbf{A})\mathbf{r}_{\mathbf{s}}\,. \tag{3.10}$$

Generally, $\mathbf{R}_{\mathbf{s}}$ is a full $d \times d$ matrix. Two important special cases are:

- *Independent sources:* the source covariance is diagonal,

$$\mathbf{R}_{\mathbf{s}} = \mathbf{\Sigma}_{\mathbf{s}} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_d^2 \end{bmatrix},$$

where the source variances (powers) σ_i^2 are possibly unequal. If $\boldsymbol{\sigma}_\mathbf{s} = \text{vecdiag}(\boldsymbol{\Sigma}_\mathbf{s})$ is a vector containing the source powers, then (3.10) becomes

$$\mathbf{r}_\mathbf{x} = (\bar{\mathbf{A}} \circ \mathbf{A})\boldsymbol{\sigma}_\mathbf{s} \,.$$

- *Independent sources with equal variances:*

$$\mathbf{R}_\mathbf{s} = \sigma_\mathbf{s}^2 \mathbf{I} \,.$$

This leads to

$$\mathbf{R}_\mathbf{x} = \sigma_\mathbf{s}^2 \mathbf{A}\mathbf{A}^{\text{H}} \,, \qquad \mathbf{r}_\mathbf{x} = \sigma_\mathbf{s}^2 (\bar{\mathbf{A}} \circ \mathbf{A})\mathbf{1}$$

where $\mathbf{1}$ is a vector with all entries equal to 1.

Next, we augment the data model with additive noise:

$$\mathbf{x}_n = \mathbf{A}_n \mathbf{s}_n + \mathbf{n}_n \,.$$

Similar to the sources, the noise is considered to be zero mean and wide-sense stationary,

$$\text{E}[\mathbf{n}_n] = \mathbf{0} \,, \qquad \mathbf{R}_\mathbf{n} = \text{E}[\mathbf{n}_n \mathbf{n}_n^{\text{H}}] \,.$$

The noise is independent from the signals, so that

$$\mathbf{R}_\mathbf{x} = \mathbf{A}\mathbf{R}_\mathbf{s}\mathbf{A}^{\text{H}} + \mathbf{R}_\mathbf{n} \,.$$

After vectoring, this becomes

$$\mathbf{r}_\mathbf{x} = (\bar{\mathbf{A}} \otimes \mathbf{A})\mathbf{r}_\mathbf{s} + \mathbf{r}_\mathbf{n} \,.$$

Usually the noise on the various sensors is considered to be independent, so that $\mathbf{R}_\mathbf{n}$ is modeled to be diagonal: $\mathbf{R}_\mathbf{n} = \boldsymbol{\Sigma}_\mathbf{n}$. Moreover, we often model the noise powers on the various antennas to be equal, i.e., spatially white noise, so that

$$\mathbf{R}_n = \sigma_\mathbf{n}^2 \mathbf{I} \,.$$

With diagonal source and noise covariance models, the vectored data model can be written as

$$\mathbf{r}_\mathbf{x} = (\bar{\mathbf{A}} \circ \mathbf{A})\boldsymbol{\sigma}_\mathbf{s} + (\mathbf{I} \circ \mathbf{I})\boldsymbol{\sigma}_\mathbf{n} \,.$$

3.2.2 Sample correlation matrices

Generally, we have a finite number of samples N, and do not have access to the true data covariance matrix. Instead we form the estimate

$$\hat{\mathbf{R}}_\mathbf{x} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_n \mathbf{x}_n^{\text{H}} \,.$$

which is called the sample correlation matrix. Using a data matrix $\mathbf{X} = [\mathbf{x}_0, \cdots, \mathbf{x}_{N-1}]$, we can also write it as

$$\hat{\mathbf{R}}_{\mathbf{x}} = \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{H}} .$$

If we define $\hat{\mathbf{R}}_{\mathbf{s}}$ in a similar way then, in the noiseless case, $\hat{\mathbf{R}}_{\mathbf{x}} = \mathbf{A}\hat{\mathbf{R}}_{\mathbf{s}}\mathbf{A}^{\mathrm{H}}$. With additive noise,

$$\hat{\mathbf{R}}_{\mathbf{x}} = \mathbf{A}\hat{\mathbf{R}}_{\mathbf{s}}\mathbf{A}^{\mathrm{H}} + \hat{\mathbf{R}}_{\mathbf{n}} + (\text{cross terms}) .$$

Since the sources and the noise are zero mean, the cross terms are zero mean, and the covariance estimate is unbiased:

$$\mathrm{E}[\hat{\mathbf{R}}_{\mathbf{x}}] = \mathbf{R}_{\mathbf{x}} = \mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^{\mathrm{H}} + \mathbf{R}_{\mathbf{n}} .$$

Thus, we can consider $\hat{\mathbf{R}}_{\mathbf{x}}$ to be equal to $\mathbf{R}_{\mathbf{x}}$ plus a zero mean error term $\mathbf{E}$ due to the finite number of samples. For large N, we expect $\mathbf{E} \to 0$. What is the (co)variance of $\hat{\mathbf{R}}_{\mathbf{x}}$? Or equivalently, of $\mathbf{E}$?

To answer this, we work with the vectored covariance matrices $\mathbf{r}_{\mathbf{x}}$ and $\hat{\mathbf{r}}_{\mathbf{x}}$.

For a matrix-valued stochastic variable $\hat{\mathbf{R}}$, its covariance matrix can be defined as the covariance of $\hat{\mathbf{r}}$, i.e.,

$$\mathrm{cov}[\hat{\mathbf{R}}] = \mathrm{cov}[\hat{\mathbf{r}}] = \mathrm{E}[(\hat{\mathbf{r}} - \mathrm{E}[\hat{\mathbf{r}}])(\hat{\mathbf{r}} - \mathrm{E}[\hat{\mathbf{r}}])^{\mathrm{H}}] .$$

Next, insert the definition of $\hat{\mathbf{r}} = \frac{1}{N}\sum \mathbf{x}_n^* \otimes \mathbf{x}_n$, and use the fact that $\mathbf{x}_i$ is independent of $\mathbf{x}_j$ for $i \neq j$ to derive

$$\begin{aligned}
\mathrm{cov}[\hat{\mathbf{R}}_{\mathbf{x}}] &= \mathrm{E}\left[(\frac{1}{N}\sum(\mathbf{x}_i^* \otimes \mathbf{x}_i) - \mathrm{E}[\mathbf{x}_i^* \otimes \mathbf{x}_i])(\frac{1}{N}\sum(\mathbf{x}_j^* \otimes \mathbf{x}_j) - \mathrm{E}[\mathbf{x}_j^* \otimes \mathbf{x}_j])^{\mathrm{H}}\right] \quad (3.11)\\
&= \frac{1}{N^2}\sum\sum \mathrm{E}\left[(\mathbf{x}_i^* \otimes \mathbf{x}_i - \mathrm{E}[\mathbf{x}_i^* \otimes \mathbf{x}_i])(\mathbf{x}_j^* \otimes \mathbf{x}_j - \mathrm{E}[\mathbf{x}_j^* \otimes \mathbf{x}_j])^{\mathrm{H}}\right]\\
&= \frac{1}{N^2}\sum \mathrm{E}\left[(\mathbf{x}_i^* \otimes \mathbf{x}_i - \mathrm{E}[\mathbf{x}_i^* \otimes \mathbf{x}_i])(\mathbf{x}_i^* \otimes \mathbf{x}_i - \mathrm{E}[\mathbf{x}_i^* \otimes \mathbf{x}_i])^{\mathrm{H}}\right]\\
&= \frac{1}{N}\left(\mathrm{E}[(\mathbf{x}^* \otimes \mathbf{x})(\mathbf{x}^* \otimes \mathbf{x})^{\mathrm{H}}] - \mathrm{E}[\mathbf{x}^* \otimes \mathbf{x}]\mathrm{E}[\mathbf{x}^* \otimes \mathbf{x}]^{\mathrm{H}}\right) \quad (3.12)\\
&= \frac{1}{N}\mathbf{C}_{\mathbf{x}} ,
\end{aligned}$$

where

$$\mathbf{C}_{\mathbf{x}} = \mathrm{E}[(\mathbf{x}_k^* \otimes \mathbf{x}_k)(\mathbf{x}_k^* \otimes \mathbf{x}_k)^{\mathrm{H}}] \; - \; \mathrm{E}[\mathbf{x}_k^* \otimes \mathbf{x}_k]\mathrm{E}[\mathbf{x}_k^* \otimes \mathbf{x}_k]^{\mathrm{H}} . \quad (3.13)$$

The first term of this expression shows that the covariance of $\hat{\mathbf{R}}_{\mathbf{x}}$ involves fourth-order correlations. These can often be described in simpler terms using cumulants. A discussion of this is deferred to Chap. 11.

For the special case where the entries x_i of $\mathbf{x}$ are zero-mean and jointly Gaussian distributed, it is known that (for arbitrary indices $a, b, c, d = 0, \cdots, M-1$)

$$\mathrm{E}[x_a\, x_b^*\, x_c\, x_d^*] = \mathrm{E}[x_a x_b^*]\mathrm{E}[x_c x_d^*] + \mathrm{E}[x_a x_d^*]\mathrm{E}[x_b^* x_c] + \mathrm{E}[x_a x_c]\mathrm{E}[x_b^* x_d^*] . \quad (3.14)$$

This follows from an expression of the 4th order (joint) cumulant in terms of moments; for Gaussian random variables this cumulant is zero. "Proper" (or circularly symmetric) complex variables are such that $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \mathbf{0}$. In this case, the last term vanishes.

The LHS of (3.14) represents a 4th order moment. Stacking in a matrix with row-index $a + M\,b$ and column-index $c + M\,d$, we can write this expression compactly as

$$\mathrm{E}[(\mathbf{x}^*\otimes\mathbf{x})(\mathbf{x}^*\otimes\mathbf{x})^{\mathrm{H}}] = \mathrm{E}[\mathbf{x}^*\otimes\mathbf{x}]\mathrm{E}[\mathbf{x}^*\otimes\mathbf{x}]^{\mathrm{H}} + \mathrm{E}[\mathbf{x}^*\mathbf{x}^{*\mathrm{H}}]\otimes\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{H}}] + \mathrm{E}[(\mathbf{x}^*\otimes\mathbf{1})(\mathbf{1}\otimes\mathbf{x})^{\mathrm{H}}]\odot\mathrm{E}[(\mathbf{1}\otimes\mathbf{x})(\mathbf{x}^*\otimes\mathbf{1})^{\mathrm{H}}]\,.$$

For proper complex variables, the last term vanishes. For this case, if we compare to (3.13), we see that

$$\mathbf{C}_{\mathbf{x}} = \mathbf{R}_{\mathbf{x}}^* \otimes \mathbf{R}_{\mathbf{x}}\,.$$

Thus, for zero mean proper complex-valued Gaussian random variables,

$$\mathrm{cov}[\hat{\mathbf{R}}_{\mathbf{x}}] = \frac{1}{N}\mathbf{R}_{\mathbf{x}}^* \otimes \mathbf{R}_{\mathbf{x}}\,. \tag{3.15}$$

while for zero mean non-proper complex variables

$$\mathrm{cov}[\hat{\mathbf{R}}_{\mathbf{x}}] = \frac{1}{N}\left[\mathbf{R}_{\mathbf{x}}^* \otimes \mathbf{R}_{\mathbf{x}} + \mathrm{E}[(\mathbf{x}^*\otimes\mathbf{1})(\mathbf{1}\otimes\mathbf{x})^{\mathrm{H}}] \odot \mathrm{E}[(\mathbf{1}\otimes\mathbf{x})(\mathbf{x}^*\otimes\mathbf{1})^{\mathrm{H}}]\,.\right]\,. \tag{3.16}$$

and for zero mean real-valued Gaussian random variables

$$\mathrm{cov}[\hat{\mathbf{R}}_{\mathbf{x}}] = \frac{1}{N}\left[\mathbf{R}_{\mathbf{x}} \otimes \mathbf{R}_{\mathbf{x}} + \mathrm{E}[(\mathbf{x}\otimes\mathbf{1})(\mathbf{1}\otimes\mathbf{x})^{\mathrm{T}}] \odot \mathrm{E}[(\mathbf{1}\otimes\mathbf{x})(\mathbf{x}\otimes\mathbf{1}^{\mathrm{T}}]\right]\,. \tag{3.17}$$

3.2.3 Spatial power spectrum estimation

By generalizing (3.6), a spatial power spectrum estimate is given by

$$\hat{P}(\theta) = \mathbf{w}(\theta)^{\mathrm{H}}\hat{\mathbf{R}}\mathbf{w}(\theta)\,.$$

A classical spectrum (equal to the periodogram in temporal power spectrum estimation) is obtained by using a matched filter: $\mathbf{w}(\theta) = \frac{1}{\sqrt{M}}\mathbf{a}(\theta)$. For example, for a single source in white noise, we have

$$\mathbf{x}_n = \mathbf{a}(\theta_0)s_n + \mathbf{n}_n\,.$$

What is the expected value and the variance of this estimate?

The expected value is straightforward:

$$\mathrm{E}[\hat{P}(\theta)] = \mathbf{w}(\theta)^{\mathrm{H}}\mathbf{R}\mathbf{w}(\theta)\,.$$

To derive the variance, for simplicity of notation, we consider a single $\mathbf{w}$.

$$\begin{aligned}
\mathrm{var}[\hat{P}] \;=\; \mathrm{E}[\,|\hat{P} - \mathrm{E}[\hat{P}]|^2] &= \mathrm{E}[\,|\mathbf{w}\hat{\mathbf{R}}\mathbf{w} - \mathbf{w}\mathbf{R}\mathbf{w}|^2] \\
&= \mathrm{E}[\,|\mathbf{w}(\hat{\mathbf{R}} - \mathbf{R})\mathbf{w}|^2] \\
&= \mathrm{E}[\,|(\mathbf{w}^*\otimes\mathbf{w})^{\mathrm{H}}(\hat{\mathbf{r}} - \mathbf{r})|^2] \\
&= \mathrm{E}[(\mathbf{w}^*\otimes\mathbf{w})^{\mathrm{H}}(\hat{\mathbf{r}} - \mathbf{r})(\hat{\mathbf{r}} - \mathbf{r})^{\mathrm{H}}(\mathbf{w}^*\otimes\mathbf{w})] \\
&= (\mathbf{w}^*\otimes\mathbf{w})^{\mathrm{H}}\mathrm{cov}[\hat{\mathbf{R}}](\mathbf{w}^*\otimes\mathbf{w})\,.
\end{aligned}$$

Assuming zero mean proper complex Gaussian sources, we can insert (3.15):

$$\mathrm{var}[\hat{P}] = \frac{1}{N}(\mathbf{w}^* \otimes \mathbf{w})^{\mathrm{H}}(\mathbf{R}^* \otimes \mathbf{R})(\mathbf{w}^* \otimes \mathbf{w}) = \frac{1}{N}\mathbf{w}^*\mathbf{R}^*\mathbf{w}^* \otimes \mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w} = \frac{1}{N}|\mathbf{w}^{\mathrm{H}}\mathbf{R}\mathbf{w}|^2 = \frac{1}{N}|\mathrm{E}[y]|^2$$

In other words, the standard deviation of the spectrum estimate is $1/\sqrt{N}$ times the expected value of the spectrum estimate itself.

This is the same result as for the periodogram [2, Ch. 8]. The result is valid for any data-independent beamformer $\mathbf{w}(\theta)$, i.e., also if we apply tapering.

3.2.4 Variance

The variance of $\hat{\mathbf{r}}$ is a vector consisting of the diagonal entries of $\mathrm{cov}[\hat{\mathbf{r}}]$. The variance of $\hat{\mathbf{R}}$ is defined as an unfolding of this vector into a matrix. Each entry of this matrix then shows the variance of the corresponding entry in $\hat{\mathbf{R}}$.

Thus, if $\mathbf{D} = \mathrm{diag}(\mathbf{R})$ and $\mathbf{d} = \mathrm{vecdiag}(\mathbf{R})$, then, for zero mean complex proper Gaussian variables,

$$\mathrm{var}[\hat{\mathbf{r}}] = \frac{1}{N}\mathrm{vecdiag}(\mathbf{R}^* \otimes \mathbf{R}) = \frac{1}{N}\mathbf{d} \otimes \mathbf{d}$$

and

$$\mathrm{var}[\hat{\mathbf{R}}] = \mathrm{vec}^{-1}(\mathrm{var}[\mathbf{r}]) = \frac{1}{N}\mathbf{d}\mathbf{d}^{\mathrm{T}}\,.$$

Some examples follow.

Independent noise If $\mathbf{x}_k = \mathbf{n}_k$ is zero mean proper symmetric Gaussian noise with variance $\Sigma_{\mathbf{n}} = \mathrm{diag}(\boldsymbol{\sigma}_{\mathbf{n}})$ (i.e., the sensors have independent noise with variance $\sigma^2_{\mathbf{n},i}$), then

$$\begin{aligned} \mathbf{R} &= \boldsymbol{\Sigma}_{\mathbf{n}} \\ \mathrm{cov}[\hat{\mathbf{R}}] &= \frac{1}{N}\boldsymbol{\Sigma}_{\mathbf{n}} \otimes \boldsymbol{\Sigma}_{\mathbf{n}} \\ \mathrm{var}[\hat{\mathbf{R}}] &= \frac{1}{N}\boldsymbol{\sigma}_{\mathbf{n}}\boldsymbol{\sigma}_{\mathbf{n}}^{\mathrm{T}}\,. \end{aligned}$$

Single point source If $\mathbf{x}_k = \mathbf{a}s_k$, where s_k is a zero mean proper complex Gaussian source with unit variance (a non-unit variance can be incorporated in $\mathbf{a}$), then

$$\begin{aligned} \mathbf{R} &= \mathbf{a}\mathbf{a}^{\mathrm{H}} \\ \mathrm{cov}[\hat{\mathbf{R}}] &= \frac{1}{N}\mathbf{a}^*\mathbf{a}^{\mathrm{T}} \otimes \mathbf{a}\mathbf{a}^{\mathrm{H}} = \frac{1}{N}(\mathbf{a}^* \otimes \mathbf{a})(\mathbf{a}^* \otimes \mathbf{a})^{\mathrm{H}} \\ \mathrm{var}[\hat{\mathbf{R}}] &= \frac{1}{N}\mathbf{a}^*\mathbf{a}^{\mathrm{T}} \odot \mathbf{a}\mathbf{a}^{\mathrm{H}}\,. \end{aligned}$$

Although $\odot$ generally does not preserve the rank of matrices, it can be shown that $\mathbf{a}^*\mathbf{a}^{\mathrm{T}} \odot \mathbf{a}\mathbf{a}^{\mathrm{H}}$ is rank 1 (see Sec. 5.1.6).

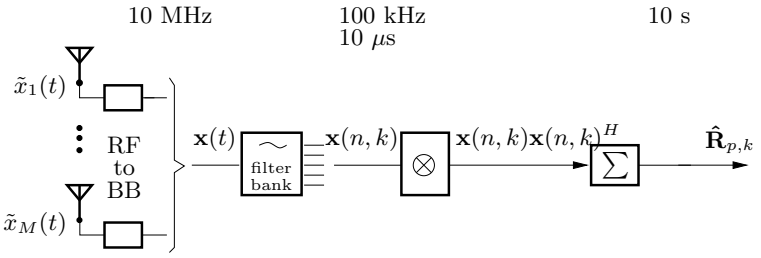


Figure 3.11. The processing chain to obtain covariance data.

3.3 APPLICATION: RADIO ASTRONOMY

In Sec. 2.5, we introduced radio astronomy. Starting from basic wave propagation, we arrived at the "Van Cittert-Zernike" measured data model (2.36) of the form

$$V(\omega, \mathbf{b}) = \int I(\omega, \boldsymbol{\zeta}) e^{-j\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}} \mathrm{d}\boldsymbol{\zeta} \tag{3.18}$$

which describes the received cross power spectral density $V(\omega, \mathbf{b})$ over a baseline $\mathbf{b}$, in terms of the image $I(\omega, \boldsymbol{\zeta})$, i.e., the intensity into the direction $\boldsymbol{\zeta}$, and the phase delays $\frac{\omega}{c}\boldsymbol{\zeta}\cdot\mathbf{b}$ in that look direction (the geometric delays).

With the tools in the present chapter, we can rewrite this into a data matrix form. Let us first consider the receiver model in a bit more detail.

3.3.1 Data acquisition

Mathematically, the correlation process is described as follows. Assume that there are M array elements (telescopes). The RF signal $\tilde{x}_j(t)$ from the jth telescope is first moved to baseband where it is denoted by $x_j(t)$, then sampled and split into narrow subbands, e.g., of 100 kHz each, such that the narrowband condition holds: the maximal geometric delay across the array should be fairly representable by a phase shift of the complex baseband signal.

For radio astronomy, the maximal geometric delay is related to the diameter of the array (usually several kilometers or nowadays up to several hundreds of kilometers). The bandwidth W such that the narrowband assumption is satisfied is then fairly small. In current systems that consist of many antennas spread over a large area, a hierarchy is made where the first group of antennas (a "station") covers only a relatively small area (a few hundred meters) such that the narrowband condition does not require very small bandwidths. The station antennas are combined via beamforming, and the beamformed output can be regarded as the output of a steered dish. Later processing stages then split the beamformed station signals into narrower bandwidths, as they are combined with antenna signals from farther away.

The resulting signal is called $x_m(n, k)$, for the mth telescope (or station), nth time bin, and for the subband frequency centered at RF frequency f_k. The M signals are stacked into a $M \times 1$

vector $\mathbf{x}(n,k)$.

A single correlation matrix is formed by "integrating" (summing) the crosscorrelation products $\mathbf{x}(n,k)\mathbf{x}^{\mathrm{H}}(n,k)$ over N subsequent samples,

$$\hat{\mathbf{R}}_{p,k} = \frac{1}{N} \sum_{n=(p-1)N}^{pN-1} \mathbf{x}(n,k)\mathbf{x}^{\mathrm{H}}(n,k)\,, \tag{3.19}$$

where p is the index of the corresponding "short-term interval" (STI) over which is correlated. The processing chain is summarized in Fig. 3.11.

The duration of an STI depends on the stationarity of the data, which is limited by factors like Earth rotation and the diameter of the array. For the Westerbork array, a typical value for the STI is 10 to 30 s; the total observation can last for up to 12 hours. The resulting number of samples N in a snapshot observation is equal to the product of bandwidth and integration time and typically ranges from 10^3 (1 s, 1 kHz) to 10^6 (10 s, 100 kHz) in radio astronomical applications.

3.3.2 Basic covariance data model

For our purposes, it is convenient to model the sky as consisting of a collection of Q spatially discrete point sources, with $s_q[n,k]$ the signal of the qth source at time sample n and frequency f_k.

For a single source, we saw in (3.5) that the received signal at an antenna array can be expressed as

$$\mathbf{x}[n,k] = \mathbf{a}_q[n,k]s_q[n,k]$$

where the array response vector $\mathbf{a}_q[n,k]$ has entries

$$a_m = e^{j\phi_m}\,, \qquad \phi_m = \frac{\omega}{c}\boldsymbol{\zeta}_q \cdot \mathbf{x}_m$$

where $\mathbf{x}_m$ is the position of the mth antenna and $\boldsymbol{\zeta}_q$ the direction vector of the qth source. For simplicity of notation, let us define normalized telescope position vectors,

$$\mathbf{z}_m[n,k] = \frac{2\pi f_k}{c}\mathbf{x}_m$$

As the earth rotates, the antenna positions are actually functions of time. We can collect them in a $3 \times M$ matrix

$$\mathbf{Z}[n,k] = [\mathbf{z}_1[n,k], \cdots, \mathbf{z}_M[n,k]]\,.$$

In this notation,

$$\mathbf{a}_q[n,k] = e^{j\mathbf{Z}(n,k)^{\mathrm{T}}\boldsymbol{\zeta}_q}\,, \tag{3.20}$$

Summing over all sources, we obtain

$$\mathbf{x}[n,k] = \sum_{q=1}^{Q} \mathbf{a}_q[n,k] s_q[n,k] + \mathbf{n}[n,k] \tag{3.21}$$

where $\mathbf{a}_q[n,k]$ is the array response vector for the qth source, consisting of the phase multiplication factors, and $\mathbf{n}[n,k]$ is an additive noise vector, due to thermal noise at the receiver. We will model $s_q[n,k]$ and $\mathbf{n}[n,k]$ as baseband complex envelope representations of zero mean wide sense stationary temporally white Gaussian random processes sampled at the Nyquist rate.

For convenience of notation, we will in future usually drop the dependence on the frequency f_k (index k) from the notation.

Previously, in (3.19), we defined correlation estimates $\hat{\mathbf{R}}_p$ as the output of the data acquisition process, where the time index p corresponds to the pth short term integration interval (STI), such that $(p-1)N \leq n \leq pN$. Due to Earth rotation, the vector $\mathbf{a}_q[n]$ changes slowly with time, but we assume that within an STI it can be considered constant and can be represented, with some abuse of notation, by $\mathbf{a}_q[p]$. In that case, $\mathbf{x}[n]$ is wide sense stationary over the STI, and a single STI autocovariance is defined as

$$\mathbf{R}_p = \mathrm{E}[\mathbf{x}[n]\,\mathbf{x}^{\mathrm{H}}[n]]\,, \qquad p = \lceil \frac{n}{N} \rceil \tag{3.22}$$

where $\mathbf{R}_p$ has size $M \times M$. Each element of $\mathbf{R}_p$ represents the interferometric correlation along the baseline vector between the two corresponding receiving elements. It is estimated by STI sample covariance matrices $\hat{\mathbf{R}}_p$ defined in (3.19), and our stationarity assumptions imply $\mathrm{E}[\hat{\mathbf{R}}_p] = \mathbf{R}_p$.

If we generalize now to Q sources and add zero mean noise, uncorrelated from antenna to antenna, as in the signal model (3.21), we obtain the covariance data model

$$\mathbf{R}_p = \mathbf{A}_p \mathbf{\Sigma_s} \mathbf{A}_p^{\mathrm{H}} + \mathbf{\Sigma_n}\,, \qquad p = 0, 1, 2, \cdots, \tag{3.23}$$

$$\begin{aligned} \text{where} \qquad \mathbf{A}_p &= [\mathbf{a}_1(p), \cdots, \mathbf{a}_Q(p)] \\ \mathbf{\Sigma_s} &= \mathrm{diag}[\sigma_{s,1}^2, \cdots, \sigma_{s,Q}^2] \\ \mathbf{\Sigma_n} &= \mathrm{E}[\mathbf{n}(p)\,\mathbf{n}^{\mathrm{H}}(p)] \;=\; \mathrm{diag}[\sigma_{n,1}^2, \cdots, \sigma_{n,M}^2]\,. \end{aligned}$$

Here, $\sigma_{s,q}^2 = \mathrm{E}[|s_q(n,k)|^2]$ is the variance of the qth source, $\mathbf{\Sigma_s}$ is the corresponding signal covariance matrix, and $\mathbf{\Sigma_n}$ is the noise covariance matrix. Noise is assumed to be independent but not evenly distributed across the array. The noise variances $\sigma_{n,j}^2$ are considered unknown until they have been calibrated. This *measurement equation* is actually a matrix version of (3.18).

Under ideal circumstances, the array response matrix $\mathbf{A}_p$ is just a phase matrix: its columns are given by the vectors $\mathbf{a}_q(p)$ in (3.20), and its entries express the phase shifts due to the geometrical delays associated with the array and source geometry. We will later generalize this and introduce directional disturbances due to non-isotropic antennas, unequal antenna gains, and disturbances due to atmospheric effects.

3.3.3 Image formation for the ideal data model

Ignoring the additive noise and using the ideal array response matrix $\mathbf{A}_p$, the measurement equation (3.23), in its simplest form, can be written as

$$(\mathbf{R}_p)_{i,j} = \sum_{q=1}^{Q} I(\boldsymbol{\zeta}_q)\, e^{j(\mathbf{z}_i(p)-\mathbf{z}_j(p))^{\mathrm{T}}\boldsymbol{\zeta}_q} \tag{3.24}$$

where $(\mathbf{R}_p)_{i,j}$ is the correlation between antennas i and j at STI interval p, $I(\boldsymbol{\zeta}_q) = \sigma_q^2$ is the brightness (power) of the source in direction $\boldsymbol{\zeta}_q$, $\mathbf{z}_i(p)$ is the normalized location vector of the ith antenna at STI p, and $\boldsymbol{\zeta}_q$ is the unit propagation vector from the qth source.

The function $I(\boldsymbol{\zeta})$ is the brightness image (or map) of interest. For our discrete point-source model, it is

$$I(\boldsymbol{\zeta}) = \sum_{q=1}^{Q} \sigma_q^2\, \delta(\boldsymbol{\zeta} - \boldsymbol{\zeta}_q) \tag{3.25}$$

where $\delta(\cdot)$ is a Kronecker delta, and the direction vector $\boldsymbol{\zeta}$ is mapped to the location of "pixels" in the image (various transformations are possible). Only the pixels $\boldsymbol{\zeta}_q$ are nonzero, and have value equal to the source variance σ_q^2.

Equation (3.24) describes the relation between the visibility model and the desired image, and it has the form of a Fourier transform; as discussed in Chap. 2.5, it is the Van Cittert-Zernike theorem [4, 5]. Image formation is essentially the inversion of this relation. We discussed this in Sec. 2.5. In the present setting, we have only a finite set of observations (as indexed by p). If we apply the inverse "discrete-space" Fourier transformation to the measured correlation data, we obtain the dirty image

$$\hat{I}_D(\boldsymbol{\zeta}) := \sum_{i,j,p} (\hat{\mathbf{R}}_p)_{ij}\; e^{-j(\mathbf{z}_i(p)-\mathbf{z}_j(p))^{\mathrm{T}}\boldsymbol{\zeta}}\,. \tag{3.26}$$

In terms of the measurement data model (3.24), the "expected value" of the image is obtained by replacing $\hat{\mathbf{R}}_p$ by $\mathbf{R}_p$, or

$$\begin{aligned} I_D(\boldsymbol{\zeta}) &:= \sum_{i,j,p} (\mathbf{R}_p)_{i,j}\; e^{-j(\mathbf{z}_i(p)-\mathbf{z}_j(p))^{\mathrm{T}}\boldsymbol{\zeta}} \\ &= \sum_{i,j,p}\sum_{q} \sigma_q^2\; e^{-j(\mathbf{z}_i(p)-\mathbf{z}_j(p))^{\mathrm{T}}(\boldsymbol{\zeta}-\boldsymbol{\zeta}_q)} \\ &= \sum_{q} I(\boldsymbol{\zeta}_q)\; B(\boldsymbol{\zeta}-\boldsymbol{\zeta}_q) \\ &= I(\boldsymbol{\zeta}) * B(\boldsymbol{\zeta}) \end{aligned} \tag{3.27}$$

where the dirty beam (or point spread function) is given by

$$B(\boldsymbol{\zeta}) := \sum_{i,j,p} e^{-j(\mathbf{z}_i(p)-\mathbf{z}_j(p))^{\mathrm{T}}\boldsymbol{\zeta}}\,. \tag{3.28}$$

This is the same result as in Sec. 2.5, but now for spatially sampled observations. Again, the dirty image $I_D(\boldsymbol{\zeta})$ is the desired image $I(\boldsymbol{\zeta})$ convolved with the dirty beam $B(\boldsymbol{\zeta})$: every point source excites a beam $B(\boldsymbol{\zeta}-\boldsymbol{\zeta}_q)$ centered at its location $\boldsymbol{\zeta}_q$. Note that $B(\boldsymbol{\zeta})$ is a known function: it only depends on the locations of the telescopes, or rather the sampled set of telescope baselines $\mathbf{z}_i(p) - \mathbf{z}_j(p)$.

3.4 NOTES

Section 3.3 is based on Van der Veen e.a. [3].

Bibliography

[1] J.G. Proakis and M. Salehi, *Communication Systems Engineering*. Prentice-Hall, 1994.

[2] M.H. Hayes, *Statistical digital signal processing and modeling*. Wiley, 1996.

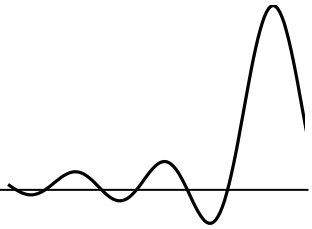
[3] A.J. van der Veen, S.J. Wijnholds, and A.M. Sardarabadi, "Signal processing for radio astronomy," in *Handbook of Signal Processing Systems, 3rd ed.*, Springer, November 2018. ISBN 978-3-319-91734-4.

[4] R.A. Perley, F.R. Schwab, and A.H. Bridle, *Synthesis Imaging in Radio Astronomy*, vol. 6 of *Astronomical Society of the Pacific Conference Series*. BookCrafters Inc., 1994.

[5] A.R. Thompson, J.M. Moran, and G.W. Swenson, *Interferometry and Synthesis in Radio Astronomy*. New York: Wiley, 2nd ed., 2001.

Chapter 4

WIDEBAND DATA MODELS



Contents

Having covered narrowband data models, in this chapter we continue and focus on wideband data models. These are used in the context of wireless (RF) communication systems, where convolutions by pulse shape functions and channel propagation delays play an important role. A data model for wireless communication consists of the following parts (see Fig. 4.1):

1. *Source model:* signal alphabet, data packets, and modulation by a pulse shape function;
2. *Physical channel:* multipath propagation over the wireless channel, based on the wave propagation model of Chapter 2;
3. *Receiver model:* reception of multiple signals at multiple antennas, sampling, beamforming, equalization and decision making. This is about algorithms, as covered in the subsequent chapters.

We start by looking at models for the physical channel.

4.1 PHYSICAL CHANNEL PROPERTIES

Wide-area multipath propagation model We consider in this section a wireless communication setting, i.e., the propagation of a signal from a transmitter ("mobile") through a medium

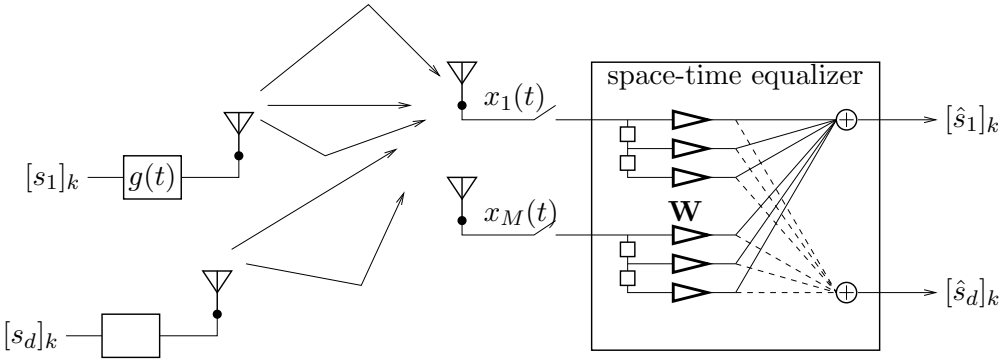


Figure 4.1. Wireless communication scenario

("channel") to a receiver ("base station"). This allows to present concepts that are to some extent also relevant in other contexts, such as microphone arrays, radar, GPS receivers, etc.

The propagation of signals through the wireless channel is fairly complicated to model. A correct treatment would require a complete description of the physical environment, and would not be very useful for the design of signal processing algorithms. To arrive at a more useful parametric model, we have to make simplifying assumptions regarding the wave propagation. Provided this model is reasonably valid, we can, in a second stage, try to derive statistical models for the parameters to obtain reasonable agreement with measurements.

The number of parameters in an accurate model can be quite high, and from a signal processing point of view, they might not be very well identifiable. For this reason, another model used in signal processing is a much less sophisticated *unparametrized* model. The radio channel is simply modeled as an FIR (finite impulse response) filter, with main parameters the impulse response length (in symbols) and the total attenuation or signal-to-noise ratio (SNR). This model is described in Section 4.3. The parametrized model is a special case, giving structure to the FIR coefficients.

Jakes' model A commonly used parametric model is a multiray scattering model, also known as Jakes' model (after Jakes [1], see also [2–6]). In this model, the signal follows on its way from the source to the receiver a number of distinct paths, referred to as multipaths. These arise from scattering, reflection, or diffraction of the radiated energy on objects that lie in the environment. The received signal from each path is much weaker than the transmitted signal due to various scattering and fading effects. Multipath propagation also results in the spreading of the signal in various dimensions: delay spread in time, Doppler spread in frequency, and angle spread in space. Each of them has a significant effect on the signal. The mean path loss, shadowing, fast fading, delay, Doppler spread and angle spread are the main channel characteristics and form the parameters of the multiray model.

The scattering of the signal in the environment can be specialized into three stages: scattering

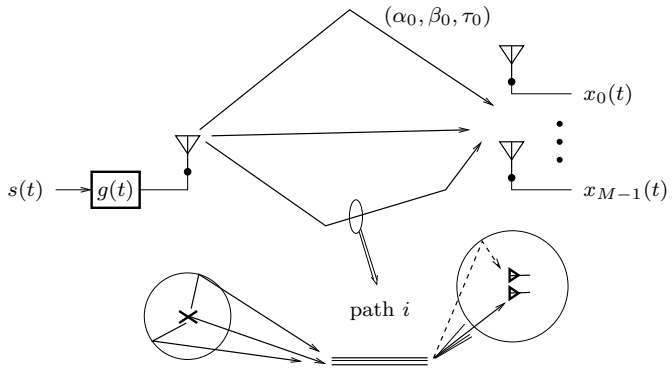


Figure 4.2. Multipath propagation model in a wireless communication setting.

local to the source at surrounding objects, reflections on distant objects of the few dominant rays that emerge out of the local clutter, and scattering local to the receiver. See Fig. 4.2.

Scatterers local to the mobile Scattering local to the mobile is caused by buildings and other objects in the direct vicinity of the mobile (at, say, a few tens of meters). Motion of the mobile and local scattering give rise to Doppler spread which causes "time-selective fading": the signal power can have significant fluctuations over time. While local scattering contributes to Doppler spread, the delay spread will usually be insignificant because of the small scattering radius. Likewise, the angle spread will also be small.

Remote scatterers Away from the cluster of local scatterers, the emerging wavefronts may then travel directly to the base or may be scattered toward the base by remote *dominant scatterers*, giving rise to specular multipath. These remote scatterers can be either terrain features (distant hills) or high rise building complexes. Remote scattering can cause significant delay and angle spreads.

Scatterers local to the base Once these multiple wavefronts reach the base station, they may be scattered further by local structures such as buildings or other structures that are in the vicinity of the base. Such scattering will be more pronounced for low elevation and below-rooftop antennas. The scattering local to the base can cause significant angle spread which can cause space-selective fading: different antennas at the base station can receive totally different signal powers. This fading is time invariant, unlike the time varying space-selective fading caused by remote scattering.

Doppler spread and time selective fading If mobiles or scatterers are moving, then the phases of each multipath component are quickly changing relative to each other, hence they add up

differently over time. As mentioned, this causes (fast) fluctuations in the received signal power over that ray (time-selective fading).

Movement also results in a *Doppler spread*, i.e., a pure CW tone is spread over a non-zero spectral bandwidth. If a source moves with a velocity of v m/s towards the receiver, then its observed frequency is increased by $f_m = v/\lambda$ [Hz], or $\omega_m = v\frac{2\pi}{\lambda}$. Likewise, if the source moves away from the receiver, its observed frequency is reduced by f_m. If it moves sideways, there is no shift in frequency.

If there is a ring of scatterers around the mobile, then seen via some reflectors, the mobile may seem to move away, while via other reflectors, it seems to approach. Thus, we obtain a distribution of Doppler shifts. If one assumes uniformly distributed scatterers, then the baseband power spectrum of the vertical electrical field component of the channel is convolved with [1, ch.1]

$$S(\omega) \;=\; \frac{3}{\omega_m}\left[1-\left(\frac{\omega}{\omega_m}\right)^2\right]^{-1/2}, \quad |\omega| < \omega_m \tag{4.1}$$

The Doppler spectrum described by (4.1) is often called the *classical* spectrum. For a mobile traveling at 100 kph, the Doppler spread is approximately $f_m = 175$ Hz in the 1900 MHz band.

Because a convolution in frequency domain translates in pointwise multiplication in time domain, and the function is non-flat in this case, Doppler spread causes time selective fading. It is usually characterized by the *coherence time* of the channel [1], i.e., the time lag over which the Doppler time function has an autocorrelation larger than 0.5. The larger the Doppler spread, the smaller the coherence time. The coherence time is in the order of $1/\omega_m$, i.e., approximately 0.9 ms for $f_m = 175$ Hz. In comparison, the burst length in a single GSM data package is 0.577 ms, so that the GSM channel can be regarded almost time-invariant during the burst, but not in between two bursts.

Signal processing model Let us ignore the local scattering for the moment, and assume that there are r rays bouncing off remote objects such as hills or tall buildings. As extension of the narrowband model (3.4), the received parametric signal model is then usually written as the convolution

$$\mathbf{x}(t) = \mathbf{h}(t) * s(t)\,, \quad \mathbf{h}(t) = \left[\sum_{i=1}^{r} \mathbf{a}(\theta_i)\beta_i g(t-\tau_i)\right], \tag{4.2}$$

where $\mathbf{x}(t)$ is a vector consisting of the M antenna outputs, $\mathbf{a}(\theta)$ is the array response vector, and the impulse response $g(t)$ collects all temporal aspects, such as pulse shaping and transmit and receive filtering. The model parameters of each ray are its (mean) angle-of-incidence θ_i, (mean) path delay τ_i, and path loss β_i. The latter parameter lumps the overall attenuation, all phase shifts, and possibly the antenna response $a_0(\theta)$ as well.

Each of the rays is itself composed of a large number of "mini-rays" due to scattering close to the source: all with roughly equal angles and delays, but arbitrary phases. This can be described by extending the model with additional parameters such as the standard deviations from the mean

Table 4.1. Typical delay, angle and Doppler spreads in cellular applications.

Environment	delay spread	angle spread	Doppler spread
Flat rural (macro)	0.5 μs	1°	190 Hz
Urban (macro)	5 μs	20°	120 Hz
Hilly (macro)	20 μs	30°	190 Hz
Mall (micro)	0.3 μs	120°	10 Hz
Indoors (pico)	0.1 μs	360°	5 Hz

angle θ_i and mean delay τ_i, which depend on the radius (aspect ratio) of the scattering region and its distance to the remote scattering object [7, 8]. For macroscopic models, the standard deviations are generally small (less than a few degrees, and a fraction of τ_i) and are usually but not always ignored.

The local scattering however has a major effect on the statistics and stationarity of β_i. For example, if all local rays have equal amplitude, then β_i is the sum of a large number of arbitrary complex numbers, each with equal modulus but random phase, which gives β_i a complex Gaussian distribution. Consequently, its amplitude has a Rayleigh distribution (hence the name Rayleigh fading). More in general, if there is a strong path with some scattering around it that causes fluctuation, then often a Rice distribution or log-normal distribution is assumed.

A second effect is that $\beta_i = \beta_i(t)$ is really (slowly) time-varying: if the source is in motion, then the Doppler shifts and/or the varying location change the phase differences among the rays, so that the sum can be totally different from one time instant to the next. The maximal Doppler shift f_D is given by the speed of the source (in m/s) divided by the wavelength of the carrier. The *coherence time* of the channel is inversely proportional to f_D, roughly by a factor of 0.2: $\beta_i(t)$ can be considered approximately constant for time intervals smaller than this time [4, 9, 10]. Angles and delays are generally assumed to be stationary over much longer periods.

A proper discussion should now present statistical models for θ_i, β_i, and τ_i. Since this is not the focus of the book, we omit further details.

Typical channel parameters Angle spread, delay spread, and Doppler spread are important characterizations of a mobile channel, as it determines the amount of equalization that is required, but also the amount of diversity that can be obtained. Measurements in macrocells indicate that up to 6 to 12 dominant paths may be present. Typical channel delay and Doppler spreads (1800 MHz) are given in table 4.1 [4, 9] (see also references in [5]). Typical angle spreads are not well known; the given values are suggested by [6].

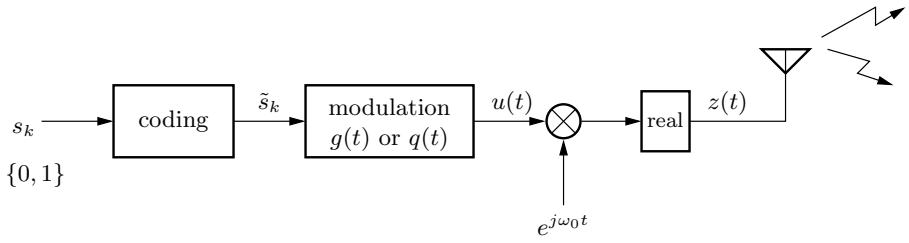


Figure 4.3. Modulation process.

4.2 SIGNAL MODULATION

Before a digital bit sequence can be transmitted over a radio channel, it has to be prepared: among other things, it has to be transformed into an analog signal in continuous time and modulated onto a carrier frequency. The various steps are shown in Fig. 4.3. The *coding* step, in its simplest form, translates the binary sequence $\{s_k\} \in \{0, 1\}$ into a sequence $\{\tilde{s}_k\}$ with another alphabet, such as $\{-1, +1\}$. A digital filter may be part of the coder as well. In linear modulation schemes, the resulting sequence is then convolved with a pulse shape function $g(t)$, whereas in phase modulation, it is convolved with some other pulse shape function $q(t)$ to yield the phase of the modulated signal. The resulting baseband signal $u(t)$ is modulated by the carrier frequency ω_0 to produce the RF signal that will be broadcast.

In this section, a few examples of coding alphabets and pulse shape functions are presented, for future reference. We do not go into the properties and reasons why certain modulation schemes are chosen; see e.g. [11] for more details.

4.2.1 Time-domain modulations

Digital alphabets The first step in the modulation process is the coding of the binary sequence $\{s_k\}$ into some other sequence $\{\tilde{s}_k\}$. The $\{\tilde{s}_k\}$ are chosen from an *alphabet* or *constellation*, which might be real or complex. There are many possibilities; common examples are BPSK (binary phase shift keying), QPSK (quadrature phase shift keying), PAM-m (pulse amplitude modulation), QAM-m (quadrature amplitude modulation), MSK (minimum-shift keying), DQPSK (differential QPSK), defined as in table 4.2. See also Fig. 4.4. Smaller constellations are more robust in the presence of noise, because of the larger distance between the symbols. Larger constellations may lead to higher bitrates, but are harder to detect in noise.

It is possible that the data rate of the output of the coder is different than the input data rate. E.g., if a binary sequence is coded into QPSK, the data rate halves. (The opposite is also possible, e.g., in CDMA systems, where each bit is coded into a sequence of 31 or more "chips".)

Table 4.2. Common digital constellations

	$\tilde{s}_k$ chosen from:	
BPSK	$\{1, -1\}$	
PAM-m	$\{-m, \cdots, -1, 1, \cdots, m\}$	
QPSK (QAM-4)	$\{1, -1, j, -j\}$	
MSK	$\{1, -1\}$,	k even
	$\{j, -j\}$,	k odd
DQPSK	$\{1, -1, j, -j\}$,	k even
	$\{e^{j\pi/4}, e^{j3\pi/4}, e^{-j3\pi/4}, e^{-j\pi/4}\}$,	k odd

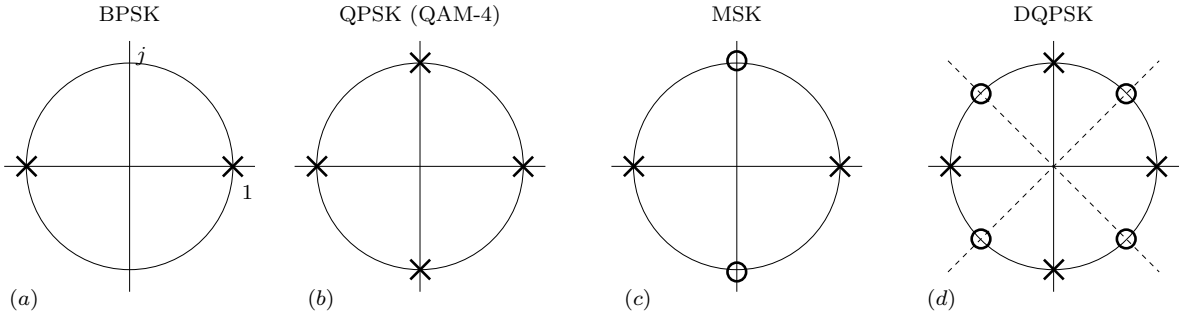


Figure 4.4. Digital constellations.

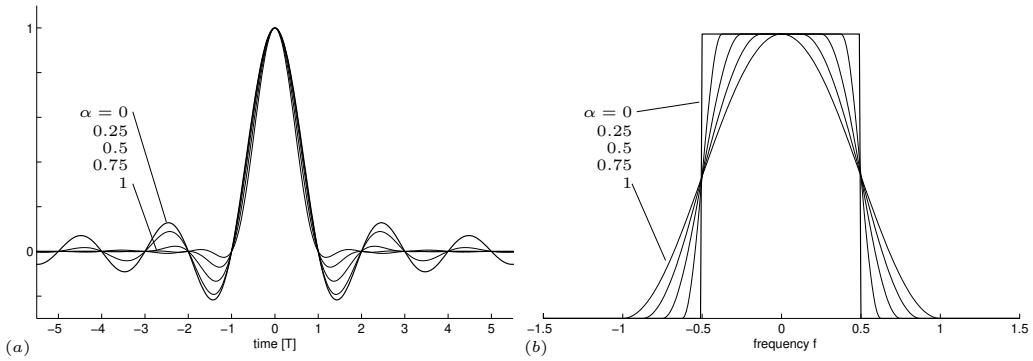


Figure 4.5. (a) Family of raised-cosine pulse shape functions, (b) corresponding spectra.

Pulse shape functions The coded digital signal $\tilde{s}(t)$ can be described as a sequence of dirac pulses,

$$\tilde{s}(t) = \sum_{-\infty}^{\infty} \tilde{s}_k \delta(t-k)\,,$$

where, for convenience, the symbol rate is normalized to $T = 1$. In linear modulation schemes, the digital dirac-pulse sequence is convolved by a pulse shape function $g(t)$:

$$u(t) = g(t) * \tilde{s}(t) = \sum_{-\infty}^{\infty} \tilde{s}_k g(t-k)\,. \tag{4.3}$$

Again, there are many possibilities. The optimum wave form is one that is both localized in time (to lie within a pulse period of length $T = 1$) and in frequency (to satisfy the Nyquist criterion when sampled at a rate $1/T = 1$). This is of course impossible, but good approximations exist. A pulse with perfect frequency localization is the sinc-pulse, defined by

$$g(t) = \frac{\sin \pi t}{\pi t}\,, \qquad G(f) = \begin{cases} 1\,, & |f| < \frac{1}{2} \\ 0\,, & \text{otherwise} \end{cases} \tag{4.4}$$

However, the pulse has very long tails in the time-domain.

Raised cosine pulseshape A modification of this pulse leads to the family of *raised-cosine* pulseshapes, with better localization properties. They are defined, for $\alpha \leq 1$, by [11, ch.6]

$$g(t) = \frac{\sin \pi t}{\pi t} \cdot \frac{\cos \alpha \pi t}{1 - 4\alpha^2 t^2}$$

with corresponding spectrum

$$G(f) = \begin{cases} 1\,, & |f| < \frac{1}{2}(1-\alpha) \\ \frac{1}{2} - \frac{1}{2}\sin(\frac{\pi}{\alpha}(|f| - \frac{1}{2}))\,, & \frac{1}{2}(1-\alpha) < |f| < \frac{1}{2}(1+\alpha) \\ 0\,, & \text{otherwise} \end{cases}$$

The spectrum is limited to $|f| \leq \frac{1}{2}(1+\alpha)$, so that α represents the excess bandwidth. For $\alpha = 0$, the pulse is identical to the sinc pulse (4.4). For other values of α, the amplitude decays more smoothly in frequency, so it is also known as the *rolloff factor*. The shape of the rolloff is that of a cosine, hence the name. In the time domain, the pulses are still infinite in extent. However, as α increases, the size of the tails diminishes. A common choice is $\alpha = 0.35$, and to truncate $g(t)$ outside the interval $[-3, 3]$.

The raised-cosine pulses are designed such that, when sampled at integer time instants, the only nonzero sample occurs at $t = 0$. Thus, $u(k) = \tilde{s}_k$, and to recover $\{\tilde{s}_k\}$ from $u(t)$ is simple, provided we are synchronized: any fractional delay $0 < \tau < 1$ results in intersymbol interference.

Phase modulation Many other modulation formats exists, in particular *phase modulations* are often used. An example is GMSK as used in the GSM system. For signal processing purposes, these are very often hard to handle. In some cases, these nonlinear modulations can be well approximated by linear modulations (e.g., GMSK), in other cases, we simply use some general properties of the resulting signal. E.g., several modulation formats are based on frequency or phase modulation and satisfy a constant-modulus property ($|s(t)| = 1$).

4.2.2 Spread spectrum signalling

In Code Division Multiple Access (CDMA) systems, instead of directly modulating a symbol sequence $\{s_k\}$ with a pulse shape function $g(t)$, the symbols are first *spread* with a user-specific code vector $\mathbf{c}$. The code vector consists of G symbols c_n called *chips*. Usually $c_n \in \{0, 1\}$ or $\{-1, 1\}$. The coded sequence is

$$\tilde{\mathbf{s}} = \begin{bmatrix} s_1\mathbf{c} \\ s_2\mathbf{c} \\ \vdots \end{bmatrix} = \mathbf{s} \otimes \mathbf{c}\,. \tag{4.5}$$

which is subsequently modulated by $g(t)$ in the usual way, cf. (4.3). Here, '$\otimes$' denotes a Kronecker product, which for vectors is defined as indicated. (Properties of Kronecker products are found in Sec. 5.1.6.) If the original sequence has a symbol duration $T = 1$, then each code chip has a duration T/G, and $g(t)$ is scaled accordingly. Thus, in frequency domain the pulse $G(f)$ occupies G times more bandwidth: this is called *spread spectrum* modulation. We can also view the combination of code $\mathbf{c}$ and pulse $g(t)$ as a new coded pulse $\tilde{g}(t)$ that now has a more complicated, user-specific form,

$$\tilde{g}(t) = \sum_{i=0}^{G-1} c_i g\left(t - \frac{i}{G}\right)\,.$$

Typical values of G are 31 till 1024. The codes are used to distinguish individual users. Instead of giving each user a dedicated time slot or frequency subband, they get a specific user code. This permits us to separate a superposition of multiple users at a (basestation) receiver. An advantage is that more than G users can be active simultaneously. A disadvantage is that, due

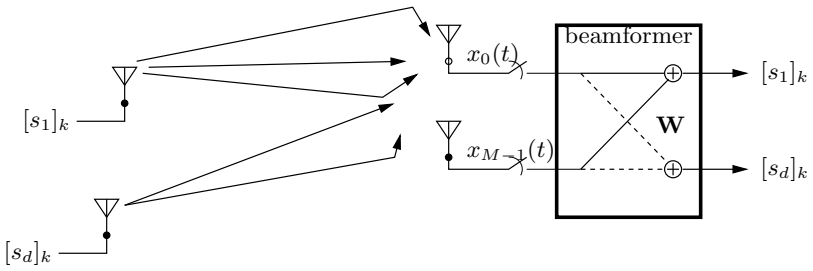


Figure 4.6. Spatial beamformer with an I-MIMO channel.

to the shorter chip duration, the convolution by the channel impulse response has more impact, and equalization is more complicated.

In practical systems like the 3rd generation (3G) mobile system UMTS, the used codes are non-periodic: they differ from symbol to symbol. This requires a simple extension of (4.5) to use symbol-specific codes $\mathbf{c}_k$:

$$\tilde{\mathbf{s}} = \begin{bmatrix} s_1\mathbf{c}_1 \\ s_2\mathbf{c}_2 \\ \vdots \end{bmatrix}$$

The factorization using Kronecker products is now not possible.

Spread spectrum is also used in GPS, with quite long codes that are different for each satellite. E.g., for the C/A code, $G = 1023 \cdot 20 = 20\,460$, while the symbol rate is at 50 bits/s. Long code lengths allow this system to operate at very low received powers.

4.3 DETERMINISTIC DATA MODELS

In Sec. 4.1, we have presented a channel model based on physical properties of the radio channel. Though useful for generating simulated data, it is not always a suitable model for identification purposes, e.g., if the number of parameters is large, if the angle spreads within a cluster are large so that parametrization in terms of directions is not possible, or if there is a large and fuzzy delay spread. In these situations, it is more appropriate to work with an unstructured model, where the channel impulse responses are posed simply as arbitrary multichannel finite impulse response (FIR) filters. It is a generalization of the physical channel model considered earlier, in the sense that at a later stage we can still specify the structure of the coefficients.

In this section, we look at deterministic data models, i.e., no stochastic considerations are used. In this case, the sampled data is directly placed in a matrix $\mathbf{X}$ which is subsequently analyzed.

4.3.1 I-MIMO model

Assume that d source signals $s_1(t), \cdots, s_d(t)$ are transmitted from d independent sources at different locations. If the delay spread is small, then what we receive at the antenna array will be a simple linear combination of these signals:

$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \cdots + \mathbf{a}_d s_d(t)$$

where as before $\mathbf{x}(t)$ is a stack of the output of the M antennas. We will usually write this in matrix form:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)\,, \quad \mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_d]\,, \quad \mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ \vdots \\ s_d(t) \end{bmatrix}.$$

Suppose we sample with a period T, normalized to $T = 1$, and collect a batch of N samples into a matrix $\mathbf{X}$, then

$$\mathbf{X} = \mathbf{A}\mathbf{S}$$

where $\mathbf{X} = [\mathbf{x}(0), \cdots, \mathbf{x}(N-1)]$ and $S = [\mathbf{s}(0), \cdots, \mathbf{s}(N-1)]$. The resulting $[\mathbf{X} = \mathbf{A}\mathbf{S}]$ model is called an *instantaneous multi-input multi-output* model, or I-MIMO for short. It is a generic linear model for source separation, valid when the delay spread of the dominant rays is much smaller than the inverse bandwidth of the signals, e.g., for narrowband signals, in line-of-sight situations or in scenarios where there is only local scattering. Even though this appears to limit its applicability, it is important to study it in its own right, since more complicated convolutive models can often be reduced (after equalization or separation into sufficiently narrow subbands) to $\mathbf{X} = \mathbf{A}\mathbf{S}$.

The objective of beamforming for source separation is to construct a left-inverse $\mathbf{W}^{\mathrm{H}}$ of $\mathbf{A}$, such that $\mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I}$ and hence $\mathbf{W}^{\mathrm{H}}\mathbf{X} = \mathbf{S}$: see Fig. 4.6. This will recover the source signals from the observed mixture. It immediately follows that in this scenario it is necessary to have $d \leq M$ to ensure interference-free reception, i.e., not more sources than sensors. If we know already (part of) $\mathbf{S}$, e.g., because of training, then we can estimate $\mathbf{W}$ via $\mathbf{W}^{\mathrm{H}} = \mathbf{S}\mathbf{X}^{\dagger} = \mathbf{S}\mathbf{X}^{\mathrm{H}}(\mathbf{X}\mathbf{X}^{\mathrm{H}})^{-1}$, where $\mathbf{X}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{X}$, here equal to its right inverse (see Chapter 5). With noise, other beamformers may be better.

Coherent multipath If we adopt the multipath propagation model, then $\mathbf{A}$ is endowed with a parametric structure: every column $\mathbf{a}_i$ is a sum of direction vectors $\mathbf{a}(\theta_{ij})$, with different fadings β_{ij}. If the ith source is received through r_i rays, then

$$\mathbf{a}_i \;=\; \sum_{j=1}^{r_i} \mathbf{a}(\theta_{ij})\beta_{ij} \;=\; [\mathbf{a}(\theta_{i1}), \cdots, \mathbf{a}(\theta_{i,r_i})] \begin{bmatrix} \beta_{i1} \\ \vdots \\ \beta_{i,r_i} \end{bmatrix} \qquad (i = 1, \cdots, d)\,.$$

If each source has only a single ray to the receiver array (a line-of-sight situation), then each $\mathbf{a}_i$ is a vector on the array manifold, and identification will be relatively straightforward. The more

general case amounts to decomposing a given **a**-vector into a sum of vectors on the manifold, which makes identification much harder.

To summarize the parametric structure in a compact way, we could collect all $\mathbf{a}(\theta_{ij})$-vectors and path attenuation coefficients β_{ij} of all rays of all sources in single matrices $\mathbf{A}_\theta$ and $\mathbf{B}$,

$$\mathbf{A}_\theta = [\mathbf{a}(\theta_{11}), \cdots, \mathbf{a}(\theta_{d,r_d})]\,, \qquad \mathbf{B} = \operatorname{diag}[\beta_{11}, \cdots, \beta_{d,r_d}]\,.$$

To sum the rays belonging to each source into the single $\mathbf{a}_i$-vector of that source, we define a selection matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{1}_{r_1} & & 0 \\ & \ddots & \\ 0 & & \mathbf{1}_{r_d} \end{bmatrix} : \quad r \times d \tag{4.6}$$

where $r = \sum_1^d r_i$ and $\mathbf{1}_m$ denotes an $m \times 1$ vector consisting of 1's. Together, this allows to write the full (noise-free) I-MIMO data model as

$$\mathbf{X} = \mathbf{AS}\,, \qquad \mathbf{A} = \mathbf{A}_\theta \mathbf{B} \mathbf{J}\,. \tag{4.7}$$

4.3.2 Convolutive model for one antenna and one source

To extend the instantaneous model to a situation with convolutive channels, let $h[k]$ be a finite impulse response (FIR) filter. The matrix equation corresponding to a convolution $x[n] = h[n] * s[n] = \sum_{k=0}^{L-1} h[k]s[n-k]$ is

$$\mathbf{x} = \mathbf{Hs} \quad \Leftrightarrow \quad \begin{bmatrix} \boxed{x[0]} \\ x[1] \\ x[2] \\ \vdots \\ \\ \vdots \\ \\ \vdots \\ x[N-1] \end{bmatrix} = \begin{bmatrix} \boxed{h[0]} & & & \mathbf{0} \\ h[1] & h[0] & & \\ h[2] & h[1] & \ddots & \\ \vdots & h[2] & \ddots & h[0] \\ h[L-1] & \vdots & \ddots & h[1] \\ & h[L-1] & \ddots & h[2] \\ & & \ddots & \vdots \\ \mathbf{0} & & & h[L-1] \end{bmatrix} \begin{bmatrix} \boxed{s[0]} \\ s[1] \\ \vdots \\ s[N_s-1] \end{bmatrix} \tag{4.8}$$

where the "box" indicates the location of time-index 0, L is the channel length, N_s is the length of the input sequence (prior and subsequent symbols are supposed to be zero; this is usually achieved by a guard interval), and $N = N_s + L - 1$ is the length of the observation (ignoring the other samples). Note that $\mathbf{H}$ has size $N_s + L - 1 \times N_s$, so $\mathbf{H}$ is always tall. If there is no guard interval, we have to drop the first $L-1$ samples of $\mathbf{x}$ since they are "contaminated" by prior symbols, and the top part of $\mathbf{H}$ has to be dropped accordingly. Likewise, we will probably

have to drop the last $L-1$ rows of $\mathbf{H}$ as well, if we have to assume that subsequent symbols $s[N_s], s[N_s+1], \cdots$ are nonzero and unknown. This will reduce the size of $\mathbf{H}$ to $N_s - L + 1 \times N_s$, and it is not tall anymore.

$\mathbf{H}$ has a Toeplitz structure: it is constant along diagonals. That structure always appears when we have time-invariant systems.

Suppose we observe $\mathbf{x}$ and know the channel matrix $\mathbf{H}$, and it is tall. The input sequence can be estimated by taking a left inverse $\mathbf{H}^\dagger$ of $\mathbf{H}$, such that $\mathbf{H}^\dagger\mathbf{H} = \mathbf{I}$. Since $\mathbf{H}$ is tall, we can usually take

$$\mathbf{H}^\dagger = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}$$

where, for now, we assume that $\mathbf{H}^{\mathrm{H}}\mathbf{H}$ is invertible. This results in

$$\hat{\mathbf{s}} = \mathbf{H}^\dagger\mathbf{x} = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{x}\,. \tag{4.9}$$

This is a block receiver: all entries of $\mathbf{s}$ are estimated simultaneously. If N_s is large, this is not very efficient.

Due to the commutativity of the convolution, we can also write $x[n] = s[n] * h[n]$, and hence

$$\mathbf{x} = \mathbf{S}\mathbf{h} \quad \Leftrightarrow \quad \begin{bmatrix} \boxed{x[0]} \\ x[1] \\ \vdots \\ \hline x[L-1] \\ \vdots \\ x[N_s-1] \\ \hline x[N_s] \\ \vdots \\ x[N_s+L-2] \end{bmatrix} = \left[\begin{array}{cccc} \boxed{s[0]} & & & \mathbf{0} \\ s[1] & s[0] & & \\ \vdots & \vdots & \ddots & \\ \hline s[L-1] & s[L-2] & \ddots & s[0] \\ \vdots & \vdots & \ddots & \vdots \\ s[N_s-1] & \vdots & \ddots & s[N_s-L] \\ \hline & s[N_s-1] & \ddots & s[N_s-L+1] \\ & & \ddots & \vdots \\ \mathbf{0} & & & s[N_s-1] \end{array}\right] \begin{bmatrix} \boxed{h[0]} \\ h[1] \\ \vdots \\ h[L-1] \end{bmatrix} \tag{4.10}$$

Now $\mathbf{S}$ has a Toeplitz structure, it has size $N_s + L - 1 \times L$. This expression can be used to estimate the channel coefficients in case we know the transmitted symbols (e.g., due to a training period), i.e., $\hat{\mathbf{h}} = \mathbf{S}^\dagger\mathbf{x}$, where $\mathbf{S}^\dagger = (\mathbf{S}^{\mathrm{H}}\mathbf{S})^{-1}\mathbf{S}^{\mathrm{H}}$. Note that $\mathbf{S}$ is tall: we need $N_s \geq 1$.

The "$\mathbf{0}$" blocks in $\mathbf{S}$ should be replaced by symbols in case the transmitter is not silent before/after the transmission of the training symbols (i.e., if there is no guard interval). Often these are unknown. To estimate $\mathbf{h}$ we should omit all rows in $\mathbf{S}$ that contain unknown entries of $s[n]$ (and also drop the corresponding entries in $\mathbf{x}$). This results in the model $\mathbf{x}' = \mathbf{S}'\mathbf{h}$, where $\mathbf{x}'$ and $\mathbf{S}'$ are the parts of $\mathbf{x}$ and $\mathbf{S}$ between the horizontal lines in (4.10). $\mathbf{S}'$ has size $N_s - L + 1 \times L$, and more samples are needed to make it have a left inverse: $N_s \geq 2L - 1$.

4.3.3 Oversampling

Since in (4.9) we aim to invert $\mathbf{H}$, we would like it to be tall. If it is not tall (e.g., due to lack of a guard interval), we can sometimes make it more tall by considering *oversampling*. In this context, oversampling means sampling faster than the symbol rate. Although it does not make sense to sample much faster than the Nyquist rate, often the Nyquist rate is higher than the symbol rate. E.g., in Fig. 4.5, we saw examples of the raised cosine pulse shape which is more compact in time than a sinc pulse, but therefore has excess bandwidth in frequency (controlled by the parameter α).

In the case of linear modulation, we can define

$$s(t) = \sum_{k=-\infty}^{\infty} s_k \, \delta(t - kT) \tag{4.11}$$

and define the modulation by a convolution with the pulse shape $g(t)$. For convenience, we normalize the symbol period to $T = 1$. Then the modulated signal is

$$u(t) = s(t) * g(t) = \sum_k g(t-k) s_k \, .$$

As before, let $x(t)$ be the baseband received signal. The impulse response of the channel from the source to the receiver, $h(t)$, is a convolution of the pulse shaping filter $g(t)$ and the actual channel response from $u(t)$ to $x(t)$. We can include any propagation delays and unknown synchronization delays in $h(t)$ as well. The data model is written compactly as the convolution $x(t) = h(t) * s(t)$.

Inserting (4.11) gives (with $T = 1$)

$$x(t) = \int h(t - t') \sum_k s_k \, \delta(t' - k) \, \mathrm{d}t' = \sum_k s_k h(t-k) \, . \tag{4.12}$$

This appears as a discrete-time convolution, even if $x(t)$ and $h(t)$ are continuous-time. An immediate consequence of the FIR assumption is that, at any given moment, at most L consecutive symbols play a role in $x(t)$. Indeed, for $t = n + \tau$, where $n \in \mathbb{Z}$ and $0 \leq \tau < 1$, the convolution (4.12) can be written as

$$x(n+\tau) = \sum_{k=0}^{L-1} h(k+\tau) s_{n-k} \, . \tag{4.13}$$

Suppose that we sample $x(t)$ at a rate of P times the symbol rate.[1] Then (4.13) shows that for all samples that fall between times n and $n+1$, the same L symbols play a role. If we define

$$\mathbf{x}[n] = \begin{bmatrix} x(n) \\ x(n + \frac{1}{P}) \\ \vdots \\ x(n + \frac{P-1}{P}) \end{bmatrix}, \qquad \mathbf{h}[k] = \begin{bmatrix} h(k) \\ h(k + \frac{1}{P}) \\ \vdots \\ h(k + \frac{P-1}{P}) \end{bmatrix} \tag{4.14}$$

[1] For the raised cosine pulses, we would select $P = 2$.

then we can write (4.13) as

$$\mathbf{x}[n] = \mathbf{h}[n] * s[n] = \sum_{k=0}^{L-1} \mathbf{h}[k] s_{n-k} \,. \tag{4.15}$$

This is the same as we had before, but now using sample *vectors* consisting of the P samples that fall within one sample period. Thus, (4.8) becomes

$$\mathbf{x} = \mathbf{H}\mathbf{s} \quad \Leftrightarrow \quad \begin{bmatrix} \boxed{\mathbf{x}[0]} \\ \mathbf{x}[1] \\ \mathbf{x}[2] \\ \vdots \\ \\ \vdots \\ \\ \vdots \\ \mathbf{x}[N-1] \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{h}[0]} & & & \mathbf{0} \\ \mathbf{h}[1] & \mathbf{h}[0] & & \\ \mathbf{h}[2] & \mathbf{h}[1] & \ddots & \\ \vdots & \mathbf{h}[2] & \ddots & \mathbf{h}[0] \\ \mathbf{h}[L-1] & \vdots & \ddots & \mathbf{h}[1] \\ & \mathbf{h}[L-1] & \ddots & \mathbf{h}[2] \\ & & \ddots & \vdots \\ \mathbf{0} & & & \mathbf{h}[L-1] \end{bmatrix} \begin{bmatrix} \boxed{s[0]} \\ s[1] \\ \vdots \\ s[N_s-1] \end{bmatrix} \tag{4.16}$$

Now, $\mathbf{H}$ is a *block*-Toeplitz matrix, where each block is a $P \times 1$ vector. We can estimate the symbols by inverting $\mathbf{H}$ as before, $\hat{\mathbf{s}} = \mathbf{H}^{\dagger}\mathbf{x} = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{x}$. Compared to the previous case, $\mathbf{H}$ is a factor P times more tall, which is usually good for inversion.

In fact, it would seem that if we take P very large, then we can make $\mathbf{H}$ as tall as we want. However, it does not make sense to sample (much) faster than the Nyquist rate. If we sample faster, $\mathbf{H}$ might be tall but at some point its columns will not become more orthogonal to each other. Thus, the condition number of $\mathbf{H}$ (see Sec. 5.4.6), an indicator of the amount of noise enhancement, converges to a constant. Said differently, by oversampling, we collect more signal energy, but we also collect more noise. A more detailed analysis is needed here, but we can expect that sampling faster than Nyquist will not give benefits.

If we define the $P \times 1$ vector

$$\mathbf{s}[n] = \begin{bmatrix} s_n \\ \vdots \\ s_n \end{bmatrix} = s_n \otimes \mathbf{1}_P$$

where $\mathbf{1}_P$ is a $P \times 1$ vector of ones, then an extension of (4.10) gives

$$\mathbf{x} = \mathcal{S}\mathbf{h} \quad \Leftrightarrow \quad \left[\begin{array}{c} \boxed{\mathbf{x}[0]} \\ \mathbf{x}[1] \\ \vdots \\ \hline \mathbf{x}[L-1] \\ \vdots \\ \mathbf{x}[N_s-1] \\ \hline \mathbf{x}[N_s] \\ \vdots \\ \mathbf{x}[N_s+L-2] \end{array}\right] = \left[\begin{array}{cccc} \boxed{\mathbf{s}[0]} & & & \mathbf{0} \\ \mathbf{s}[1] & \mathbf{s}[0] & & \\ \vdots & \vdots & \ddots & \\ \hline \mathbf{s}[L-1] & \mathbf{s}[L-2] & \ddots & \mathbf{s}[0] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}[N_s-1] & \vdots & \ddots & \mathbf{s}[N_s-L] \\ \hline & \mathbf{s}[N_s-1] & \ddots & \mathbf{s}[N_s-L+1] \\ & & \ddots & \vdots \\ \mathbf{0} & & & \mathbf{s}[N_s-1] \end{array}\right] \left[\begin{array}{c} \boxed{h[0]} \\ h[1] \\ \vdots \\ h[L-1] \end{array}\right] \tag{4.17}$$

where the symbol matrix $\mathcal{S}$ has size $(N_s + L - 1)P \times L$. Clearly, $\mathcal{S}$ has many repeated entries: we can write $\mathcal{S} = \mathbf{S} \otimes \mathbf{1}_P$.

Another way to stack the data is

$$\mathbf{X} = [\mathbf{x}[0] \quad \cdots \quad \mathbf{x}[N-1]] = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-1) \\ x(\frac{1}{P}) & x(1+\frac{1}{P}) & \cdot & \\ \vdots & & & \vdots \\ x(\frac{P-1}{P}) & \cdot & \cdots & x(N-1+\frac{P-1}{P}) \end{bmatrix}. \tag{4.18}$$

$\mathbf{X}$ has size $P \times N$; its nth column $\mathbf{x}[n]$ contains the P samples taken during the nth symbol period. Based on the FIR assumption, it follows that $\mathbf{X}$ has a factorization

$$\mathbf{X} = \mathbf{H}\mathbf{S} \tag{4.19}$$

where

$$\mathbf{H} = [\mathbf{h}[0] \quad \mathbf{h}[1] \quad \cdots \quad \mathbf{h}[L-1]] = \begin{bmatrix} h(0) & h(1) & \cdots & h(L-1) \\ h(\frac{1}{P}) & \cdot & & \cdot \\ \vdots & & & \vdots \\ h(\frac{P-1}{P}) & \cdot & \cdots & h(L-\frac{1}{P}) \end{bmatrix} : \quad P \times L \tag{4.20}$$

$$\mathbf{S} = \left[\begin{array}{ccc|ccc|ccc} s_0 & s_1 & \cdots & s_{L-1} & \cdots & s_{N_s-2} & s_{N_s-1} & & \mathbf{0} \\ & s_0 & \cdots & \cdots & \cdots & \cdots & s_{N_s-2} & s_{N_s-1} & \\ & & \ddots & s_1 & \cdots & \cdots & \cdots & \ddots & \ddots \\ \mathbf{0} & & & s_0 & \cdots & \cdots & s_{N_s-L} & \cdots & s_{N_s-2} \; s_{N_s-1} \end{array}\right] : \quad L \times N\,,$$

and $N = N_s + L - 1$. This factorization is readily derived from a transpose of (4.10). It is seen in the definition of $\mathbf{H}$ that we have "folded" samples of $h(t)$ into a matrix. As a result of

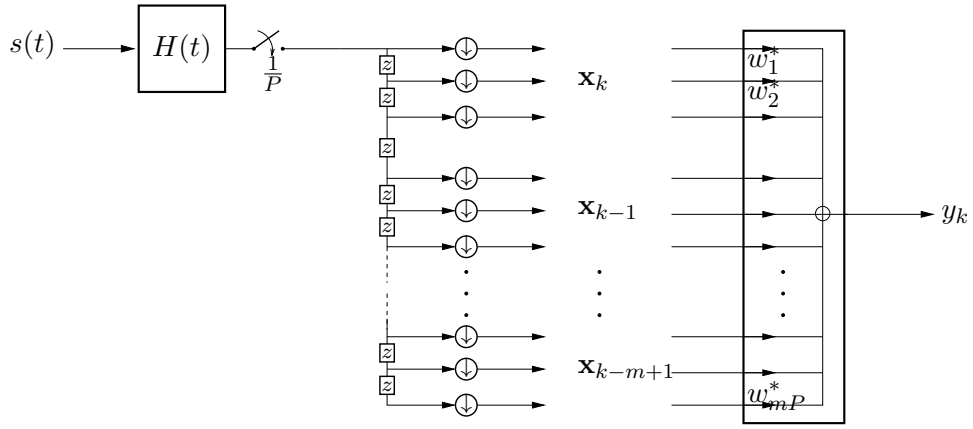


Figure 4.7. Equalizer

the different organization of the received data into a matrix, not a vector, we have avoided the Kronecker-repetition of the symbols in $\mathcal{S} = \mathbf{S} \otimes \mathbf{1}_P$ that was present in (4.16).

A problem with using this factorization to estimate $\mathbf{S}$ compared to the estimation based on (4.16) is that the Toeplitz structure of $\mathbf{S}$ is not enforced: in the presence of noise, $\hat{\mathbf{S}} = \mathbf{H}^\dagger \mathbf{X}$ is not Toeplitz. The redundancy in $\mathbf{S}$ is not exploited. Also, usually P is not very large (e.g., $P = 2$), and therefore, usually $\mathbf{H}$ is not tall.

4.3.4 Stacking and linear equalization

A linear equalizer in the present context can be written as a vector $\mathbf{w}$ which combines the rows of $\mathbf{X}$ to generate an output $\mathbf{y} = \mathbf{w}^{\mathrm{H}}\mathbf{X}$. If we consider the model $\mathbf{X} = \mathbf{HS}$, then we would require $\mathbf{w}^{\mathrm{H}}\mathbf{H} = [0, \cdots, 0, 1, 0, \cdots 0]$, so that equalization by $\mathbf{w}$ results in $\mathbf{y}$ equal to one of the rows of $\mathbf{S}$. In the noise-free model of (4.20), it doesn't really matter which row of $\mathbf{S}$ is reconstructed: we have L options, and they only differ by a delay. Since we only combine the P samples of $x(t)$ in one symbol period, the equalizer length is one symbol period.

Often, it is much better to filter over multiple sample periods. For a linear equalizer with a length of m symbol periods, we have to augment $\mathbf{X}$ with $m - 1$ horizontally shifted copies of itself:

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}[0] & \mathbf{x}[1] & \cdots & \mathbf{x}[N-m] \\ \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \cdots \\ \cdots & \cdots & \cdots & \mathbf{x}[N-2] \\ \mathbf{x}[m-1] & \cdots & \mathbf{x}[N-2] & \mathbf{x}[N-1] \end{bmatrix} : \quad mP \times N - m + 1 \,.$$

Each column of $\mathcal{X}$ is a regression vector: the memory of the filter. Using $\mathcal{X}$, a linear equalizer over m symbol periods can be written as $\mathbf{y} = \mathbf{w}^{\mathrm{H}}\mathcal{X}$, which combines mP snapshots: see Fig. 4.7.

The augmented data matrix $\mathcal{X}$ has a factorization

$$\mathcal{X} \;=\; \mathcal{H}\mathcal{S} \;=\; \begin{bmatrix} \mathbf{0} & \boxed{\mathbf{H}} \\ \ddots & \ddots \\ \boxed{\mathbf{H}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} s_{m-1} & \ddots & s_{N-2} & s_{N-1} \\ \ddots & \ddots & \ddots & s_{N-2} \\ s_{-L+2} & s_{-L+3} & \ddots & \ddots \\ s_{-L+1} & s_{-L+2} & \ddots & s_{N-L-m+1} \end{bmatrix} \tag{4.21}$$

where $\mathcal{H} = \mathcal{H}_m$ has size $mP \times L + m - 1$. $\mathcal{H}$ has a block-Hankel structure: it is constant along antidiagonals. $\mathcal{S}$ has the same structure as $\mathbf{S}$ in (4.20) but size $L + m - 1 \times N - m + 1$.

In this factorization, oversampling is not essential: we can also have $P = 1$. In that case, $\mathcal{H}$ will not be tall for any m so that perfect equalization for finite m is not possible. The reason is that we try to invert an FIR channel by an FIR filter! Generally, without oversampling we will need an ARMA filter to do this. A common problem is that the ARMA filter may easily become unstable (e.g., if the FIR filter is non-minimum phase: zeros outsize the unit circle).

4.3.5 Multiple antennas and multiple sources: FIR-MIMO model

Instead of oversampling, we may also consider the use of multiple antennas. In (4.14), we defined $\mathbf{x}[n]$ as a stack of P samples. We can also define $\mathbf{x}[n]$ to be a stack of M antenna outputs at time n. Likewise, $\mathbf{h}[k]$ in (4.14) simply becomes an arbitrary vector, e.g., the sum of array response vectors for multipath components arriving at a delay of k samples. The convolution model (4.15) remains unchanged. That means that all the subsequent steps, i.e., the stacking of $\mathbf{x}[n]$ into $\mathbf{X}$ and $\mathcal{X}$, and the resulting factorization models, are unchanged.

We can also consider oversampling together with multiple antennas. In that case, each vector $\mathbf{x}[n]$ will have size MP. The stacking and factorization models are unchanged, except that $\mathcal{H}_m$ will have size $mMP \times L + m - 1$. It is now much easier to have $\mathcal{H}_m$ tall.

In the I-MIMO model, we considered multiple sources. In a more general FIR-MIMO model, we can also do this. This models d sources arriving at an antenna array with M antennas, and convolutive channels, and oversampling by a factor P: see Fig. 4.8. This extension to FIR-MIMO is straightforward extension, although the notation becomes a bit cluttered. As simplifying assumption, we could start by assuming the d sources have the same symbol rate, so that the oversampling rate P has the same meaning. Assume that the ith source received on the jth antenna has an FIR channel $h_{ij}(t)$ of length L_j symbols. If for the ith source, we have a model $\mathcal{X} = \mathcal{H}_i\mathcal{S}_i$ as in (4.21), then with d sources we can write

$$\mathcal{X} = \mathcal{H}\mathcal{S}\,, \qquad \mathcal{H} = [\mathcal{H}_1, \cdots, \mathcal{H}_d]\,, \qquad \mathcal{S} = \begin{bmatrix} \mathcal{S}_1 \\ \vdots \\ \mathcal{S}_d \end{bmatrix}.$$

If the d have the same channel lengths L, then we could also rearrange this to arrive at a model as in (4.21), but now with block matrices $\mathbf{H}$ of size $MP \times dL$, and d-dimensional vectors $\mathbf{s}_k$ in $\mathcal{S}$. The m shifts of $\mathbf{H}$ to the left in $\mathcal{H}$ then are each over d positions.

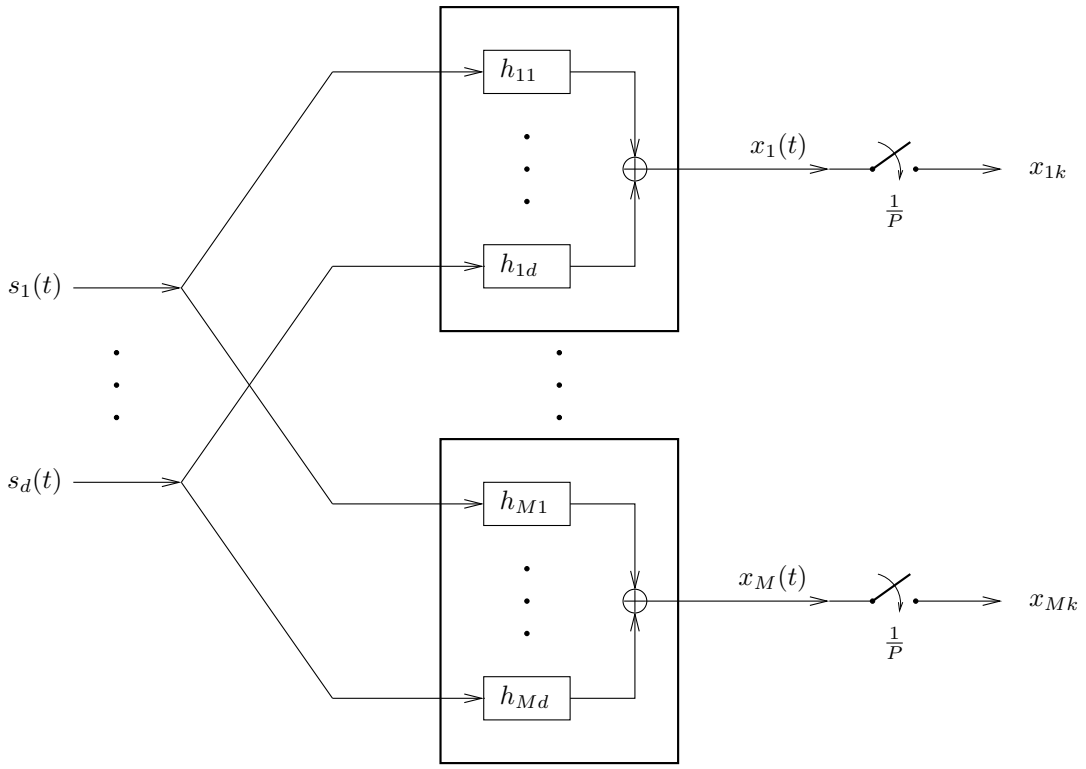


Figure 4.8. Multiuser convolutive channel model. Input signals $s_1(t), \cdots, s_d(t)$ are synchronized dirac-pulse sequences.

In this general case, $\mathcal{H}$ has size $mMP \times d(L+m-1)$. A *necessary* condition for space-time equalization (the output $\mathbf{y}$ is equal to a row of $\mathcal{S}$) is that $\mathcal{H}$ is tall, which gives minimal conditions on m in terms of M, P, d, L:

$$mMP \geq d(L+m-1) \qquad \Rightarrow \qquad m(MP-d) \geq d(L-1)$$

which implies

$$MP > d\,, \qquad m \geq \frac{d(L-1)}{MP-d}\,.$$

4.3.6 Connection to the parametric multipath model

For a *single* source, recall the multipath propagation model (4.2), valid for specular multipath with small cluster angle spread:

$$\mathbf{h}(t) = \sum_{i=1}^{r} \mathbf{a}(\theta_i)\beta_i g(t-\tau_i) \tag{4.22}$$

where $g(t)$ is the pulse shape function by which the signals are modulated. In this model, there are r distinct propagation paths, each parameterized by $(\theta_i, \tau_i, \beta_i)$, where θ_i is the direction-of-arrival (DOA), τ_i is the path delay, and $\beta_i \in \mathbb{C}$ is the complex path attenuation (fading). The vector-valued function $\mathbf{a}(\theta)$ is the array response vector for an array of M antenna elements to a signal from direction θ.

Suppose as before that $\mathbf{h}(t)$ has finite duration and is zero outside an interval $[0, L)$. Consequently, $g(t-\tau_i)$ has the same support for all τ_i. At this point, we can define a parametric "time manifold" vector function $\mathbf{g}(\tau)$, collecting LP samples of $g(t-\tau)$:

$$\mathbf{g}(\tau) = \begin{bmatrix} g(0-\tau) \\ g(\frac{1}{P}-\tau) \\ \vdots \\ g(L-\frac{1}{P}-\tau) \end{bmatrix}, \quad 0 \leq \tau \leq \max \tau_i\,.$$

If we also construct a vector $\mathbf{h}$ with samples of $\mathbf{h}(t)$,

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}(0) \\ \mathbf{h}(\frac{1}{P}) \\ \vdots \\ \mathbf{h}(L-\frac{1}{P}) \end{bmatrix}$$

then it is straightforward to verify that (4.22) gives

$$\mathbf{h} = \sum_{i=1}^{r} (\mathbf{g}_i \otimes \mathbf{a}_i)\beta_i = [\mathbf{g}_1 \otimes \mathbf{a}_1, \cdots, \mathbf{g}_r \otimes \mathbf{a}_r] \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_r \end{bmatrix}$$
$$\mathbf{g}_i = \mathbf{g}(\tau_i)\,, \quad \mathbf{a}_i = \mathbf{a}(\theta_i)\,.$$

Thus, the multiray channel vector is a weighted sum of vectors on the *space-time manifold* $\mathbf{g}(\tau) \otimes \mathbf{a}(\theta)$. Because of the Kronecker product, this is a vector in an LPM-dimensional space, with more distinctive characteristics than the M-dimensional $\mathbf{a}(\theta)$-vector in a scenario without delay spread. The connection of $\mathbf{h}$ with $\mathbf{H}$ as in (4.20) is that $\mathbf{h} = \mathrm{vec}(\mathbf{H})$, i.e., $\mathbf{h}$ is a stacking of all columns of $\mathbf{H}$ in a single vector.

We can define, much as before, parametric matrix functions

$$\mathbf{A}_\theta = [\mathbf{a}(\theta_1) \cdots \mathbf{a}(\theta_r)]\,, \quad \mathbf{G}_\tau = [\mathbf{g}(\tau_1) \cdots \mathbf{g}(\tau_r)]\,, \quad \mathbf{B} = \mathrm{diag}[\beta_1 \cdots \beta_r]$$

$$\mathbf{G}_\tau \circ \mathbf{A}_\theta := [\mathbf{g}_1 \otimes \mathbf{a}_1, \cdots, \mathbf{g}_r \otimes \mathbf{a}_r]$$

$(\mathbf{G}_\tau \circ \mathbf{A}_\theta)$ is a columnwise Kronecker product known as the *Khatri-Rao product*; its properties are discussed in Sec. 5.1.6. This gives $\mathbf{h} = (\mathbf{G}_\tau \circ \mathbf{A}_\theta)\mathbf{B}\mathbf{1}_r$.

Extending now to d sources, we get that the $MP \times dL$-sized matrix $\mathbf{H}$ in (4.20) can be rearranged into an $MPL \times d$ matrix

$$\mathbf{H}' = [\mathbf{h}_1, \cdots, \mathbf{h}_d] = (\mathbf{G}_\tau \circ \mathbf{A}_\theta)\mathbf{B}\mathbf{J}\,. \tag{4.23}$$

where $\mathbf{J}$ is the selection matrix defined in (4.6) that sums the rays into channel vectors. $(\mathbf{G}_\tau \circ \mathbf{A}_\theta)$ now plays the same role as $\mathbf{A}_\theta$ in Sec. 4.3.1. Each of its columns is a vector on the space-time manifold.

4.3.7 Summary

A summary of the noise-free data models developed so far is

$$\begin{array}{ll} \text{I-MIMO:} & \mathbf{X} = \mathbf{A}\mathbf{S}\,, \quad \mathbf{A} = \mathbf{A}_\theta\mathbf{B}\mathbf{J} \\ \text{FIR-MIMO:} & \mathcal{X} = \mathcal{H}\mathcal{S}\,, \quad \mathcal{H} \leftrightarrow \mathbf{H}' = (\mathbf{G}_\tau \circ \mathbf{A}_\theta)\mathbf{B}\mathbf{J} \end{array} \tag{4.24}$$

The first part of these model equations is generally valid for linear time-invariant channels, whereas the second part is a consequence of the adopted multiray model in the form of a *parametric channel model*.

Based on this model, the received data matrix $\mathbf{X}$ or $\mathcal{X}$ has several *structural properties*. In several combinations, these are often strong enough to allow to find the factors $\mathbf{A}$ (or $\mathbf{H}$) and $\mathbf{S}$ (or $\mathcal{S}$), even from knowledge of $\mathbf{X}$ or $\mathcal{X}$ alone. Very often, this will be in the form of a collection of beamformers (or space-time equalizers) $\{\mathbf{w}_i\}_1^d$ such that each beamformed output $\mathbf{w}_i^{\mathrm{H}}\mathbf{X} = \mathbf{s}_i$ is equal to one of the source signals, so that it must have the properties of that signal.

One of the most powerful "structures", on which most systems today rely to a large extent, is knowledge of part of the transmitted message (a training sequence), so that several columns of $\mathcal{S}$ are known. Along with the received signal $\mathcal{X}$, this allows to estimate $\mathbf{H}$. Very often, an unparameterized FIR model is assumed here. The algorithms are using a *temporal reference*. Algorithms that do not use this are called *blind*. Examples of this will be discussed in the coming chapters.

4.4 FREQUENCY-DOMAIN DATA MODELS

TBD: for wideband data, consider STFT, split into narrow subbands; this translates a wideband model into a set of narrowband models. These models can be processed independently, or (better) jointly.

Diagonalization of Hankel matrix (if circulant).

4.4.1 OFDM

TBD

4.5 APPLICATION: RADIO ASTRONOMY

4.5.1 Instrument design

New instruments are designed to achieve higher performance:

- *Higher resolution* implies longer baselines. We will see that this results in shorter integration time (due to earth rotation), and more (narrower) subbands.
- *Higher sensitivity* implies a larger number of antennas (typically grouped in stations), longer observing times, and better calibration (direction dependent).
- *Higher survey speed* requires a larger total bandwidth, a larger field-of-view, multiple beams, and direction dependent calibration.

This results in larger data sets, higher computational demands, and the need for better calibration and imaging algorithms.

The performance of a radio telescope depends on many parameters: the spatial resolution depends on the diameter of the instrument, the number of spatial samples on the number of STI's, the finite sample noise in a single STI depends on the number of samples N that we average in that STI, etc. How should these parameters be designed?

As it turns out, a lot depends on the non-stationarity introduced by the rotation of the earth, and constraints resulting from a requirement to satisfy the narrowband assumption. Essentially, we can average samples that differ by phase factors $\phi = exp(-j2\pi f\tau)$ if they satisfy:

- *Narrowband condition:* $f\tau$ should be approximately constant for $f \in (f_{\min}, f_{\max})$ and all geometric delays τ. If $W = f_{\max} - f_{\min}$ is the bandwidth, this translates to

$$W\tau \ll 1\,.$$

  For omni-directional antennas, the maximal geometric delay is $\tau_{\max} = D/c$, for source signals arriving in the same direction as the baseline. For directional antennas, D is scaled by $\sin(\theta)$,

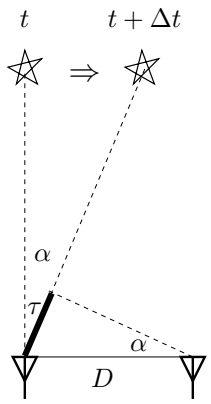


Figure 4.9. Due to earth rotation, the stars appear to move relative to the array.

where the maximal θ depends on the field of view.[2] Thus,

$$W \ll \frac{c}{D \sin\theta)} \tag{4.25}$$

This condition determines maximum processing bandwidth, as a function of the array diameter D. A signal with a larger total bandwidth should first be split into sufficiently narrow subbands ("channels").

– *Stationarity condition:* $f\tau$ should be approximately constant while the baselines move (due to earth rotation).
This determines the maximum processing time (STI), also a function of the array diameter, because for longer baselines, τ changes faster.

The latter condition is worked out as follows. The earth rotation rate is $\omega_e = \frac{2\pi}{1\text{day}} = 7.27 \cdot 10^{-5}$ rad/s. The sky appears to move with this angular speed. "A day" is taken here to be a sidereal day, i.e., taking into account that the earth makes an extra revolution over the course of a year; it is about 4 minutes shorter than 24 hours.

As shown in Fig. 4.9, over a small time period Δt, the earth rotates over a small angle $\alpha = \omega_e \, \Delta t$. If initially we had $\tau = 0$, we now have

$$\tau = \frac{D}{c} \sin(\alpha) \approx \frac{D}{c} \alpha$$

Thus, the rate of change of τ due to earth rotation is

$$\frac{\mathrm{d}\tau}{\mathrm{d}t} = \frac{D}{c} \frac{\mathrm{d}\alpha}{\mathrm{d}t} = \frac{D}{c} \omega_e$$

[2] This requires some elaboration.

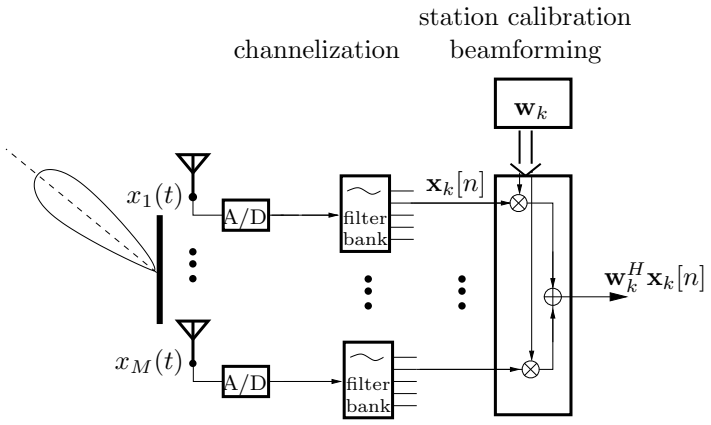


Figure 4.10. Station data processing

Integrating phases $\phi = e^{-j2\pi f\tau}$ coherently over a period T requires, at the largest frequency $f_{\max}$,

$$f_{\max} \cdot \frac{\mathrm{d}\tau}{\mathrm{d}t} T \ll 1 \quad \Rightarrow \quad T \ll \frac{c}{D f_{\max} \omega_e} = \frac{\lambda_{\min}}{D\,\omega_e} \tag{4.26}$$

This condition limits the STI. It depends on the observing frequency and the array diameter.

As example, we can look at the design of a first phase of the Square Kilometre Array (SKA), i.e., SKA1-Low: a low-frequency aperture array planned for around 2021-2023. Initial SKA1 design objectives were specified in 2013 [12], but several of these numbers were scaled down later. The architecture of the instrument is similar to that of LOFAR (commissioned in 2007), but at a larger scale.

Generally, for the lower frequencies, the idea is that simple non-steerable antennas are grouped into stations. The beamformed output of a station mimics that of a steerable dish. Next, the station output signals are combined (correlated) at a central location.

The initial SKA1 design called for 131,072 antennas, divided over 512 stations each with 256 dual-polarized antennas. For SKA1-Low, the frequency range is 50–300 MHz, sampled using log-periodic (somewhat directional) antennas. The maximal baseline was set at 100 km (later scaled down to 65 km), and each station has a diameter of 35 m.

The objective of station data processing is to produce beamformed outputs. An important consideration is that this will reduce the raw datarate by a factor equal to the number of antennas in a station. If budget permits, a station can produce multiple beamformed outputs, increasing the survey speed.

The number of coarse frequency channels that are needed is determined by the maximal bandwidth that satisfies the narrowband condition. This depends on the station diameter. For this

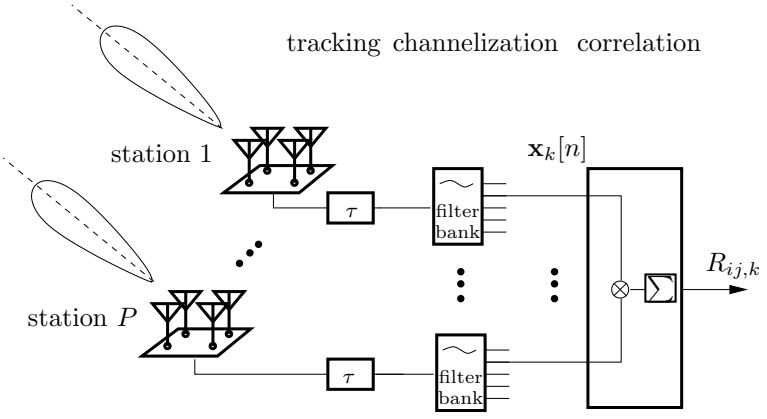


Figure 4.11. Central signal processing.

example, (4.25) gives, for a station with omnidirectional antennas,

$$D = 35 \text{ m} \qquad \Rightarrow \qquad W_{chan} \ll 8.6 \text{ MHz}.$$

In actuality, narrower subbands are proposed e.g. to facilitate station calibration.

Next, at a central location, the beamformed data from the stations are correlated, and averaged over short time intervals. With P stations, the correlation matrices have size $P \times P$. For the SKA1 example, $P = 512$.

The number of frequency channels is again determined by the narrowband condition, but this time depends on the instrument diameter. For example:

$$B = 100 \text{ km} \qquad \Rightarrow \qquad W_{chan} \ll 4.2 \text{ kHz}.$$

For the selected SKA1 design parameters, we could choose 250,000 channels with $W_{chan} = 1$ kHz. However, this result is valid if the array aperture B is filled with omnidirectional antennas. In reality, it is filled with stations that have beamformed outputs, and the beams limit the field of view to

$$\sin(\theta) = \frac{\lambda}{D}$$

Using (4.25) gives

$$W_{chan} \ll \frac{cD}{B\lambda} \ll \frac{D}{B} f_{\min}$$

Eq. (4.26) gives ?????????? TBD

$$T < 1200 \frac{D}{B}$$

With $B = 100$ km (the longest baseline) and $D = 35$ m (station diameter), the maximal integration time is $T < 0.4$ sec.

Thus, the output of the central signal processing step are complex correlation matrices of size 512×512 times 250,000 channels, each 0.4 sec, times 4 polarizations.

Depending on P, it can happen that the the correlator produces several times more output "data" than flows in: the input vectors of size P are transformed into matrices of size $P \times P$, and then averaged. If the STI T is too short, it is more efficient to work with the original data (**X**-matrices) rather than correlated data (**R**-matrices). The correlation matrices will be rank-deficient, indicasting redundancy.

"Baseline dependent averaging" (analyzed in [13]) exploits the fact that up to 90% of the baselines are short and can be integrated over longer times and larger bandwidths. This may significantly reduce the datarates.

The hardware bottleneck is perhaps not the required computational complexity (flops) but the required bandwidth to transport all the data. In particular, a seemingly trivial operation is this: data comes in from the various stations, each with a large number of subbands, and has to be rearranged such that for each subband the data for all stations is together. This does not involve computations but is a massive communication operation nonetheless.

Resolution So far, we have seen that B and D determine the number of subband channels and the integration length. They also determine the resolution. To see this, consider first a single station. If the antennas are placed sufficiently dense in a rectangular or circular aperture with diameter D, then we have seen in (2.21) that the beamwidth is $\theta_s \sim \lambda/D$. This determines the instantaneous field of view (FOV) of a station, which acts as a dish element in the entire array. If the array has a diameter B, then its beamwidth is $\theta \sim \lambda/B$. This beam partitions the FOV. Let

$$N := \frac{\theta_s}{\theta} = \frac{B}{D}$$

The field of view is covered by N beams in each dimension: we have a resolution of N "pixels" in each dimension. Thus, without superresolution techniques, we can expect to create an image of size $N \times N$ pixels. For the SKA1-Low example: $D = 35$ m, $B = 100$ km, resulting in $N \sim 3000$.

Interestingly, N is independent of frequency, but the FOV is frequency dependent. Thus, although the resulting image is size $N \times N$, the area on the sky which the image covers is frequency dependent. Lower frequencies (larger λ) cover a larger area.

Number of subband channels For the complete instrument, the longest baseline B determines the narrowband constraint. The maximum subband bandwidth is (taking into account the reduced field of view)

$$W_{chan} \ll \frac{D}{B} f_{\min} .$$

For example, we can set

$$W_{chan} = 0.1 \cdot \frac{D}{B} f_{\min} .$$

For a total bandwidth W_{tot}, the number of subband channels is proportional to

$$N_{chan} = \frac{W_{tot}}{W_{chan}} = 10 \cdot \frac{B}{D} \frac{W_{tot}}{f_{\min}}$$

Thus, in first approximation, $N \sim \frac{B}{D}$ determines each dimension of the image, and the number of frequency bins, i.e., the entire "image cube". The main drivers for complexity are thus P^2 and the ratios $\left(\frac{B}{D}\right)^3$ (the instrument to station diameter) and $\frac{W_{tot}}{f_{\min}}$ (the fractional bandwidth).

4.6 NOTES

Section 4.5 is based on Van der Veen e.a. [14].

Bibliography

[1] W.C. Jakes, ed., *Microwave Mobile Communications*. New York: John Wiley, 1974.

[2] F. Adachi and etal, "Cross correlation between the envelopes of 900 MHz signals received at a mobile radio base station site," *IEE Proceedings*, vol. 133, pp. 506–512, Oct. 1986.

[3] W.C.Y. Lee, "Effects on correlation between two mobile radio basestation antennas," *IEEE Tr. Comm.*, vol. 21, pp. 1214–1224, Nov. 1973.

[4] W.C.Y. Lee, *Mobile Communications Design Fundamentals*. New York: John Wiley, 1993.

[5] B. Sklar, "Rayleigh fading channels in mobile digital communication systems, part I: Characterization," *IEEE Communications Magazine*, vol. 35, pp. 90–100, July 1997.

[6] A.J. Paulraj and C.B. Papadias, "Space-time processing for wireless communications," *IEEE Signal Proc. Mag.*, vol. 14, pp. 49–83, November 1997.

[7] T. Trump and B. Ottersten, "Estimation of nominal direction of arrival and angular spread using an array of sensors," *Signal Processing*, vol. 50, pp. 57–69, April 1996.

[8] B. Ottersten, "Array processing for wireless communications," in *Proc. IEEE workshop on Stat. Signal Array Proc.*, (Corfu), pp. 466–473, June 1996.

[9] T.S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice Hall, 1996.

[10] K. Pahlavan and A.H. Levesque, "Wireless data communications," *Proc. IEEE*, vol. 82, pp. 1398–1430, September 1994.

[11] E.A. Lee and D.G. Messerschmitt, *Digital Communication*. Boston: Kluwer Publishers, 1988.

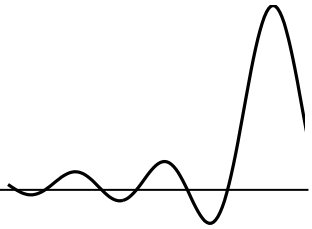
[12] P.E. Dewdney, W. Turner, R. Millenaar, R. McCool, J. Lazio, and T.J. Cornwell, "SKA1 system baseline design," Tech. Rep. SKA-TEL-SKO-DD-001, SKA Office, 2013.

[13] S.J. Wijnholds, A.G. Willis, and S. Salvini, "Baseline-dependent averaging in radio interferometry," *Monthly Notices of the Royal Astronomical Society*, vol. 476, pp. 2029–2039, Feb. 2018.

[14] A.J. van der Veen, S.J. Wijnholds, and A.M. Sardarabadi, "Signal processing for radio astronomy," in *Handbook of Signal Processing Systems, 3rd ed.*, Springer, November 2018. ISBN 978-3-319-91734-4.

Chapter 5

LINEAR ALGEBRA BACKGROUND



Contents

TBD: notation: N vs n vs d; check duplicate material (pseudo-inverse)

Throughout the book, several linear algebra concepts such as subspaces, QR factorizations, singular value decompositions and eigenvalue decompositions play an omni-important role. This chapter gives a brief review of the most important properties as needed here.[1]

An extensive tutorial to linear algebra in relation to signal processing can be found in Moon and Stirling [1]. Suitable reference books on advanced matrix algebra are Golub and Van Loan [2], and Horn and Johnson [3].

5.1 BASICS

5.1.1 Notation

A bold-face letter, such as $\mathbf{x}$, denotes a vector (usually a column vector, but occasionally a row vector). Matrices are written with capital bold letters. A matrix $\mathbf{A}$ has entries a_{ij}, and columns

[1] On a first reading, the more advanced topics should probably be skipped until needed in a future chapter, as indicated there.

$\mathbf{a}_j$, and we can write

$$\mathbf{A} = [a_{ij}] = [\mathbf{a}_1 \quad \mathbf{a}_2 \ \cdots \ \mathbf{a}_N] \,.$$

The $M \times M$ identity matrix is denoted by $\mathbf{I}_M$, or $\mathbf{I}$ for short. A matrix or vector with only zero entries is denoted by $\mathbf{0}_{M\times N}$, or $\mathbf{0}$ for short.

Complex conjugate is denoted by an overbar, the transpose of a matrix is denoted by $\mathbf{A}^{\mathrm{T}} = [a_{ji}]$. For complex matrices, the complex conjugate (= hermitian) transpose is $\mathbf{A}^{\mathrm{H}} := \overline{\mathbf{A}}^{\mathrm{T}}$.

5.1.2 Matrix products

A matrix $\mathbf{A}$ and a vector $\mathbf{x}$ can be multiplied if their sizes match: the number of columns of $\mathbf{A}$ should equal the number of entries in $\mathbf{x}$. In that case

$$\mathbf{y} = \mathbf{A}\mathbf{x} = [\mathbf{a}_1 \quad \mathbf{a}_2 \ \cdots \ \mathbf{a}_N] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_N x_N$$

In general,

$$y_i = \sum_j A_{ij} x_j \,, \qquad i = 1, \cdots, M$$

Likewise, two matrices $\mathbf{A}$ and $\mathbf{B}$ can be multiplied if their "inner dimensions" match (i.e., the number of columns of $\mathbf{A}$ equals the number of rows of $\mathbf{B}$. In that case

$$\mathbf{C} = \mathbf{A}\mathbf{B} \quad \Leftrightarrow \quad C_{ij} = \sum_k A_{ik} B_{kj}$$

5.1.3 Inner product and norms

The inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$ of equal size is

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{b}^{\mathrm{H}} \mathbf{a} \,.$$

The inner product is used to define a vector norm $\|\mathbf{a}\|$ by

$$\|\mathbf{a}\|^2 = \mathbf{a}^{\mathrm{H}} \mathbf{a} = \sum_i |a_i|^2$$

This satisfies the required properties of a norm, i.e.,

$$\begin{aligned} \|\mathbf{a}\| &\geq 0 \\ \|\mathbf{a}\| &= 0 \quad \Leftrightarrow \quad \mathbf{a} = \mathbf{0} \,. \end{aligned}$$

It also satisfies the triangle inequality:

$$\mathbf{c} = \mathbf{a} + \mathbf{b} \quad \Rightarrow \quad \|\mathbf{c}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|$$

with equality only if $\mathbf{a}$ is parallel to $\mathbf{b}$.

The inner product satisfies the inequality

$$|\mathbf{b}^{\mathrm{H}}\mathbf{a}| \leq \|\mathbf{a}\| \, \|\mathbf{b}\|$$

and this allows to define the angle θ between two vectors via

$$\mathbf{b}^{\mathrm{H}}\mathbf{a} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos(\theta) \, .$$

5.1.4 Matrix norms

The *induced matrix 2-norm* of a matrix $\mathbf{A}$ (also called the spectral norm, or the operator norm) is

$$\| \, \mathbf{A} \, \| := \max_{\mathbf{x}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$$

It represents the largest magnification of a vector $\mathbf{x}$ that can be obtained by applying $\mathbf{A}$ to it. Another expression for this is

$$\| \, \mathbf{A} \, \|^2 = \max_{\mathbf{x}} \frac{\mathbf{x}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{x}}{\mathbf{x}^{\mathrm{H}}\mathbf{x}}$$

The *Frobenius norm* of $\mathbf{A}$ represents the energy contained in its entries:

$$\| \, \mathbf{A} \, \|_{\mathrm{F}} = (\sum |A_{ij}|^2)^{1/2}$$

5.1.5 Trace

For a square matrix $\mathbf{A}$, the trace of $\mathbf{A}$ is the sum of its diagonal entries:

$$\mathrm{tr}[\mathbf{A}] = \sum_i A_{ii}$$

It has several properties. One is

$$\mathrm{tr}[\mathbf{A}\mathbf{B}] = \mathrm{tr}[\mathbf{B}\mathbf{A}]$$

The Frobenius norm of $\mathbf{A}$ is

$$\|\mathbf{A}\|_{\mathrm{F}}^2 = \sum_{ij} |A_{ij}|^2 = \mathrm{tr}[\mathbf{A}^{\mathrm{H}}\mathbf{A}]$$

since the ith diagonal entry of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is given by $\sum_j A_{ji}^* A_{ji}$.

5.1.6 Kronecker products and the vec operator

For a matrix, $\mathrm{vec}(\cdot)$ denotes the stacking of the columns of a matrix into a vector. Conversely, $\mathrm{vec}^{-1}(\cdot)$ is the inverse of $\mathrm{vec}(\cdot)$: the construction of a matrix out of a vector. This is not unambiguous: the matrix dimensions must be made clear from the context.

For a vector, $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix with the entries of $\mathbf{v}$ on the diagonal. For a matrix, $\mathrm{vecdiag}(\mathbf{A})$ is a vector consisting of the diagonal entries of $\mathbf{A}$.

For two matrices $\mathbf{A}$ and $\mathbf{B}$, the Kronecker product is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1N}\mathbf{B} \\ \vdots & & \vdots \\ a_{M1}\mathbf{B} & \cdots & a_{MN}\mathbf{B} \end{bmatrix},$$

and the Schur-Hadamard product as

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1N}b_{1N} \\ \vdots & & \vdots \\ a_{M1}b_{M1} & \cdots & a_{MN}b_{MN} \end{bmatrix},$$

provided $\mathbf{A}$ and $\mathbf{B}$ have the same size.

A rank-one matrix has the form $\mathbf{a}\mathbf{b}^{\mathrm{T}}$. It can be written using the Kronecker product as

$$\mathrm{vec}(\mathbf{a}\mathbf{b}^{\mathrm{T}}) = \mathbf{b} \otimes \mathbf{a},$$

and similarly, for complex vectors,

$$\mathrm{vec}(\mathbf{a}\mathbf{b}^{\mathrm{H}}) = \overline{\mathbf{b}} \otimes \mathbf{a}. \tag{5.1}$$

This can be readily shown by writing the products in full:

$$\mathbf{a}\mathbf{b}^{\mathrm{H}} = \begin{bmatrix} a_1b_1 & a_1b_2 & \cdots & a_1b_N \\ a_2b_1 & a_2b_2 & \cdots & a_2b_N \\ \vdots & \vdots & & \vdots \\ a_Mb_1 & a_Mb_2 & \cdots & a_Mb_N \end{bmatrix}, \qquad \overline{\mathbf{b}} \otimes \mathbf{a} = \left[\begin{array}{c} b_1a_1 \\ b_1a_2 \\ \vdots \\ b_1a_M \\ \hline b_2a_1 \\ b_2a_2 \\ \vdots \\ b_2a_M \\ \hline \vdots \\ \hline b_Na_1 \\ b_Na_2 \\ \vdots \\ b_Na_M \end{array}\right]$$

We will also often use the more general matrix identity

$$\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})\mathrm{vec}(\mathbf{B})\,.$$

This can be proven using (5.1), by writing $\mathbf{ABC}$ as a sum of rank-one components of the form $(\mathbf{a}_i\mathbf{c}_j^{\mathrm{T}})b_{ij}$, where $\mathbf{a}_i$ is the ith column of $\mathbf{A}$, and $\mathbf{c}_j^{\mathrm{T}}$ the jth row of $\mathbf{C}$.

$\circ$ denotes the Khatri-Rao product, i.e., a column-wise Kronecker product:

$$\mathbf{A} \circ \mathbf{B} := [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots]\,.$$

This forms a submatrix of $\mathbf{A} \otimes \mathbf{B}$.

Notable properties of Kronecker products are (for matrices and vectors of compatible sizes):

$$\begin{aligned}
\mathrm{vec}(\mathbf{ab}^{\mathrm{H}}) &= \mathbf{b}^* \otimes \mathbf{a} && (5.2)\\
(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= \mathbf{AC} \otimes \mathbf{BD} && (5.3)\\
(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \circ \mathbf{D}) &= \mathbf{AC} \circ \mathbf{BD} && (5.4)\\
(\mathbf{A} \circ \mathbf{B})^{\mathrm{H}}(\mathbf{C} \circ \mathbf{D}) &= \mathbf{A}^{\mathrm{H}}\mathbf{C} \odot \mathbf{B}^{\mathrm{H}}\mathbf{D} && (5.5)\\
(\mathbf{a}^{\mathrm{H}} \otimes \mathbf{B})\mathbf{C} &= \mathbf{a}^{\mathrm{H}} \otimes \mathbf{BC} && (5.6)\\
\mathrm{vec}(\mathbf{ABC}) &= (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})\mathrm{vec}(\mathbf{B}) && (5.7)\\
\mathrm{vec}(\mathbf{A}\,\mathrm{diag}(\mathbf{b})\,\mathbf{C}) &= (\mathbf{C}^{\mathrm{T}} \circ \mathbf{A})\mathbf{b} && (5.8)\\
[\mathbf{a} \otimes \mathbf{b}][\mathbf{c} \otimes \mathbf{d}]^{\mathrm{H}} &= \mathbf{ac}^{\mathrm{H}} \otimes \mathbf{bd}^{\mathrm{H}} && (5.9)\\
&= \mathbf{a} \otimes \mathbf{bc}^{\mathrm{H}} \otimes \mathbf{d}^{\mathrm{H}} && (5.10)\\
&= \mathbf{c}^{\mathrm{H}} \otimes \mathbf{ad}^{\mathrm{H}} \otimes \mathbf{b}\\
\mathrm{tr}(\mathbf{AB}) &= \mathrm{vec}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}})\mathrm{vec}(\mathbf{B}) \;=\; \mathrm{vec}^{\mathrm{H}}(\mathbf{A}^{\mathrm{H}})\mathrm{vec}(\mathbf{B}) && (5.11)\\
& && (5.12)\\
\mathrm{tr}(\mathbf{ABCD}) &= \mathrm{vec}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}})(\mathbf{D}^{\mathrm{T}} \otimes \mathbf{B})\mathrm{vec}(\mathbf{C}) \;=\; \mathrm{vec}^{\mathrm{H}}(\mathbf{A}^{\mathrm{H}})(\mathbf{D}^{\mathrm{T}} \otimes \mathbf{B})\mathrm{vec}(\mathbf{C}) && (5.13)\\
\mathrm{tr}(\mathbf{A} \otimes \mathbf{B}) &= \mathrm{tr}(\mathbf{A})\mathrm{tr}(\mathbf{B}) && (5.14)
\end{aligned}$$

Let $\mathbf{A}$ be a $P \times Q$ matrix. Clearly, $\mathrm{vec}(\mathbf{A})$ and $\mathrm{vec}(\mathbf{A}^{\mathrm{T}})$ contain the same elements, but organized differently: they are related by a permutation matrix. This matrix is called the commutation matrix, and denoted by $\mathbf{K}_{P,Q}$:

$$\mathrm{vec}(\mathbf{A}^T) = \mathbf{K}_{P,Q}\mathrm{vec}(\mathbf{A})\,.$$

For any $P \times Q$ matrix $\mathbf{A}$ and $M \times N$ matrix $\mathbf{B}$ we have

$$\begin{aligned}
(\mathbf{A} \otimes \mathbf{B})\mathbf{K}_{Q,N} &= \mathbf{K}_{P,M}(\mathbf{B} \otimes \mathbf{A}) && (5.15)\\
(\mathbf{A} \circ \mathbf{B}) &= \mathbf{K}_{P,M}(\mathbf{B} \circ \mathbf{A}), && (5.16)
\end{aligned}$$

where $Q = N$ for (5.16).

Pointwise multiplication by $\odot$ generally does not preserve rank: the rank of $\mathbf{A} \odot \mathbf{B}$ can be higher than that of $\mathbf{A}$ or $\mathbf{B}$. For example, if $\mathbf{B} = \mathbf{11}^{\mathrm{T}}$, then $\mathbf{B}$ has rank 1, while $\mathbf{A} \odot \mathbf{B} = \mathbf{A}$ has rank

equal to $\mathbf{A}$, possibly larger than 1. If $\mathbf{A} = \mathbf{a}\mathbf{a}^{\mathrm{H}}$ has rank 1, and $\mathbf{B} = \mathbf{I}$, then $\mathbf{A} \odot \mathbf{B} = \mathrm{diag}(\mathbf{A})$ can have full rank while $\mathbf{A}$ has rank 1. An exception is that $\mathbf{a}\mathbf{b}^{\mathrm{H}} \odot \mathbf{c}\mathbf{d}^{\mathrm{H}}$ does have rank 1. This is shown by considering

$$\mathbf{a}\mathbf{b}^{\mathrm{H}} \otimes \mathbf{c}\mathbf{d}^{\mathrm{H}} = (\mathbf{a} \otimes \mathbf{c})(\mathbf{b} \otimes \mathbf{d})^{\mathrm{H}},$$

(which is rank 1) and noting that $\mathbf{a}\mathbf{b}^{\mathrm{H}} \odot \mathbf{c}\mathbf{d}^{\mathrm{H}}$ is a submatrix of $\mathbf{a}\mathbf{b}^{\mathrm{H}} \otimes \mathbf{c}\mathbf{d}^{\mathrm{H}}$.

5.2 SUBSPACES

The space $\mathcal{H}$ spanned by a collection of vectors $\{\mathbf{x}_k\}$

$$\mathcal{H} := \{\alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n \mid \alpha_i \in \mathbb{C}\,,\ \forall i\}$$

is called a *linear subspace*.

Important examples of subspaces are

Range (column span) of $\mathbf{A}$:	$\mathrm{ran}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} :\ \mathbf{x} \in \mathbb{C}^N\}$
Kernel (nullspace) of $\mathbf{A}$:	$\ker(\mathbf{A}) = \{\mathbf{x} \in \mathbb{C}^N :\ \mathbf{A}\mathbf{x} = 0\}$

One routinely shows that

$$\begin{array}{lcll} \mathrm{ran}(\mathbf{A}) & \oplus & \ker(\mathbf{A}^{\mathrm{H}}) & = & \mathbb{C}^M \\ \mathrm{ran}(\mathbf{A}^{\mathrm{H}}) & \oplus & \ker(\mathbf{A}) & = & \mathbb{C}^N \end{array}$$

Here, $\mathcal{H}_1 \oplus \mathcal{H}_2$ denotes the direct sum of two linearly independent subspaces, namely $\{\mathbf{x}_1 + \mathbf{x}_2 \mid \mathbf{x}_1 \in \mathcal{H}_1,\ \mathbf{x}_2 \in \mathcal{H}_2\}$.

5.2.1 Linear independence

A collection of vectors $\{\mathbf{x}_i\}$ is called linearly independent if

$$\alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n = 0 \quad \Leftrightarrow \quad \alpha_1 = \cdots = \alpha_n = 0\,.$$

5.2.2 Basis

An independent collection of vectors $\{\mathbf{x}_i\}$ that together span a subspace is called a *basis* for that subspace.

If the vectors are orthogonal ($\mathbf{x}_i^{\mathrm{H}}\mathbf{x}_j = 0,\ i \neq j$), it is an *orthogonal basis*.

If moreover, the vectors have norm 1: $\|\mathbf{x}_i\| = 1$, the basis is called orthonormal.

The basis for a subspace is not unique.

Often, we stack the basis vectors $\mathbf{x}_i$ into a matrix $\mathbf{X}$ and, with abuse of terminology, call that matrix a basis.

5.2.3 Rank

The *rank* of a matrix $\mathbf{X}$ is the number of independent columns (or rows) of $\mathbf{X}$.

A prototype rank-1 matrix is $\mathbf{X} = \mathbf{a}\mathbf{b}^{\mathrm{H}}$, a prototype rank-2 matrix is $\mathbf{X} = \mathbf{a}_1\mathbf{b}_1^{\mathrm{H}} + \mathbf{a}_2\mathbf{b}_2^{\mathrm{H}}$, etcetera: a rank-$d$ matrix is

$$\mathbf{X} = \mathbf{a}\mathbf{a}_1\mathbf{b}_1^{\mathrm{H}} + \mathbf{a}_2\mathbf{b}_2^{\mathrm{H}} + \cdots + \mathbf{a}_d\mathbf{b}_d^{\mathrm{H}} = \mathbf{A}\mathbf{B}^{\mathrm{H}}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d]$ and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_d]$. Thus, a matrix factorization $\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{H}}$ where the "inner dimension" is d shows that the rank of $\mathbf{X}$ is (at most) d. This decomposition is called a dyadic decomposition; it is not unique.

The rank cannot be larger than the smallest size of the matrix (when it is equal, the matrix is full rank, otherwise it is rank deficient). A tall matrix is said to have full column rank if the rank is equal to the number of columns: the columns are independent. Similary, a wide matrix has full row rank if its rank equals the number of rows.

5.2.4 Unitary matrix and isometry

A real (square) matrix $\mathbf{U}$ is called an orthogonal matrix if $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$, and $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}$.

Likewise, a complex matrix $\mathbf{U}$ is unitary if $\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{I}$, $\mathbf{U}\mathbf{U}^{\mathrm{H}} = \mathbf{I}$.

A unitary matrix looks like a rotation and/or a reflection. Its norm is $\| \mathbf{U} \| = 1$, and its columns are orthonormal.

A tall matrix $\hat{\mathbf{U}}$ is called an isometry if $\hat{\mathbf{U}}^{\mathrm{H}}\hat{\mathbf{U}} = \mathbf{I}$. Its columns are an orthonormal basis of a subspace (not the complete space), its norm is $\| \hat{\mathbf{U}} \| = 1$. There is an orthogonal complement $\hat{\mathbf{U}}^{\perp}$ of $\hat{\mathbf{U}}$ such that $\mathbf{U} = [\hat{\mathbf{U}} \;\; \hat{\mathbf{U}}^{\perp}]$ is square and unitary.

5.2.5 Projection

A square matrix $\mathbf{P}$ is a projection if $\mathbf{P}\mathbf{P} = \mathbf{P}$. It is an orthogonal projection if also $\mathbf{P}^{\mathrm{H}} = \mathbf{P}$.

The norm of an orthogonal projection is $\| \mathbf{P} \| = 1$. For an isometry $\hat{\mathbf{U}}$, the matrix $\mathbf{P} = \hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathrm{H}}$ is an orthogonal projection onto the space spanned by the columns of $\hat{\mathbf{U}}$. This is the general form of an orthogonal projection.

Suppose $\mathbf{U} = [\underbrace{\hat{\mathbf{U}}}_{d} \quad \underbrace{\hat{\mathbf{U}}^{\perp}}_{M-d}]$ is unitary. Then,

1. from $\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{I}_m$:

$$\hat{\mathbf{U}}^{\mathrm{H}}\hat{\mathbf{U}} = \mathbf{I}_d\,, \qquad \hat{\mathbf{U}}^{\mathrm{H}}\hat{\mathbf{U}}^{\perp} = 0\,, \qquad (\hat{\mathbf{U}}^{\perp})^{\mathrm{H}}\hat{\mathbf{U}}^{\perp} = \mathbf{I}_{M-d}\,.$$

2. from $\mathbf{U}\mathbf{U}^{\mathrm{H}} = \mathbf{I}_M$:

$$\hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathrm{H}} + \hat{\mathbf{U}}^{\perp}(\hat{\mathbf{U}}^{\perp})^{\mathrm{H}} = \mathbf{I}_M\,, \qquad \hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathrm{H}} = \mathbf{P}_c\,, \qquad \hat{\mathbf{U}}^{\perp}(\hat{\mathbf{U}}^{\perp})^{\mathrm{H}} = \mathbf{P}_c^{\perp} \;=\; \mathbf{I} - \mathbf{P}_c$$

This shows that any vector $\mathbf{x} \in \mathbb{C}^M$ can be decomposed into $\mathbf{x} = \hat{\mathbf{x}} + \hat{\mathbf{x}}^\perp$, where $\hat{\mathbf{x}} \perp \hat{\mathbf{x}}^\perp$,

$$\hat{\mathbf{x}} = \mathbf{P}_c \mathbf{x} \in \mathrm{ran}(\hat{\mathbf{U}}), \qquad \hat{\mathbf{x}}^\perp = \mathbf{P}_c^\perp \mathbf{x} \in \mathrm{ran}(\hat{\mathbf{U}}^\perp)$$

The matrices $\hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathrm{H}} = \mathbf{P}_c$ and $\hat{\mathbf{U}}^\perp(\hat{\mathbf{U}}^\perp)^{\mathrm{H}} = \mathbf{P}_c^\perp$ are the orthogonal projectors onto the column span of $\mathbf{X}$ and its orthogonal complement in $\mathbb{C}^M$ respectively.

Similarly, we can find a matrix $\hat{\mathbf{V}}^{\mathrm{H}}$ whose rows span the row span of $\mathbf{X}$, and augment it with a matrix $\hat{\mathbf{V}}^\perp$ to a unitary matrix $\mathbf{V}$:

$$\mathbf{V} = \begin{matrix} \\ N \updownarrow \end{matrix} \begin{matrix} \overset{d}{\leftrightarrow} & \overset{N-d}{\leftrightarrow} \\ \end{matrix} \left[\begin{array}{cc} \hat{\mathbf{V}} & \hat{\mathbf{V}}^\perp \end{array} \right].$$

The matrices $\hat{\mathbf{V}}\hat{\mathbf{V}}^{\mathrm{H}} = \mathbf{P}_r$ and $\hat{\mathbf{V}}^\perp(\hat{\mathbf{V}}^\perp)^{\mathrm{H}} = \mathbf{P}_r^\perp$ are orthogonal projectors onto the original subspaces in $\mathbb{C}^N$ spanned by the columns of $\hat{\mathbf{V}}$ and $\hat{\mathbf{V}}^\perp$, respectively. The columns of $\hat{\mathbf{V}}^\perp$ span the kernel (or nullspace) of $\mathbf{X}$, i.e., the space of vectors $\mathbf{a}$ for which $\mathbf{X}\mathbf{a} = \mathbf{0}$.

5.3 THE QR FACTORIZATION

Let $\mathbf{X}: \ N \times N$ be a square matrix of full rank. Then there is a decomposition $\mathbf{X} = \mathbf{QR}$,

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_N \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ 0 & r_{22} & \cdots & r_{2N} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & r_{NN} \end{bmatrix}$$

The interpretation is that $\mathbf{q}_1$ is a normalized vector with the same direction as $\mathbf{x}_1$, similarly $[\mathbf{q}_1 \ \ \mathbf{q}_2]$ is an isometry spanning the same space as $[\mathbf{x}_1 \ \ \mathbf{x}_2]$, etcetera.

If $\mathbf{X}: \ M \times N$ is a tall matrix ($M \geq N$), then there is a decomposition

$$\mathbf{X} = \mathbf{QR} = [\hat{\mathbf{Q}} \quad \hat{\mathbf{Q}}^\perp] \begin{bmatrix} \hat{\mathbf{R}} \\ \mathbf{0} \end{bmatrix} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$$

Here, $\mathbf{Q}$ is a unitary matrix, $\hat{\mathbf{R}}$ is upper triangular and square. $\mathbf{R}$ is upper triangular with $M - N$ zero rows added. $\mathbf{X} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ is called an "economy-size" QR.

If $\hat{\mathbf{R}}$ is nonsingular (all entries on the main diagonal are invertible), then $d = N$, the columns of $\hat{\mathbf{Q}}$ form a basis of the column span of $\mathbf{X}$, and $\mathbf{P}_c = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathrm{H}}$. If $\hat{\mathbf{R}}$ is rank-deficient, then this is not true: the column span of $\hat{\mathbf{Q}}$ is too large. However, the QR factorization can be used as a start in the estimation of an orthogonal basis for the column span of $\mathbf{X}$. Although this has sometimes been attempted, it is numerically not very robust to use the QR directly to estimate the rank of a matrix. (Modifications such as a "rank-revealing QR" do exist.)

Likewise, for a "wide" matrix ($M \leq N$) we can define an RQ factorization

$$\mathbf{X} = \mathbf{RQ} = [\hat{\mathbf{R}} \quad \mathbf{0}] \begin{bmatrix} \hat{\mathbf{Q}} \\ \hat{\mathbf{Q}}^{\perp} \end{bmatrix}$$

(for different $\mathbf{Q}$ and $\mathbf{R}$). Now, $\mathbf{X}$ and $\hat{\mathbf{R}}$ have the same singular values and *left* singular vectors.

5.4 THE SINGULAR VALUE DECOMPOSITION (SVD)

For a given (complex) matrix $\mathbf{X}$ of size $m \times n$, where we assume $m > n$, the SVD is defined by

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}, \qquad \mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_m], \quad \mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} \end{bmatrix}, \quad \mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_n] \tag{5.17}$$

where $\mathbf{U} : m \times m$ and $\mathbf{V} : n \times n$ are orthogonal matrices, and $\mathbf{\Sigma}$ is a diagonal matrix of size $m \times n$ containing the singular values in descending order. These are non-negative (real) numbers. Note that $\mathbf{\Sigma}$ has a block of $m - n$ "zero" rows at the bottom.

Any matrix $\mathbf{X}$ has this decomposition [2]. If $\mathbf{X}$ is real-valued, then the factors are also real-valued. Algorithms to compute the SVD are iterative and of complexity $O(m^2 n)$, but with a large scale factor: think about $20m^2 n$. This is much more complex than a QR factorization. In fact, a QR factorization is usually applied as a preprocessing step to compute the SVD.

Since $\mathbf{\Sigma}$ has $m - n$ rows with zeros, and often m can be very large, it is inefficient to keep so many columns of $\mathbf{U}$ that are anyway not used (they are multiplied by the zeros). Thus, we can also define the "economy-size" SVD, where

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}, \qquad \mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_n], \quad \mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}, \quad \mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_n] \tag{5.18}$$

where $\mathbf{U} : m \times n$ is a tall matrix of the same size as $\mathbf{X}$, and $\mathbf{V}$ and $\mathbf{\Sigma}$ are $n \times n$. Note that $\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{I}$ but $\mathbf{U}\mathbf{U}^{\mathrm{H}} \neq \mathbf{I}$ because it is an $m \times m$ matrix of rank n.

By using these properties, we can readily show:

$$\begin{aligned} \mathbf{\Sigma} &= \mathbf{U}^{\mathrm{H}}\mathbf{X}\mathbf{V} \\ \mathbf{U}\mathbf{\Sigma} &= \mathbf{X}\mathbf{V} \\ \mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} &= \mathbf{U}^{\mathrm{H}}\mathbf{X}\,. \end{aligned}$$

The singular values give important information on the dominant directions in the column span and row span of $\mathbf{X}$. This is seen by writing out the matrix equations, which gives the dyadic decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} = \mathbf{u}_1\sigma_1\mathbf{v}_1^{\mathrm{H}} + \mathbf{u}_2\sigma_2\mathbf{v}_2^{\mathrm{H}} + \cdots + \mathbf{u}_n\sigma_n\mathbf{v}_n^{\mathrm{H}}\,. \tag{5.19}$$

Each term of the form $\mathbf{u}_k\sigma_k\mathbf{v}_k^{\mathrm{H}}$ is a rank-1 matrix. If σ_1 is large, then the corresponding component $\mathbf{u}_1\mathbf{v}_1^{\mathrm{H}}$ is dominantly present in $\mathbf{X}$, and $\mathbf{u}_1$ is the dominant direction in the column span of $\mathbf{X}$. In fact, $\mathbf{u}_1\sigma_1\mathbf{v}_1^{\mathrm{H}}$ is the best rank-1 approximation of $\mathbf{X}$ (in the Least Squares sense).

If σ_n is zero, then one dimension is missing in the matrix: it is rank deficient by order 1. In general, if $\mathbf{X}$ is of rank d, then only d singular values $\sigma_1, \cdots, \sigma_d$ are nonzero. This is also seen from (5.19) because it will then consist of the sum of d rank-1 components. The best rank-d approximation of a matrix $\mathbf{X}$ is obtained by setting $\sigma_{d+1} = \cdots = \sigma_n = 0$.

Suppose $\mathbf{X}$ has rank d, with $d < n$. Similar to the economy-size SVD, we can write

$$\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\mathrm{H}}\,, \qquad \hat{\mathbf{U}} = [\mathbf{u}_1\ \cdots\ \mathbf{u}_d]\,, \quad \hat{\mathbf{\Sigma}} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_d \end{bmatrix}, \quad \hat{\mathbf{V}} = [\mathbf{v}_1\ \cdots\ \mathbf{v}_d] \tag{5.20}$$

where $\hat{\mathbf{\Sigma}}$ now has size $d \times d$, and only the nonzero singular values are kept.

Since $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ are "tall" isometric matrices, we can complement them with orthonormal columns $\mathbf{u}_{d+1}, \cdots, \mathbf{u}_m$ and $\mathbf{v}_{d+1}, \cdots, \mathbf{v}_n$, respectively, to square unitary matrices,

$$\mathbf{U} = [\hat{\mathbf{U}}\,, \mathbf{U}^{\perp}]\,, \qquad \mathbf{V} = [\hat{\mathbf{V}}\,, \mathbf{V}^{\perp}]\,.$$

We can augment $\hat{\mathbf{\Sigma}}$ accordingly with zero entries along the diagonal and elsewhere, to arrive at the original decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$ in (5.17). Thus, in $\mathbf{\Sigma}$, the number of nonzero singular values shows the rank of $\mathbf{X}$.

The columns of $\hat{\mathbf{U}}$ provide an orthonormal basis for the column span of $\mathbf{X}$. Likewise, the columns of $\hat{\mathbf{V}}$ are an orthonormal basis for the column span of $\mathbf{X}^{\mathrm{H}}$. The complementary matrices also have a meaning: since $\hat{\mathbf{V}}^{\mathrm{H}}\mathbf{V}^{\perp} = \mathbf{0}$, and hence $\mathbf{X}\mathbf{V}^{\perp} = \mathbf{0}$, it is seen that the columns of $\mathbf{V}^{\perp}$ span the null space of $\mathbf{X}$. Likewise, the columns of $\mathbf{U}^{\perp}$ span the left null space, $\mathbf{U}^{\perp H}\mathbf{X} = \mathbf{0}$.

The SVD of $\mathbf{X}$ in (5.20) reveals the behavior of the map $\mathbf{b} = \mathbf{X}\mathbf{a}$: $\mathbf{a}$ is projected onto the column span of $\hat{\mathbf{V}}$ and rotated in n-space (by $\mathbf{V}^{\mathrm{H}}\mathbf{a}$), then scaled (by the entries of $\hat{\mathbf{\Sigma}}$), and finally rotated in m-space (by $\hat{\mathbf{U}}$) to give $\mathbf{b}$.

5.4.1 Norms and the SVD

Recall the Frobenius norm of $\mathbf{X}$:

$$\|\mathbf{X}\|_{\mathrm{F}}^2 = \sum_{ij} |X_{ij}|^2 = \mathrm{tr}[\mathbf{X}^{\mathrm{H}}\mathbf{X}]$$

The latter expression shows that multiplication of $\mathbf{X}$ by a unitary matrix does not change the Frobenius norm. Thus, since $\mathbf{\Sigma} = \mathbf{U}^{\mathrm{H}}\mathbf{X}\mathbf{V}$, we find that

$$\|\mathbf{X}\|_{\mathrm{F}}^2 = \sum_i \sigma_i^2 \,.$$

Recall the "induced 2 norm" or matrix 2-norm $\|\mathbf{X}\|_2$, which measures how much a matrix can increase the 2-norm of a vector $\mathbf{v}$:

$$\|\mathbf{X}\| = \max_{\mathbf{v}} \frac{\|\mathbf{X}\mathbf{v}\|}{\|\mathbf{v}\|} \tag{5.21}$$

Without loss of generality, we may normalize the vectors $\mathbf{v}$ such that $\|\mathbf{v}\| = 1$. We can also insert the SVD. We then obtain

$$\|\mathbf{X}\|^2 = \max_{\|\mathbf{v}\|=1} \|\mathbf{X}\mathbf{v}\|^2 = \max_{\|\mathbf{v}\|=1} \mathbf{v}^{\mathrm{H}}(\mathbf{X}^{\mathrm{H}}\mathbf{X})\mathbf{v} = \max_{\|\mathbf{v}\|=1} \mathbf{v}^{\mathrm{H}}(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{H}})\mathbf{v} \,.$$

From this we can deduce that the vector $\mathbf{v}$ that maximizes the norm is given by $\mathbf{v} = \mathbf{v}_1$, the dominant right singular vector. The matrix 2-norm of $\mathbf{X}$ is then seen to be equal to σ_1. The fact that the 2-norm is attained on the dominant singular vector $\mathbf{v}_1$ gives a recursive way to prove the existence of the SVD: find $\mathbf{v}_1$ on which the norm is attained, and the corresponding σ_1 and $\mathbf{u}_1$ using

$$\mathbf{X}\mathbf{v}_1 = \mathbf{u}_1\sigma_1 \,,$$

then subtract (or project out) this rank-1 component and consider the residual $\mathbf{X}'$, and repeat [2].

An important property that follows from the definition of the norm (5.21) is

$$\|\mathbf{X}\mathbf{v}\| \leq \|\mathbf{X}\|\|\mathbf{v}\| \qquad \forall \mathbf{v}$$

where the maximum is only achieved for $\mathbf{v} = \alpha\mathbf{v}_1$.

For an arbitrary matrix $\mathbf{X}$, perhaps of full rank, the best rank-d approximant $\hat{\mathbf{X}}$ is obtained by computing $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$, and then setting all but the first d singular values in $\mathbf{\Sigma}$ equal to zero:

$$\hat{\mathbf{X}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\mathrm{H}} \,,$$

The approximation error in Frobenius norm and operator norm is given by

$$\begin{aligned} \|\,\mathbf{X} - \hat{\mathbf{X}}\,\|_{\mathrm{F}}^2 &= \sum_{i=d+1}^{M} \sigma_i^2 \\ \|\,\mathbf{X} - \hat{\mathbf{X}}\,\|^2 &= \sigma_{d+1}^2 \end{aligned}$$

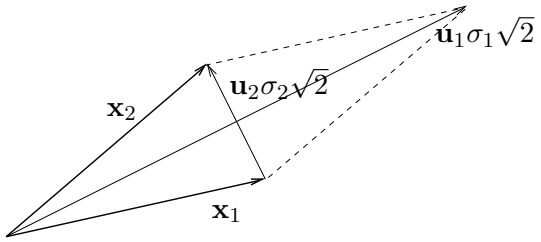


Figure 5.1. Construction of the left singular vectors and values of the matrix $X = [\mathbf{x}_1 \;\; \mathbf{x}_2]$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ have equal length.

5.4.2 QR and the SVD

The QR factorization can be used as a start in the computation of the SVD of a tall matrix $\mathbf{X}$. We first compute

$$\mathbf{X} = \hat{\mathbf{Q}}\hat{\mathbf{R}}\,.$$

The next step is to continue with an SVD of $\hat{\mathbf{R}}$:

$$\hat{\mathbf{R}} = \hat{\mathbf{U}}_R\hat{\mathbf{\Sigma}}_R\hat{\mathbf{V}}_R^{\mathrm{H}}\,,$$

so that the SVD of $\mathbf{X}$ is

$$\mathbf{X} = (\hat{\mathbf{Q}}\hat{\mathbf{U}}_R)\hat{\mathbf{\Sigma}}_R\hat{\mathbf{V}}_R^{\mathrm{H}}\,,$$

The preprocessing by QR in computing the SVD is useful because it reduces the size from $\mathbf{X}$ to that of $\hat{\mathbf{R}}$, and obviously, $\mathbf{X}$ and $\hat{\mathbf{R}}$ have the same singular values and *right* singular vectors. A QR decomposition is much easier to compute than the SVD (which requires an iterative process).

As mentioned before, the QR decomposition gives only a poor indication of the rank (although *rank-revealing QR decompositions* have been proposed). The SVD, on the other hand, is the standard tool to determine the rank.

Example 5.1. Figure 5.1 shows the construction of the left singular vectors of a matrix $\mathbf{X} = [\mathbf{x}_1 \;\; \mathbf{x}_2]$, whose columns $\mathbf{x}_1$ and $\mathbf{x}_2$ are of equal length. The largest singular vector $\mathbf{u}_1$ is in the direction of the sum of $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e., the "common" direction of the two vectors, and the corresponding singular value σ_1 is equal to $\sigma_1 = \| \mathbf{x}_1 + \mathbf{x}_2 \|/\sqrt{2}$. On the other hand, the smallest singular vector $\mathbf{u}_2$ is dependent on the difference $\mathbf{x}_2 - \mathbf{x}_1$, as is its corresponding singular value: $\sigma_2 = \| \mathbf{x}_2 - \mathbf{x}_1 \|/\sqrt{2}$. If $\mathbf{x}_1$ and $\mathbf{x}_2$ become more aligned, then σ_2 will be smaller and $\mathbf{X}$ will be closer to a singular matrix. Clearly, $\mathbf{u}_2$ is the most sensitive direction for perturbations on $\mathbf{x}_1$ and $\mathbf{x}_2$.

An example of such a matrix could be $\mathbf{A} = [\mathbf{a}(\phi_1) \;\; \mathbf{a}(\phi_2)]$, where $\mathbf{a}(\phi) = [1 \;\; \phi \;\; \phi^2 \;\; \cdots \;\; \phi^{M-1}]^{\mathrm{T}}$, where ϕ is for example related to the direction at which a signal hits an antenna array, or to the time difference to a reference signal. If

Table 5.1. Example 5.4.2: Singular values of $\mathbf{X}_{M,N}$.

$M = 3,$	$\sigma_1 = 3.44$	$M = 3,$	$\sigma_1 = 4.86$
$N = 3$	$\sigma_2 = 0.44$	$N = 6$	$\sigma_2 = 0.63$
$M = 6,$	$\sigma_1 = 4.73$		
$N = 3$	$\sigma_2 = 1.29$		

two directions are close together, then $\phi_1 \approx \phi_2$ and $\mathbf{a}(\phi_1)$ points in about the same direction as $\mathbf{a}(\phi_2)$, which will be the direction of $\mathbf{u}_1$. The smallest singular value, σ_2, is dependent on the difference of the directions of $\mathbf{a}(\phi_1)$ and $\mathbf{a}(\phi_2)$.

For further illustration, consider the following small numerical experiment. Let $\phi_1 = 1$, $\phi_2 = \exp(j\pi \cdot 0.1)$, and construct $M \times N$ matrices $\mathbf{X} = [\mathbf{a}(\phi_1) \;\; \mathbf{a}(\phi_2)]\mathbf{S}$, where $\mathbf{S}\mathbf{S}^{\mathrm{H}} = N\mathbf{I}$. Since $(1/\sqrt{N})\mathbf{S}$ is co-isometric, the singular values of $\mathbf{X}$ are those of $\mathbf{A} = [\mathbf{a}(\phi_1) \;\; \mathbf{a}(\phi_2)]$ times $\sqrt{N}$. The two non-zero singular values of $\mathbf{X}$ for some values of M, N are given in Table 5.1. It is seen that doubling M almost triples the smallest singular value, whereas doubling N only increases the singular values by a factor $\sqrt{2}$, which is because the matrices have larger size.

5.4.3 Matrix inversion using the SVD

If $\mathbf{X}$ is full column rank, then the left inverse of $\mathbf{X}$ is $\mathbf{X}^\dagger = (\mathbf{X}^{\mathrm{H}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{H}}$. Inserting $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$, we obtain that this can also be written as

$$\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}} \tag{5.22}$$

which is a slightly more general expression. Essentially, we are inverting the singular values here. We can easily verify that $\mathbf{X}^\dagger\mathbf{X} = \mathbf{I}$ and

$$\mathbf{X}\mathbf{X}^\dagger = \mathbf{U}\mathbf{U}^{\mathrm{H}}$$

If we define $\mathbf{P} = \mathbf{U}\mathbf{U}^{\mathrm{H}}$ then we see that $\mathbf{P}$ is an orthogonal projection, because $\mathbf{P}\mathbf{P} = \mathbf{P}$ and $\mathbf{P}^{\mathrm{H}} = \mathbf{P}$. It is a projection onto the column span of $\mathbf{X}$.

The largest singular value of the pseudo-inverse is σ_n^{-1}. It follows that the matrix 2-norm of $\mathbf{X}^\dagger$ is given by σ_n^{-1}. If σ_n is very small, it shows that Σ^{-1} and $\mathbf{X}^\dagger$ have a very large norm, and should not be used: in applications, this leads to noise enhancement.

5.4.4 Connection to the eigenvalue decomposition

If we take the SVD of $\mathbf{X}$ and "square" it to $\mathbf{X}^{\mathrm{H}}\mathbf{X}$, we obtain

$$\mathbf{X}^{\mathrm{H}}\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{H}}$$

Matrix $\mathbf{X}^{\mathrm{H}}\mathbf{X}$ is a symmetric matrix; the decomposition is recognized as the eigenvalue decomposition of the symmetric matrix $\mathbf{X}^{\mathrm{H}}\mathbf{X}$, where the eigenvalues are given by the entries of $\mathbf{\Sigma}^2$, and the eigenvectors by the columns of $\mathbf{V}$. (The eigenvalue problem is discussed later in Sec. 5.6.)

Similarly, $\mathbf{X}\mathbf{X}^{\mathrm{H}}$ has eigenvalue decomposition

$$\mathbf{X}\mathbf{X}^{\mathrm{H}} = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^{\mathrm{H}}$$

The point of the SVD is that it gives similar information as we obtain from an eigenvalue decomposition, but (*i*) it is applicable to any matrix (e.g., non-square matrices), and (*ii*) it always exists, whereas the eigenvalue decomposition only exists for "regular" matrices. Also, there are numerically very robust algorithms to compute the decomposition.

5.4.5 Rank reduction using the SVD; Moore-Penrose pseudo-inverse

Equation (5.22) shows that, when we invert a matrix $\mathbf{X}$, the smallest singular values of $\mathbf{X}$ become the largest singular values of $\mathbf{X}^{\dagger}$. Sometimes, if a matrix is almost rank deficient (σ_n is very small), that small component will dominate the inverse, which can give rise to numerical problems, as we will see later. In that case, we propose to first approximate $\mathbf{X}$ to a lower rank d, by setting all singular values below a certain threshold ϵ equal to zero:

$$\hat{\mathbf{X}} = \mathbf{u}_1\sigma_1\mathbf{v}_1^{\mathrm{H}} + \mathbf{u}_2\sigma_2\mathbf{v}_2^{\mathrm{H}} + \cdots + \mathbf{u}_d\sigma_d\mathbf{v}_d^{\mathrm{H}}$$

which we can write as (economy-size SVD notation)

$$\hat{\mathbf{X}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\mathrm{H}}\,, \qquad \hat{\mathbf{U}} = [\mathbf{u}_1 \cdots \mathbf{u}_d]\,, \quad \hat{\mathbf{\Sigma}} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_d \end{bmatrix}, \quad \hat{\mathbf{V}} = [\mathbf{v}_1 \cdots \mathbf{v}_d]\,.$$

This is called the Truncated SVD.

The corresponding approximate inverse is

$$\hat{\mathbf{X}}^{\dagger} = \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}^{\mathrm{H}}\,.$$

This is called the Moore-Penrose pseudo-inverse of $\hat{\mathbf{X}}$. It satisfies the projection properties:

$$\hat{\mathbf{X}}\hat{\mathbf{X}}^{\dagger} = \hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathrm{H}} = \mathbf{P}_c\,, \qquad \hat{\mathbf{X}}^{\dagger}\hat{\mathbf{X}} = \hat{\mathbf{V}}\hat{\mathbf{V}}^{\mathrm{H}} = \mathbf{P}_r$$

where $\mathbf{P}_c$ is a projection onto the dominant column span of $\mathbf{X}$, and $\mathbf{P}_r$ a projection onto the dominant row span.

The largest singular value of the Moore-Penrose (truncated) pseudo-inverse is σ_d^{-1}, whereas without truncation it was σ_n^{-1}. This gives a way to control the norm of the inverse, by inverting only dominant directions in $\mathbf{X}$, and projecting away the other dimensions.

This pseudo-inverse (matlab: `pinv`) is commonly used if we are not sure if a matrix is full rank. Typically, we compare the singular values of $\mathbf{X}$ to a threshold (ϵ) and replace them by 0 if they are below the threshold, leading to $\hat{\mathbf{X}}$. Next, we compute $\hat{\mathbf{X}}^\dagger$ by inverting the non-zero singular values.

5.4.6 The condition number

The condition number of $\mathbf{X}$ is defined by

$$c(\mathbf{X}) := \frac{\sigma_1}{\sigma_n}$$

Thus, we always have $c(\mathbf{X}) \geq 1$. If it is large, then $\mathbf{X}$ is hard to invert (and $\mathbf{X}^\dagger$ is sensitive to small changes). The smallest condition number for a matrix is $c = 1$, which is achieved for an orthogonal matrix.

When we compute the inverse of a matrix, its condition number is very important. Indeed, the condition number gives the relative sensitivity of the solution of a linear systems of equations. Let us suppose that we wish to solve a system of equations $\mathbf{Ax} = \mathbf{b}$, where we take $\mathbf{A} : n \times n$ square. We have

$$\mathbf{Ax} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

Now, if we perturb the data vector $\mathbf{b}$ by a noise vector $\mathbf{e}$, we obtain

$$\mathbf{b}' = \mathbf{b} + \mathbf{e} \quad \Rightarrow \quad \mathbf{x}' = \mathbf{x} + \mathbf{A}^{-1}\mathbf{e}$$

Define $\sigma_1 = \| \mathbf{A} \|, \quad \sigma_n^{-1} = \| \mathbf{A}^{-1} \|$, and use $\|\mathbf{Ax}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$. Then

$$\begin{array}{rcl} \| \mathbf{A}^{-1}\mathbf{e} \| & \leq & \sigma_n^{-1} \| \mathbf{e} \| \\ \| \mathbf{b} \| & \leq & \sigma_1 \| \mathbf{x} \| \\ \dfrac{\|\mathbf{x}' - \mathbf{x}\|}{\|\mathbf{x}\|} & \leq & \sigma_n^{-1} \dfrac{\| \mathbf{e} \|}{\|\mathbf{x}\|} \ \leq \ \sigma_n^{-1}\sigma_1 \dfrac{\| \mathbf{e} \|}{\|\mathbf{b}\|} \end{array}$$

This measures the relative change in the solution vector $\mathbf{x}$, and shows that any error in $\mathbf{b}$ is potentially magnified by a factor equal to the condition number.

If a matrix has a poor condition number, the usual strategy is not to invert it directly, but to do a rank reduction to rank d, and compute the pseudo-inverse as shown before. This will avoid noise enhancement due to the inversion of non-important components.

5.5 PSEUDO-INVERSE AND THE LEAST SQUARES PROBLEM

5.5.1 The pseudo-inverse

Consider a rank-d $M \times N$ matrix $\mathbf{X}$. In general, since $\mathbf{X}$ may be rank-deficient or non-square, the inverse of $\mathbf{X}$ does not exist; i.e., for a given vector $\mathbf{b}$, we cannot always find a vector $\mathbf{a}$ such that $\mathbf{b} = \mathbf{Xa}$.

If $\mathbf{X}$ is tall but of full rank, the *pseudo-inverse* of $\mathbf{X}$ is $\mathbf{X}^\dagger = (\mathbf{X}^{\mathrm{H}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{H}}$. It satisfies

$$\begin{aligned} \mathbf{X}^\dagger X &= \mathbf{I}_N \\ \mathbf{X}\mathbf{X}^\dagger &= \mathbf{P}_c \end{aligned}$$

Thus, $\mathbf{X}^\dagger$ is an inverse on the "short space", and $\mathbf{X}\mathbf{X}^\dagger$ is a projection onto the column span of $\mathbf{X}$. It is easy to verify that the solution to $\mathbf{b} = \mathbf{X}\mathbf{a}$ is given by $\mathbf{a} = \mathbf{X}^\dagger\mathbf{b}$.

If $\mathbf{X}$ is rank deficient, then $\mathbf{X}^{\mathrm{H}}\mathbf{X}$ is not invertible, and there is no exact solution to $\mathbf{b} = \mathbf{X}\mathbf{a}$. In this case, we can resort to the Moore-Penrose pseudo-inverse of $\mathbf{X}$, also denoted by $\mathbf{X}^\dagger$. It can be defined in terms of the "economy size" SVD $\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\mathrm{H}}$ (equation (5.18)) as

$$\mathbf{X}^\dagger = \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}^{\mathrm{H}} \,.$$

This pseudo-inverse satisfies the properties

1. $\mathbf{X}\mathbf{X}^\dagger\mathbf{X} = \mathbf{X}$
2. $\mathbf{X}^\dagger\mathbf{X}\mathbf{X}^\dagger = \mathbf{X}^\dagger$
3. $\mathbf{X}\mathbf{X}^\dagger = \mathbf{P}_c$
4. $\mathbf{X}^\dagger\mathbf{X} = \mathbf{P}_r$

which constitute the Moore-Penrose inverse in the traditional way.

These equations show that, in order to make the problem $\mathbf{b} = \mathbf{X}\mathbf{a}$ solvable, a solution can be forced to an approximate problem by projecting $\mathbf{b}$ onto the column space of $\mathbf{X}$:

$$\mathbf{b}' = \mathbf{P}_c\mathbf{b} \,,$$

after which $\mathbf{b}' = \mathbf{X}\mathbf{a}$ has solution

$$\mathbf{a} = \mathbf{X}^\dagger\mathbf{b}' \,.$$

The projection is in fact implicitly done by just taking $\mathbf{a} = \mathbf{X}^\dagger\mathbf{b}$: from properties 1 and 3 of the list above, we have that

$$\mathbf{a} = \mathbf{X}^\dagger\mathbf{b}' = \mathbf{X}^\dagger\mathbf{X}\mathbf{X}^\dagger\mathbf{b} = \mathbf{X}^\dagger\mathbf{b}$$

It can be shown that this solution $\mathbf{a}$ is the solution of the (Least Squares) minimization problem

$$\min_{\mathbf{a}} \| \mathbf{b} - \mathbf{X}\mathbf{a} \|^2 \,,$$

where $\mathbf{a}$ is chosen to have minimal norm if there is more than one solution (the latter requirement translates to $\mathbf{a} = \mathbf{P}_r\mathbf{a}$).

Some other properties of the pseudo-inverse are

- The norm of $\mathbf{X}^\dagger$ is $\| \mathbf{X}^\dagger \| = \sigma_d^{-1}$.
- The *condition number* of $\mathbf{X}$ is $c(\mathbf{X}) := \frac{\sigma_1}{\sigma_d}$.

  If it is large, then $\mathbf{X}$ is hard to invert ($\mathbf{X}^\dagger$ is sensitive to small changes).

5.5.2 Total Least Squares

Now, suppose that instead of a single vector $\mathbf{b}$ we are given an $(M \times N)$-dimensional matrix $\mathbf{Y}$, the columns of which are not all in the column space of the matrix $\mathbf{X}$. We want to force solutions to $\mathbf{XA} = \mathbf{Y}$. Clearly, we can use a least squares approximation $\hat{\mathbf{Y}} = \mathbf{P}_{\mathbf{X}}\mathbf{Y}$ to force the columns of $\hat{\mathbf{Y}}$ to be in the d-dimensional column space of $\mathbf{X}$. This is reminiscent to the LS application above, but just one way to arrive at $\mathbf{X}$ and $\mathbf{Y}$ having a common column space, in this case by only modifying $\mathbf{Y}$. There is another way, called Total Least Squares (TLS) which is effectively described as projecting both $\mathbf{X}$ and $\mathbf{Y}$ onto some subspace that lies between them, and that is "closest" to the column spaces of the two matrices. To implement this method, we compute the SVD

$$[\mathbf{X} \quad \mathbf{Y}] \;=\; [\hat{\mathbf{U}} \quad \hat{\mathbf{U}}^{\perp}]\,\mathbf{\Sigma}\begin{bmatrix} \hat{\mathbf{V}}^{\mathrm{H}} \\ (\hat{\mathbf{V}}^{\perp})^{\mathrm{H}} \end{bmatrix} \;=\; \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\,[\hat{\mathbf{V}}_1^{\mathrm{H}} \quad \hat{\mathbf{V}}_2^{\mathrm{H}}] + \hat{\mathbf{U}}^{\perp}\hat{\mathbf{\Sigma}}^{\perp}(\hat{\mathbf{V}}^{\perp})^{\mathrm{H}}$$

and define the projection $\mathbf{P}_c = \hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathrm{H}}$. We now take the TLS (column space) approximations to be $\hat{\mathbf{X}} = \mathbf{P}_c\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}_1^{\mathrm{H}}$ and $\hat{\mathbf{Y}} = \mathbf{P}_c\mathbf{Y} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}_2^{\mathrm{H}}$, where $\hat{\mathbf{V}}_1$ and $\hat{\mathbf{V}}_2$ are the partitions of $\hat{\mathbf{V}}$ corresponding to $\mathbf{X}$ and $\mathbf{Y}$ respectively. $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ have the same column span defined by $\hat{\mathbf{U}}$, and are in fact solutions to

$$\min_{[\hat{\mathbf{X}}\;\hat{\mathbf{Y}}]\ \text{rank}\ N} \| [\mathbf{X} \;\; \mathbf{Y}] - [\hat{\mathbf{X}} \;\; \hat{\mathbf{Y}}] \|_{\mathrm{F}}^2$$

and $\mathbf{A}$ satisfying $\hat{\mathbf{X}}\mathbf{A} = \hat{\mathbf{Y}}$ is obtained as $\mathbf{A} = \hat{\mathbf{X}}^{\dagger}\hat{\mathbf{Y}}$. This $\mathbf{A}$ is the TLS solution of $\mathbf{XA} \approx \mathbf{Y}$. Instead of asking for rank N, we might even insist on a lower rank d.

5.6 THE EIGENVALUE PROBLEM

The *eigenvalue problem* for a matrix $\mathbf{A}$ is

$$\mathbf{Ax} = \lambda\mathbf{x} \quad \Leftrightarrow \quad (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{5.23}$$

Any λ that makes $\mathbf{A} - \lambda\mathbf{I}$ singular is called an eigenvalue, the corresponding $\mathbf{x}$ is the eigenvector (invariant vector). It has an arbitrary norm usually set equal to 1.

We can collect the eigenvectors in a matrix:

$$\begin{aligned} \mathbf{A}[\mathbf{x}_1 \;\; \mathbf{x}_2 \;\cdots] \;&=\; [\mathbf{x}_1 \;\; \mathbf{x}_2 \;\cdots]\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} \\ \Leftrightarrow \qquad \mathbf{AT} \;&=\; \mathbf{T\Lambda} \end{aligned}$$

It is common to normalize the eigenvectors (columns of $\mathbf{T}$) to have unit norm.

A *regular* matrix $\mathbf{A}$ has an *eigenvalue decomposition:*

$$\mathbf{A} = \mathbf{T\Lambda T}^{-1}, \qquad \mathbf{T} \text{ invertible }, \mathbf{\Lambda} \text{ diagonal} \tag{5.24}$$

This decomposition might not exist if eigenvalues are repeated. A classical example of a matrix that does not have an eigenvalue decomposition is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

5.6.1 Schur decomposition

Suppose $\mathbf{T}$ has a QR factorization $\mathbf{T} = \mathbf{Q}\mathbf{R}_T$, so that $\mathbf{T}^{-1} = \mathbf{R}_T^{-1}\mathbf{Q}^{\mathrm{H}}$. Then

$$\mathbf{A} \;=\; \mathbf{Q}\mathbf{R}_T\mathbf{\Lambda}\mathbf{R}_T^{-1}\mathbf{Q}^{\mathrm{H}} \;=\; \mathbf{Q}\mathbf{R}\mathbf{Q}^{\mathrm{H}}$$

The factorization

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

with $\mathbf{Q}$ unitary and $\mathbf{R}$ upper triangular, is called a *Schur decomposition*. One can show that this decomposition always exists (although it is not unique); if $\mathbf{A}$ is hermitian, then $\mathbf{R} = \mathbf{R}^{\mathrm{H}}$ is upper triangular implies that $\mathbf{R}$ is diagonal, and in this case the Schur decomposition coincides with the eigenvalue decomposition (and the SVD). For non-hermitian $\mathbf{A}$, the Schur decomposition avoids the inversion of the eigenvalue matrix $\mathbf{T}$, which might be ill-conditioned (or even non-invertible in some cases).

$\mathbf{R}$ has the eigenvalues of $\mathbf{A}$ on the diagonal. $\mathbf{Q}$ gives information about "eigen-subspaces" (invariant subspaces), but doesn't contain eigenvectors.

5.6.2 Connection to the SVD

Suppose we compute the SVD of a matrix $\mathbf{X}$, and then consider $\mathbf{X}\mathbf{X}^{\mathrm{H}}$:

$$\begin{aligned} \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} \qquad \Rightarrow \qquad \mathbf{X}\mathbf{X}^{\mathrm{H}} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathrm{H}} \\ &= \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^{\mathrm{H}} \\ &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}} \end{aligned}$$

This shows that the eigenvalues of $\mathbf{X}\mathbf{X}^{\mathrm{H}}$ are the singular values of $\mathbf{X}$, squared (hence real). The eigenvectors of $\mathbf{X}\mathbf{X}^{\mathrm{H}}$ are equal to the left singular vectors of $\mathbf{X}$ (hence $\mathbf{U}$ is unitary). Since the SVD always exists, the eigenvalue decomposition of $\mathbf{X}\mathbf{X}^{\mathrm{H}}$ always exists (in fact it exists for any Hermitian matrix $\mathbf{C} = \mathbf{C}^{\mathrm{H}}$).

Historically, the SVD was derived out of frustration that the eigenvalue decomposition does not always exist. By generalizing the eigenvector matrix $\mathbf{T}$ to two unitary matrices $\mathbf{U}, \mathbf{V}$, a decomposition was found that does always exist. Despite this connection, the decompositions are generally different and have different applications.

5.7 THE GENERALIZED EIGENVALUE DECOMPOSITION

For two matrices $\mathbf{A}$ and $\mathbf{B}$, the generalized eigenvalue problem is

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x} \quad \Leftrightarrow \quad (\mathbf{A} - \lambda\mathbf{B})\mathbf{x} = \mathbf{0}$$

This is a generalization of (5.23) where we had $\mathbf{B} = \mathbf{I}$. The solutions λ_i are called the generalized eigenvalues. The set of matrices $\mathbf{A} - \lambda\mathbf{B}$ (for any λ) or the pair $(\mathbf{A}, \mathbf{B})$ is called a *matrix pencil*.

In the above formulation, $\mathbf{A}$ and $\mathbf{B}$ have the same size but could be rectangular. If $\mathbf{A}$ is square and $\mathbf{B}$ is invertible, then we can immediately return to the usual eigenvalue decomposition, by considering

$$\mathbf{B}^{-1}\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

Thus, the generalized eigenvalues of a pair $(\mathbf{A}, \mathbf{B})$ are the eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$. Generally, we try to avoid inverting $\mathbf{B}$, for numerical reasons, or because $\mathbf{A}$ and $\mathbf{B}$ might have structure that is otherwise lost. E.g., if $\mathbf{B}$ is banded, its inverse is not a band.

As in (5.24), we can collect the eigenvectors in a matrix $\mathbf{T}$, such that

$$\mathbf{A}\mathbf{T} = \mathbf{B}\mathbf{T}\mathbf{\Lambda}$$

where $\mathbf{\Lambda}$ is diagonal. We can also write the solution as a joint matrix decomposition

$$\begin{cases} \mathbf{A} &= \mathbf{F}\mathbf{\Lambda}_A\mathbf{T}^{-1} \\ \mathbf{B} &= \mathbf{F}\mathbf{\Lambda}_B\mathbf{T}^{-1} \end{cases} \tag{5.25}$$

where $\mathbf{\Lambda}_B^{-1}\mathbf{\Lambda}_A$ are the generalized eigenvalues, and $\mathbf{F}$ is an (invertible?) matrix with unit-norm columns. Indeed, from $\mathbf{A}\mathbf{T} = \mathbf{B}\mathbf{T}\mathbf{\Lambda}$, after having found $\mathbf{T}$ we can set $\mathbf{W} = \mathbf{A}\mathbf{T}$, and then normalize the columns of $\mathbf{W}$ to find $\mathbf{W} = \mathbf{F}\mathbf{\Lambda}_A$. This decomposition is called the Generalized Eigenvalue Decomposition (GEV).

The form (5.25) shows an application of the GEV, namely a joint diagonalization of two matrices $(\mathbf{A}, \mathbf{B})$. This application is studied in more detail in Chap. 9.

The existence of this decomposition is similar to that of the eigenvalue decomposition: in many cases, it does not exist, and/or $\mathbf{F}$ and $\mathbf{T}$ are not invertible. However, if $\mathbf{A}$ and $\mathbf{B}$ are hermitian and one of them (typically $\mathbf{B}$) is positive definite, the decomposition exists. For more general cases, numerical algorithms are more complicated and might run into problems.

Just as the SVD is connected to the eigenvalue decomposition, the generalized eigenvalue decomposition leads to a Generalized SVD (GSVD):

$$\begin{cases} \mathbf{A} &= \mathbf{F}\mathbf{\Sigma}_A\mathbf{U}^{\mathrm{H}} \\ \mathbf{B} &= \mathbf{F}\mathbf{\Sigma}_B\mathbf{V}^{\mathrm{H}} \end{cases} \tag{5.26}$$

where $\mathbf{U}, \mathbf{V}$ are unitary, $\mathbf{\Sigma}_A, \mathbf{\Sigma}_B$ are diagonal and nonnegative real, and $\mathbf{F}$ is an invertible matrix with unit-norm columns. This decomposition always exists; the matrices $\mathbf{A}$ and $\mathbf{B}$ do not need

to be square, they can even have a different number of columns. Note that if $\mathbf{B}$ is square and invertible, then $\mathbf{B}^{-1}\mathbf{A} = \mathbf{V}\mathbf{\Sigma}_B^{-1}\mathbf{\Sigma}_A\mathbf{U}^{\mathrm{H}}$ constitutes an SVD of $\mathbf{B}^{-1}\mathbf{A}$.

Actually, various definitions of the GSVD exist. In the usual formulation [2],

$$\begin{cases} \mathbf{A} &= \mathbf{U}\mathbf{\Sigma}_A\mathbf{X}^{-1} \\ \mathbf{B} &= \mathbf{V}\mathbf{\Sigma}_B\mathbf{X}^{-1} \end{cases}$$

where $\mathbf{U}, \mathbf{V}$ are unitary, $\mathbf{X}$ is invertible, and $\mathbf{\Sigma}_A^2 + \mathbf{\Sigma}_B^2 = \mathbf{I}$, which gives a connection to the CS decomposition ("cosine-sine"). However, this formulation has problems in case there is a vector $\mathbf{x}$ in the nullspace of both $\mathbf{A}$ and $\mathbf{B}$. In our applications, we often have that case. Therefore, we will use (5.26).

The Generalized Schur Decomposition (GSD), also called the QZ decomposition, is

$$\begin{cases} \mathbf{A} &= \mathbf{Q}\mathbf{R}_A\mathbf{Z}^{\mathrm{H}} \\ \mathbf{B} &= \mathbf{Q}\mathbf{R}_B\mathbf{Z}^{\mathrm{H}} \end{cases} \tag{5.27}$$

where $\mathbf{Q}, \mathbf{Z}$ are unitary, and $\mathbf{R}_A, \mathbf{R}_B$ are upper triangular. This decomposition always exists. It follows from the GEV (5.25) by inserting a QR decomposition for $\mathbf{F}$ and another one for $\mathbf{T}^{-1}$. The generalized eigenvalues of $(\mathbf{A}, \mathbf{B})$ are found by the ratios of the diagonal entries of $\mathbf{R}_A$ and $\mathbf{R}_B$. The advantage of this decomposition is that it is more stable to compute as it involves only unitary matrices. This facilitates its computation using 2×2 Givens rotations. Generally, the QZ algorithm is used: the core of this consists of an iteration where the QR decomposition of $\mathbf{B}^{-1}\mathbf{A}$ is implicitly computed, without forming the product.

5.8 NOTES

A widely-used reference book on linear algebra is Golub & Van Loan [2]. More advanced properties are found in Horn and Johnson [3]. An extensive tutorial to linear algebra in relation to signal processing can be found in Moon and Stirling [1].

Bibliography

[1] T. K. Moon and W. C. Stirling, *Mathematical methods and algorithms for signal processing.* Prentice Hall, 2000.

[2] G. Golub and C. Van Loan, *Matrix Computations.* The Johns Hopkins University Press, 1989.

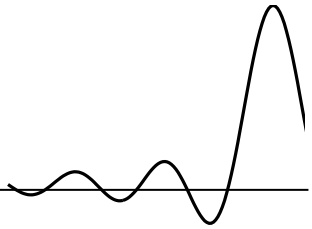
[3] R. Horn and C. Johnson, *Matrix Analysis.* Cambridge, NY: Cambridge Univ. Press, 1985.

Part II

METHODS AND ALGORITHMS

Chapter 6

SPATIAL PROCESSING TECHNIQUES



Contents

In this chapter, we look at elementary receiver schemes: the matched filter and Wiener filter in their non-adaptive forms. They are suitable if we have a good estimate of the channel, or if we know a segment of the transmitted data, e.g., because of a training sequence. These receivers are most simple in the context of narrowband antenna array processing, and hence we place the discussion first in this scenario. The matched filter is shown to maximize the output signal-to-noise ratio (in the case of a single signal in noise), whereas the Wiener receiver maximizes the output signal-to-interference plus noise (in the case of several sources in noise). We also look at the application of these receivers as non-parametric beamformers for direction-of-arrival estimation. Improved accuracy is possible using parametric data models and subspace-based techniques: a prime example is the MUSIC algorithm.

General references to this chapter are [1–7].

6.1 DETERMINISTIC APPROACH TO MATCHED AND WIENER FILTERS

6.1.1 Data model and assumptions

In this chapter, we consider a simple array signal processing model of the form

$$\mathbf{x}_k \;=\; \sum_{i=1}^{d} \mathbf{a}_i s_{i,k} + \mathbf{n}_k \;=\; \mathbf{A}\mathbf{s}_k + \mathbf{n}_k \,. \tag{6.1}$$

We assume that signals are received by M antennas, and that the antenna outputs (after demodulation, sampling, A/D conversion) are stacked into vectors $\mathbf{x}_k$. According to the model, $\mathbf{x}_k$ is a linear combination of d narrowband source signals $s_{i,k}$ and noise $\mathbf{n}_k$. Initially, we will consider an even simpler case where there is only one signal in noise. In all cases, we assume that the noise covariance matrix

$$\mathbf{R}_\mathbf{n} := \mathrm{E}[\mathbf{n}\mathbf{n}^{\mathrm{H}}]$$

is known, up to a scalar which represents the noise power. The most simple situation is spatially white noise, for which

$$\mathbf{R}_\mathbf{n} = \sigma^2 \mathbf{I} \,.$$

Starting from the data model (6.1), let us assume that we have collected N sample vectors. If we store the samples in an $M \times N$ matrix $\mathbf{X} = [\mathbf{x}_1 \,, \cdots \,, \mathbf{x}_N]$, then we obtain that $\mathbf{X}$ has a decomposition

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N} \tag{6.2}$$

where the rows of $\mathbf{S} \in \mathbb{C}^{d \times N}$ contain the samples of the source signals. Note that we can choose to put the source powers in either $\mathbf{A}$ or $\mathbf{S}$, or even in a separate factor $\mathbf{B}$. Here we will assume they are absorbed in $\mathbf{A}$, thus the sources have unit powers. Sources may be considered either stochastic (with probability distributions) or deterministic. If they are stochastic, we assume they are zero mean, independent and hence uncorrelated,

$$\mathrm{E}[\mathbf{s}_k \mathbf{s}_k^{\mathrm{H}}] = \mathbf{I} \,.$$

If they are considered deterministic ($\mathrm{E}[\mathbf{s}_k] = \mathbf{s}_k$), we will assume similarly that

$$\lim_{N \to \infty} \; \tfrac{1}{N} \mathbf{S}\mathbf{S}^{\mathrm{H}} \;=\; \mathbf{I} \,.$$

The objective of beamforming is to construct a receiver weight vector $\mathbf{w}$ such that the output is

$$y_k = \mathbf{w}^{\mathrm{H}} \mathbf{x}_k = \hat{s}_k \tag{6.3}$$

is an estimate of one of the original sources. Which beamformer is "the best" depends on the optimality criterion, of which there are many. It also makes a difference if we wish to receive only a single signal, as in (6.3), or all d signals jointly,

$$\mathbf{y}_k = \mathbf{W}^{\mathrm{H}} \mathbf{x}_k = \hat{\mathbf{s}}_k \tag{6.4}$$

where $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_d]$.

We will first look at purely deterministic techniques to estimate the beamformers: here no explicit statistical assumptions are made on the data. The noise is viewed as a perturbation on the noise-free data $\mathbf{AS}$, and the perturbations are assumed to be small and equally important on all entries. Then we will look at more statistically oriented techniques. The noise will be modeled as a stochastic sequence with a joint Gaussian distribution. Still, we have a choice whether we consider $\mathbf{s}_k$ to be a deterministic sequence (known or unknown), or if we associate a probabilistic distribution to it, for example Gaussian or belonging to a certain alphabet such as $\{+1, -1\}$. In the latter case, we can often improve on the linear receiver (6.3) or (6.4) by taking into account that the output of the beamformer should belong to this alphabet (or should have a certain distribution). The resulting receivers will then contain some non-linear components.

In this chapter, we only consider the most simple cases, resulting in the classical linear beamformers.

6.1.2 Algebraic (purely deterministic) approach

Noiseless case Let us first consider the noiseless case, and a situation where we have collected N samples. Our data model thus is

$$\mathbf{X} = \mathbf{AS}\,.$$

Our objective will be to construct a linear beamforming matrix $\mathbf{W}$ such that

$$\mathbf{W}^{\mathrm{H}}\mathbf{X} = \mathbf{S}\,.$$

We consider two cases:

1. $\mathbf{A}$ is known, for example we know the directions of the sources and have set $\mathbf{A} = [\mathbf{a}(\theta_1)\ \cdots\ \mathbf{a}(\theta_d)]$,
2. $\mathbf{S}$ is known, for example we have selected a segment of the data which contains a training sequence for all sources. Alternatively, for discrete alphabet sources (e.g., $\mathbf{S}_{ij} \in \{\pm 1\}$) we can be in this situation via *decision feedback*.

In both cases, the problem is easily solved. If $\mathbf{A}$ is known, then we set

$$\mathbf{W}^{\mathrm{H}} = \mathbf{A}^{\dagger}\,, \qquad \mathbf{S} = \mathbf{W}^{\mathrm{H}}\mathbf{X}\,.$$

Here, $\mathbf{A}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$. If $M \geq d$ and the columns of $\mathbf{A}$ are linearly independent, then $\mathbf{A}^{\dagger}$ is equal to the left inverse

$$\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{H}}\,.$$

Note that, indeed, under these assumptions $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is invertible and $\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{I}$. If $M < d$ then we cannot recover the sources exactly: $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is not invertible (it is a $d \times d$ matrix with maximal rank M), so that $\mathbf{A}^{\dagger}\mathbf{A} \neq \mathbf{I}$.

If $\mathbf{S}$ is known, then we take

$$\mathbf{W}^{\mathrm{H}} = \mathbf{S}\mathbf{X}^{\dagger}\,, \qquad \mathbf{A} = (\mathbf{W}^{\mathrm{H}})^{\dagger}\,.$$

where $\mathbf{X}^{\dagger}$ is a right inverse of $\mathbf{X}$. If $N \geq d$ and the rows of $\mathbf{X}$ are linearly independent,[1] then

$$\mathbf{X}^{\dagger} = \mathbf{X}^{\mathrm{H}}(\mathbf{X}\mathbf{X}^{\mathrm{H}})^{-1}\,.$$

This is verified by $\mathbf{X}\mathbf{X}^{\dagger} = \mathbf{I}$. In both cases, we obtain a beamformer which exactly cancels all interference, i.e., $\mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I}$.

Noisy case In the presence of additive noise, we have $\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}$. Two types of linear least-squares (LS) minimization problems can now be considered. The first is based on minimizing the model fitting error,

$$\min_{\mathbf{S}} \ \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2\,, \qquad \text{or} \qquad \min_{\mathbf{A}} \ \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2 \tag{6.5}$$

with $\mathbf{A}$ or $\mathbf{S}$ known, respectively. The second type of minimization problem is based on minimizing the output error,

$$\min_{\mathbf{S}} \ \|\mathbf{W}^{\mathrm{H}}\mathbf{X} - \mathbf{S}\|_{\mathrm{F}}^2\,, \qquad \text{or} \qquad \min_{\mathbf{W}} \ \|\mathbf{W}^{\mathrm{H}}\mathbf{X} - \mathbf{S}\|_{\mathrm{F}}^2\,, \tag{6.6}$$

also with $\mathbf{A}$ or $\mathbf{S}$ known, respectively. The minimization problems are straightforward to solve, and in the same way as before.

Deterministic model matching For (6.5) with $\mathbf{A}$ known we obtain

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} \ \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2 \qquad \Rightarrow \qquad \hat{\mathbf{S}} = \mathbf{A}^{\dagger}\mathbf{X}\,, \tag{6.7}$$

so that again $\mathbf{W}^{\mathrm{H}} = \mathbf{A}^{\dagger}$. This is known as the *Zero-Forcing solution*, because $\mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I}$: all interfering sources are canceled. As will be shown later, the ZF beamformer maximizes the Signal-to-Interference power Ratio (SIR) at the output: in this case (known $\mathbf{A}$) it is infinity. Note however that

$$\mathbf{W}^{\mathrm{H}}\mathbf{X} = \mathbf{S} + \mathbf{A}^{\dagger}\mathbf{N}\,.$$

The noise contribution at the output is $\mathbf{A}^{\dagger}\mathbf{N}$, and if $\mathbf{A}^{\dagger}$ is large, the output noise will be large. To get a better insight for this, introduce the "economy-size" singular value decomposition of $\mathbf{A}$,

$$\mathbf{A} = \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}}$$

where we take $\mathbf{U}_A : m \times d$ with orthonormal columns, $\mathbf{\Sigma}_A : d \times d$ diagonal containing the nonzero singular values of $\mathbf{A}$, and $\mathbf{V}_A : d \times d$ unitary. Since

$$\mathbf{A} = \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}} \qquad \Rightarrow \qquad \mathbf{A}^{\dagger} = \mathbf{V}_A\mathbf{\Sigma}_A^{-1}\mathbf{U}_A^{\mathrm{H}}\,,$$

[1] In the present noiseless case, note that there are only d linearly independent rows in $\mathbf{S}$ and $\mathbf{X}$, so for linear independence of the rows of $\mathbf{X}$ we need $M = d$. With noise, $\mathbf{X}$ will have full row rank M.

$\mathbf{A}^\dagger$ is large if $\mathbf{\Sigma}_A^{-1}$ is large, i.e., if $\mathbf{A}$ is ill conditioned.

Similarly, for (6.5) with $\mathbf{S}$ known we obtain

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2 \qquad \Rightarrow \qquad \hat{\mathbf{A}} = \mathbf{X}\mathbf{S}^\dagger = \mathbf{X}\mathbf{S}^{\mathrm{H}}(\mathbf{S}\mathbf{S}^{\mathrm{H}})^{-1}\,. \tag{6.8}$$

This does not specify the beamformer, but staying in the same context of minimizing $\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2$, it is natural to take again a Zero-Forcing beamformer so that $\mathbf{W}^{\mathrm{H}} = \hat{\mathbf{A}}^\dagger$. Asymptotically for zero mean noise independent of the sources, this gives $\hat{\mathbf{A}} \to \mathbf{A}$: we converge to the true $\mathbf{A}$-matrix.

Example 6.1. The ZF beamformer satisfies $\mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I}$. Let $\mathbf{w}_1$ be the first column of $\mathbf{W}$, it is the beamformer to receive the first signal. Then

$$\mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I} \qquad \Rightarrow \qquad \mathbf{w}_1^{\mathrm{H}}[\mathbf{a}_2\,, \cdots, \mathbf{a}_d] = [\mathbf{0}\,, \cdots, \mathbf{0}]$$

so that

$$\mathbf{w}_1 \perp \{\mathbf{a}_2\,,\ \cdots\,,\ \mathbf{a}_d\}\,.$$

Thus, $\mathbf{w}_1$ projects out all other sources, except source 1,

$$\begin{aligned} \mathbf{w}_1^{\mathrm{H}}\mathbf{x}(t) &= \textstyle\sum_{i=1}^d \mathbf{w}_1^{\mathrm{H}}\mathbf{a}_i s_i(t) + \mathbf{w}_1^{\mathrm{H}}\mathbf{n}(t) \\ &= s_1(t) + \mathbf{w}_1^{\mathrm{H}}\mathbf{n}(t)\,. \end{aligned}$$

The effect on the noise is not considered. In ill-conditioned cases ($\mathbf{A}$ is ill-conditioned so that its inverse $\mathbf{W}$ may have large entries), $\mathbf{w}_1$ might give a large amplification of the noise.

Deterministic output error minimization The second optimization problem (6.6) minimizes the difference of the output signals to $\mathbf{S}$. For known $\mathbf{S}$, we obtain

$$\mathbf{W}^{\mathrm{H}} = \arg\min_{\mathbf{W}} \|\ \mathbf{W}^{\mathrm{H}}\mathbf{X} - \mathbf{S}\ \|_{\mathrm{F}}^2 = \mathbf{S}\mathbf{X}^\dagger\,. \tag{6.9}$$

Note that $\mathbf{X}^\dagger = \mathbf{X}^{\mathrm{H}}(\mathbf{X}\mathbf{X}^{\mathrm{H}})^{-1}$, so that

$$\mathbf{W}^{\mathrm{H}} = \tfrac{1}{N}\mathbf{S}\mathbf{X}^{\mathrm{H}}(\tfrac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{H}})^{-1} \ = \ \hat{\mathbf{R}}_{xs}^{\mathrm{H}}\hat{\mathbf{R}}_{\mathbf{x}}^{-1}\,, \qquad \mathbf{W} \ = \ \hat{\mathbf{R}}_{\mathbf{x}}^{-1}\hat{\mathbf{R}}_{xs}\,.$$

$\hat{\mathbf{R}}_{\mathbf{x}} := \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{H}}$ is the sample data covariance matrix, and $\hat{\mathbf{R}}_{xs} := \frac{1}{N}(\mathbf{X}\mathbf{S}^{\mathrm{H}})$ is the sample correlation between the sources and the received data.

With known $\mathbf{A}$, note that we cannot solve the minimization problem (6.6) since we can fit any $\mathbf{S}$. We have to put certain assumptions on $\mathbf{S}$, for example the fact that the rows of $\mathbf{S}$ (the signals) are statistically independent from each other and the noise, and hence for large N

$$\tfrac{1}{N}\mathbf{S}\mathbf{S}^{\mathrm{H}} \to \mathbf{I}\,, \qquad \tfrac{1}{N}\mathbf{S}\mathbf{N}^{\mathrm{H}} \to 0$$

(we assumed that the source powers are incorporated in $\mathbf{A}$), so that

$$\hat{\mathbf{R}}_{xs} = \tfrac{1}{N}\mathbf{X}\mathbf{S}^{\mathrm{H}} = \tfrac{1}{N}\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{H}} + \tfrac{1}{N}\mathbf{N}\mathbf{S}^{\mathrm{H}} \quad \rightarrow \quad \mathbf{A}\,.$$

Asymptotically,

$$\mathbf{W} \;\rightarrow\; \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{A}\,,$$

where $\mathbf{R}_{\mathbf{x}} = \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{H}}]$ is the true data covariance matrix.[2] With finite samples, we would set

$$\mathbf{W} = \hat{\mathbf{R}}_{\mathbf{x}}^{-1}\mathbf{A}\,.$$

This is known as the Linear Minimum Mean Square Error (LMMSE) or Wiener receiver. This beamformer maximizes the Signal-to-Interference-plus-Noise Ratio (SINR) at the output. Since it does not cancel all interference, $\mathbf{W}^{\mathrm{H}}\mathbf{A} \neq \mathbf{I}$, the output source estimates are not unbiased. However, it produces estimates of $\mathbf{S}$ with minimal deviation, which is often more relevant.

6.2 STOCHASTIC APPROACH TO MATCHED AND WIENER FILTERS

6.2.1 Performance criteria

Let us now define some performance criteria, based on elementary stochastic assumptions on the data. For the case of a single signal in noise,

$$\mathbf{x}_k = \mathbf{a}s_k + \mathbf{n}_k\,, \qquad y_k = \mathbf{w}^{\mathrm{H}}\mathbf{x}_k = (\mathbf{w}^{\mathrm{H}}\mathbf{a})s_k + (\mathbf{w}^{\mathrm{H}}\mathbf{n}_k)\,.$$

We make the assumptions

$$\mathrm{E}[|s_k|^2] = 1\,, \qquad \mathrm{E}[s_k\mathbf{n}_k^{\mathrm{H}}] = 0\,, \qquad \mathrm{E}[\mathbf{n}_k\mathbf{n}_k^{\mathrm{H}}] = \mathbf{R}_{\mathbf{n}}\,,$$

so that

$$\mathrm{E}[|y|^2] = (\mathbf{w}^{\mathrm{H}}\mathbf{a})(\mathbf{a}^{\mathrm{H}}\mathbf{w}) + \mathbf{w}^{\mathrm{H}}\mathbf{R}_{\mathbf{n}}\mathbf{w}\,.$$

The Signal to Noise Ratio (SNR) at the output can then be defined as

$$\mathrm{SNR}_{out}(\mathbf{w}) = \frac{\mathrm{E}[|(\mathbf{w}^{\mathrm{H}}\mathbf{a})s_k|^2]}{\mathrm{E}[|\mathbf{w}^{\mathrm{H}}\mathbf{n}_k|^2]} = \frac{\mathbf{w}^{\mathrm{H}}\mathbf{a}\mathbf{a}^{\mathrm{H}}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{R}_{\mathbf{n}}\mathbf{w}}\,.$$

With d signals (signal 1 of interest, the others considered interferers), we can write

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \mathbf{a}_1 s_{1,k} + \mathbf{A}'\mathbf{s}'_k + \mathbf{n}_k\,, \qquad y = \mathbf{w}^{\mathrm{H}}\mathbf{x}_k = (\mathbf{w}^{\mathrm{H}}\mathbf{a}_1)s_{1,k} + \mathbf{w}^{\mathrm{H}}\mathbf{A}'\mathbf{s}'_k + (\mathbf{w}^{\mathrm{H}}\mathbf{n}_k)\,,$$

[2] We thus see that even if we adopt a deterministic framework, we cannot avoid to make certain stochastic assumptions on the data.

where $\mathbf{A}'$ contains the columns of $\mathbf{A}$ except for the first one, and similarly for $\mathbf{s}'_k$. Now we can define two criteria: the Signal to Interference Ratio (SIR), and the Signal to Interference plus Noise Ratio (SINR):

$$\begin{aligned} \mathrm{sir}_1(\mathbf{w}) &:= \frac{\mathbf{w}^{\mathrm{H}}(\mathbf{a}_1\mathbf{a}_1^{\mathrm{H}})\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{w}} = \frac{\mathbf{w}^{\mathrm{H}}(\mathbf{a}_1\mathbf{a}_1^{\mathrm{H}})\mathbf{w}}{\mathbf{w}^{\mathrm{H}}(\mathbf{A}\mathbf{A}^{\mathrm{H}} - \mathbf{a}_1\mathbf{a}_1^{\mathrm{H}})\mathbf{w}} \\ \mathrm{sinr}_1(\mathbf{w}) &:= \frac{\mathbf{w}^{\mathrm{H}}(\mathbf{a}_1\mathbf{a}_1^{\mathrm{H}})\mathbf{w}}{\mathbf{w}^{\mathrm{H}}(\mathbf{A}'\mathbf{A}'^{\mathrm{H}} + \mathbf{R_n})\mathbf{w}} = \frac{\mathbf{w}^{\mathrm{H}}(\mathbf{a}_1\mathbf{a}_1^{\mathrm{H}})\mathbf{w}}{\mathbf{w}^{\mathrm{H}}(\mathbf{A}\mathbf{A}^{\mathrm{H}} - \mathbf{a}_1\mathbf{a}_1^{\mathrm{H}} + \mathbf{R_n})\mathbf{w}} . \end{aligned} \tag{6.10}$$

For the Zero-Forcing receiver, we have by definition (for known $\mathbf{A}$)

$$\mathbf{W}^{\mathrm{H}}\mathbf{A} = \mathbf{I} \qquad \Rightarrow \qquad \mathbf{w}_1^{\mathrm{H}}\mathbf{A} = [1,\ 0,\ \cdots,\ 0] \qquad \Rightarrow \qquad \mathbf{w}^{\mathrm{H}}\mathbf{a}_1 = 1\,, \quad \mathbf{w}^{\mathrm{H}}\mathbf{A}' = [0,\ \cdots,\ 0]\,,$$

and it follows that $\mathrm{sir}_1(\mathbf{w}_1) = \infty$. When $\mathbf{W}$ is estimated from a known $\mathbf{S}$, the ZF receiver still maximizes the SIR, but it is not infinity anymore.

Note that (6.10) defines only the performance with respect to the first signal. If we want to receive all signals, we need to define a performance vector, with entries for each signal,

$$\begin{aligned} \mathrm{SIR}(\mathbf{W}) &:= [\mathrm{sir}_1(\mathbf{w}_1) \quad \cdots \quad \mathrm{sir}_d(\mathbf{w}_d)] \\ \mathrm{SINR}(\mathbf{W}) &:= [\mathrm{sinr}_1(\mathbf{w}_1) \quad \cdots \quad \mathrm{sinr}_d(\mathbf{w}_d)]\,. \end{aligned}$$

In graphs, we would usually plot only the worst performance of each vector, or the average of each vector.

6.2.2 Stochastic derivations (white noise)

We now show how the same ZF and Wiener receivers can be derived when starting from a stochastic formulation, but considering the signals deterministic.

Stochastic model matching Assume a model with d sources,

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k \quad (k = 1, \cdots, N) \qquad \Leftrightarrow \qquad \mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}\,.$$

Suppose that $\mathbf{s}_k$ is deterministic, and that the noise samples are independent and identically distributed in time (temporally white), and spatially white ($\mathbf{R_n} = \mathbf{I}$) and jointly complex Gaussian distributed, so that $\mathbf{n}_k$ has a probability density

$$\mathbf{n}_k \sim \mathcal{CN}(0, \sigma^2\mathbf{I}) \qquad \Leftrightarrow \qquad p(\mathbf{n}_k) = \frac{1}{\sqrt{\pi}\sigma} e^{-\frac{\|\mathbf{n}_k\|^2}{\sigma^2}}\,.$$

Because of temporal independence, the probability distribution of N samples is the product of the individual probability distributions,

$$p(\mathbf{N}) = \prod_{k=1}^{N} \frac{1}{\sqrt{\pi}\sigma} e^{-\frac{\|\mathbf{n}_k\|^2}{\sigma^2}}\,.$$

Since $\mathbf{n}_k = \mathbf{x}_k - \mathbf{A}\mathbf{s}_k$, the probability to receive a certain vector $\mathbf{x}_k$ (with a known or deterministic $\mathbf{s}_k$) is thus

$$p(\mathbf{x}_k|\mathbf{s}_k) = \frac{1}{\sqrt{\pi}\sigma} e^{-\frac{\|\mathbf{x}_k - \mathbf{A}\mathbf{s}_k\|^2}{\sigma^2}}$$

and hence

$$p(\mathbf{X}|\mathbf{S}) = \prod_{k=1}^{N} \frac{1}{\sqrt{\pi}\sigma} e^{-\frac{\|\mathbf{x}_k - \mathbf{A}\mathbf{s}_k\|^2}{\sigma^2}} = \left(\frac{1}{\sqrt{\pi}\sigma}\right)^N e^{-\frac{\sum_{k=1}^{N}\|\mathbf{x}_k - \mathbf{A}\mathbf{s}_k\|^2}{\sigma^2}} = \left(\frac{1}{\sqrt{\pi}\sigma}\right)^N e^{-\frac{\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2}{\sigma^2}} .$$

$p(\mathbf{X}|\mathbf{S})$ is called the *likelihood* of receiving a certain data matrix $\mathbf{X}$, for a certain transmitted data matrix $\mathbf{S}$. It is of course a probability density function, but in the likelihood interpretation we regard it as a function of $\mathbf{S}$, for an actual received data matrix $\mathbf{X}$. The *Deterministic Maximum Likelihood* technique estimates $\mathbf{S}$ as that matrix that maximizes the likelihood of having received the actual received $\mathbf{X}$, thus

$$\hat{\mathbf{S}} = \arg\max_{\mathbf{S}} \left(\frac{1}{\sqrt{\pi}\sigma}\right)^N e^{-\frac{\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2}{\sigma^2}} . \tag{6.11}$$

If we take the negative logarithm of $p(\mathbf{X}|\mathbf{S})$, we obtain what is called the negative log-likelihood function. Since it is a monotonously growing function, taking the logarithm does not change the location of the maximum. The maximization problem then becomes a minimization over $\text{const} + \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2/\sigma^2$, or

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_{\mathrm{F}}^2 . \tag{6.12}$$

This is the same model fitting problem as we had before in (6.5). Thus, the deterministic ML problem is equivalent to the LS model fitting problem in the case of white Gaussian noise.

Stochastic output error minimization In a statistical framework, the output error problem (6.6) becomes

$$\min_{\mathbf{w}} \mathrm{E}[|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k - s_k|^2] .$$

The cost function is known as the Linear Minimum Mean Square Error. It can be worked out as follows:

$$\begin{aligned} J(\mathbf{w}) &= \mathrm{E}[|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k - s_k|^2] \\ &= \mathbf{w}^{\mathrm{H}}\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathrm{H}}]\mathbf{w} - \mathbf{w}^{\mathrm{H}}\mathrm{E}[\mathbf{x}\bar{s}_k] - \mathrm{E}[s_k\mathbf{x}^{\mathrm{H}}]\mathbf{w} + \mathrm{E}[|s_k|^2] . \end{aligned}$$

At this point, note that there is a question whether we regard s_k as a stochastic variable or deterministic. If s_k is stochastic with $\mathrm{E}[|s_k|^2] = 1$, then

$$J(\mathbf{w}) = \mathbf{w}^{\mathrm{H}}\mathbf{R}_{\mathbf{x}}\mathbf{w} - \mathbf{w}^{\mathrm{H}}\mathbf{a} - \mathbf{a}^{\mathrm{H}}\mathbf{w} + 1 .$$

If s_k is deterministic, then $J = J_k$ depends on s_k, and we need to work with an average over N samples, $\bar{J} = \frac{1}{N}\sum_k J_k$. For large N and i.i.d. aassumptions on $\mathbf{s}_k$, the result will be the same.

Now differentiate with respect to $\mathbf{w}$. This is a bit tricky since $\mathbf{w}$ is complex and functions of complex variables may not be differentiable (a simple example of a non-analytic function is $f(z) = \bar{z}$). There are various approaches (e.g. [1, 8]). A consistent approach is to regard $\mathbf{w}$ and $\mathbf{w}^*$ as *independent* variables. Let $\mathbf{w} = \mathbf{u} + j\mathbf{v}$ with $\mathbf{u}$ and $\mathbf{v}$ real-valued, then the complex gradients to $\mathbf{w}$ and $\mathbf{w}^*$ are defined as [8]

$$\nabla_{\mathbf{w}} J = \frac{1}{2}\left(\nabla_{\mathbf{u}} J + j\nabla_{\mathbf{v}} J\right) = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial u_1} J \\ \vdots \\ \frac{\partial}{\partial u_d} J \end{bmatrix} + \frac{1}{2} j \begin{bmatrix} \frac{\partial}{\partial v_1} J \\ \vdots \\ \frac{\partial}{\partial v_d} J \end{bmatrix}$$

$$\nabla_{\mathbf{w}^*} J = \frac{1}{2}\left(\nabla_{\mathbf{u}} J - j\nabla_{\mathbf{v}} J\right) = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial u_1} J \\ \vdots \\ \frac{\partial}{\partial u_d} J \end{bmatrix} - \frac{1}{2} j \begin{bmatrix} \frac{\partial}{\partial v_1} J \\ \vdots \\ \frac{\partial}{\partial v_d} J \end{bmatrix}$$

with properties

$$\nabla_{\mathbf{w}} \mathbf{w}^{\mathrm{H}} \mathbf{a} = \mathbf{0}\,, \quad \nabla_{\mathbf{w}} \mathbf{a}^{\mathrm{H}} \mathbf{w} = \mathbf{a}^*\,, \quad \nabla_{\mathbf{w}} \mathbf{w}^{\mathrm{H}} \mathbf{R} \mathbf{w} = \mathbf{R}^{\mathrm{T}} \mathbf{w}^*$$
$$\nabla^*_{\mathbf{w}} \mathbf{w}^{\mathrm{H}} \mathbf{a} = \mathbf{a}\,, \quad \nabla^*_{\mathbf{w}} \mathbf{a}^{\mathrm{H}} \mathbf{w} = \mathbf{0}\,, \quad \nabla^*_{\mathbf{w}} \mathbf{w}^{\mathrm{H}} \mathbf{R} \mathbf{w} = \mathbf{R}\mathbf{w}$$

It can further be shown that for a stationary point, it is necessary and sufficient that either $\nabla_{\mathbf{w}} J = \mathbf{0}$ or that $\nabla^*_{\mathbf{w}} J = \mathbf{0}$: the two are equivalent. Since the latter expression is more simple, and because it specifies the maximal rate of change, we keep from now on the definition for the gradient

$$\nabla J(\mathbf{w}) \equiv \nabla^*_{\mathbf{w}} J(\mathbf{w})\,, \tag{6.13}$$

and we obtain

$$\nabla J(\mathbf{w}) = \mathbf{R}_{\mathbf{x}} \mathbf{w} - \mathbf{a}\,.$$

The minimum of $J(\mathbf{w})$ is attained for

$$\nabla_{\mathbf{w}} J = \mathbf{0} \qquad \Rightarrow \qquad \mathbf{w} = \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{a}\,.$$

We thus obtain the Wiener receiver.

The LMMSE cost function is also called Minimum Variance. This is in fact a misnomer: the expression is not really that of a variance because the error $\mathrm{E}[\mathbf{w}^{\mathrm{H}} \mathbf{x}_k - s_k] \neq 0$. In fact, for the Wiener receiver, a single signal in noise, and s_k considered deterministic ($\mathrm{E}[s_k] = s_k$),

$$\begin{aligned} \mathrm{E}[y_k] &= \mathrm{E}[\mathbf{w}^{\mathrm{H}} \mathbf{x}_k] \\ &= \mathrm{E}[\mathbf{a}^{\mathrm{H}} \mathbf{R}_{\mathbf{x}}^{-1} (\mathbf{a} s_k + \mathbf{n}_k)] \\ &= \mathbf{a}^{\mathrm{H}} \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{a} s_k \\ &= \mathbf{a}^{\mathrm{H}} (\mathbf{a}\mathbf{a}^{\mathrm{H}} + \sigma^2 \mathbf{I})^{-1} \mathbf{a} s_k \\ &= \mathbf{a}^{\mathrm{H}} \mathbf{a} (\mathbf{a}^{\mathrm{H}} \mathbf{a} + \sigma^2)^{-1} s_k \\ &= \frac{\mathbf{a}^{\mathrm{H}} \mathbf{a}}{\mathbf{a}^{\mathrm{H}} \mathbf{a} + \sigma^2} s_k\,. \end{aligned}$$

Thus, the expected value of the output is not s_k, but a scaled-down version of it.

6.2.3 Colored noise

Let us now see what changes in the above when the noise is not white, but has a variance

$$\mathrm{E}[\mathbf{n}\mathbf{n}^{\mathrm{H}}] = \mathbf{R}_{\mathbf{n}}\,.$$

We assume that we know the variance. In that case, we can *prewhiten* the data with a square-root factor $\mathbf{R}_{\mathbf{n}}^{-1/2}$:

$$\begin{array}{rcccccc} \mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k \qquad \Rightarrow \qquad \underbrace{\mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{x}_k}_{\underline{\mathbf{x}}_k} & = & \underbrace{\mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{A}}_{\underline{\mathbf{A}}}\,\mathbf{s}_k & + & \underbrace{\mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{n}_k}_{\underline{\mathbf{n}}_k} \\ & = & \underline{\mathbf{A}}\mathbf{s}_k & + & \underline{\mathbf{n}}_k \end{array}$$

Note that now

$$\underline{\mathbf{R}}_{\mathbf{n}} = \mathrm{E}[\underline{\mathbf{n}}_k\underline{\mathbf{n}}_k{}^{\mathrm{H}}] = \mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{R}_{\mathbf{n}}\mathbf{R}_{\mathbf{n}}^{-1/2} = \mathbf{I}$$

so that the noise $\underline{\mathbf{n}}_k$ is white. At this point, we are back on familiar grounds. The ZF equalizer becomes

$$\begin{aligned} \mathbf{s}_k = \underline{\mathbf{A}}^{\dagger H}\underline{\mathbf{x}}_k = (\underline{\mathbf{A}}^{\mathrm{H}}\mathbf{A})^{-1}\underline{\mathbf{A}}^{\mathrm{H}}\underline{\mathbf{x}}_k = (\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathbf{n}}^{-1}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathbf{n}}^{-1}\mathbf{x}_k \\ \Rightarrow \qquad \mathbf{W} = \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{A}(\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathbf{n}}^{-1}\mathbf{A})^{-1} \end{aligned} \tag{6.14}$$

The Wiener receiver on the other hand will be the same, since $\mathbf{R}_{\mathbf{n}}$ is not used at all in the derivation. This can also be checked:

$$\begin{array}{rcl} \underline{\mathbf{W}} & = & \underline{\mathbf{R}}_{\mathbf{x}}^{-1}\underline{\mathbf{A}} = (\mathbf{R}_n^{-1/2}\mathbf{R}_{\mathbf{x}}\mathbf{R}_{\mathbf{n}}^{-1/2})^{-1}\mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{A} = \mathbf{R}_{\mathbf{n}}^{1/2}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{A} \\ \Rightarrow \quad \mathbf{W} & = & \mathbf{R}_{\mathbf{n}}^{-1/2}\underline{\mathbf{W}} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{A}\,. \end{array}$$

6.3 OTHER INTERPRETATIONS OF MATCHED FILTERING

6.3.1 Maximum Ratio Combining

Consider a special case of the previous, a single signal in white noise,

$$\mathbf{x}_k = \mathbf{a}s_k + \mathbf{n}_k\,, \qquad \mathrm{E}[\mathbf{n}_k\mathbf{n}_k^{\mathrm{H}}] = \sigma^2\mathbf{I}\,.$$

As we showed before, the ZF beamformer is given by

$$\mathbf{w} = \mathbf{a}(\mathbf{a}^{\mathrm{H}}\mathbf{a})^{-1} = \gamma_1\mathbf{a}$$

where γ_1 is a scalar. Since a scalar multiplication does not change the output SNR, the optimal beamformer for s in this case is given by

$$\mathbf{w}_{MF} = \mathbf{a}$$

which is known as a *matched filter* or a *classical beamformer*. It is also known as *Maximum Ratio Combining* (MRC).

With non-white noise,

$$\mathbf{x}_k = \mathbf{a}s_k + \mathbf{n}_k\,, \qquad \mathrm{E}[\mathbf{n}\mathbf{n}^{\mathrm{H}}] = \mathbf{R}_{\mathbf{n}}$$

we have seen in (6.14) that

$$\mathbf{w} = \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a}(\mathbf{a}^{\mathrm{H}}\mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a})^{-1} = \gamma_2\mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a}\,.$$

Thus, the matched filter in non-white noise is

$$\mathbf{w}_{MF} = \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a}\,.$$

We can proceed similarly with the Wiener receiver. In white noise,

$$\begin{aligned}\mathbf{w} &= \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a} \\ &= (\mathbf{a}\mathbf{a}^{\mathrm{H}} + \sigma^2\mathbf{I})^{-1}\mathbf{a} \\ &= \mathbf{a}(\mathbf{a}^{\mathrm{H}}\mathbf{a} + \sigma^2)^{-1} \qquad \sim \quad \mathbf{a}\,.\end{aligned}$$

It is equal to a multiple of the matched filter. In colored noise, we whiten to apply the white noise result:

$$\begin{aligned}\mathbf{w} &= \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a} \\ &= (\mathbf{a}\mathbf{a}^{\mathrm{H}} + \mathbf{R}_{\mathbf{n}})^{-1}\mathbf{a} \\ &= \mathbf{R}_{\mathbf{n}}^{-1/2}(\underline{\mathbf{a}}\underline{\mathbf{a}}^{\mathrm{H}} + \mathbf{I})^{-1}\underline{\mathbf{a}} \qquad (\underline{\mathbf{a}} = \mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{a}) \\ &= \mathbf{R}_{\mathbf{n}}^{-1/2}\underline{\mathbf{a}}(\underline{\mathbf{a}}^{\mathrm{H}}\underline{\mathbf{a}} + 1)^{-1} \\ &\sim \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a}\,.\end{aligned}$$

This is equal to a multiple of the matched filter for colored noise.

The colored noise case is relevant also for the following reason: with more than one signal, we can write the model as

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \mathbf{a}_1 s_k + (\mathbf{A}'\mathbf{s}'_k + \mathbf{n}_k)\,.$$

This is of the form

$$\mathbf{x}_k = \mathbf{a}s_k + \mathbf{n}_k\,, \qquad \mathbf{R}_{\mathbf{n}} = \mathbf{A}'\mathbf{A}'^{H} + \sigma^2\mathbf{I}$$

where the noise is now not white, but colored due to the contribution of the interfering sources.

The conclusion is quite interesting:

> For the reception of a single source out of interfering sources plus noise, the Zero-Forcing receiver, Matched Filter or MRC: $\mathbf{w} = \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a}$, and the Wiener receiver: $\mathbf{w} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}$, are asymptotically all equal to a scalar multiple of each other, and hence will asymptotically give the same performance.

It should be stressed that this equivalence is only an asymptotic result (large N), because the interfering sources are not regarded deterministic sources, but stochastic. In finite samples, the corresponding receivers

$$\mathbf{w}_{ZF} = \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{a}\,, \qquad \mathbf{w}_{Wiener} = \hat{\mathbf{R}}_{\mathbf{x}}^{-1}\mathbf{a}$$

will be different. Note that $\mathbf{R_n}$ is assumed to be known, whereas $\hat{\mathbf{R}}_\mathbf{x}$ is estimated from the received data.

The above are examples of *non-joint receivers*: the interference is lumped together with the noise, and there might as well be many more interferers than antennas. Improved performance may be possible by a *joint* estimation of the collection of receivers for all sources.

Example 6.2. Consider a single source in white noise:

$$\mathbf{x}(t) = \mathbf{a}(\theta)s(t) + \mathbf{n}(t)\,, \qquad \mathbf{R_n} = \sigma^2\mathbf{I}\,.$$

Suppose the signal is normalized to have power $\mathrm{E}[|s|^2] = \sigma_s^2$. Then

$$\mathrm{SNR}_{in} = \frac{\sigma_s^2}{\sigma^2}\,.$$

This is the SNR at each element of the array. Suppose all entries of $\mathbf{a}(\theta)$ have unit norm, $|a_i(\theta)| = 1$. With M antennas, $\mathbf{a}(\theta)^{\mathrm{H}}\mathbf{a}(\theta) = M$.

If we choose the matched filter, or MRC, i.e., $\mathbf{w} = \mathbf{a}(\theta)$, then

$$y(t) = \mathbf{w}^{\mathrm{H}}\mathbf{x}(t) = \mathbf{a}^{\mathrm{H}}\mathbf{a}s(t) + \mathbf{a}^{\mathrm{H}}\mathbf{n}(t) = Ms(t) + \mathbf{a}^{\mathrm{H}}\mathbf{n}(t)$$

then

$$\mathrm{SNR}_{out} = \frac{M^2\sigma_s^2}{\mathbf{a}^{\mathrm{H}}\sigma^2\mathbf{I}\mathbf{a}} = \frac{M^2\sigma_s^2}{M\sigma^2} = M\cdot\mathrm{SNR}_{in}\,.$$

The factor M is the array gain.

6.3.2 Maximizing the output SNR

For a single signal in noise, the matched filter $\mathbf{w} = \mathbf{R_n^{-1}}\mathbf{a}$ maximizes the output SNR. This is derived as follows. Similarly as in the preceding example, we have

$$\mathbf{x}(t) = \mathbf{a}s(t) + \mathbf{n}(t)\,.$$

Define $\mathrm{E}[|s|^2] = \sigma_s^2$, then $\mathbf{R_x} = \mathbf{R_s} + \mathbf{R_n}$, with

$$\mathbf{R_s} = \sigma_s^2\mathbf{a}\mathbf{a}^{\mathrm{H}}\,, \qquad \mathbf{R_n} = \mathrm{E}[\mathbf{n}\mathbf{n}^{\mathrm{H}}]\,.$$

The output SNR after beamforming is equal to

$$\mathrm{SNR}_{out}(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{H}}\mathbf{R_s}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{R_n}\mathbf{w}}\,.$$

We now would like to find the beamformer that maximizes SIR_{out}, i.e.,

$$\mathbf{w} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^{\mathrm{H}}\mathbf{R_s}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{R_n}\mathbf{w}}\,.$$

The expression is known as a Rayleigh quotient, and the solution is known to be given by the solution of the eigenvalue equation

$$\mathbf{R_n^{-1}R_s w} = \lambda_{\max}\mathbf{w}\,. \tag{6.15}$$

This can be seen as follows: suppose that $\mathbf{R_n} = \mathbf{I}$, then the equation is

$$\max_{\mathbf{w}} \ \mathbf{w}^{\mathrm{H}}\mathbf{R_s w}\,.$$

Introduce an eigenvalue decomposition for $\mathbf{R_s} = \mathbf{U\Lambda U}^{\mathrm{H}}$, then

$$\max_{\mathbf{w}} \ (\mathbf{w}^{\mathrm{H}}\mathbf{U})\mathbf{\Lambda}(\mathbf{U}^{\mathrm{H}}\mathbf{w})\,.$$

Let λ_1 be the largest eigenvalue ($1,1$-entry of $\mathbf{\Lambda}$), then it is clear that the maximum of the expression is given by choosing $\mathbf{w}^{\mathrm{H}}\mathbf{U} = [1 \quad 0 \cdots 0]$. Thus, the optimal $\mathbf{w}$ is the eigenvector corresponding to the largest eigenvalue, and satisfies the eigenvalue equation $\mathbf{R_s w} = \lambda_1 \mathbf{w}$. If $\mathbf{R_n} \neq \mathbf{I}$, then we can first whiten the noise to obtain the result in (6.15).

The solution of (6.15) can be found in closed form, by inserting $\mathbf{R_s} = \sigma_s^2\mathbf{a}\mathbf{a}^{\mathrm{H}}$. We obtain

$$\begin{aligned}
& & \mathbf{R_n^{-1}R_s w} &= \lambda_{\max}\mathbf{w} \\
\Leftrightarrow & & \sigma_s^2\mathbf{R_n^{-1}a}\mathbf{a}^{\mathrm{H}}\mathbf{w} &= \lambda_{\max}\mathbf{w} \\
\Leftrightarrow & \sigma_s^2(\mathbf{R_n^{-1/2}a})(\mathbf{a}^{\mathrm{H}}\mathbf{R_n^{-1/2}})(\mathbf{R_n^{1/2}w}) &= \lambda_{\max}(\mathbf{R_n^{1/2}w}) \\
\Leftrightarrow & & \sigma_s^2\underline{\mathbf{a}}\underline{\mathbf{a}}^{\mathrm{H}}\underline{\mathbf{w}} &= \lambda_{\max}\underline{\mathbf{w}} \\
\Leftrightarrow & & \underline{\mathbf{w}} &= \underline{\mathbf{a}}\,, \quad \lambda_{\max} = \sigma_s^2\underline{\mathbf{a}}^{\mathrm{H}}\underline{\mathbf{a}}
\end{aligned}$$

and it follows that

$$\mathbf{w} = \mathbf{R_n^{-1}a}$$

which is, as promised, the matched filter in colored noise.

6.3.3 LCMV – MVDR – GSC – Capon

A related technique for beamforming is the so-called *Linearly constrained Minimum Variance* (LCMV), also known as *Minimum Variance Distortionless Response* (MVDR), *Generalized Sidelobe Canceling* (GSC), and *Capon beamforming* (in the French literature). In this technique, it is again assumed that we have a single source in colored noise (this might contain other interferers as well),

$$\mathbf{x}_k = \mathbf{a}s_k + \mathbf{n}_k\,.$$

If $\mathbf{a}$ is known, then the idea is that we constrain the beamformer $\mathbf{w}$ to

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

i.e., we have a fixed response towards the source. The remaining freedom is used to minimize the total output power ("response" or "variance") after beamforming,

$$\min_{\mathbf{w}} \ \mathbf{w}^{\mathrm{H}}\mathbf{R_x w} \qquad \text{such that} \quad \mathbf{w}^{\mathrm{H}}\mathbf{a} = 1\,.$$

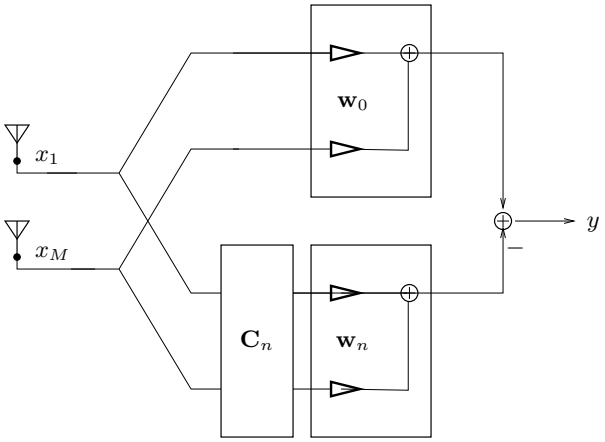


Figure 6.1. The Generalized Sidelobe Canceler

The solution can be found in closed form using Lagrange multipliers and is given by

$$\mathbf{w} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}(\mathbf{a}^{\mathrm{H}}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a})^{-1}\,.$$

Thus, $\mathbf{w}$ is a scalar multiple of the Wiener receiver.

This case may be generalized by introducing a constraint matrix $\mathbf{C}:\ \ M \times L\ (M > L)$ and L-dimensional vector $\mathbf{f}$, and asking for $\mathbf{C}^{\mathrm{H}}\mathbf{w} = \mathbf{f}$. The solution to

$$\min_{\mathbf{w}}\ \mathbf{w}^{\mathrm{H}}\mathbf{R}_{\mathbf{x}}\mathbf{w} \qquad \text{such that} \quad \mathbf{C}^{\mathrm{H}}\mathbf{w} = \mathbf{f}$$

is given by

$$\mathbf{w} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{C}(\mathbf{C}^{\mathrm{H}}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{C})^{-1}\mathbf{f}\,.$$

Generalized Sidelobe Canceler The generalized sidelobe canceler (GSC) represents an alternative formulation of the LCMV problem, which provides insight, is useful for analysis, and can simplify LCMV beamformer implementation. Essentially, it is a technique to convert a constrained minimization problem into an unconstrained form. Suppose we decompose the weight vector $\mathbf{w}$ into two orthogonal components, $\mathbf{w}_0$ and $-\mathbf{v}$ ($\mathbf{w} = \mathbf{w}_0 - \mathbf{v}$), that lie in the range and null space of $\mathbf{C}$ and $\mathbf{C}^{\mathrm{H}}$, respectively. These subspaces span the entire space so this decomposition can be used to represent any $\mathbf{w}$. Since $\mathbf{C}^{\mathrm{H}}\mathbf{v} = \mathbf{0}$, we must have

$$\mathbf{w}_0 = \mathbf{C}(\mathbf{C}^{\mathrm{H}}\mathbf{C})^{-1}\mathbf{f} \tag{6.16}$$

if $\mathbf{w}$ is to satisfy the constraints. (6.16) is the minimum norm solution to the under-determined system $\mathbf{C}^{\mathrm{H}}\mathbf{w}_0 = \mathbf{f}$. The vector $\mathbf{v}$ is a linear combination of the columns of an $M \times (M - L)$ matrix $\mathbf{C}_n$, and $\mathbf{v} = \mathbf{C}_n\mathbf{w}_n$; provided the columns of $\mathbf{C}_n$ form a basis for the null space of $\mathbf{C}$. The matrix $\mathbf{C}_n$ can be obtained from $\mathbf{C}$ using any of several orthogonalization procedures, for

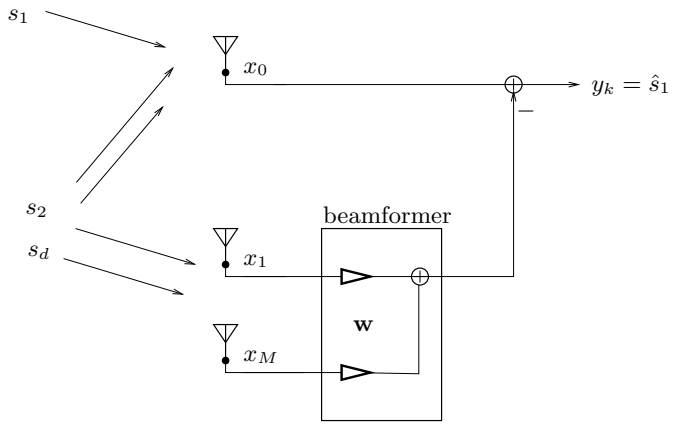


Figure 6.2. The Multiple Sidelobe Canceller: interference is estimated from a reference antenna array and subtracted from the primary antenna x_0.

example the QR factorization or the SVD. The structure of the beamformer using the weight vector $\mathbf{w} = \mathbf{w}_0 - \mathbf{C}_n\mathbf{w}_n$ is depicted in Fig. 6.1. The choice for $\mathbf{w}_0$ and $\mathbf{C}_n$ implies that $\mathbf{w}$ satisfies the constraints independent of $\mathbf{w}_n$ and reduces the LCMV problem to the unconstrained problem

$$\min_{\mathbf{w}_n} \left[\mathbf{w}_0 - \mathbf{C}_n\mathbf{w}_n\right]^{\mathrm{H}} \mathbf{R}_{\mathbf{x}} [\mathbf{w}_0 - \mathbf{C}_n\mathbf{w}_n] \,.$$

The solution is

$$\mathbf{w}_n = (\mathbf{C}_n^{\mathrm{H}}\mathbf{R}_{\mathbf{x}}\mathbf{C}_n)^{-1}\mathbf{C}_n^{\mathrm{H}}\mathbf{R}_{\mathbf{x}}\mathbf{w}_0 \,.$$

The primary advantage of this implementation stems from the fact that the weights $\mathbf{w}_n$ are unconstrained and a data independent beamformer $\mathbf{w}_0$ is implemented as an integral part of the adaptive beamformer. The unconstrained nature of the adaptive weights permits much simpler adaptive algorithms to be employed and the data independent beamformer is useful in situations where adaptive signal cancellation occurs.

Example 6.3. *Reference channels – Multiple sidelobe canceler*

A special case of the LCMV is that where there is a primary channel $x_0(t)$, receiving a signal of interest plus interferers and noise, and a collection of reference antennas $\mathbf{x}(t)$, receiving only interference and noise. For example, in hands-free telephony in a car, we may have a microphone close to the speaker, and other microphones further away from the speaker and closer to the engine and other noise sources. Or we may have a directional antenna (parabolic dish) and an array of omnidirectional antennas. The objective is to subtract from the primary channel a linear combination of the reference antennas such that the output power is minimized. If indeed the signal of interest is not present on the reference antennas, the SINR of this signal will be

improved. (If the signal *is* present at the reference antennas, then of course it will be canceled as well!)

Call the primary sensor signal x_0 and the reference signal vector $\mathbf{x}$. Then the objective is

$$\min_{\mathbf{w}} \mathrm{E}\|x_0 - \mathbf{w}^{\mathrm{H}}\mathbf{x}\|^2 .$$

The solution of this problem is given by

$$\mathbf{w} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a} \qquad \mathbf{a} := \mathrm{E}[\mathbf{x}\bar{x}_0] .$$

This technique is called the Multiple Sidelobe Canceler (Applebaum 1976). It is a special case of the LCMV beamformer, which becomes clear if we construct a joint data vector

$$\mathbf{x}' = \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} , \qquad \mathbf{w}' = \begin{bmatrix} 1 \\ -\mathbf{w} \end{bmatrix} , \qquad \mathbf{c} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$$

The constraint is $(\mathbf{w}')^{\mathrm{H}}\mathbf{c} = 1$.

6.4 PREWHITENING FILTER STRUCTURE

Subspace-based prefiltering In the noise-free case with less sources than sensors, $\mathbf{X} = \mathbf{AS}$ is rank deficient: its rank is d (the number of signals) rather than m (the number of sensors). As a consequence, once we have found a beamformer $\mathbf{w}$ such that $\mathbf{w}^{\mathrm{H}}\mathbf{X} = \mathbf{s}$, one of the source signals, then we can add any vector $\mathbf{w}_0$ such that $\mathbf{w}_0^{\mathrm{H}}\mathbf{X} = \mathbf{0}$ to $\mathbf{w}$, and obtain the same output. The beamforming solutions are not unique.

The desired beamforming solutions are all in the column span of $\mathbf{A}$. Indeed, any component orthogonal to this span will not contribute at the output. The most easy way to ensure that our solutions will be in this span is by performing a *dimension-reducing* prefiltering. Let $\mathbf{F}$ be any $M \times d$ matrix such that $\mathrm{span}(\mathbf{F}) = \mathrm{span}(\mathbf{A})$. Then all beamforming matrices $\mathbf{W}$ in the column span of $\mathbf{A}$ are given by

$$\mathbf{W} = \mathbf{F}\underline{\mathbf{W}}$$

where $\underline{\mathbf{W}}$ is a $d \times d$ matrix, nonsingular if the beamformers are linearly independent. We will use the underscore to denote prefiltered variables. Thus, the prefiltered noisy data matrix is

$$\underline{\mathbf{X}} := \mathbf{F}^{\mathrm{H}}\mathbf{X}$$

with structure

$$\underline{\mathbf{X}} = \underline{\mathbf{A}}\mathbf{S} + \underline{\mathbf{N}} , \qquad \text{where} \quad \underline{\mathbf{A}} := \mathbf{F}^{\mathrm{H}}\mathbf{A} , \;\; \underline{\mathbf{N}} := \mathbf{F}^{\mathrm{H}}\mathbf{N} .$$

$\underline{\mathbf{X}}$ has only d channels, and is such that $\mathbf{W}^{\mathrm{H}}\mathbf{X} = \underline{\mathbf{W}}^{\mathrm{H}}\underline{X}$. Thus, the columns of $\underline{\mathbf{W}}$ are d-dimensional beamformers on the prefiltered data $\underline{\mathbf{X}}$, and for any choice of $\underline{\mathbf{W}}$ the columns of the effective beamformer $\mathbf{W}$ are all in the column span of $\mathbf{A}$, as desired.

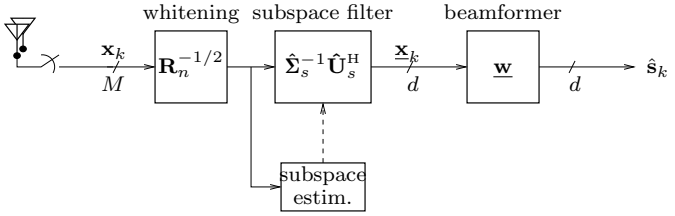


Figure 6.3. Beamforming prefiltering structure

To describe the column span of $\mathbf{A}$, introduce the "economy-size" singular value decomposition of $\mathbf{A}$,

$$\mathbf{A} = \mathbf{U}_A \mathbf{\Sigma}_A \mathbf{V}_A^{\mathrm{H}}$$

where we take $\mathbf{U}_A : m \times d$ with orthonormal columns, $\mathbf{\Sigma}_A : d \times d$ diagonal containing the nonzero singular values of $\mathbf{A}$, and $\mathbf{V}_A : d \times d$ unitary. Also let $\mathbf{U}_A^{\perp}$ be the orthonormal complement of $\mathbf{U}_A$. The columns of $\mathbf{U}_A$ are an orthonormal basis of the column span of $\mathbf{A}$. The point is that even if $\mathbf{A}$ is unknown, $\mathbf{U}_A$ can be estimated from the data, as described below (and in more detail in section 6.5).

We assume that the noise is spatially white, with covariance matrix $\sigma^2\mathbf{I}$. Let $\hat{\mathbf{R}}_{\mathbf{x}} = \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{H}}$ be the noisy sample data covariance matrix, with eigenvalue decomposition

$$\hat{\mathbf{R}}_{\mathbf{x}} \;=\; \hat{\mathbf{U}}\hat{\mathbf{\Lambda}}\hat{\mathbf{U}}^{\mathrm{H}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}^2\hat{\mathbf{U}}^{\mathrm{H}}\,. \tag{6.17}$$

Here, $\hat{\mathbf{U}}$ is $M \times M$ unitary, and $\hat{\mathbf{\Sigma}}$ is $M \times M$ diagonal. Equivalently, these factors follow from an SVD of the data matrix $\mathbf{X}$ directly:

$$\tfrac{1}{\sqrt{N}}\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\mathrm{H}}$$

We collect the d largest singular values into a $d \times d$ diagonal matrix $\hat{\mathbf{\Sigma}}_s$, and collect the corresponding d eigenvectors into $\hat{\mathbf{U}}_s$. Asymptotically, $\mathbf{R}_{\mathbf{x}}$ satisfies $\mathbf{R}_{\mathbf{x}} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \sigma^2\mathbf{I}$, with eigenvalue decomposition

$$\mathbf{R}_{\mathbf{x}} \;=\; \mathbf{U}_A\mathbf{\Sigma}_A^2\mathbf{U}_A^{\mathrm{H}} + \sigma^2\mathbf{I} \;=\; \mathbf{U}_A(\mathbf{\Sigma}_A^2 + \sigma^2 I)\mathbf{U}_A^{\mathrm{H}} \;+\; \sigma^2\mathbf{U}_A^{\perp}\mathbf{U}_A^{\perp H}\,. \tag{6.18}$$

Since $\hat{\mathbf{R}}_{\mathbf{x}} \to \mathbf{R}_{\mathbf{x}}$ as the number of samples N grows, we have that $\hat{\mathbf{U}}_s\hat{\mathbf{\Sigma}}_s^2\hat{\mathbf{U}}_s^{\mathrm{H}} \to \mathbf{U}_A(\mathbf{\Sigma}_A^2 + \sigma^2\mathbf{I})\mathbf{U}_A^{\mathrm{H}}$, so that $\hat{\mathbf{U}}_s$ is an asymptotically unbiased estimate of $\mathbf{U}_A$. Thus $\mathbf{U}_A$ and also $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$ can be estimated consistently from the data, by taking sufficiently many samples. In contrast, $\mathbf{V}_A$ cannot be estimated like this: this factor is on the "inside" of the factorization $\mathbf{A}\mathbf{S} = \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}}\mathbf{S}$ and as long as $\mathbf{S}$ is unknown, any unitary factor can be exchanged between $\mathbf{V}_A$ and $\mathbf{S}$.

Even if we choose $\mathbf{F}$ to have the column span of $\hat{\mathbf{U}}_s$, there is freedom left. As we will show, a natural choice is to combine the dimension reduction with a whitening of the data covariance matrix, i.e., such that $\underline{\mathbf{R}}_{\mathbf{x}} := \frac{1}{N}\underline{\mathbf{X}}\underline{\mathbf{X}}^{\mathrm{H}}$ becomes unity: $\underline{\mathbf{R}}_{\mathbf{x}} = \mathbf{I}$. This is achieved if we define $\mathbf{F}$ as

$$\mathbf{F} = \hat{\mathbf{U}}_s\hat{\mathbf{\Sigma}}_s^{-1}\,. \tag{6.19}$$

Without dimension reduction, $\mathbf{F} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}$ is a square root factor[3] of $\hat{\mathbf{R}}_{\mathbf{x}}^{-1}$, i.e., $\hat{\mathbf{R}}_{\mathbf{x}}^{-1} = \mathbf{F}\mathbf{F}^{\mathrm{H}}$.

If the noise is colored with covariance matrix $\sigma^2\mathbf{R}_{\mathbf{n}}$, where we know $\mathbf{R}_{\mathbf{n}}$ but perhaps not the noise power σ^2, then we first whiten the noise by computing $\mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{X}$, and continue as in the white noise case, by computing an SVD of $\mathbf{R}_{\mathbf{n}}^{-1/2}\mathbf{X}$. The resulting prewhitening/dimension reducing filter is then

$$\mathbf{F} = \mathbf{R}_{\mathbf{n}}^{-1/2}\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}^{-1}\,.$$

The structure of the resulting beamformer is shown in Fig. 6.3.

After this preprocessing, the Wiener filter is simply given by

$$\underline{\mathbf{W}} = \underline{\mathbf{A}}$$

at least asymptotically. Indeed,

$$\mathbf{W} = \mathbf{F}\underline{\mathbf{W}} = \mathbf{F}\mathbf{F}^{\mathrm{H}}\mathbf{A}$$

and asymptotically $\mathbf{F}\mathbf{F}^{\mathrm{H}} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{P}_A$. Since $\mathbf{P}_A\mathbf{A} = \mathbf{A}$, the result follows. For finite samples, the dimension reduction gives a slight difference.

Direct matched filtering Another choice for $\mathbf{F}$ that reduces dimensions and that is often taken if (an estimate of) $\mathbf{A}$ is known is by simply setting

$$\mathbf{F} = \mathbf{A}$$

The output after this filter becomes

$$\underline{\mathbf{X}} = \mathbf{A}^{\mathrm{H}}\mathbf{X} = (\mathbf{A}^{\mathrm{H}}\mathbf{A})\mathbf{S} + \mathbf{A}^{\mathrm{H}}\mathbf{N}$$

The noise is now non-white, it has covariance $\mathbf{A}^{\mathrm{H}}\mathbf{A}$.

We can whiten it by multiplying by a factor $(\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1/2}$. It is more convenient to introduce an SVD $\mathbf{A} = \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}}$, and use a non-symmetrical factor $\mathbf{\Sigma}_A^{-1}\mathbf{V}_A^{\mathrm{H}}$. Note that $\mathbf{\Sigma}_A^{-1}\mathbf{V}_A^{\mathrm{H}}\mathbf{A}^{\mathrm{H}} = \mathbf{U}_A^{\mathrm{H}}$. This gives

$$\underline{\mathbf{X}} = \mathbf{U}_A^{\mathrm{H}}\mathbf{X} = (\mathbf{U}_A^{\mathrm{H}}\mathbf{A})\mathbf{S} + \mathbf{U}_A^{\mathrm{H}}\mathbf{N} = (\mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}})\mathbf{S} + \mathbf{U}_A^{\mathrm{H}}\mathbf{N}\,.$$

The noise is white again, and $\underline{\mathbf{A}} = \mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}}$. If we subsequently want to apply a Wiener receiver in this prefiltered domain, it is given by

$$\underline{\mathbf{W}} = (\underline{\mathbf{A}}\underline{\mathbf{A}}^{\mathrm{H}} + \sigma^2\mathbf{I})^{-1}\underline{\mathbf{A}} = (\mathbf{\Sigma}_A^2 + \sigma^2\mathbf{I})^{-1}\mathbf{\Sigma}_A\mathbf{V}_A^{\mathrm{H}}$$

[3]Square root factors are usually taken symmetric, i.e., $\hat{\mathbf{R}}_{\mathbf{x}}^{1/2}\hat{\mathbf{R}}_{\mathbf{x}}^{1/2} = \hat{\mathbf{R}}_{\mathbf{x}}$ and $\hat{\mathbf{R}}_{\mathbf{x}}^{1/2\,H} = \hat{\mathbf{R}}_{\mathbf{x}}^{1/2}$, but this is not necessary. $\mathbf{F}$ is a non-symmetric factor.

Conclusion

We can do the following forms of prefiltering:

- $\mathbf{F} = \mathbf{A}$. After this the noise is nonwhite.
- $\mathbf{F} = \mathbf{A}(\mathbf{A}^{\mathrm{H}}\mathbf{A})^{-1/2} = \mathbf{U}_A = \mathbf{U}_s$. After this the noise is white, the Wiener receiver is obtained by setting $\underline{\mathbf{W}} = \underline{\mathbf{R}}_\mathbf{x}^{-1}\underline{\mathbf{A}}$.
- $\mathbf{F} = \hat{\mathbf{\Sigma}}_s^{-1}\hat{\mathbf{U}}_s$. The noise becomes nonwhite, but the data is whitened, $\underline{\hat{\mathbf{R}}}_\mathbf{x} = \mathbf{I}$. The Wiener receiver is obtained by $\underline{\mathbf{W}} = \underline{\mathbf{A}}$.

6.5 EIGENVALUE ANALYSIS OF $\mathbf{R}_\mathbf{x}$

So far, we have looked at the receiver problem from a rather restricted viewpoint: the beamformers were based on the situation where there is a single source in noise. In the next section we will also consider beamforming algorithms that can handle more sources. These are based on an eigenvalue analysis of the data covariance matrix, which is introduced in this section.

Let us first consider the covariance matrix due to d sources and no noise,

$$\mathbf{R}_\mathbf{x} = \mathbf{A}\mathbf{R}_\mathbf{s}\mathbf{A}^{\mathrm{H}}$$

where $\mathbf{R}_\mathbf{x}$ has size $M \times M$, $\mathbf{A}$ has size $M \times d$ and $\mathbf{R}_\mathbf{s}$ has size $d \times d$. If $d < M$, then the rank of $\mathbf{R}_\mathbf{x}$ is d since $\mathbf{A}$ has only d columns. Thus, we can estimate the number of narrow-band sources from a rank analysis. This is also seen from an eigenvalue analysis: let

$$\mathbf{R}_\mathbf{x} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}}$$

be an eigenvalue decomposition of $\mathbf{R}_\mathbf{x}$, where the $M \times M$ matrix $\mathbf{U}$ is unitary ($\mathbf{U}\mathbf{U}^{\mathrm{H}} = \mathbf{I}$, $\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{I}$) and contains the eigenvectors, and the $M \times M$ diagonal matrix $\mathbf{\Lambda}$ contains the corresponding eigenvalues in non-increasing order ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M \geq 0$). Since the rank is d, there are only d nonzero eigenvalues. We can collect these in a $d \times d$ diagonal matrix $\mathbf{\Lambda}_s$, and the corresponding eigenvectors in a $M \times d$ matrix $\mathbf{U}_s$, so that

$$\mathbf{R}_\mathbf{x} = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^{\mathrm{H}}\,. \tag{6.20}$$

The remaining $M - d$ eigenvectors from $\mathbf{U}$ can be collected in a matrix $\mathbf{U}_n$, and they are orthogonal to $\mathbf{U}_s$ since $\mathbf{U} = [\mathbf{U}_s \;\; \mathbf{U}_n]$ is unitary. The subspace spanned by the columns of $\mathbf{U}_s$ is called the *signal subspace*, the orthogonal complement spanned by the columns of $\mathbf{U}_n$ is known as the *noise subspace* (although this is a misnomer since here there is no noise yet and later the noise will be everywhere and not confined to the subspace). Thus, in the noise-free case,

$$\mathbf{R}_\mathbf{x} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}} = [\mathbf{U}_s \quad \mathbf{U}_n] \left[\begin{array}{c|c} \mathbf{\Lambda}_s & 0 \\ \hline 0 & 0 \end{array}\right] \begin{bmatrix} \mathbf{U}_s^{\mathrm{H}} \\ \mathbf{U}_n^{\mathrm{H}} \end{bmatrix}$$

In the presence of white noise,

$$\mathbf{R_x} \;=\; \mathbf{A}_s\mathbf{R_s}\mathbf{A}_s^{\mathrm{H}} + \sigma^2\mathbf{I}_M\,.$$

In this case, $\mathbf{R_x}$ is full rank: its rank is always M. However, we can still detect the number of sources by looking at the eigenvalues of $\mathbf{R_x}$. Indeed, the eigenvalue decomposition is derived as (expressed in terms of the previous decomposition (6.20) and using the fact that $\mathbf{U} = [\mathbf{U}_s \quad \mathbf{U}_n]$ is unitary: $\mathbf{U}_s\mathbf{U}_s^{\mathrm{H}} + \mathbf{U}_n\mathbf{U}_n^{\mathrm{H}} = \mathbf{I}_M$)

$$\begin{aligned}
\mathbf{R_x} &= \mathbf{A}_s\mathbf{R_s}\mathbf{A}_s^{\mathrm{H}} + \sigma^2\mathbf{I}_M \\
&= \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^{\mathrm{H}} + \sigma^2(\mathbf{U}_s\mathbf{U}_s^{\mathrm{H}} + \mathbf{U}_n\mathbf{U}_n^{\mathrm{H}}) \\
&= \mathbf{U}_s(\mathbf{\Lambda}_s + \sigma^2\mathbf{I}_q)\mathbf{U}_s^{\mathrm{H}} + \mathbf{U}_n(\sigma^2\mathbf{I}_{M-d})\mathbf{U}_n^{\mathrm{H}} \\
&= [\mathbf{U}_s \quad \mathbf{U}_n]\left[\begin{array}{c|c} \mathbf{\Lambda}_s + \sigma^2\mathbf{I}_q & 0 \\ \hline 0 & \sigma^2\mathbf{I}_{M-d} \end{array}\right]\left[\begin{array}{c} \mathbf{U}_s^{\mathrm{H}} \\ \mathbf{U}_n^{\mathrm{H}} \end{array}\right] \\
&=: \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}}
\end{aligned} \tag{6.21}$$

hence $\mathbf{R_x}$ has $M-d$ eigenvalues equal to σ^2, and d that are larger than σ^2. *Thus, we can detect the number of signals d by comparing the eigenvalues of $\mathbf{R_x}$ to a threshold defined by σ^2.*

A physical interpretation of the eigenvalue decomposition can be as follows. The eigenvectors give an orthogonal set of "directions" (spatial signatures) present in the covariance matrix, sorted in decreasing order of dominance. The eigenvalues give the power of the signal coming from the corresponding directions, or the power of the output of a beamformer matched to that direction. Indeed, let the i'th eigenvector be $\mathbf{u}_i$, then this output power will be

$$\mathbf{u}_i^{\mathrm{H}}\mathbf{R}\mathbf{u}_i = \lambda_i\,.$$

The first eigenvector, $\mathbf{u}_1$, is always pointing in the direction from which most energy is coming. The second one, $\mathbf{u}_2$, points in a direction orthogonal to $\mathbf{u}_1$ from which most of the remaining energy is coming, etcetera.

If only (spatially white) noise is present but no sources, then there is no dominant direction, and all eigenvalues are equal to the noise power. If there is a single source with power σ_s^2 and spatial signature $\mathbf{a}$, normalized to $\|\mathbf{a}\|^2 = p$, then the covariance matrix is $\mathbf{R_x} = \sigma_s^2\mathbf{a}\mathbf{a}^{\mathrm{H}} + \sigma^2\mathbf{I}$. It follows from the previous that there is only one eigenvalue larger than σ^2. The corresponding eigenvector is $\mathbf{u}_1 = \mathbf{a}\frac{1}{\|\mathbf{a}\|}$, and is in the direction of $\mathbf{a}$. The power coming from that direction is

$$\lambda_1 = \mathbf{u}_1^{\mathrm{H}}\mathbf{R}\mathbf{u}_1 = M\sigma_s^2 + \sigma^2\,.$$

Since there is only one source, the power coming from any other direction orthogonal to $\mathbf{u}_1$ is σ^2, the noise power. Since $\mathbf{u}_1 = \mathbf{a}\frac{1}{\|\mathbf{a}\|}$,

$$\frac{\mathbf{a}^{\mathrm{H}}\mathbf{R_x}\mathbf{a}}{\mathbf{a}^{\mathrm{H}}\mathbf{a}} = \frac{\mathbf{u}_1^{\mathrm{H}}\mathbf{R_x}\mathbf{u}_1}{\mathbf{u}_1^{\mathrm{H}}\mathbf{u}_1} = \lambda_1\,.$$

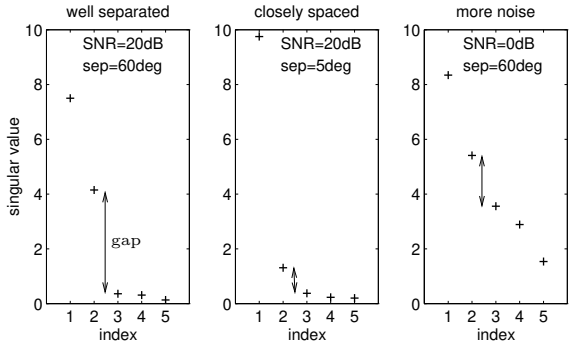


Figure 6.4. Behavior of singular values.

Thus, the result of using the largest eigenvector as a beamformer is the same as the output power of a matched filter where the **a**-vector of the source is known.

With more than one source, this generalizes. Suppose there are two sources with powers σ_1 and σ_2, and spatial signatures $\mathbf{a}_1$ and $\mathbf{a}_2$. If the spatial signatures are orthogonal, $\mathbf{a}_1^{\mathrm{H}}\mathbf{a}_2 = 0$, then $\mathbf{u}_1$ will be in the direction of the strongest source, number 1 say, and λ_1 will be the corresponding power, $\lambda_1 = M\sigma_1^2 + \sigma^2$. Similarly, $\lambda_2 = M\sigma_2^2 + \sigma^2$.

In general, the spatial signatures are not orthogonal to each other. In that case, $\mathbf{u}_1$ will point into the direction that is common to both $\mathbf{a}_1$ and $\mathbf{a}_2$, and $\mathbf{u}_2$ will point in the remaining direction orthogonal to $\mathbf{u}_1$. The power λ_1 coming from direction $\mathbf{u}_1$ will be larger than before because it combines power from both sources, whereas λ_2 will be smaller.

Example 6.4. Instead of the eigenvalue decomposition of $\hat{\mathbf{R}}_{\mathbf{x}}$, we may also compute the singular value decomposition of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$$

Here, $\mathbf{U} : M \times M$ and $\mathbf{V} : N \times N$ are unitary, and $\mathbf{\Sigma} : M \times N$ is diagonal. Since

$$\mathbf{R}_{\mathbf{x}} = \tfrac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{H}} = \tfrac{1}{N}\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^{\mathrm{H}}$$

it is seen that $\mathbf{U}$ contains the eigenvectors of $\hat{\mathbf{R}}_{\mathbf{x}}$, whereas $\frac{1}{N}\mathbf{\Sigma}^2 = \mathbf{\Lambda}$ are the eigenvalues. Thus, the two decompositions give the same information (numerically, it is often better to compute the SVD).

Figure 6.4 shows singular values of $\mathbf{A}$ for $d = 2$ sources, a uniform linear array with $M = 5$ antennas, and $N = 10$ samples, for

1. well separated angles: large gap between signal and noise singular values,
2. signals from close directions, resulting in a small signal singular value,
3. increased noise level, increasing noise singular values.

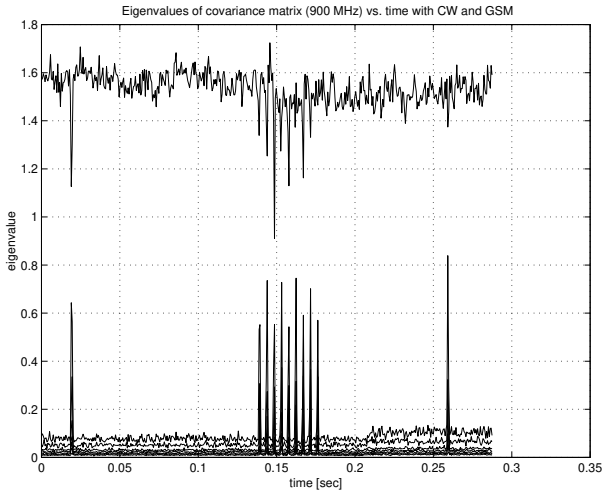


Figure 6.5. Eigenstructure as a function of time

Example 6.5. The covariance matrix eigenvalue structure can be nicely illustrated on data collected at the Westerbork telescope array. We selected a narrow band slice (52 kHz) of a GSM uplink data file, around 900 MHz. In this subband we have two sources: a continuous narrow band (sine wave) signal which leaked in from a local oscillator, and a weak GSM signal. From this data we computed a sequence of short term data covariance matrices $\hat{\mathbf{R}}_{\mathbf{x}}^{0.5ms}$ based on 0.5 ms averages. Figure 6.5 shows the time evolution of the eigenvalues of these matrices. The largest eigenvalue is due to the CW signal and is always present. The GSM source is intermittent: at time intervals where it is present the number of large eigenvalues increases to two. The remaining eigenvalues are at the noise floor, σ^2. The small step in the noise floor after 0.2 s is due to a periodically switched calibration noise source at the input of the telescope front ends.

6.6 BEAMFORMING AND DIRECTION ESTIMATION

In the previous sections, we have assumed that the source matrix $\mathbf{S}$ or the array matrix $\mathbf{A}$ is known. We can now generalize the situation and only assume that the array response is known as a function of the direction parameter θ. Then the directions of arrival (DOA's) of the signals are estimated and used to generate the beamformer weights. The beamformers are in fact the same as we derived in the previous section, except that we specify them in terms of $\mathbf{a}(\theta)$ and subsequently scan θ to find directions where there is "maximal response" (e.g., in the sense of maximal output SNR).

6.6.1 The classical beamformer

The weights in a data independent beamformer are designed so the beamformer response approximates a desired response independent of the array data or data statistics. The design objective—approximating a desired response—is the same as that for classical FIR filter design.

In spatial filtering one is often interested in receiving a signal arriving from a known location point θ_0. Assuming the signal is narrowband, a common choice for the beamformer weight vector is the array response vector $\mathbf{a}(\theta_0)$. This is called the *classical beamformer*, or the Bartlett beamformer; it is precisely the same as the matched filter assuming spatially white noise.

In direction finding using classical beamforming, we estimate the directions of the sources as those that maximize the output power of the beamformer when pointing in a scanning direction θ (and normalizing the output by the array gain):

$$\hat{\theta} = \max_{\theta} \frac{\mathbf{a}(\theta)^{\mathrm{H}} \mathbf{R}_{\mathbf{x}} \mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\mathrm{H}} \mathbf{a}(\theta)} \,.$$

The expression is a *spatial spectrum* estimator. An example of the spectrum obtained this way is shown in Fig. 6.6, see also Chap. 3. With only N samples available, we replace $\mathbf{R}_{\mathbf{x}}$ by the sample covariance matrix, $\hat{\mathbf{R}}_{\mathbf{x}}$. For multiple signals we choose the d largest local maxima.

This technique is equivalent to maximizing the output SNR in case there is only 1 signal in white noise. If the noise is colored, the denominator should actually be replaced by $\mathbf{a}(\theta)^{\mathrm{H}} \mathbf{R}_{\mathbf{n}} \mathbf{a}(\theta)$. If the noise is white but there are interfering sources, our strategy before was to lump the interferers with the noise. However, in the present situation we do not know the interfering directions or $\mathbf{a}(\theta_2), \cdots, \mathbf{a}(\theta_d)$, so this is impossible. This shows that with multiple sources, the classical beamforming technique gives a bias to the direction estimate.

6.6.2 The MVDR

As discussed before, in the MVDR technique we try to minimize the output power, while constraining the power towards the direction θ:

$$\hat{\theta} = \min_{\theta} \; \mathbf{w}^{\mathrm{H}} \hat{\mathbf{R}}_{\mathbf{x}} \mathbf{w} \qquad \text{subject to} \quad \mathbf{w}^{\mathrm{H}} \mathbf{a}(\theta) = 1 \,.$$

This yields

$$\mathbf{w} = \frac{\hat{\mathbf{R}}_{\mathbf{x}}^{-1} \mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\mathrm{H}} \hat{\mathbf{R}}_{\mathbf{x}}^{-1} \mathbf{a}(\theta)}$$

and thus the direction estimate is

$$\hat{\theta} = \min_{\theta} \; \frac{1}{\mathbf{a}(\theta)^{\mathrm{H}} \hat{\mathbf{R}}_{\mathbf{x}}^{-1} \mathbf{a}(\theta)} \,.$$

To make a spectral graph as in the classical beamformer, the expression is inverted to obtain a search for maxima,

$$\hat{\theta} = \max_{\theta} \; \mathbf{a}(\theta)^{\mathrm{H}} \hat{\mathbf{R}}_{\mathbf{x}}^{-1} \mathbf{a}(\theta) \,.$$

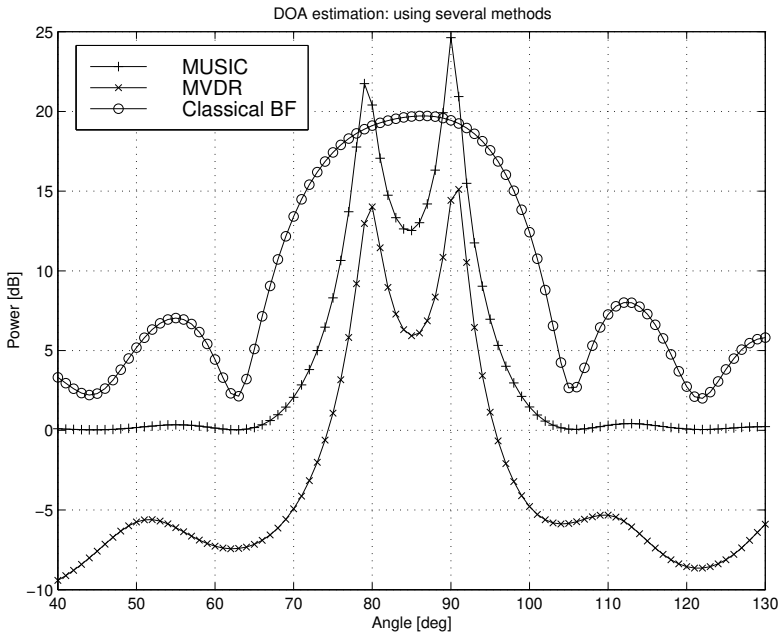


Figure 6.6. Spatial spectra corresponding to the classical beamformer, MVDR, and MUSIC. The DOA's are estimated as the maxima of the spectra.

For multiple signals choose again the d largest local maxima. The MVDR is also illustrated in Fig. 6.6.

6.6.3 The AAR

TBD: make notation uniform

A problem with the MVDR and other adaptive beamformers is that the output noise power is not spatially uniform. Consider the data model $\mathbf{R} = \mathbf{A}\mathbf{\Sigma}_s\mathbf{A}^H + \mathbf{\Sigma}_n$, where $\mathbf{\Sigma}_n = \sigma_n^2\mathbf{I}$ is the noise covariance matrix, then at the output of the beamformer the noise power is

$$\begin{aligned}\sigma_y^2(\mathbf{p}) &= \mathbf{w}(\mathbf{p})^H\mathbf{R}_n\mathbf{w}(\mathbf{p}) \\ &= \frac{\mathbf{a}(\mathbf{p})^H\mathbf{R}^{-1}(\sigma_n^2\mathbf{I})\mathbf{R}^{-1}\mathbf{a}(\mathbf{p})}{[\mathbf{a}(\mathbf{p})^H\mathbf{R}^{-1}\mathbf{a}(\mathbf{p})]^2} \\ &= \sigma_n^2\frac{\mathbf{a}(\mathbf{p})^H\mathbf{R}^{-2}\mathbf{a}(\mathbf{p})}{[\mathbf{a}(\mathbf{p})^H\mathbf{R}^{-1}\mathbf{a}(\mathbf{p})]^2}\,.\end{aligned}$$

Thus, the output noise power is direction dependent.

As a remedy to this, a related beamformer which satisfies the constraint $\mathbf{w}(\mathbf{p})^H\mathbf{w}(\mathbf{p}) = 1$ (and therefore has spatially uniform output noise) is obtained by using a different scaling of the

MVDR beamformer:

$$\mathbf{w}(\mathbf{p}) = \mu \mathbf{R}^{-1}\mathbf{a}(\mathbf{p})\,, \qquad \mu = \frac{1}{\mathbf{a}(\mathbf{p})^H \mathbf{R}^{-2}\mathbf{a}(\mathbf{p})}\,.$$

This beamformer is known as the "Adapted Angular Response" (AAR) [9]. The resulting image is

$$I_{AAR}(\mathbf{p}) = \mathbf{w}(\mathbf{p})^H \mathbf{R}\mathbf{w}(\mathbf{p}) = \frac{\mathbf{a}(\mathbf{p})^H \mathbf{R}^{-1}\mathbf{a}(\mathbf{p})}{[\mathbf{a}(\mathbf{p})^H \mathbf{R}^{-2}\mathbf{a}(\mathbf{p})]^2}\,.$$

It has a high resolution and suppresses sidelobe interference under the white noise constraint. It was proposed for use in radio astronomy image formation in [10], the resulting image was called LS-MVI.

6.6.4 The CLEAN algorithm

TBD (here? or separate section on deconvolution in context of RA)

6.6.5 The MUSIC algorithm

The classical beamformer and the MVDR have a poor performance in cases where there are several closely spaced sources. We now consider more advanced techniques based on the eigenvalue decomposition of the covariance matrix, viz. equation (6.21),

$$\begin{aligned} \mathbf{R}_\mathbf{x} &= \mathbf{A}_s \mathbf{R}_\mathbf{s} \mathbf{A}_s^{\mathrm{H}} + \sigma^2 \mathbf{I}_M \\ &= \mathbf{U}_s(\mathbf{\Lambda}_s + \sigma^2 \mathbf{I}_q)\mathbf{U}_s^{\mathrm{H}} + \mathbf{U}_\mathbf{n}(\sigma^2 \mathbf{I}_{M-d})\mathbf{U}_\mathbf{n}^{\mathrm{H}} \end{aligned}$$

As discussed before, the eigenvalues give information on the number of sources (by counting how many eigenvalues are larger than σ^2). However, the decomposition shows more than just the number of sources. Indeed, *the columns of* $\mathbf{U}_s$ *span the same subspace as the columns of* $\mathbf{A}$. This is clear in the noise-free case (6.20), but the decomposition (6.21) shows that the eigenvectors contained in $\mathbf{U}_s$ and $\mathbf{U}_n$ respectively are the same as in the noise-free case. Thus,

$$\mathrm{span}(\mathbf{U}_s) = \mathrm{span}(\mathbf{A})\,, \qquad \mathbf{U}_n^{\mathrm{H}}\mathbf{A} = 0\,. \tag{6.22}$$

Given a correlation matrix $\hat{\mathbf{R}}_\mathbf{x}$ estimated from the data, we compute its eigenvalue decomposition. From this we can detect the rank d from the number of eigenvalues larger than σ^2, and we can estimate $\mathbf{U}_s$ and hence the subspace spanned by the columns of $\mathbf{A}$. Although we cannot directly identify each individual column of $\mathbf{A}$, its subspace estimate can nonetheless be used to determine the directions, since we know that

$$\mathbf{A} = [\mathbf{a}(\theta_1)\,,\ \cdots\,,\ \mathbf{a}(\theta_d)]$$

If $\mathbf{a}(\theta)$ is known as a function of θ, then we can select the unknown parameters $[\theta_1\,,\ \cdots\,,\ \theta_d]$ to make the estimate of $\mathbf{A}$ *fit* the subspace $\mathbf{U}_s$. Several algorithms are based on this idea. Below we

discuss an effective algorithm that is widely used, the MUSIC (Multiple SIgnal Classification) algorithm.

Note that it is crucial that the noise is spatially white. For colored noise, an extension (whitening) is possible but we have to know the coloring.

Assume that $d < M$. Since $\text{col}(\mathbf{U}_s) = \text{col}\{\mathbf{a}(\theta_1), \cdots, \mathbf{a}(\theta_d)\}$, we have

$$\mathbf{U}_n^{\text{H}}\, \mathbf{a}(\theta_i) \;=\; 0\,, \qquad (1 \le i \le d) \tag{6.23}$$

The MUSIC algorithm estimates the directions of arrival by choosing the d lowest local minima of the cost function

$$J_{MUSIC}(\theta) = \frac{\|\hat{\mathbf{U}}_n^{\text{H}}\mathbf{a}(\theta)\|^2}{\|\mathbf{a}(\theta)\|^2} = \frac{\mathbf{a}(\theta)^{\text{H}}\hat{\mathbf{U}}_n\hat{\mathbf{U}}_n^{\text{H}}\mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\text{H}}\mathbf{a}(\theta)} \tag{6.24}$$

where $\hat{\mathbf{U}}_n^{\text{H}}$ is the sample estimate of the noise subspace, obtained from an eigenvalue decomposition of $\hat{\mathbf{R}}_{\mathbf{x}}$. To obtain a 'spectral-like' graph as before (it is called a pseudo-spectrum), we plot the inverse of $J_{MUSIC}(\theta)$. See Fig. 6.6. Note that this eigenvalue technique gives a higher resolution than the original classical spectrum, also because its sidelobes are much more flat.

Note, very importantly, that as long as the number of sources is smaller than the number of sensors ($d < M$), the eigenvalue decomposition of the true $\mathbf{R}_{\mathbf{x}}$ allows to estimate *exactly* the DOAs. This means that if the number of samples N is large enough, we can obtain estimates with arbitrary precision. Thus, in contrast to the beamforming techniques, the MUSIC algorithm provides *statistically consistent* estimates.

An important limitation is still the failure to resolve closely spaced signals in small samples and at low SNR scenarios. This loss of resolution is more pronounced for highly correlated signals. In the limiting case of coherent signals, the property (6.23) is violated because the rank of $\mathbf{R}_{\mathbf{x}}$ becomes smaller than the number of sources (the dimension of $\mathbf{U}_n$ is too large), and the method fails to yield consistent estimates. To remedy this problem, techniques such as "spatial smoothing" as well as extensions of the MUSIC algorithm have been derived.

6.7 APPLICATIONS TO TEMPORAL MATCHED FILTERING

In the previous sections, we have looked at matched filtering in the context of array signal processing. Let us now look at how this applies to temporal filtering.

No intersymbol interference We start with a fairly simple case, namely the reception of a symbol sequence $s(t)$ convolved with a pulse shape function $g(t)$:

$$x(t) = g(t) * s(t)$$

The symbol sequence is modeled as a sequence of delta pulses, $s(t) = \sum s_k \delta(t - kT)$. The symbol period T will be assumed to be normalized to $T = 1$. We will first assume that the pulse shape

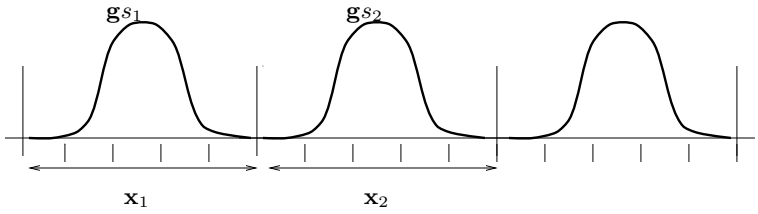


Figure 6.7. No intersymbol interference

function has a duration of less than T, so that $g(t)$ has support only on the interval $[0, 1\rangle$. We sample $x(t)$ at a rate P, where P is the (integer) *oversampling rate*. The samples of $x(t)$ are stacked in vectors

$$\mathbf{x}_k = \begin{bmatrix} x(k) \\ x(k+\frac{1}{P}) \\ \vdots \\ x(k+\frac{P-1}{P}) \end{bmatrix}$$

See also Fig. 6.7. If we are sufficiently synchronized, this means that

$$\mathbf{x}_k = \mathbf{g}s_k \quad \Leftrightarrow \quad \begin{bmatrix} x(k) \\ x(k+\frac{1}{P}) \\ \vdots \\ x(k+\frac{P-1}{P}) \end{bmatrix} = \begin{bmatrix} g(0) \\ g(\frac{1}{P}) \\ \vdots \\ g(\frac{P-1}{P}) \end{bmatrix} s_k \tag{6.25}$$

or

$$\mathbf{X} = \mathbf{g}\mathbf{s}\,, \qquad \mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_N]\,, \quad \mathbf{s} = [s_1 \quad s_2 \quad \cdots \quad s_N]\,.$$

The matched filter in this context is simply $\mathbf{g}^{\mathrm{H}}$. It has a standard interpretation as a convolution or *integrate-and-dump filter*. Indeed, $y_k = \mathbf{g}^{\mathrm{H}}\mathbf{x}_k = \sum_{i=0}^{P-1} g(i)x(k+\frac{i}{P})$. This can be viewed as a convolution by the reverse filter $g_r(t) := g(T-t)$:

$$y_k \;=\; \mathbf{g}^{\mathrm{H}}\mathbf{x}_k \;=\; \sum_{i=1}^{P} g_r(\tfrac{i}{P})x(k+1-\tfrac{i}{P})$$

If P is very large, the summation becomes an integral

$$y_k = \int_0^{\mathrm{T}} g(t)x(kT+t)\,dt = \int_0^{\mathrm{T}} g_r(t)x(\,(k+1)T\,-t)\,dt\,.$$

With intersymbol interference In practise, pulse shape functions are often a bit larger than 1 symbol period. Also, we might not be able to achieve perfect synchronization. Thus let us define a shift of $\mathbf{g}$ over some delay τ, and assume for simplicity that the result has support on

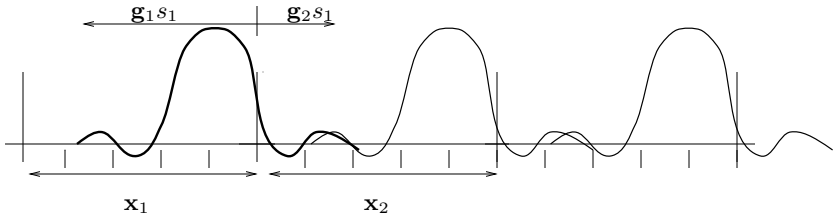


Figure 6.8. With intersymbol interference

$[0, 2T\rangle$ (although with pulse shapes longer than a symbol period, it would in fact be more correct to have a support of $[0, 3T\rangle$):

$$\mathbf{g}_\tau := \begin{bmatrix} g(0-\tau) \\ g(\frac{1}{P}-\tau) \\ \vdots \\ g(2-\frac{1}{P}-\tau) \end{bmatrix}$$

Now, $\mathbf{g}_\tau$ is spread over two symbol periods, and we can define

$$\mathbf{g}_\tau = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}$$

After convolution of $g(t-\tau)$ by the symbol sequence $s(t)$, sampling at rate P, and stacking, we obtain that the resuling sample vectors $\mathbf{x}_k$ are the sum of two symbol sequences (see Fig. 6.8):

$$\mathbf{x}_k = \mathbf{g}_1 s_k + \mathbf{g}_2 s_{k-1} \qquad \Leftrightarrow \qquad \begin{bmatrix} x(k) \\ x(k+\frac{1}{P}) \\ \vdots \\ x(k+\frac{P-1}{P}) \end{bmatrix} = \begin{bmatrix} g(0-\tau) \\ g(\frac{1}{P}-\tau) \\ \vdots \\ g(1-\frac{1}{P}-\tau) \end{bmatrix} s_k + \begin{bmatrix} g(1-\tau) \\ g(\frac{1}{P}-\tau) \\ \vdots \\ g(2-\frac{1}{P}-\tau) \end{bmatrix} s_{k-1}$$

or in matrix form

$$\mathbf{X} = \mathbf{G}_\tau \mathbf{S} \qquad \Leftrightarrow \qquad [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_N] \;=\; [\mathbf{g}_1 \quad \mathbf{g}_2] \begin{bmatrix} s_1 & s_2 & \cdots & s_N \\ s_0 & s_1 & \cdots & s_{N-1} \end{bmatrix}$$

In this case, there is *intersymbol interference*: a sample vector $\mathbf{x}_k$ contains the contributions of more than a single symbol.

A matched filter in this context would be $\mathbf{G}_\tau^{\mathrm{H}}$, at least if $\mathbf{G}_\tau$ is tall: $P \geq 2$. In the current situation (impulse response length including fractional delay shorter than 2 symbols) this is the case as soon as we do any amount of oversampling. After matched filtering, the output $\mathbf{y}_k$ has two entries, each containing a mixture of the symbol sequence and one shift of this sequence. The mixture is given by

$$\mathbf{G}_\tau^{\mathrm{H}} \mathbf{G}_\tau = \begin{bmatrix} \mathbf{g}_1^{\mathrm{H}} \mathbf{g}_1 & \mathbf{g}_1^{\mathrm{H}} \mathbf{g}_2 \\ \mathbf{g}_2^{\mathrm{H}} \mathbf{g}_1 & \mathbf{g}_2^{\mathrm{H}} \mathbf{g}_2 \end{bmatrix}$$

Thus, if $\mathbf{g}_1$ is not orthogonal to $\mathbf{g}_2$, the two sequences will be mixed and further equalization ('beamformer' on $\mathbf{y}_k$) will be necessary. The matched filter in this case only serves to make the output more compact (2 entries) in case P is large.

More in general, we can stack the sample vectors to obtain

$$\mathcal{X} = \mathcal{G}_\tau \mathcal{S} \qquad \Leftrightarrow \qquad \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{N-1} \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{g}_1 & \mathbf{g}_2 \\ \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} s_2 & s_3 & \cdots & s_N \\ s_1 & s_2 & \cdots & s_{N-1} \\ s_0 & s_1 & \cdots & s_{N-2} \end{bmatrix}$$

$\mathcal{G}_\tau$ is tall if $2P \geq 3$. It is clear that for any amount of oversampling ($P > 1$) this is satisfied.

We can imagine several forms of filtering based on this model.

1. *Matched filtering by* $\mathcal{G}_\tau$. The result after matched filtering is $\mathbf{y}_k = \mathcal{G}_\tau^{\mathrm{H}} \left[\begin{smallmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{smallmatrix} \right]$, a vector with 3 entries, and containing the contributions of 3 symbols, mixed via $\mathcal{G}_\tau^{\mathrm{H}} \mathcal{G}_\tau$ (a 3×3 matrix).

2. *Matched filtering by* $\mathbf{g}_\tau$. This is a more common operation, and equal to performing integrate-and-dump filtering after a synchronization delay by τ. The data model is regarded as a signal of interest (the center row of $\mathcal{S}$, premultiplied by $\mathbf{g}_\tau$: the center column of $\mathcal{G}_\tau$),

$$\mathcal{X} \;=\; \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} s_1 & s_2 & \cdots & s_{N-1} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{g}_2 \\ \mathbf{g}_1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} s_2 & s_3 & \cdots & s_N \\ s_0 & s_1 & \cdots & s_{N-2} \end{bmatrix}$$

   The second term is regarded as part of the noise. As such, it has a covariance matrix

$$\mathbf{R}_{\mathbf{n}} = \begin{bmatrix} \mathbf{g}_2 \mathbf{g}_2^{\mathrm{H}} & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_1 \mathbf{g}_1^{\mathrm{H}} \end{bmatrix}$$

   The result after matched filtering is a 1-dimensional sequence $\{y_k\}$,

$$y_k = (\mathbf{g}_\tau^{\mathrm{H}} \mathbf{g}_\tau) s_k + \mathbf{g}_\tau^{\mathrm{H}} \mathbf{n}_k$$

   where the noise at the output due to ISI has variance

$$\mathbf{g}_\tau^{\mathrm{H}} \mathbf{R}_{\mathbf{n}} \mathbf{g}_\tau = \begin{bmatrix} \mathbf{g}_1^{\mathrm{H}} & \mathbf{g}_2^{\mathrm{H}} \end{bmatrix} \begin{bmatrix} \mathbf{g}_2 \mathbf{g}_2^{\mathrm{H}} & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_1 \mathbf{g}_1^{\mathrm{H}} \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = 2 |\mathbf{g}_1^{\mathrm{H}} \mathbf{g}_2|^2$$

   If $\mathbf{g}_1$ is not orthogonal to $\mathbf{g}_2$, then the noise due to ISI is not zero. Since these vectors are dependent on τ, this will generally be the case. With temporally white noise added to the samples, there will also be a contribution $\sigma^2(\mathbf{g}_1^{\mathrm{H}} \mathbf{g}_1 + \mathbf{g}_2^{\mathrm{H}} \mathbf{g}_2)$ to the output noise variance.[4]

[4] In actuality, the noise will not be white but shaped by the receiver filter.

3. *Zero-forcing filtering* and selection of one output. This solution can be regarded as the matched filter of item 1, followed by a de-mixing step (multiplication by $(\mathcal{G}_\tau^{\mathrm{H}}\mathcal{G}_\tau)^{-1}$), and selection of one of the outputs. The resulting filter is

$$\mathbf{w} = \mathcal{G}_\tau(\mathcal{G}_\tau^{\mathrm{H}}\mathcal{G}_\tau)^{-1}\begin{bmatrix}0\\1\\0\end{bmatrix} = (\mathcal{G}_\tau\mathcal{G}_\tau^{\mathrm{H}})^\dagger\mathbf{g}_\tau$$

   and the output will be $[s_1 \;\; s_2 \;\; \cdots$. Note that in principle we could select also one of the other outputs, this would give only a shift in the output sequence (starting with $[s_0 \;\; s_1 \;\; \cdots]$ or $[s_2 \;\; s_3 \;\; \cdots]$). With noise, however, reconstructing the center sequence is likely to give the best performance since it carries the most energy.

4. *Wiener filtering.* This is

$$\mathbf{w} = \hat{\mathbf{R}}_{\mathcal{X}}^{-1}\mathbf{g}_\tau\,.$$

   Under noise-free conditions, this is asympotically equal to $\mathbf{w} = (\mathcal{G}\mathcal{G}^{\mathrm{H}})^\dagger\mathbf{g}_\tau$, i.e., the zero-forcing filter. In the presence of noise, however, it is more simply implemented by direct inversion of the data covariance matrix. Among the linear filtering schemes considered here, the Wiener filter is probably the preferred filter since it maximizes the output SINR.

   As we have seen before, the Wiener filter is asymptotically also equal to a scaling of $\mathbf{R}_{\mathbf{n}}^{-1}\mathbf{g}_\tau$, i.e., the result of item 2, taking the correlated ISI-noise into account. (This equivalence can however only be shown if there is some amount of additive noise as well, or else $\mathbf{R}_{\mathbf{n}}$ and $\mathbf{R}_{\mathcal{X}}$ are not invertible.)

Delay estimation In general, the delay τ by which the data is received is unknown and has to be estimated from the data as well. This is a question very related to that of the DOA estimation considered in the previous section. Indeed, in an ISI-free model $\mathbf{x}_k = \mathbf{g}_\tau s_k$, the problem is similar to $\mathbf{x}_k = \mathbf{a}(\theta)s_k$, but for a different functional. The traditional technique in communications is to use the "classical beamformer": scan the matched filter over a range of τ, and take that τ that gives the peak response. As we have seen in the previous sections, this is optimal if there is only a single component in noise, i.e., no ISI. With ISI, the technique relies on a sufficient orthogonality of the columns of $\mathcal{G}_\tau$. This is however not guaranteed, and the resolution may be poor.

We may however also use the MUSIC algorithm. This is implemented here as follows: compute the SVD of $\mathcal{X}$, or the eigenvalue decomposition of $\mathbf{R}_{\mathcal{X}}$. In either case, we obtain a basis $\mathbf{U}_s$ for the column span of $\mathcal{X}$. In noise-free conditions or asymptotically for a large number of samples, we know that the rank of $\mathcal{X}$ is 3, so that $\mathbf{U}_s$ has 3 columns, and that

$$\mathrm{span}\{\mathbf{U}_s\} = \mathrm{span}\{\mathcal{G}_\tau\} = \mathrm{span}\{\begin{bmatrix}\mathbf{0} & \mathbf{g}_1 & \mathbf{g}_2\\ \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{0}\end{bmatrix}\}$$

Thus, $\mathbf{g}_\tau$ is in the span of $\mathbf{U}_s$. Therefore,

$$\mathbf{g}_\tau \;\perp\; \mathbf{U}_n = (\mathbf{U}_s)^\perp$$

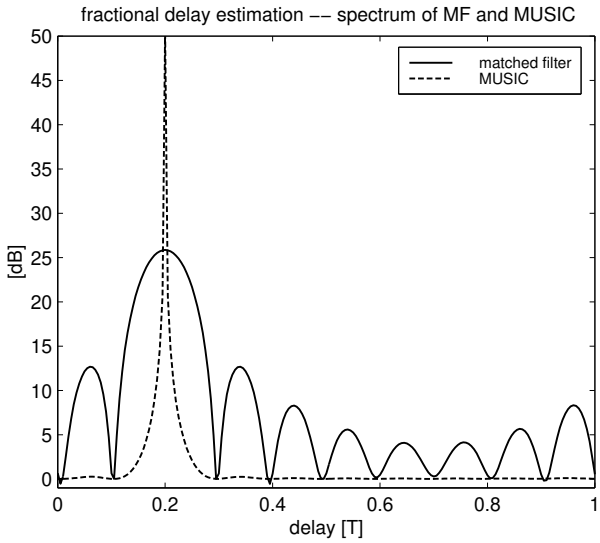


Figure 6.9. Delay estimation: spectra corresponding to the matched filter and MUSIC. The true delay is $0.2T$.

Thus, if we look at the MUSIC cost function (viz. (6.24))

$$J_{MUSIC}(\tau) = \frac{\mathbf{g}(\tau)^{\mathrm{H}}\hat{\mathbf{U}}_n\hat{\mathbf{U}}_n^{\mathrm{H}}\mathbf{g}(\tau)}{\mathbf{g}(\tau)^{\mathrm{H}}\mathbf{g}(\tau)}$$

it will be exactly zero when τ matches the true delay. Figure 6.9 shows the inverse of $J_{MUSIC}(\tau)$, compared to scanning the matched filter. It is obvious that the MUSIC provides a much higher resolution.

Bibliography

[1] D.H. Johnson and D.E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Prentice-Hall, 1993.

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

[3] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs (NJ): Prentice-Hall, 1992.

[4] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*. New-York: Wiley-Interscience, 1980.

[5] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Processing Magazine*, vol. 13, pp. 67–94, July 1996.

[6] B.D. van Veen and K.M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, pp. 4–24, Apr. 1988.

[7] L.L. Scharf, *Statistical Signal Processing.* Reading, MA: Addison-Wesley, 1991.

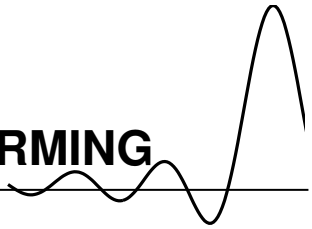
[8] D.H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc., parts F and H*, vol. 130, pp. 11–16, Feb. 1983.

[9] G. B. Borgiotti and L. J. Kaplan, "Supperresolution of uncorrelated interference sources by using adaptive array techniques," *IEEE Trans. Antennas Propagat.*, vol. 27, p. 842–845, 1979.

[10] C. Ben-David and A. Leshem, "Parametric high resolution techniques for radio astronomical imaging," *IEEE J. Sel. Topics in Signal Processing*, vol. 2, pp. 670–684, Oct. 2008.

Chapter **7**

WEIGHTED LEAST SQUARES BEAMFORMING



Contents

7.1 MAXIMUM LIKELIHOOD FORMULATION TO DIRECTION FINDING

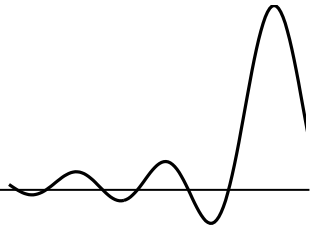
7.2 COVARIANCE MATCHING; WEIGHTED SUBSPACE FITTING

7.3 GAUSS-NEWTON SOLVER

7.4 APPLICATION TO RADIO ASTRONOMY IMAGING

Chapter 8

DIRECTION FINDING: THE ESPRIT ALGORITHM



Contents

In Chapter 6, we have looked at the MVDR and MUSIC algorithms for direction finding. It was seen that MUSIC provides high-resolution estimates for the directions-of-arrival (DOAs). However, these algorithms need a search over the parameter α, and extensive calibration data (i.e., the function $\mathbf{a}(\alpha)$ for a finely sampled range of α). In this present chapter, we look at the ESPRIT algorithm for direction estimation. This algorithm does not require a search or calibration data, but assumes a special array configuration that allows to solve for the DOAs algebraically, by solving an eigenvalue problem. The same algorithm applies to delay estimation and to frequency estimation.

8.1 PRELUDE: SHIFT-INVARIANCE

In this chapter, we will be involved in estimating the parameter θ of vectors with the following polynomial or "Vandermonde" structure

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ \theta \\ \theta^2 \\ \vdots \\ \theta^N \end{bmatrix}, \quad |\theta| = 1\,.$$

The phase of θ provides either the angle-of-arrival of signals, its frequency, or its relative delay, depending on the interpretation of a phase shift in the application. The structure of $\mathbf{a}(\theta)$ is rich, and there are several ways to estimate θ from it after it has been perturbed by noise. A simple (but statistically suboptimal) method is to look for the ratios of entries of $\mathbf{a}$ with their neighbor: each ratio $\frac{a_{i+1}}{a_i}$ is equal to θ. To obtain a good estimate in the presence of noise, we would rather take the average over these ratios:

$$\hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} \frac{a_{i+1}}{a_i} \,. \tag{8.1}$$

This usually provides a very reasonable estimate of θ. The property which has been used here is that of shift-invariance of the vector: if we shift $\mathbf{a}(\theta)$ over one position, we obtain the same vector, but multiplied by θ. Indeed, define the subvectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^{N-1} \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} \theta \\ \theta^2 \\ \vdots \\ \theta^{N} \end{bmatrix}.$$

Then the shift-invariance is expressed as

$$\mathbf{y} = \mathbf{x}\theta \quad \Rightarrow \quad \theta = (\mathbf{x}^{\mathrm{H}}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{H}}\mathbf{y} = \mathbf{x}^{\dagger}\mathbf{y} \,. \tag{8.2}$$

If the entries of $\mathbf{x}$ are on the unit circle, then $(\mathbf{x}^{\mathrm{H}}\mathbf{x})^{-1} = \frac{1}{N}$ and $(\mathbf{x}^{\mathrm{H}})_i = \frac{1}{a_i}$, and the two estimates of θ are the same.[1] The "algorithm" to compute θ in (8.2) is readily extended to superpositions of multiple vectors $\mathbf{a}(\theta_i)$ of the same form, and this is the principle underlying many subspace-based algorithms for harmonic retrieval, direction finding, and rational system identification. The prototype algorithm for this is the ESPRIT algorithm, which was originally proposed for direction finding.

8.2 DIRECTION ESTIMATION USING THE ESPRIT ALGORITHM

As in previous chapters, we assume that all signals are narrowband with respect to the propagation delay across the array, so that this delay translates to a phase shift. We consider a simple propagation scenario, in which there is no multipath and sources have only one ray towards the receiving antenna array. Since no delays are involved, all measurements are simply instantaneous linear combinations of the source signals. Each source has only one ray, so that the data model is

$$\mathbf{X} = \mathbf{A}\mathbf{S} \,.$$

[1] Otherwise, the estimates are slightly different: the ratios in (8.1) should be weighted by $|a_i|^2$ to obtain the same result. This deemphasizes ratios with a poor SNR.

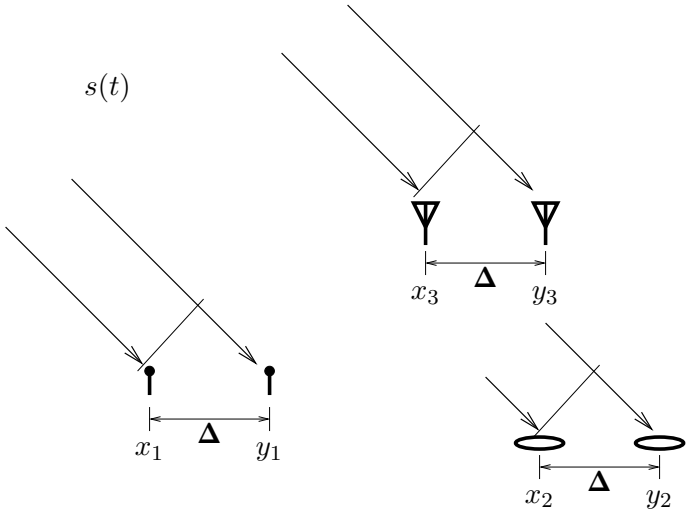


Figure 8.1. Array geometry: sensor doublets.

$\mathbf{A} = [\mathbf{a}(\alpha_1), \cdots, \mathbf{a}(\alpha_d)]$ contains the array response vectors. The rows of $\mathbf{S}$ contain the signals, multiplied by the fading parameters (amplitude scalings and phase rotations).

Computationally attractive ways to compute $\{\alpha_i\}$ and hence $\mathbf{A}$ are possible for certain regular antenna array configurations for which $\mathbf{a}(\alpha)$ becomes a *shift-invariant* or similar recursive structure. This is the basis for the ESPRIT algorithm (Roy, Kailath and Paulraj 1987 [1]).

8.2.1 Array geometry

The constraint on the array geometry imposed by ESPRIT is that of *sensor doublets*: the array consists of two subarrays, denoted by

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \end{bmatrix}, \qquad \mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_M(t) \end{bmatrix}$$

where each sensor y_i has an identical response as x_i, and is spaced at a constant displacement vector $\mathbf{\Delta}$ (wavelengths) from x_i. It is important that the displacement vector is the same for all sensor pairs (both in length and in direction). The antenna response $a_i(\alpha)$ for the pair (x_i, y_i) is arbitrary and may be different for other pairs.

8.2.2 Data model

For the pair $(x_i(t), y_i(t))$, we have the model

$$\begin{aligned} x_i(t) &= \sum_{k=1}^{d} a_i(\alpha_k) s_k(t) \\ y_i(t) &= \sum_{k=1}^{d} a_i(\alpha_k) e^{j2\pi\Delta\sin(\alpha_k)} s_k(t) \;=\; \sum_{k=1}^{d} a_i(\alpha_k)\theta_k s_k(t) \end{aligned}$$

where $\theta_k = e^{j2\pi\Delta\sin(\alpha_k)}$ is the phase rotation due to the propagation of the signal from the x-antenna to the corresponding y-antenna.

In terms of the vectors $\mathbf{x}$ and $\mathbf{y}$, we have

$$\begin{aligned} \mathbf{x}(t) &= \sum_{k=1}^{d} \mathbf{a}(\alpha_k) s_k(t) \\ \mathbf{y}(t) &= \sum_{k=1}^{d} \mathbf{a}(\alpha_k)\theta_k s_k(t) \end{aligned} \tag{8.3}$$

The ESPRIT algorithm does not assume any structure on $\mathbf{a}(\alpha)$. It will instead use the phase relation between $\mathbf{x}(t)$ and $\mathbf{y}(t)$.

If we collect N samples in matrices $\mathbf{X}$ and $\mathbf{Y}$, we obtain the data model

$$\begin{aligned} \mathbf{X} &= \mathbf{AS} \\ \mathbf{Y} &= \mathbf{A\Theta S} \end{aligned} \tag{8.4}$$

where

$$\mathbf{A} = [\mathbf{a}(\alpha_1) \quad \cdots \quad \mathbf{a}(\alpha_d)], \qquad \mathbf{\Theta} = \begin{bmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_d \end{bmatrix}, \qquad \theta_i = e^{j2\pi\Delta\sin(\alpha_i)}.$$

One special case in which the shift-invariant structure occurs is that of a uniform linear array (ULA) with $M+1$ antennas. For such an array, with interelement spacing Δ wavelengths, we have seen that

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^M \end{bmatrix}, \quad \theta = e^{j2\pi\Delta\sin(\alpha)}. \tag{8.5}$$

If we now split the array into two overlapping subarrays, the first ($\mathbf{x}$) containing antennas 1 to

M, and the second ($\mathbf{y}$) antennas 2 to $M+1$, we obtain

$$\mathbf{a}_x = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^{M-1} \end{bmatrix}, \qquad \mathbf{a}_y = \begin{bmatrix} \theta \\ \theta^2 \\ \vdots \\ \theta^{M} \end{bmatrix} = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^{M-1} \end{bmatrix} \theta$$

which gives precisely the model (8.3), where $\mathbf{a}$ in (8.3) is one entry shorter than in (8.5).

8.2.3 Algorithm

Given the data $\mathbf{X}$ and $\mathbf{Y}$, we first stack all data in a single matrix $\mathbf{Z}$ of size $2M \times N$ with model

$$\mathbf{Z} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \mathbf{A}_z \mathbf{S}\,, \qquad \mathbf{A}_z = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\boldsymbol{\Theta} \end{bmatrix}.$$

(In the case of a ULA with $M+1$ antennas, we stack the available antenna outputs vertically but do not duplicate the antennas; $\mathbf{Z}$ will then have size $M+1 \times N$). Since $\mathbf{Z}$ has rank d, we compute an (economy-size) SVD

$$\mathbf{Z} = \hat{\mathbf{U}}_z \hat{\boldsymbol{\Sigma}}_z \hat{\mathbf{V}}_z^{\mathrm{H}}\,, \tag{8.6}$$

where $\hat{\mathbf{U}}_z : 2M \times d$ has d columns which together span the column space of $\mathbf{Z}$. The same space is spanned by the columns of $\mathbf{A}_z$, so that there must exist a $d \times d$ invertible matrix $\mathbf{T}$ that maps one basis into the other, i.e., such that

$$\hat{\mathbf{U}}_z = \mathbf{A}_z \mathbf{T} = \begin{bmatrix} \mathbf{A}\mathbf{T} \\ \mathbf{A}\boldsymbol{\Theta}\mathbf{T} \end{bmatrix} \tag{8.7}$$

If we now split $\hat{\mathbf{U}}_z$ into two $M \times d$ matrices in the same way as $\mathbf{Z}$,

$$\hat{\mathbf{U}}_z = \begin{bmatrix} \hat{\mathbf{U}}_x \\ \hat{\mathbf{U}}_y \end{bmatrix}$$

then we obtain that

$$\begin{cases} \hat{\mathbf{U}}_x &= \mathbf{A}\mathbf{T} \\ \hat{\mathbf{U}}_y &= \mathbf{A}\boldsymbol{\Theta}\mathbf{T} \end{cases}$$

For $M \geq d$, $\hat{\mathbf{U}}_x$ is "tall", and if we assume that $\mathbf{A}$ has full column rank, then $\hat{\mathbf{U}}_x$ has a left-inverse

$$\hat{\mathbf{U}}_x^{\dagger} := (\hat{\mathbf{U}}_x^{\mathrm{H}} \hat{\mathbf{U}}_x)^{-1} \hat{\mathbf{U}}_x^{\mathrm{H}}\,.$$

It is straightforward to verify that

$$\hat{\mathbf{U}}_x^{\dagger} = (\mathbf{T}^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \mathbf{A}\mathbf{T})^{-1} \mathbf{T}^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} = \mathbf{T}^{-1} \mathbf{A}^{\dagger}$$

so that

$$\hat{\mathbf{U}}_x^\dagger \hat{\mathbf{U}}_y = \mathbf{T}^{-1}\boldsymbol{\Theta}\mathbf{T}\,.$$

The matrix on the left hand side is known from the data. Since $\boldsymbol{\Theta}$ is a diagonal matrix, the matrix product on the right hand side is recognized as an eigenvalue equation: $\mathbf{T}^{-1}$ contains the eigenvectors of $\hat{\mathbf{U}}_x^\dagger \hat{\mathbf{U}}_y$ (scaled arbitrarily to unit norm), and the entries of $\boldsymbol{\Theta}$ on the diagonal are the eigenvalues. Hence we can simply compute the eigenvalue decomposition of $\hat{\mathbf{U}}_x^\dagger \hat{\mathbf{U}}_y$, take the eigenvalues $\{\theta_i\}$ (they should be on the unit circle), and compute the DOAs α_i from each of them. This comprises the ESPRIT algorithm.

Note that the SVD of $\mathbf{Z}$ in (8.6) along with the definition of $\mathbf{T}$ in (8.7) as $\hat{\mathbf{U}}_z = \mathbf{A}_z\mathbf{T}$ implies that

$$\begin{aligned} & \mathbf{Z} = \hat{\mathbf{U}}_z\hat{\boldsymbol{\Sigma}}_z\hat{\mathbf{V}}_z^{\mathrm{H}}\,, \qquad \mathbf{Z} = \mathbf{A}_z\mathbf{S} = \mathbf{A}_z\mathbf{T}\mathbf{T}^{-1}\mathbf{S} \\ \Rightarrow \quad & \mathbf{T}^{-1}\mathbf{S} = \hat{\boldsymbol{\Sigma}}_z\hat{\mathbf{V}}_z^{\mathrm{H}} = \hat{\mathbf{U}}_z^{\mathrm{H}}\mathbf{Z} \\ \Rightarrow \quad & \mathbf{S} = \mathbf{T}\hat{\mathbf{U}}_z^{\mathrm{H}}\mathbf{Z} \end{aligned}$$

Hence, after having obtained $\mathbf{T}$ from the eigenvectors, a zero-forcing beamformer $\mathbf{W}$ on $\mathbf{Z}$ such that $\mathbf{S} = \mathbf{W}^{\mathrm{H}}\mathbf{Z}$ is given by

$$\mathbf{W} = \hat{\mathbf{U}}_z\mathbf{T}^{\mathrm{H}}\,.$$

Thus, source separation is straightforward in this case and essentially reduced to an SVD and an eigenvalue problem.

If the two subarrays are spaced by at most half a wavelength, then the DOAs are directly recovered from the diagonal entries of $\boldsymbol{\Theta}$, otherwise they are ambiguous (two different values of α give the same θ). Such an ambiguity does not prevent the construction of the beamformer $\mathbf{W}$ from $\mathbf{T}$, and source separation is possible nonetheless. Because the rows of $\mathbf{T}$ are determined only up to a scaling, the correct scaling of the rows of $\mathbf{S}$ cannot be recovered unless we know the average power of each signal or the array manifold $\mathbf{A}$. This is of course inherent in the problem definition.

With noise, essentially the same algorithm is used. If we assume that the number of sources d is known, then we compute the SVD of the noisy $\mathbf{Z}$, and set $\hat{\mathbf{U}}_Z$ equal to the principal d left singular vectors. This is the best estimate of the subspace spanned by the columns of $\mathbf{A}$, and asymptotically (infinite samples) identical to it. Thus, for infinitely many samples we obtain the correct directions: the algorithm is asymptotically unbiased (*consistent*). For finite samples, an estimated eigenvalue $\hat{\theta}$ will not be on the unit circle, but we can easily map it to the unit circle by dividing by $|\hat{\theta}|$.

Compared to the beamforming algorithms in Chap. 6, which locate sources due to their peaks in a spatial power spectrum, the ESPRIT algorithm finds the exact directions of arrival under noise-free conditions, or asymptotically as the number of samples grows large. This is due to the parametric assumptions: an exact number of point sources, less than the number of antennas. Under these conditions, the sources may be arbitrarily close. On the other hand, the algorithm will fail if some of the sources are diffuse: such sources must be modeled as part of the noise. If the noise is not white, the noise covariance must be estimated and whitened.

8.2.4 Extension for a ULA

There are many important refinements and extensions to this algorithm. If we have a uniform linear array, we can use the fact that the solutions θ should be on the unit circle, i.e.,

$$\bar{\theta} = \theta^{-1}$$

along with the structure of $\mathbf{a}(\theta)$ in (8.5):

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^M \end{bmatrix} \quad \Rightarrow \quad \mathbf{\Pi}\bar{\mathbf{a}}(\theta) =: \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix} \begin{bmatrix} 1 \\ \bar{\theta} \\ \vdots \\ \bar{\theta}^M \end{bmatrix} = \begin{bmatrix} \bar{\theta}^M \\ \bar{\theta}^{M-1} \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^M \end{bmatrix} \theta^{-M} = \mathbf{a}(\theta)\theta^{-M} \,.$$

Thus, if we construct an extended data matrix

$$\mathbf{Z}_e \;=\; [\mathbf{Z}\,, \quad \mathbf{\Pi}\bar{\mathbf{X}}]$$

then this will double the number of observations but will not increase the rank, since

$$\mathbf{Z}_e \;=\; \mathbf{A}_z[\mathbf{S}\,, \quad \mathbf{\Theta}^{-1}\mathbf{S}]\,.$$

Using this structure, it is also possible to transform $\mathbf{Z}_e$ to a real-valued matrix, by simple linear operations on its rows and columns [2, 3]. As we saw in chapter 6, there are many other direction finding algorithms that are applicable. For the case of a ULA in fact a better algorithm is known to be MODE [4]. Although ESPRIT is statistically suboptimal, its performance is usually quite adequate. Its interest lies also in its straightforward generalization to more complicated estimation problems in which shift-invariance structure is present.

8.2.5 Noise whitening

TBD:

The SVD estimates the column span of $\mathbf{A}$, which is the essential first step. If there is white noise, this does not affect this estimate (asympotically). Alternatively, we can work on an EVD of $\mathbf{R}_x$.

If the noise covariance $\mathbf{R}_n$ is not white, it must be estimated and taken into account: whiten $\mathbf{X}$ by working with $\mathbf{R}_n^{-1/2}\mathbf{X}$, or using a GEV $(\mathbf{R}_x, \mathbf{R}_n)$.

(This is common in all subspace-based algorithms and should be mentioned earlier and more prominently.)

8.2.6 Performance

Figure 8.2 shows the results of a simulation with 2 sources with directions $-10°,\ 10°$, a ULA($\frac{\lambda}{2}$) with 6 antennas, and $N = 40$ samples. The first graph shows the mean value, the second the

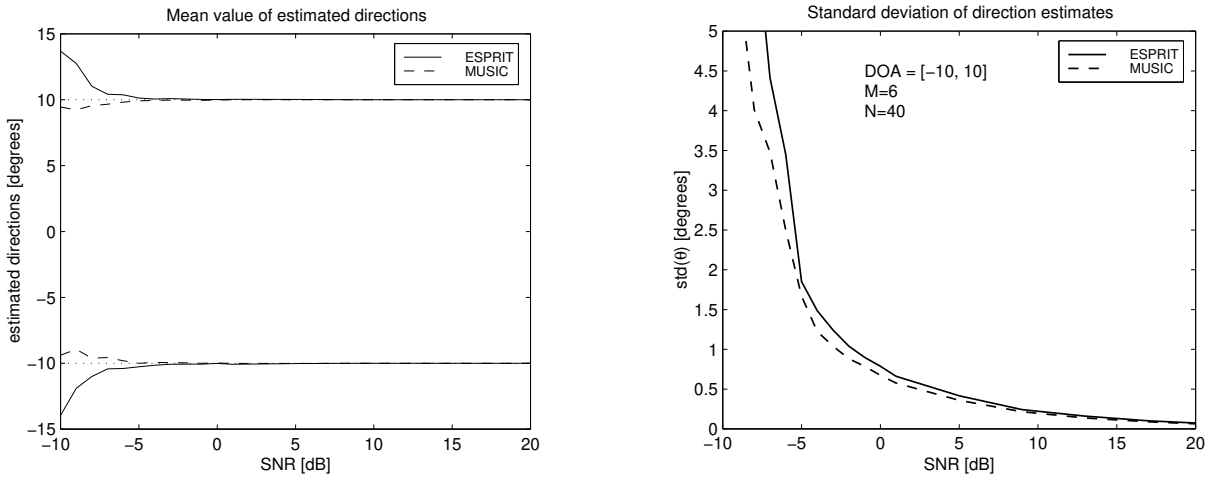


Figure 8.2. Mean and standard deviations of ESPRIT and MUSIC estimates as function of SNR

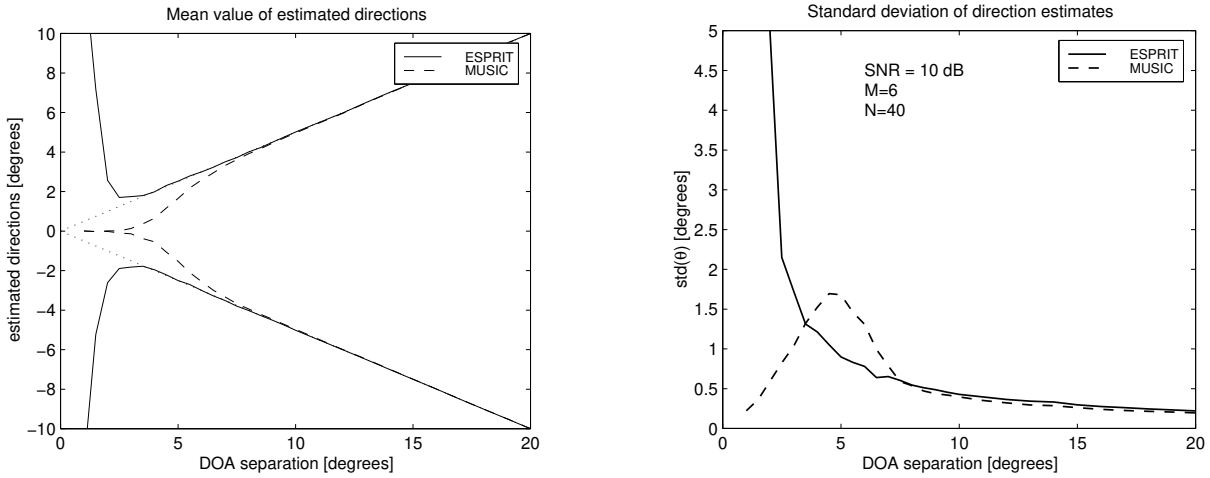


Figure 8.3. Mean and standard deviations of ESPRIT and MUSIC estimates as function of DOA separation

standard deviation (averaged over the two sources), which indicates the accuracy of an individual estimate. For sufficient SNR, the performance of both algorithms is approximately the same.

Figure 8.3 shows the same for varying separation of the two sources, with an SNR of 10 dB. For small separation, the performance of ESPRIT drops because the matrix $\mathbf{A}$ drops in rank: it appears to have only 1 independent column rather than 2. If we select two singular vectors, then this subspace will not be shift-invariant, and the algorithm produces bad estimates: both the mean value and the standard deviation explode. MUSIC, on the other hand, selects the null space and scans for vectors orthogonal to it. If we ask for 2 vectors, it will in this case produce two times the same vector since there is only a single maximum in the MUSIC spectrum. It is seen that the estimates become biased towards a direction centered between the two sources ($= 0°$), but that the standard deviation gets smaller since the algorithm consistenly picks this center.

The performance of both ESPRIT and MUSIC is noise limited: without noise, the correct DOAs are obtained. With noise and asymptotically many samples, $N \to \infty$, the correct DOAs are obtained as well, since the subspace spanned by $\hat{\mathbf{U}}_z$ is asymptotically identical to that obtained in the noise-free case, the span of the columns of $\mathbf{A}$.

8.2.7 Extension to coherent multipath

In the above, we assumed that there was no multipath: each source had only one path to the antenna array. However, the $\mathbf{X} = \mathbf{AS}$ model is also valid if sources have multiple rays towards the array, as long as the delay differences are small compared to the signal bandwidth, so that they can be represented by phase shifts. This is known as coherent multipath (see also Sec. 4.3.1).

Let d be the number of sources, r_i the number of rays belonging to source i, and $r = \sum_1^d r_i$ the total number of rays (assumed to be distinct). In that case, a more detailed model is

$$\mathbf{X} = (\mathbf{A}_\theta \mathbf{B} \mathbf{J}) \mathbf{S} \tag{8.8}$$

where $\mathbf{A}_\theta : M \times r$ is the Vandermonde matrix associated to the DOAs of the rays, and $\mathbf{J} : r \times d$ is a selection matrix which adds groups of rays to source signals, e.g.,

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

in case of two sources, each with two rays. $\mathbf{B}$ is a diagonal scaling matrix representing the different amplitudes (fadings) of each ray, including phase offsets. Because the rank of $\mathbf{X}$ is still d, the SVD of $\mathbf{X}$ can retrieve only a d-dimensional subspace $\hat{\mathbf{U}}$, so that

$$\hat{\mathbf{U}} = (\mathbf{A}_\theta \mathbf{B} \mathbf{J}) \mathbf{T}\,, \qquad \mathbf{S} = \mathbf{T} \hat{\mathbf{U}}^{\mathrm{H}} \mathbf{X}\,.$$

It is clear that blind beamforming is more challenging now: we try to find $\mathbf{T}$ such that each column of $\hat{\mathbf{U}}$ is represented by a sum of r Vandermonde vectors, rather than only d vectors, and r is not known.

To solve this problem algebraically using ESPRIT-type techniques,[2] we first try to restore the rank to r. This is possible if the number of antennas M is sufficiently large, in fact $M \geq r + \max(r_i)$. In that case, we can form a block-Hankel matrix out of $\hat{\mathbf{U}}$ by taking vertical shifts of it:

$$\mathcal{U}_m := [\hat{\mathbf{U}}^{(1)} \quad \hat{\mathbf{U}}^{(2)} \quad \cdots \quad \hat{\mathbf{U}}^{(m)}] : \quad (M-m+1) \times md\,. \tag{8.9}$$

Here, $\hat{\mathbf{U}}^{(i)}$ is a submatrix of $\hat{U}$ consisting of its i-th till $M-m+i$-th row, and m is known as the *spatial smoothing factor* [5, 6]. With the above model, we have that $\mathcal{U}_m$ satisfies the factorization

$$\begin{array}{rcl} \mathcal{U}_m & = & \mathbf{A}'_\theta \mathbf{B}[\mathbf{JT} \quad \mathbf{\Theta JT} \quad \cdots \quad \mathbf{\Theta}^{m-1}\mathbf{JT}] \\ & =: & \mathbf{A}'_\theta \mathbf{B}\mathcal{T}\,, \end{array} \tag{8.10}$$

where $\mathbf{A}'_\theta$ consists of the top $M-m+1$ rows of $\mathbf{A}_\theta$. If $M-m+1 \geq r$ and $m \geq \max(r_i)$, the factors in the above factorization can be shown to have full rank r, so that $\mathcal{U}_m$ has rank r.

At this point, the structure of $\mathcal{U}_m$ in (8.10) shows that we have reduced the problem to an $[\mathbf{X} = \mathbf{AS}]$-type problem without multipath, which can be solved using the ESPRIT algorithm. Thus we compute an SVD of $\mathcal{U}_m$,

$$\mathcal{U}_m = \hat{\mathbf{U}}_u \hat{\mathbf{\Sigma}}_u \hat{\mathbf{V}}_u\,,$$

where $\hat{\mathbf{U}}_u$ contains the dominant r singular vectors of $\mathcal{U}_m$. From (8.10) it follows that there is an invertible $r \times r$ matrix $\mathbf{R}$ such that

$$\hat{\mathbf{U}}_u = \mathbf{A}'_\theta \mathbf{BR}\,, \qquad \mathcal{T} = (\mathbf{R}\hat{\mathbf{U}}_u^{\mathrm{H}})\mathcal{U}_m\,.$$

We continue in the same way as before to compute $\mathbf{R}$: with

$$\hat{\mathbf{U}}_x = \mathbf{J}_x \hat{\mathbf{U}}_u\,, \qquad \hat{\mathbf{U}}_y = \mathbf{J}_y \hat{\mathbf{U}}_u\,,$$

the data model satisfies the eigenvalue equation

$$\hat{\mathbf{U}}_x^\dagger \hat{\mathbf{U}}_y = \mathbf{R}^{-1}\mathbf{\Theta R} \tag{8.11}$$

which gives both $\mathbf{\Theta}$ and $\mathbf{R}$, up to scaling of its rows. At this point, we have recovered $\mathcal{T} = (\mathbf{R}\hat{\mathbf{U}}_u^{\mathrm{H}})\mathcal{U}_m$, up to multiplication at the left by an arbitrary diagonal matrix. The next objective is to estimate $\mathbf{T}$ from the structure of $\mathcal{T}$ in (8.10). This is now a much simpler task: we have available m matrices of size $r \times d$, after correction by suitable powers of $\mathbf{\Theta}^{-1}$ all equal to $\mathbf{JT}$. The structure of $\mathbf{J}$ ensures that this matrix has only d distinct rows, which are the d rows of $\mathbf{T}$. Hence, it suffices to estimate these d unique rows, which is a simple clustering problem if the rows of $\mathbf{T}$ are sufficiently different. This determines both $\mathbf{T}$ and $\mathbf{J}$, i.e., the assignment of rays to sources. With $\mathbf{T}$ in hand, we have our blind beamformer as before: $\mathbf{W}^{\mathrm{H}} = \mathbf{T}\hat{\mathbf{U}}^{\mathrm{H}}$.

[2] Other techniques such as MODE are directly applicable to the coherent case without modifications.

8.3 DELAY ESTIMATION USING ESPRIT

A channel matrix $\mathbf{H}$ can be estimated from training sequences, or sometimes "blindly" (without training). Very often, we do not need to know the details of $\mathbf{H}$ if our only purpose is to recover the signal matrix $\mathbf{S}$. But there are several situations as well where it is interesting to pose a multipath propagation model, and try to resolve the individual propagation paths. This would give information on the available delay and angle spread, for the purpose of diversity. It is often assumed that the directions and delays of the paths do not change quickly, only their powers (fading parameters), so that it makes sense to estimate these parameters. If the channel is well-characterized by this parametrized model, then fitting the channel estimate to this model will lead to a more accurate receiver. Another application is wireless localization.

8.3.1 Principle

Let us consider first the simple case already introduced in Sec. 6.7. Assume we have a vector $\mathbf{g}_0$ corresponding to N samples of an FIR pulse shape function $g(t)$, sampled with period T above the Nyquist rate,

$$g(t) \quad \leftrightarrow \quad \mathbf{g}_0 = \begin{bmatrix} g(0) \\ g(T) \\ \vdots \\ g((N-1)T) \end{bmatrix} .$$

Similarly, we can consider a delayed version of $g(t)$:

$$g(t-\tau) \quad \leftrightarrow \quad \mathbf{g}_\tau = \begin{bmatrix} g(0-\tau) \\ g(T-\tau) \\ \vdots \\ g((N-1)T--\tau) \end{bmatrix} .$$

The number of samples N is chosen such that at the maximal possible delay, $g(t-\tau)$ has support only on the interval $[0, NT\rangle$ symbols.

Given $\mathbf{g}_\tau$ and knowing $\mathbf{g}_0$, how do we estimate τ? Note here that τ does not have to be a multiple of T, so that $\mathbf{g}_\tau$ is not exactly a shift of the samples in $\mathbf{g}_0$. A simple "pattern matching" with entry-wise shifts of $\mathbf{g}_0$ will not give an exact result.

We can however make use of the fact that a Fourier transformation maps a delay to a certain phase progression. Let

$$\tilde{g}(\omega_i) = \sum_{k=0}^{N-1} e^{-j\omega_i k} g(kT) , \qquad \omega_i = i\frac{2\pi}{N} , \qquad i = 0, 1, \cdots, N-1 .$$

In matrix-vector form, this can be written as

$$\tilde{\mathbf{g}}_0 = \mathcal{F}\, \mathbf{g}_0 , \qquad \tilde{\mathbf{g}}_\tau = \mathcal{F}\, \mathbf{g}_\tau$$

where $\mathcal{F}$ denotes the DFT matrix of size $N \times N$, defined by

$$\mathcal{F} := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \phi & \cdots & \phi^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & \phi^{N-1} & \cdots & \phi^{(N-1)^2} \end{bmatrix}, \quad \phi = e^{-j\frac{2\pi}{N}}. \tag{8.12}$$

If τ is an integer multiple of T, then it is straightforward to see that the Fourier transform $\tilde{\mathbf{g}}_\tau$ of the sampled version of $g(t-\tau)$ is given by

$$\tilde{\mathbf{g}}_\tau \;=\; \tilde{\mathbf{g}}_0 \odot \begin{bmatrix} 1 \\ \phi^{\tau/T} \\ (\phi^{\tau/T})^2 \\ \vdots \\ (\phi^{\tau/T})^{N-1} \end{bmatrix} \;=\; \mathrm{diag}(\tilde{\mathbf{g}}_0) \cdot \begin{bmatrix} 1 \\ \phi^{\tau/T} \\ (\phi^{\tau/T})^2 \\ \vdots \\ (\phi^{\tau/T})^{N-1} \end{bmatrix} \tag{8.13}$$

where $\odot$ represents entrywise multiplication of the two vectors. The same holds true for any τ if $g(t)$ is bandlimited and sampled at or above the Nyquist rate.

Thus, we will assume that $g(t)$ is bandlimited and sampled at such a rate that (8.13) is valid even if τ is not an integer multiple of T. The next step is to do a deconvolution of $g(t)$ in frequency domain, by entrywise dividing $\tilde{\mathbf{g}}_\tau$ by $\tilde{\mathbf{g}}_0$. Obviously, this can be done only on intervals where $\tilde{\mathbf{g}}_0$ is nonzero. Pulse shapes are bandlimited, and if we sample above Nyquist, some entries of $\tilde{\mathbf{g}}_0$ will be close to zero. If necessary, a selection matrix has to be applied to select only the nonzero interval.

Next, we factor $\mathrm{diag}(\tilde{\mathbf{g}}_0)$ out of $\tilde{\mathbf{g}}_\tau$ and obtain

$$\mathbf{z} := \{\mathrm{diag}(\tilde{\mathbf{g}}_0)\}^{-1}\tilde{\mathbf{g}}_\tau\,, \qquad (N \times 1) \tag{8.14}$$

which satisfies the model

$$\mathbf{z} = \mathbf{f}(\phi)\,, \qquad \mathbf{f}(\phi) := \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{N-1} \end{bmatrix}, \qquad \phi := e^{-j\frac{2\pi}{N}\frac{\tau}{T}}. \tag{8.15}$$

Note that $\mathbf{f}(\phi)$ has the same structure as $\mathbf{a}(\theta)$ for a ULA. Hence, we can apply the ESPRIT algorithm in the same way as before to estimate ϕ from $\mathbf{z}$, and subsequently τ. In the present case, we simply split $\mathbf{z}$ into two subvectors $\mathbf{x}$ and $\mathbf{y}$, one a shift of the other, and from the model $\mathbf{y} = \mathbf{x}\phi$ we can obtain $\phi = \mathbf{x}^\dagger\mathbf{y}$, from which we can compute τ.

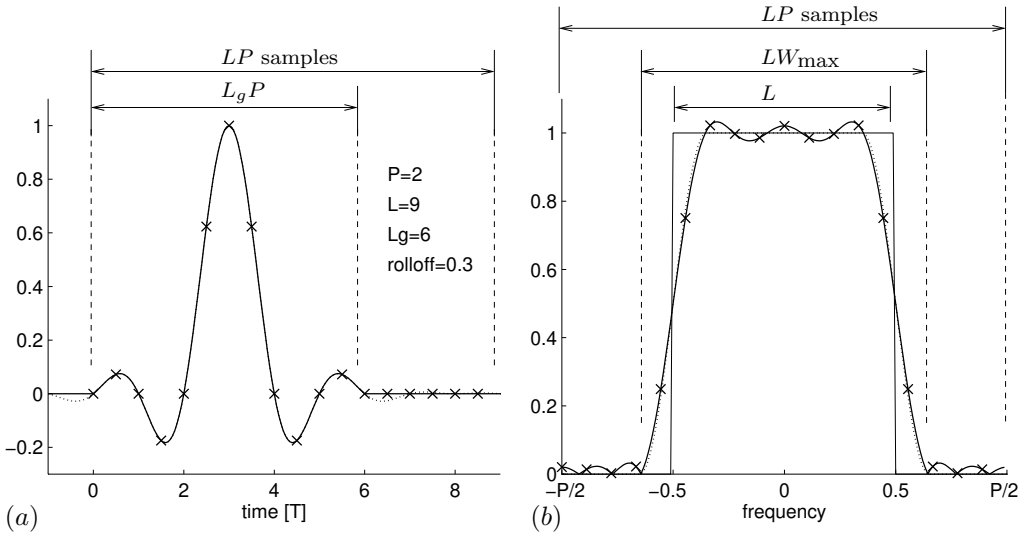


Figure 8.4. Definition of parameters: (a) time domain, (b) frequency domain.

Oversampled pulse shapes There are some details that we skipped in the preceding discussion. First of all, we assumed $g(t)$ has a representation as an FIR filter. Because of the truncation to length N, the spectrum of $g(t)$ widens and sampling at a rate $1/T$ introduces some aliasing due to spectral folding. This will eventually lead to a small bias in the delay estimate. To avoid this, we can oversample the channel.

To give a specific example, assume that $g(t)$ is a raised cosine pulse, as in Fig. 8.4. For convenience of notation we normalize the time axis and set $T = 1/P$, where P is the oversampling factor (in the figure, $P = 2$), so that in the DFT frequency domain we have N samples within the fundamental interval $-P/2 < F < P/2$. Clearly, $g(t)$ is bandlimited, and only $L = N/P$ frequency domain samples are significant. In the deconvolution step, we cannot divide out $\tilde{\mathbf{g}}_0$, because we will be dividing by small numbers.

Let $\mathbf{J}_{\tilde{\mathbf{g}}} : L \times N$ be a selection matrix for $\tilde{\mathbf{g}}$, such that $\mathbf{J}_{\tilde{\mathbf{g}}}\tilde{\mathbf{g}}$ has the desired entries. For later use, we require that the selected frequencies appear in increasing order, which with the definition of the DFT in (8.12) means that the final $\lceil L/2 \rceil$ samples of $\tilde{\mathbf{g}}_0$ should be moved up front: $\mathbf{J}_{\tilde{\mathbf{g}}}$ has the form

$$\mathbf{J}_{\tilde{\mathbf{g}}} = \begin{bmatrix} 0 & 0 & I_{\lceil L/2 \rceil} \\ I_{\lfloor L/2 \rfloor} & 0 & 0 \end{bmatrix} : \quad L \times N \,.$$

Next, we can factor $\mathrm{diag}(\mathbf{J}_{\tilde{\mathbf{g}}}\tilde{\mathbf{g}}_0)$ out of $\mathbf{J}_{\tilde{\mathbf{g}}}\tilde{\mathbf{g}}_\tau$ and obtain

$$\mathbf{z} := \{\mathrm{diag}(\mathbf{J}_{\tilde{\mathbf{g}}}\tilde{\mathbf{g}}_0)\}^{-1}\mathbf{J}_{\tilde{\mathbf{g}}}\tilde{\mathbf{g}}_\tau \,, \qquad (L \times 1) \tag{8.16}$$

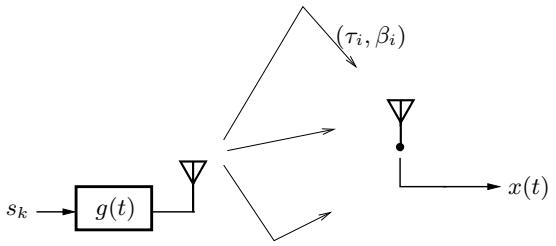


Figure 8.5. Multiray propagation channel

which satisfies the model

$$\mathbf{z} = \mathbf{f}(\phi)\, \phi^{-\lceil L/2 \rceil}\,, \qquad \mathbf{f}(\phi) := \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{L-1} \end{bmatrix}, \qquad \phi := e^{-j\frac{2\pi}{N}\frac{\tau}{T}}\,. \tag{8.17}$$

We are essentially back to (8.15), although the vector is a bit shorter.

8.3.2 Multipath channel model estimation

We will now build on the above principle. Consider a multipath channel which consists of r delayed copies of $g(t)$, as in Fig. 8.5,[3] so that the impulse response is

$$h(t) = \sum_{i=1}^{r} \beta_i g(t-\tau_i) \qquad \Leftrightarrow \qquad \mathbf{h} = \sum_{i=1}^{r} \mathbf{g}_{\tau_i}\beta_i = [\mathbf{g}_{\tau_1}\,, \cdots, \mathbf{g}_{\tau_r}] \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_r \end{bmatrix} =: \mathbf{G}_\tau \mathbf{b}\,.$$

We assume that we know $\mathbf{h}$ (e.g., from a channel identification using a training sequence). Also the pulse shape $g(t)$ is known. The unknowns are the parameters $\{\tau_i\}$ and $\{\beta_i\}$. Our objective is to estimate these parameters.

As before, we can introduce the DFT transformation and the deconvolution by the known pulse shape,

$$\mathbf{z} := \{\mathrm{diag}(\tilde{\mathbf{g}})\}^{-1}\, \mathcal{F}\mathbf{h}\,, \qquad (N \times 1)\,.$$

The vector $\mathbf{z}$ has model

$$\mathbf{z} = \mathbf{F}\mathbf{b}\,, \qquad \mathbf{F} = [\mathbf{f}(\phi_1)\,, \cdots, \mathbf{f}(\phi_r)]\,, \qquad \mathbf{f}(\phi) := \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{N-1} \end{bmatrix}.$$

[3] As in Sec. 4.3.6, but for a single receiver antenna.

Since there are now multiple components in $\mathbf{F}$ and only a single vector $\mathbf{z}$, we cannot simply estimate the parameters from this single vector by splitting it in $\mathbf{x}$ and $\mathbf{y}$: this would allow only to estimate a model with a single component. However, we can use the shift-invariance of the vectors $\mathbf{f}(\cdot)$ to construct a matrix out of $\mathbf{z}$ as

$$\mathbf{Z} = [\mathbf{z}^{(0)}\,,\ \mathbf{z}^{(1)}\,,\ \cdots\,,\ \mathbf{z}^{(m-1)}]\,, \quad (N-m+1 \times m) \tag{8.18}$$

where

$$\mathbf{z}^{(i)} := \begin{bmatrix} z_{i+1} \\ z_{i+2} \\ \vdots \\ z_{N-m+i} \end{bmatrix}$$

is a subvector of $\mathbf{z}$ containing the $i+1$-st till the $N-m+i$-th entry. If we define $\mathbf{f}(\phi)^{(i)}$ similarly, then

$$\mathbf{f}(\phi)^{(i)} = \begin{bmatrix} \phi^i \\ \phi^{i+1} \\ \phi^{i+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \end{bmatrix} \phi^i =: \mathbf{f}'(\phi)\phi^i\,.$$

Thus, $\mathbf{Z}$ has the model

$$\mathbf{Z} = \mathbf{F}'\mathbf{B}\,, \qquad \mathbf{F}' = [\mathbf{f}'(\phi_1)\,, \cdots\,, \mathbf{f}'(\phi_r)]$$

$$\mathbf{B} = [\mathbf{b} \quad \mathbf{\Phi b} \quad \mathbf{\Phi}^2\mathbf{b} \quad \cdots \quad \mathbf{\Phi}^{m-1}\mathbf{b}]\,, \qquad \mathbf{\Phi} = \begin{bmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_r \end{bmatrix}$$

where $\mathbf{F}'$ is a submatrix of $\mathbf{F}$ of size $N-m+1 \times r$, and $\mathbf{B}$ has size $r \times m$. Since each column of $\mathbf{F}'$ has the required shift-invariant structure, this is a model of the form that can be used by ESPRIT: split $\mathbf{Z}$ into $\mathbf{X}$ and $\mathbf{Y}$,

$$\mathbf{Z} = \left[\begin{array}{c} \mathbf{X} \\ \hline * \, * \, * \end{array}\right] = \left[\begin{array}{c} * \, * \, * \\ \hline \mathbf{Y} \end{array}\right]$$

where $\mathbf{X}$ contains all but the last rows of $\mathbf{Z}$, and $\mathbf{Y}$ contains all but the first. Subsequently compute the eigenvalue decomposition

$$\mathbf{X}^\dagger\mathbf{Y} = \mathbf{T}^{-1}\mathbf{\Phi}\mathbf{T}\,.$$

This determines $\mathbf{\Phi}$ as the eigenvalues of $\mathbf{X}^\dagger\mathbf{Y}$, from which the delays $\{\tau_i\}$ can be estimated.

This algorithm produces high-resolution estimates of the delays, in case the parametrized model holds with good accuracy for $\mathbf{h}$. There is however one condition to check. ESPRIT requires that the factorization $\mathbf{Z} = \mathbf{F}'\mathbf{B}$ is a low-rank factorization, i.e., if $\mathbf{F}'$ is strictly tall ($N-m+1 > r$) and $\mathbf{B}$ is square or wide ($r \leq m$). These conditions imply

$$r \leq \frac{1}{2}N\,.$$

Thus, there is a limit on the number of rays that can be estimated: not more than half the number of samples in frequency domain. If this condition cannot be satisfied, we need to use multiple antennas. This is discussed in Sec. 9.3.

8.4 FREQUENCY ESTIMATION

The ESPRIT algorithm can also be used to estimate frequencies. Consider a signal $x(t)$ which is the sum of d harmonic components,

$$x(t) = \sum_{i=1}^{d} \beta_i e^{j\omega_i t} \tag{8.19}$$

Suppose that we uniformly sample this signal with period T (satisfying the Nyquist criterion, here $-\pi \leq \omega_i T < \pi$), and have available $x(T),\ x(2T),\ \cdots,\ x(NT)$. We can then collect the samples in a data matrix $\mathbf{Z}$ with m rows,

$$\mathbf{Z} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots \\ x_2 & x_3 & x_4 & \cdots \\ \vdots & \vdots & & \vdots \\ x_m & x_{m+1} & \cdots & x_N \end{bmatrix}, \qquad x_k = x(kT)\,.$$

From (8.19), we see that this matrix satisfies the model

$$\mathbf{Z} = \mathbf{AS} := \begin{bmatrix} 1 & \cdots & 1 \\ \phi_1 & \cdots & \phi_d \\ \phi_1^2 & \cdots & \phi_d^2 \\ \vdots & & \vdots \\ \phi_1^{m-1} & \cdots & \phi_d^{m-1} \end{bmatrix} \begin{bmatrix} \beta_1\phi_1 & \beta_1\phi_1^2 & \cdots \\ \vdots & \vdots & \\ \beta_d\phi_d & \beta_d\phi_d^2 & \cdots \end{bmatrix}$$

where $\phi_i = e^{j\omega_i T}$. Since the model is the same as before, we can estimate the phase factors $\{\phi_i\}$ as before using ESPRIT, and from these the frequencies $\{\omega_i\}$ follow uniquely, since the Nyquist condition was assumed to hold.

The parameter m has to be chosen larger than d. A larger m will give more accurate estimates, however if N is fixed then the number of columns of $\mathbf{Z}$ $(= N - m + 1)$ will get smaller and there is a tradeoff. For a single sinusoid in noise, one can show that the most accurate estimate is obtained by making $\mathbf{Z}$ rectangular with 2 times more columns than rows, $m = \frac{N}{3}$.

8.5 SYSTEM IDENTIFICATION

Linear time-invariant (LTI) systems can be represented using state-space models. This is in particular convenient in the case of systems with multiple inputs and multiple outputs (MIMO). The time-invariance gives rise to a shift invariance property, which allows to identify the state-space matrices.

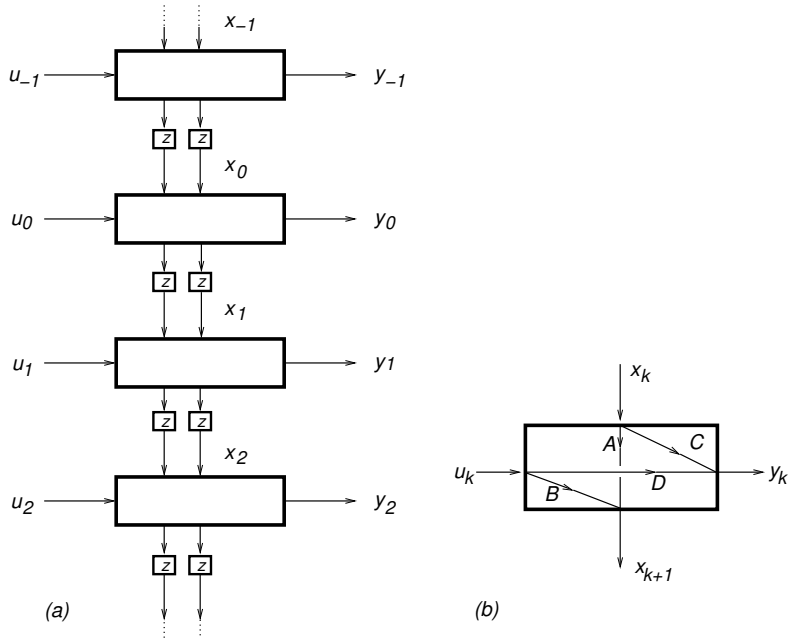


Figure 8.6. LTI state space model. (a) Mapping of an input sequence $\{u_i\}$ to an output sequence $\{y_i\}$ using an intermediate state sequence $\{x_i\}$. The state dimension is $d = 2$. Due to causality, the signal flow is from top to bottom. The delay operator z^{-1} denotes a time shift here. (b) The operation at a particular time instant k is a linear map from input u_k and current state $\mathbf{x}_k$ to output y_k and next state $\mathbf{x}_{k+1}$.

8.5.1 State space model

8.5.2 State space representation

The familiar state space model used to describe causal LTI systems is (for a system with a scalar input u_k and a scalar output y_k),

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \\ y_k &= \mathbf{C}\mathbf{x}_k + Du_k\,. \end{aligned} \tag{8.20}$$

Here, $\mathbf{x}_k$ is the state vector (assumed to have d entries), $\mathbf{A}$ is a $d \times d$ state transition matrix, $\mathbf{B}$ and C^{T} are $d \times 1$ vectors, and D is a scalar (see Fig. 8.6). The integer d is called the state dimension or system order. All finite dimensional linear systems can be described in this way.

The representation (8.20) is not at all unique. An equivalent system representation (yielding the same input-output relationship) is obtained by applying a state transformation $\mathbf{R}$ (an invertible

$d \times d$ matrix) to define a new state vector $\mathbf{x}'_k = \mathbf{R}\mathbf{x}_k$. The equivalent system is

$$\begin{aligned} \mathbf{x}'_{k+1} &= \mathbf{A}'\mathbf{x}'_k + \mathbf{B}'u_k \\ y_k &= \mathbf{C}'x'_k + Du_k \end{aligned}$$

where the new state space quantities are given by

$$\begin{bmatrix} \mathbf{A}' & \mathbf{B}' \\ \mathbf{C}' & D \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{-1} & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & D \end{bmatrix} \begin{bmatrix} \mathbf{R} & \\ & 1 \end{bmatrix}.$$

The eigenvalues of $\mathbf{A}$ remain invariant under this transformation since $\mathbf{R}^{-1}\mathbf{A}\mathbf{R}$ is a similarity transformation. The eigenvalues of $\mathbf{A}$ are directly related to the poles of the system; for stability, they are required to be bounded by 1.

The impulse response of this system is

$$\mathbf{h} = [\cdots \quad 0 \quad \boxed{D} \quad \mathbf{CB} \quad \mathbf{CAB} \quad \mathbf{CA}^2\mathbf{B} \quad \cdots]^{\mathrm{H}}. \tag{8.21}$$

The *realization problem* is to find a state space representation that matches a given impulse response. As pointed out above, this representation is not unique.

8.5.3 Hankel operator

The solution to the realization problem in a subspace context calls for the Hankel matrix, defined from the impulse response as

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots \\ h_2 & h_3 & & \\ h_3 & & \ddots & \\ \vdots & & & \end{bmatrix}. \tag{8.22}$$

The Hankel structure is recognized: $\mathbf{H}$ is constant along the anti-diagonals.

Let us define the controllability operator $\mathbf{C}$ and observability operator $\mathcal{O}$ as

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix}; \qquad \mathcal{C} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \cdots]. \tag{8.23}$$

Then, using (8.21) and comparing to (8.22) shows that $\mathbf{H}$ has a factorization as

$$\mathbf{H} = \mathcal{O}\mathcal{C}$$

For a minimal realization, $\mathcal{C}$ and $\mathcal{O}$ have by definition full rank d. Since $\mathbf{H}$ is an outer product of rank d matrices, it must be of rank d itself. Even for minimal realizations, there is of course

an ambiguity in this factorization. With $\mathbf{R}$ an invertible $d \times d$ matrix, we can also factor $\mathbf{H}$ as $\mathbf{H} = \mathcal{O}'\mathcal{C}' = \mathcal{O}\mathbf{R} \cdot \mathbf{R}^{-1}\mathcal{C}$, corresponding to a state space model that has undergone a state transformation by $\mathbf{R}$ as described above. Factorizations modulo $\mathbf{R}$ lead to equivalent systems.

$\mathcal{C}$ and $\mathcal{O}$ have a shift-invariance structure. E.g., if we let $\mathcal{O}^{\uparrow}$ denote $\mathcal{O}$ with its top row removed (thus, shifted upwards), then (8.23) shows that

$$\mathcal{O}^{\uparrow} = \mathcal{O}\mathbf{A}$$

Likewise, if $\mathcal{C}^{\rightarrow}$ denotes $\mathcal{C}$ with its first column removed (thus, shifted to the left), then

$$\mathcal{C}^{\rightarrow} = \mathbf{A}\mathcal{C}$$

This shift-invariance carries over to $\mathbf{H}$.

$$\begin{array}{rcccl} \mathbf{H}^{\uparrow} & = & \mathcal{O}^{\uparrow}\mathcal{C} & = & \mathcal{O}\mathbf{A} \cdot \mathcal{C} \\ \mathbf{H}^{\leftarrow} & = & \mathcal{O}\mathcal{C}^{\leftarrow} & = & \mathcal{O} \cdot \mathbf{A}\mathcal{C}\,. \end{array}$$

Thus it is seen that shifting $\mathbf{H}$ upwards or to the left is equivalent to a multiplication by $\mathbf{A}$ in the center of the factorization.

8.5.4 Realization scheme

Using the above two properties of the Hankel operator $\mathbf{H}$ — i.e., that it is of finite rank with some minimal factorization $\mathbf{H} = \mathcal{O}\mathcal{C}$, and that it is shift-invariant — we will show how to obtain a state space realization as in equation (8.20) from a given impulse response.

1. Given the impulse reponse, construct the Hankel matrix $\mathbf{H}$ as in (8.22). Determine the rank d, and any factorization $\mathbf{H} = \mathcal{O}\mathcal{C}$, where $\mathcal{O}$ and $\mathcal{C}$ are of full rank d. The SVD is a robust tool for doing this.

2. At this point, we know that $\mathcal{C}$ and $\mathcal{O}$ have the shift-invariant structure of equation (8.23). Use this property to derive

   $$\mathcal{O}\mathbf{A} = \mathcal{O}^{\uparrow} \qquad \Rightarrow \qquad \mathbf{A} = \mathcal{O}^{\dagger}\mathcal{O}^{\uparrow}$$

   Because $\mathcal{O}$ is of full row rank d, we have $\mathcal{O}^{\dagger} = (\mathcal{O}^{\mathrm{H}}\mathcal{O})^{-1}\mathcal{O}^{\mathrm{H}}$. This determines $\mathbf{A}$. The matrices $\mathbf{B}$, $\mathbf{C}$ and D follow simply as

   $$\begin{array}{rcl} \mathbf{B} & = & \mathcal{C}_{(:,1)} \\ \mathbf{C} & = & \mathcal{O}_{(1,:)} \\ D & = & h_0 \end{array}$$

   where the subscript $(:, 1)$ denotes the first column of the associated matrix, and $(1, :)$ the first row.

In practice, $\mathbf{H}$ should have finite size. This issue can be dealt with relatively easily. Further, in practice we may not have the impulse response, but only a single input signal $\mathbf{u}$ plus its corresponding output signal $\mathbf{y}$. With some effort, we can adapt the algorithm to this situation. See Verhaegen [7, 8] for details.

8.6 REAL PROCESSING

TBD: unitary ESPRIT [2, 3]

8.7 NOTES

The ESPRIT algorithm was originally proposed by Roy and Kailath in [9, 10]. See [11, 12] for overviews.

Delay estimation using ESPRIT was proposed in [13].

Bibliography

[1] R. Roy and T. Kailath, "ESPRIT – Estimation of Signal Parameters via Rotational Invariance Techniques," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 37, pp. 984–995, July 1989.

[2] M. Haardt and J. Nossek, "Unitary ESPRIT: how to obtain increased estimation accuracy with a reduced computational burden," *IEEE Trans. Signal Proc.*, vol. 43, pp. 1232–1242, May 1995.

[3] M. Zoltowski, M. Haardt, and C. Mathews, "Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via Unitary ESPRIT," *IEEE Trans. Signal Proc.*, vol. 44, pp. 316–328, February 1996.

[4] P. Stoica and K. Sharman, "Maximum Likelihood methods for direction-of-arrival estimation," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 38, pp. 1132–1143, July 1990.

[5] T. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust. Speech Signal Proc.*, vol. 33, pp. 806–811, April 1985.

[6] U. Pillai and B. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 37, pp. 8–15, January 1989.

[7] M. Verhaegen and P. Dewilde, "Subspace model identification. Part 1: The Output Error state space model identification class of algorithms," *Int. J. Control*, vol. 56, no. 5, pp. 1187–1210, 1992.

[8] M. Verhaegen and P. Dewilde, "Subspace model identification. Part 2: Analysis of the elementary Output-Error state-space model identification algorithm," *Int. J. Control*, vol. 56, no. 5, pp. 1211–1241, 1992.

[9] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT—a subspace rotation approach to estimation of parameters of cisoids in noise," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 34, pp. 1340–1342, Oct. 1986.

[10] R. Roy, *ESPRIT*. PhD thesis, Stanford Univ., Stanford, CA, 1987.

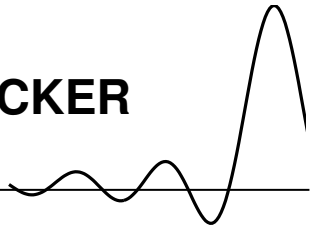
[11] F. Li and R. Vaccaro, "Analytical performance prediction of subspace-based algorithms for DOA estimation," in *SVD and Signal Processing, II: Algorithms, Analysis and Applications* (R. Vaccaro, ed.), pp. 243–260, Elsevier, 1991.

[12] A. van der Veen, E. Deprettere, and A. Swindlehurst, "Subspace based signal analysis using singular value decomposition," *Proceedings of the IEEE*, vol. 81, pp. 1277–1308, Sept. 1993.

[13] A. van der Veen, M. Vanderveen, and A. Paulraj, "Joint angle and delay estimation using shift-invariance properties," *subm. IEEE Signal Processing Letters*, Aug. 1996.

Chapter 9

JOINT DIAGONALIZATION AND KRONECKER PRODUCT STRUCTURES



Contents

In the previous chapter, we have seen how direction finding of narrowband sources using a ULA, delay estimation, and frequency estimation, all lead to a similar data model that shows shift-invariance, and can be solved using the ESPRIT algorithm. In this chapter, we extend this to a number of *joint* estimation techniques: determining the two-dimensional directions of arrival, joint angle-delay estimation, and joint angle-frequency estimation. The data models related to these applications have the same Khatri-Rao product structure, as well as the shift-invariance structure. Therefore, the estimation algorithms can be based on an extension of the ESPRIT algorithm to two dimensions. In many cases, the shift invariance is not even needed to enable source separation: the Khatri-Rao product structure suffices.

9.1 JOINT AZIMUTH AND ELEVATION ESTIMATION

9.1.1 The data model

As an extension of the ESPRIT-related doublet array, consider M sensor triplets, each composed of three identical sensors with unknown gain and phase patterns, which may vary from triplet to triplet. For every triplet, the displacement vectors $\mathbf{d}_{xy}$ and $\mathbf{d}_{xz}$ between its components are required to be the same. See Fig. 9.1.

This way, collecting N time-domain samples and assigning the three sensors of each triplet to each of the data matrices $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$, respectively, three identical although displaced arrays are

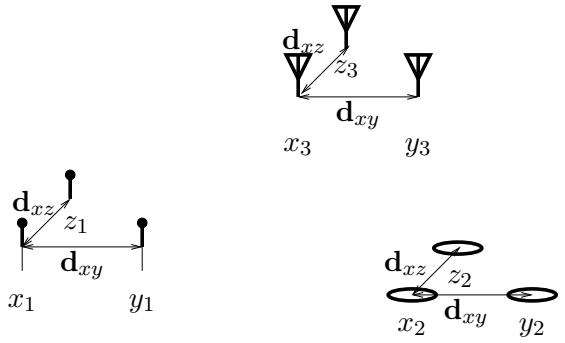


Figure 9.1. Sensor array consisting of triplets

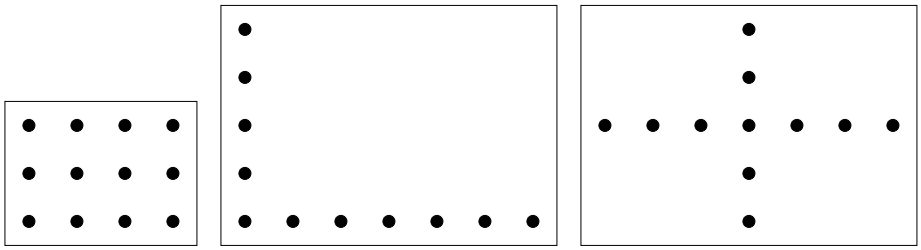


Figure 9.2. Possible array configurations: a uniform rectangular array (URA), an L-shaped array, and a +-shaped array.

obtained. Impinging on every array are d narrowband non-coherent signals $s_k(t)$. In a direct extension of the data models in Chap. 8, we obtain the data model (ignoring the noise for the moment)

$$\left\{ \begin{array}{rcl} \mathbf{X} & = & \mathbf{A}_x\mathbf{S} \\ \mathbf{Y} & = & \mathbf{A}_y\mathbf{S} \\ \mathbf{Z} & = & \mathbf{A}_z\mathbf{S} \end{array} \right. = \left\{ \begin{array}{rcl} \mathbf{X} & = & \mathbf{AS} \\ \mathbf{Y} & = & \mathbf{A\Phi S} \\ \mathbf{Z} & = & \mathbf{A\Theta S} \end{array} \right. \quad \Leftrightarrow \quad \left[\begin{array}{c} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{array} \right] = \left[\begin{array}{c} \mathbf{A} \\ \mathbf{A\Phi} \\ \mathbf{A\Theta} \end{array} \right] \mathbf{S}\,. \tag{9.1}$$

We write $\mathbf{A} = \mathbf{A}_x$ for brevity of notation. We will not assume a more detailed structure of $\mathbf{A}$: all structure in the problem is obtained by the assumption of shift invariance in the relation of $\mathbf{A}_x$ to $\mathbf{A}_y$ and $\mathbf{A}_z$.

Fig. 9.2 shows some other antenna configurations that lead to the required shift invariance: a Uniform Rectangular Array (URA), an L-shaped array, and a +-shaped array. The latter two have a larger aperture for the same number of antennas, but it is sparsely filled, and the number of baselines that are used is less: $\mathbf{A}$ has fewer rows. Hence, it is hard to say a priori which geometry is preferred.

Due to the shift-invariance of the array, $\mathbf{A}_y = \mathbf{A}_x\mathbf{\Phi}$ and $\mathbf{A}_z = \mathbf{A}_x\mathbf{\Theta}$, where $\mathbf{\Phi}$ and $\mathbf{\Theta}$ are diagonal

matrices with entries

$$\begin{aligned} \phi_k &= e^{-j\frac{\omega_0}{c}\mathbf{d}_{xy}\cdot\boldsymbol{\zeta}_k} \\ \theta_k &= e^{-j\frac{\omega_0}{c}\mathbf{d}_{xz}\cdot\boldsymbol{\zeta}_k}\,, \quad k = 1, \cdots, d\,, \end{aligned} \tag{9.2}$$

in which $\boldsymbol{\zeta}_k$ is the propagation direction vector of the kth signal, ω_0 is the carrier frequency of the d signals, and c is the propagation velocity.

$\mathbf{S}$ is the signal matrix ($d \times N$). The matrices $\mathbf{A}$ and $\mathbf{S}$ are unknown, and are not rank-deficient by assumption. The matrices
bPhi and
bTheta are diagonal and contain the phase shifts (9.2) for each signal:

$$\begin{aligned} \boldsymbol{\Phi} &= \mathrm{diag}(\phi_1, \phi_2, \cdots, \phi_d) \\ \boldsymbol{\Theta} &= \mathrm{diag}(\theta_1, \theta_2, \cdots, \theta_d) \end{aligned}$$

The DOA problem is to estimate $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ from $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. From these matrices, the 2D angles of arrival can directly be computed.

At this point, of course, the ESPRIT algorithm can be applied separately to $(\mathbf{X}, \mathbf{Y})$ and $(\mathbf{X}, \mathbf{Z})$. This will produce two sets of angles. However, the angles are listed in random order. How can the correct pairs be found?

9.1.2 Preprocessing: estimating the column span

Since there are d sources, we would like to reduce the problem to matrices of size $d \times d$. As before, this is done by computing an SVD. First, construct the combined data matrix

$$\mathbf{K} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix}.$$

In view of the model (9.1), we know that without noise, $\mathbf{K}$ has rank d. Therefore, we can compute the 'economy-size' SVD,

$$\mathbf{K} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}$$

where $\mathbf{U}$ has d columns. In the case of noise, we need to actually compute the truncated SVD where we take the dominant d singular components. Also, if we assume array configurations as shown in Fig. 9.2, where $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ share some of the same elements, then the SVD is computed on a data matrix $\mathbf{K}$ that contains only the unique elements.

Partitioning $\mathbf{U}$ in the same way as $\mathbf{K}$ gives

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_x \\ \mathbf{U}_y \\ \mathbf{U}_z \end{bmatrix} \boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}\,.$$

The d columns of $\mathbf{U}$ span the signal subspace. Comparing to the model (9.1), we find that there must be a $d \times d$ invertible matrix $\mathbf{T}$ that maps one basis of the subspace to the other:

$$\begin{bmatrix} \mathbf{U}_x \\ \mathbf{U}_y \\ \mathbf{U}_z \end{bmatrix} = \begin{bmatrix} \mathbf{A}_x \\ \mathbf{A}_y \\ \mathbf{A}_z \end{bmatrix} \mathbf{T} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\boldsymbol{\Phi} \\ \mathbf{A}\boldsymbol{\Theta} \end{bmatrix} \mathbf{T} \tag{9.3}$$

This implies

$$\mathbf{U}^{\mathrm{H}}\mathbf{K} = \boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} = \mathbf{T}^{-1}\mathbf{S}$$

so that the separating beamformer (such that $\mathbf{W}^{\mathrm{H}}\mathbf{K} = \mathbf{S}$) is

$$\mathbf{W}^{\mathrm{H}} = \mathbf{T}\mathbf{U}^{\mathrm{H}}.$$

The source separation problem thus reduces to find $\mathbf{T}$.

9.1.3 Joint diagonalization

Equation (9.3) shows that $\mathbf{A} = \mathbf{U}_x\mathbf{T}^{-1}$, so that

$$\begin{cases} \mathbf{U}_y &= \mathbf{U}_x\,\mathbf{T}^{-1}\boldsymbol{\Phi}\mathbf{T} \\ \mathbf{U}_z &= \mathbf{U}_x\,\mathbf{T}^{-1}\boldsymbol{\Theta}\mathbf{T} \end{cases} \tag{9.4}$$

Assuming $\mathbf{U}_x$ has a left inverse (this requires at least $M \geq d$), compute $\mathbf{M}_y = \mathbf{U}_x^{\dagger}\mathbf{U}_y$ and $\mathbf{M}_z = \mathbf{U}_x^{\dagger}\mathbf{U}_z$, both of size $d \times d$ Then these have model

$$\begin{cases} \mathbf{M}_y &= \mathbf{T}^{-1}\boldsymbol{\Phi}\mathbf{T} \\ \mathbf{M}_z &= \mathbf{T}^{-1}\boldsymbol{\Theta}\mathbf{T} \end{cases} \tag{9.5}$$

This shows that both $\mathbf{M}_y$ and $\mathbf{M}_z$ are jointly diagonalized by the same $\mathbf{T}$. (Here, we mean diagonalization by a similarity transform; we will see another form of diagonalization later on.)

These two equations are redundant: already one of the two will allow us to compute $\mathbf{T}$. If the eigenvalues $\boldsymbol{\Phi}$ are distinct, then we can compute $\mathbf{T}$ from an eigenvalue decomposition of $\mathbf{M}_y$; its eigenvector matrix $\mathbf{T}$ is unique up to a permutation and a scaling of its columns; this will translate to an unknown scaling and permutation of the rows of $\mathbf{S}$. The scaling can be fixed by prior knowledge on the powers of the sources, or on the norms of the columns of $\mathbf{A}$. In this case, we can compute $\mathbf{T}$ from $\mathbf{M}_y$ and apply it to $\mathbf{M}_z$ to find $\boldsymbol{\Theta} = \mathbf{T}\mathbf{M}_z\mathbf{T}^{-1}$.

Similarly, if the eigenvalues $\boldsymbol{\Theta}$ are distinct, we can compute $\mathbf{T}$ from an eigenvalue decomposition of $\mathbf{M}_z$, and then use $\mathbf{T}$ to compute $\boldsymbol{\Phi}$.

In either case, we find the correct correspondence between the entries of $\boldsymbol{\Phi}$ and those of $\boldsymbol{\Theta}$, i.e., one pair for each source. This correct pairing does not happen if we compute the two eigenvalue decompositions separately, as generally the eigenvalues will appear in random order.

With noise, we use the SVD to compute the dominant subspace $\mathbf{U}$ and proceed as above to find $\mathbf{M}_y$ and $\mathbf{M}_z$. However, now there is not a single $\mathbf{T}$ that exactly diagonalizes both matrices. We would aim to find a single $\mathbf{T}$ to diagonalize as much as possible both matrices. This Joint Approximate Diagonalization problem has several formulations and several algorithms have been proposed, and a decent treatment warrants a separate chapter.

In one formulation, we can propose a QR decomposition of $\mathbf{T}^{-1} = \mathbf{QR}$ (where $\mathbf{Q}$ is unitary and $\mathbf{R}$ is upper triangular), so that

$$\begin{cases} \mathbf{M}_y &= \mathbf{Q}\mathbf{R}_y\mathbf{Q}^{\text{H}} \\ \mathbf{M}_z &= \mathbf{Q}\mathbf{R}_z\mathbf{Q}^{\text{H}} \end{cases} \tag{9.6}$$

where $\mathbf{R}_y$ and $\mathbf{R}_z$ are upper triangular. This translates the problem into a joint Schur decomposition. Since $\mathbf{Q}$ is unitary, it can be composed of 2×2 rotations (called Jacobi rotations), which leads to numerically stable algorithms. The main diagonals of $\mathbf{R}_y$ and $\mathbf{R}_z$ give us $\mathbf{\Phi}$ and $\mathbf{\Theta}$, respectively.

9.2 CONNECTION TO THE KHATRI-RAO PRODUCT STRUCTURE

Recall the model (9.1)

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\mathbf{\Phi} \\ \mathbf{A}\mathbf{\Theta} \end{bmatrix} \mathbf{S}$$

If we define a matrix $\mathbf{F}$ from the diagonals of $\mathbf{\Phi}$ and $\mathbf{\Theta}$ as

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \phi_1 & \phi_2 & \cdots & \phi_d \\ \theta_1 & \theta_2 & \cdots & \theta_d \end{bmatrix}$$

then we can write this compactly as

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = (\mathbf{F} \circ \mathbf{A})\mathbf{S}$$

where $\circ$ denotes the Khatri-Rao product (column-wise Kronecker product); see Sec. 5.1.6 for its definition and some properties.

Likewise, we can write (9.3) compactly as

$$\begin{bmatrix} \mathbf{U}_x \\ \mathbf{U}_y \\ \mathbf{U}_z \end{bmatrix} = (\mathbf{F} \circ \mathbf{A})\mathbf{T}$$

Note that this Khatri-Rao product structure is the *only* property that was needed to derive the joint diagonalization model (9.5), via (9.4) and subsequently removing one matrix ($\mathbf{U}_x$) by inversion. Thus, whenever we have this structure, we can transform it into joint diagonalization.

Further, note that here we expanded $\mathbf{F}$ into its rows; each row leading to a matrix of the form shown in (9.4). But the form $\mathbf{U} = (\mathbf{F} \circ \mathbf{A})\mathbf{T}$ is multilinear: we can also expand along $\mathbf{T}$ or $\mathbf{A}$, and arrive at a joint diagonalization model. E.g., expanding $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_d]$ and likewise $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_d]$ gives

$$\mathbf{u}_k = (\mathbf{F} \circ \mathbf{A})\mathbf{t}_k \qquad \Leftrightarrow \qquad \mathbf{U}_k = \mathbf{A}\mathbf{D}_k\mathbf{F}^{\mathrm{T}}\,, \qquad k = 1, \cdots, d,$$

where $\mathbf{U}_k$ is a $M \times 3$ matrix such that $\mathrm{vec}(\mathbf{U}_k) = \mathbf{u}_k$, and $\mathbf{D}_k$ is a diagonal matrix such that $\mathrm{diag}(\mathbf{D}_k) = \mathbf{t}_k$. We used (5.8) to derive this. If $\mathbf{F}^{\mathrm{T}}$ is square and invertible *(but here it is not; this requires some preprocessing)* then if we premultiply the set of matrices with a left inverse of $\mathbf{U}_1$, where $\mathbf{U}_1^{\dagger} = \mathbf{F}^{-\mathrm{T}}\mathbf{D}_1^{-1}\mathbf{A}^{\dagger}$, then

$$\mathbf{U}_1^{\dagger}\mathbf{U}_k = \mathbf{F}^{-\mathrm{T}}(\mathbf{D}_1^{-1}\mathbf{D}_k)\mathbf{F}^{\mathrm{T}}\,, \qquad k = 2, \cdots d\,.$$

This is indeed another joint diagonalization model, now involving d matrices. Likewise, we can expand along $\mathbf{A}$ and obtain a joint diagonalization model with M matrices.

We have seen here that inverting one term ($\mathbf{U}_x$ or $\mathbf{U}_1$) and applying it to the other components leads to the desired result. That makes the first component "special" in some sense. If it is unreliable due to noise, or if it is poorly conditioned, then that carries over to all other components.

It is possible to avoid the inversion and replace it by correlation, as follows. Consider again (9.3), viz.

$$\begin{bmatrix} \mathbf{U}_x \\ \mathbf{U}_y \\ \mathbf{U}_z \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}\boldsymbol{\Phi} \\ \mathbf{A}\boldsymbol{\Theta} \end{bmatrix} \mathbf{T}\,.$$

This time, premultiply by[1] $\mathbf{U}_x^{\mathrm{H}}$:

$$\begin{bmatrix} \mathbf{U}_x^{\mathrm{H}}\mathbf{U}_x \\ \mathbf{U}_x^{\mathrm{H}}\mathbf{U}_y \\ \mathbf{U}_x^{\mathrm{H}}\mathbf{U}_z \end{bmatrix} = \begin{bmatrix} \mathbf{T}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A} \\ \mathbf{T}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\boldsymbol{\Phi} \\ \mathbf{T}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\boldsymbol{\Theta} \end{bmatrix} \mathbf{T} \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{M}_x \\ \mathbf{M}_y \\ \mathbf{M}_z \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{B}\boldsymbol{\Phi} \\ \mathbf{B}\boldsymbol{\Theta} \end{bmatrix} \mathbf{T}$$

where $\mathbf{B} = \mathbf{T}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is a square invertible matrix (assuming $\mathbf{A}$ is tall and full rank). In other words, we have three $d \times d$ matrices of the form

$$\begin{cases} \mathbf{M}_x &= \mathbf{B}\mathbf{T} \\ \mathbf{M}_y &= \mathbf{B}\boldsymbol{\Phi}\mathbf{T} \\ \mathbf{M}_z &= \mathbf{B}\boldsymbol{\Theta}\mathbf{T} \end{cases}$$

[1] This still singles out one data matrix and transfers its noise over to the other matrices. It would be better to compute the column span of $\mathbf{A}$ from an SVD of $[\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z]$, i.e., stacked in a block row, and use this joint estimate to reduce the dimensions to size $d \times d$.

This is also a joint diagonalization problem, but now "by congruence" and not by similarity.

Also this problem has been well studied and several algorithms have been proposed. One technique to proceed is to insert QR factorizations $\mathbf{B} = \mathbf{QR}$ and $\mathbf{T} = \mathbf{R}'\mathbf{Z}$ (where $\mathbf{R}, \mathbf{R}'$ are upper triangular and $\mathbf{Q}, \mathbf{Z}$ are unitary matrices). Then the problem has the form

$$\left\{ \begin{array}{rcl} \mathbf{M}_x & = & \mathbf{Q}\mathbf{R}_x\mathbf{Z} \\ \mathbf{M}_y & = & \mathbf{Q}\mathbf{R}_y\mathbf{Z} \\ \mathbf{M}_z & = & \mathbf{Q}\mathbf{R}_z\mathbf{Z} \end{array} \right. \tag{9.7}$$

where $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ are upper triangular. We thus need to find unitary matrices $\mathbf{Q}, \mathbf{Z}$ to make $\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z$ upper triangular. From the main diagonals of $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$, we can recover $\mathbf{\Phi}$ and $\mathbf{\Theta}$.

This problem is a "joint" generalized Schur decomposition, see Sec. 5.7. The matrices $\mathbf{Q}, \mathbf{Z}$ can be found using a generalization of the QZ algorithm. Note that a good starting point for the iteration is available by first computing the solution to a single generalized eigenvalue problem.

Comparing (9.7) to (9.6), we see that two unitary matrices $\mathbf{Q}, \mathbf{Z}$ are used instead of only one $\mathbf{Q}$. At the same time, three matrices $\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z$ are available, rather than two. The number of degrees of freedom in two unitary matrices is about equal to that of a single general matrix. Thus, in the present case, the number of equations and number of unknowns has about the same balance.

We have seen that the Khatri-Rao structure of the form $\mathbf{X} = (\mathbf{F} \circ \mathbf{A})\mathbf{T}$ is the root to the joint diagonalization model. This structure is an instance of a more general *canonical polyadic decomposition* (CPD) of the data matrix, in this case of a third order tensor. A CPD aims to find a low multi-linear rank approximation of a given tensor. The model in the present context is similar to *parallel factor analysis* (PARAFAC). A CPD is more general because it allows more than 3 dimensions, gives exact conditions on the dimensions in relation to the rank such that there is a 'unique' decomposition, and allows for sparse representations. The tensor framework also gives access to other decompositions such as a block term decomposition (BTD).

9.3 JOINT ANGLE AND DELAY ESTIMATION

A second application that leads to a joint diagonalization problem is the following. In Sec. 8.3, we studied the multipath estimation problem. Starting from a channel estimate $\mathbf{h}(t)$, we want to estimate the individual path delays, directions of arrival, and path gains of each ray, as shown in Fig. 9.3. With multiple antennas, the channel model is

$$\mathbf{h}(t) = \sum_{i=1}^{r} \mathbf{a}(\alpha_i)\beta_i g(t - \tau_i) \, .$$

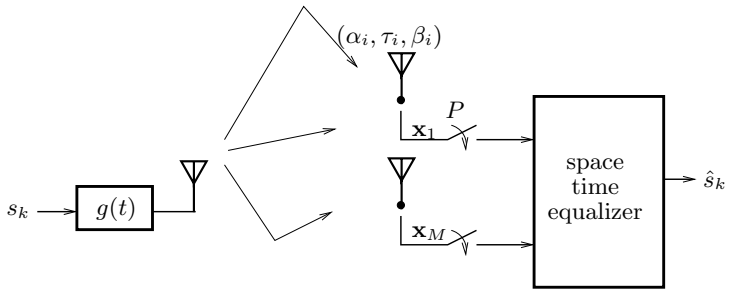


Figure 9.3. Multiray propagation channel

Here, the pulse shape $g(t)$ is known, and the antenna response vector $\mathbf{a}(\alpha)$ is known as function of α. We assume a ULA with interelement spacing Δ wavelengths, so that

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ \theta \\ \vdots \\ \theta^{M-1} \end{bmatrix}, \quad \theta = e^{j2\pi\Delta\sin(\alpha)}.$$

Assume $\mathbf{h}(t)$ is sampled above the Nyquist rate and that we collect N samples. Also assume that the entire support of each $g(t-\tau_i)$ is contained in these samples. We stack the samples of $\mathbf{h}(t)$ into a vector as before,

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbf{h}(0) \\ \mathbf{h}(T) \\ \vdots \\ \mathbf{h}((N-1)T) \end{bmatrix} = \sum_{i=1}^{r} [\mathbf{g}_{\tau_i} \otimes \mathbf{a}(\alpha_i)]\beta_i = [\mathbf{G} \circ \mathbf{A}]\mathbf{b}. \tag{9.8}$$

The N samples of $g(t-\tau_i)$ are stacked in the vector $\mathbf{g}_{\tau_i}$, and we will assume that the entire support of each $g(t-\tau_i)$ is contained in these samples.

The equation shows the Khatri-Rao structure, which in the previous section we established to be the root of the joint diagonalization model. How does this work out here?

- We can rearrange $\mathbf{h}$ into an $M \times N$-matrix $\mathbf{H}$:

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{N-1} \end{bmatrix} \quad \Leftrightarrow \quad \mathbf{H} = [\mathbf{h}_0 \;\; \mathbf{h}_1 \;\; \cdots \;\; \mathbf{h}_{N-1}]$$

  Since $\mathbf{h} = \mathrm{vec}(\mathbf{H})$, we find (with property (5.8))

$$\mathbf{H} = \mathbf{A}\mathrm{diag}(\mathbf{b})\mathbf{G}^{\mathrm{T}}. \tag{9.9}$$

With just a single matrix $\mathbf{H}$, we do not have sufficient information to uniquely determine its factorization: there is no "joint" diagonalization.

- Alternatively, expand $\mathbf{G}$ into rows $\mathbf{g}_i^{\mathrm{T}}$. This gives

$$\mathbf{h}_i = \mathbf{A}\mathrm{diag}(\mathbf{g}_i)\mathbf{b}\,, \qquad i = 0, \cdots, N-1$$

Here, joint diagonalization also doesn't work because we just have single vectors $\mathbf{h}_i$, not matrices.

- Expanding on the rows of $\mathbf{A}$, we reach the same conclusion.

Thus, before we can proceed, we need to find a way to expand a single vector into a matrix. We have seen in Chap. 8 that if $\mathbf{A}$ corresponds to a ULA (or a doublet structure is sufficient), we can apply spatial smoothing to do this. Alternatively, after a DFT and deconvolution, we can use the similar structure resulting in $\mathbf{G}$ to do a similar smoothing.

Indeed, this is what we did in Sec. 8.3 on single-antenna data. There, we applied a DFT to the time domain samples in $\mathbf{h}$, resulting in a vector $\mathbf{z}$, and then in (8.18) constructed a matrix $\mathbf{Z}$ from m shifts of $\mathbf{z}$ so that

$$\mathbf{Z} = [\mathbf{z}^{(0)}\,,\ \mathbf{z}^{(1)}\,,\ \cdots\,,\ \mathbf{z}^{(m-1)}]\,, \qquad M(N-m+1)\times m\,. \tag{9.10}$$

Extending the results in Sec. 8.3 to multiple antennas, we see that $\mathbf{Z}$ has a model

$$\mathbf{Z} = [\mathbf{F}\circ\mathbf{A}]\mathbf{B}\,, \qquad \mathbf{F} = [\mathbf{f}(\phi_1)\,,\cdots,\mathbf{f}(\phi_r)] \tag{9.11}$$

where

$$\mathbf{f}(\phi) = \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{N-1} \end{bmatrix}\,, \qquad \phi := e^{-j\frac{2\pi}{N}\frac{\tau}{T}}$$

$$\mathbf{B} = [\mathbf{b} \quad \mathbf{\Phi b} \quad \mathbf{\Phi}^2\mathbf{b} \quad \cdots \quad \mathbf{\Phi}^{m-1}\mathbf{b}]\,, \qquad \mathbf{\Phi} = \begin{bmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_r \end{bmatrix}\,.$$

Therefore, the shift invariance in the time domain (after the DFT) allows us to expand a single vector $\mathbf{b}$ to a matrix $\mathbf{B}$.

In Sec. 8.3, we applied ESPRIT to only a shift along the time domain. With a ULA, we can also expand using the shift invariance of $\mathbf{A}$. This leads to a joint diagonalization with two matrices, and allows us to jointly estimate both delays and angles of arrival.

Thus consider $\mathbf{Z}$ in (9.10), and assume m is large enough such that $\mathbf{Z}$ has rank r (in the noise-free case). As usual, we proceed by computing the (truncated) SVD of $\mathbf{Z}$,

$$\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}},$$

where we truncate to rank r, i.e., $\mathbf{U}$ has r columns. Comparing to the model (9.11) we see

$$\mathbf{U} = (\mathbf{F} \circ \mathbf{A})\mathbf{T}, \quad \mathbf{B} = (\mathbf{T}\mathbf{U}^{\mathrm{H}})\mathbf{Z}.$$

To estimate $\mathbf{T}$, we form two types of selection matrices: a pair to select submatrices of $\mathbf{F}$, and a pair to select from $\mathbf{A}$,

$$\begin{aligned} \mathbf{J}_{x\phi} &:= [\mathbf{I}_{N-1} \quad \mathbf{0}_1] \otimes \mathbf{I}_M, & \mathbf{J}_{x\theta} &:= \mathbf{I}_N \otimes [\mathbf{I}_{M-1} \quad \mathbf{0}_1], \\ \mathbf{J}_{y\phi} &:= [\mathbf{0}_1 \quad \mathbf{I}_{N-1}] \otimes \mathbf{I}_M, & \mathbf{J}_{y\theta} &:= \mathbf{I}_N \otimes [\mathbf{0}_1 \quad \mathbf{I}_{M-1}]. \end{aligned}$$

To estimate $\mathbf{\Phi}$, we take submatrices consisting of the first and respectively last $M(N-1)$ rows of $\mathbf{U}$, i.e.,

$$\mathbf{U}_{x\phi} = \mathbf{J}_{x\phi}\mathbf{U}, \qquad \mathbf{U}_{y\phi} = \mathbf{J}_{y\phi}\mathbf{U},$$

whereas to estimate $\mathbf{\Theta}$ we stack, for all N blocks, its first and respectively last $M-1$ rows:

$$\mathbf{U}_{x\theta} = \mathbf{J}_{x\theta}\mathbf{U}, \qquad \mathbf{U}_{y\theta} = \mathbf{J}_{y\theta}\mathbf{U}.$$

These new data matrices have the structure

$$\left\{ \begin{array}{lcl} \mathbf{U}_{x\phi} &=& \mathbf{A}'\mathbf{T} \\ \mathbf{U}_{y\phi} &=& \mathbf{A}'\mathbf{\Phi}\mathbf{T} \end{array} \right. \qquad \left\{ \begin{array}{lcl} \mathbf{U}_{x\theta} &=& \mathbf{A}''\mathbf{T} \\ \mathbf{U}_{y\theta} &=& \mathbf{A}''\mathbf{\Theta}\mathbf{T} \end{array} \right. \tag{9.12}$$

If dimensions are such that these are low-rank factorizations, then

$$\begin{array}{lcl} \mathbf{U}_{x\phi}^{\dagger}\mathbf{U}_{y\phi} &=& \mathbf{T}^{-1}\mathbf{\Phi}\mathbf{T} \\ \mathbf{U}_{x\theta}^{\dagger}\mathbf{U}_{y\theta} &=& \mathbf{T}^{-1}\mathbf{\Theta}\mathbf{T} \end{array} \tag{9.13}$$

This is again a joint diagonalization problem, where a single matrix $\mathbf{T}$ can diagonalize two data matrices. Having found the eigenvalue matrices $\mathbf{\Phi}$ and $\mathbf{\Theta}$, we can retrieve the delays and angles of each ray. The correct pairing of angles to delays follows simply from the fact that they share the same eigenvectors.

More multipath With a straightforward extension of this approach, we can estimate the multipath parameters of d sources, where each source is received via a superposition of rays, each with its own angle θ_i, delay τ_i, and fading β_i. The corresponding data model is

$$\mathbf{X} = (\mathbf{G} \circ \mathbf{A})\mathbf{B}\mathbf{J}\mathbf{S}, \tag{9.14}$$

where $\mathbf{B}$ is the diagonal matrix containing all fading parameters, and $\mathbf{J}$ is a $r \times d$ selection matrix which assigns each ray to one of the sources. A similar model was derived in (4.23).

The presence of $\mathbf{J}$ and $\mathbf{S}$ requires some additional processing steps: we try to estimate $r > d$ components from a rank-d matrix. This is the same problem as we encountered in the "coherent multipath" problem in Sec. 8.2.7, and we can proceed in the same way.

In summary, an important property of the joint processing is that it allows us to simultaneously identify parameters of many more rays than we have antennas: by combining with the time domain, we extend $\mathbf{A}$ to $\mathbf{G} \circ \mathbf{A}$.

By combining with Sec. 9.1, the algorithm has an elegant extension to the estimation of delays and both azimuth and elevation angles. This results in a joint diagonalization problem of *three* matrices. Similar generalizations occur if we have a non-uniform array with multiple baselines.

Exploiting fading diversity At the start of the section, we mentioned the multipath model (9.9)

$$\mathbf{h} = [\mathbf{G} \circ \mathbf{A}]\mathbf{b}\,.$$

Since only a single channel vector is available, we needed to exploit shift invariance of $\mathbf{A}$ or (after the DFT) $\mathbf{G}$ to expand this to a matrix that admits joint diagonalization.

However, in mobile communication we often experience fast fading. In this case, angles and delays of $\mathbf{h}$ remain more or less constant over time (in the order of microseconds), but $\mathbf{b}$ fluctuates. If we obtain multiple channel estimates $\mathbf{h}_k$ with constant angles and delays, but each with different fading amplitudes $\mathbf{b}_k$, then

$$[\mathbf{h}_1, \mathbf{h}_2, \cdots] = [\mathbf{G} \circ \mathbf{A}]\mathbf{B}\,, \qquad \mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots]\,.$$

By unvectoring the $\mathbf{h}_k$, we immediately obtain a joint diagonalization model of the form

$$\mathbf{H}_k = \mathbf{A}\mathrm{diag}(\mathbf{b}_k)\mathbf{G}^{\mathrm{T}}\,, \qquad k = 1, 2, \cdots\,.$$

Joint diagonalization algorithms will allow us to estimate the columns of $\mathbf{A}$ and the rows of $\mathbf{G}$, without making any further assumptions on the structure of $\mathbf{A}$ or $\mathbf{G}$: we do not need shift invariance.

Fading diversity can also be exploited in other applications. As a simple example, consider d independent narrowband unit-power sources impinging on an antenna array. In the kth transmission block, the received data is

$$\mathbf{x}_k[n] = \mathbf{A}\mathrm{diag}(\mathbf{b}_k)\mathbf{s}_k[n]$$

(ignoring the noise), where $\mathbf{b}_k$ are the complex amplitudes, including the source powers. Subject to fading, we assume that these are different for each k. Then the correlation matrix of $\mathbf{x}_k[n]$ is

$$\mathbf{R}_k = \mathbf{A}\mathrm{diag}(\mathbf{b}_k)^2\mathbf{A}^{\mathrm{H}}\,.$$

This again leads to a joint diagonalization problem. We do not need to make assumptions on the structure of $\mathbf{A}$ to be able to separate the sources.

9.4 JOINT ANGLE AND FREQUENCY ESTIMATION

A somewhat different scenario than what we considered before, which however leads to the same type of data models (and thus the same beamforming algorithms), is the following. Suppose that we observe a frequency band of interest, and want to separate all sources that are present. Assume that the sources are narrowband, typically with different carrier frequencies, but that the spectra might be partly overlapping. The objective is to construct a beamformer to separate the sources based on differences in angles or carrier frequencies. This is a problem of *joint angle-frequency estimation* [1, 2]. We will assume that the sample rates in this application are much higher than the data rates of each source, and that there is only coherent multipath, although generalizations are possible.

Suppose that the narrowband signals have a bandwidth of less than $\frac{1}{T}$, so that they can be sampled with a period T to satisfy the Nyquist rate. We normalize to $T = 1$. Also assume that the bandwidth of the band to be scanned is P times larger: after demodulation to IF we have to sample at rate P. Without multipath, the data model of the modulated sources at the receiver is

$$\mathbf{x}(t) = \sum_{1}^{d} \mathbf{a}(\theta_i)\beta_i e^{j\frac{2\pi}{P}f_i t} s_i(t)$$

where f_i is the residual modulation frequency of the i-th source ($-\frac{P}{2} \leq f_i < \frac{P}{2}$). In matrix form this is written as

$$\mathbf{x}(t) = \mathbf{A}_\theta \mathbf{B} \mathbf{\Phi}^t \mathbf{s}(t) \tag{9.15}$$

where

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1 & & 0 \\ & \ddots & \\ 0 & & \phi_d \end{bmatrix}, \quad \phi_i = e^{j\frac{2\pi}{P}f_i}.$$

Since P can be quite large (order 100, say), it would be very expensive to construct a full data matrix of all samples. In fact, it is sufficient to subsample: collect m subsequent samples at rate P, then wait till the next period before sampling again, resulting in a data matrix X of size $mM \times N$,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(0) & \mathbf{x}(1) & \cdots & \mathbf{x}(N-1) \\ \mathbf{x}(\frac{1}{P}) & \mathbf{x}(1+\frac{1}{P}) & \cdots & \mathbf{x}(N-1+\frac{1}{P}) \\ \vdots & \vdots & & \vdots \\ \mathbf{x}(\frac{m-1}{P}) & \mathbf{x}(1+\frac{m-1}{P}) & \cdots & \mathbf{x}(N-1+\frac{m-1}{P}) \end{bmatrix}.$$

With the model of $\mathbf{x}(t)$ in (9.15), we find that $\mathbf{X}$ has a factorization

$$\mathbf{X} = \begin{bmatrix} \mathbf{A}_\theta \mathbf{B}\mathbf{s}(0) & \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}^P \mathbf{s}(1) & \cdots \\ \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}\mathbf{s}(\frac{1}{P}) & \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}^{P+1}\mathbf{s}(1+\frac{1}{P}) & \cdots \\ \vdots & \vdots & \\ \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}^{m-1}\mathbf{s}(\frac{m-1}{P}) & \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}^{P+m-1}\mathbf{s}(1+\frac{m-1}{P}) & \cdots \end{bmatrix}$$

Let us assume at this point that $P \gg m$. In that case, $s(t)$ is relatively bandlimited with respect to the observed band, which allows to make the crucial assumption that

$$\mathbf{s}(t) \approx \mathbf{s}(t + \tfrac{1}{P}) \approx \cdots \approx \mathbf{s}(t + \tfrac{m-1}{P})$$

so that the model of $\mathbf{X}$ simplifies to

$$\begin{aligned} \mathbf{X} &\approx \begin{bmatrix} \mathbf{A}_\theta \\ \mathbf{A}_\theta \mathbf{\Phi} \\ \vdots \\ \mathbf{A}_\theta \mathbf{\Phi}^{m-1} \end{bmatrix} \mathbf{B} \begin{bmatrix} \mathbf{s}_0 & \mathbf{\Phi}^P \mathbf{s}_1 & \cdots & \mathbf{\Phi}^{(N-1)P} \mathbf{s}_{N-1} \end{bmatrix} \\ &= (\mathbf{F}_\phi \circ \mathbf{A}_\theta)\, \mathbf{B}\, (\mathbf{F}_P \odot \mathbf{S})\,. \end{aligned}$$

$\mathbf{F}_\phi$ is as in (9.11), and only has a different interpretation: ϕ is now related to the carrier frequency. $\mathbf{F}_P$ is similar to $\mathbf{F}_\phi$ except for a transpose and different powers, and the pointwise multiplication represents the modulation on the signals. Obviously, beamforming will not remove this modulation but after estimating $\mathbf{\Phi}$, we can easily correct for it.

If we do consider coherent multipath, the data model becomes

$$\mathbf{X} = (\mathbf{F}_\phi \circ \mathbf{A}_\theta)\mathbf{B}\mathbf{J}(\mathbf{F}_P \odot \mathbf{S})\,. \tag{9.16}$$

The column span of this model has precisely the same structure as $\mathbf{X}$ in (9.14) before, and hence we can use the same algorithm to find the beamformer.

If sources are assumed not to have equal carrier frequencies and $m > d$, we can separate them based on the structure of $\mathbf{F}_\phi$ only. In this case we do not need the array structure and an arbitrary array can be used, but we do not recover the DOAs. If frequencies can be close, however, we will have to separate the signals based on differences in angles as well. It is then also necessary to restore the rank of $\mathbf{X}$ to r by spatial smoothing.

9.5 MULTIPLE INVARIANCES

TBD

Direct extension: Using both short and long baselines to improve resolution

Swindlehurst: MI-ESPRIT [**?**]

Lemma (in context of JAFE)

9.6 NOTES

Section 9.1 discussed the use of antenna triplets to derive the 2D ESPRIT algorithm. Instead of triplets, we can also consider two ULAs oriented in two different directions, e.g., in an L-shape

or a +-shape. Extensions to more general 2-D arrays on which the ESPRIT algorithm works are straightforward to derive, see e.g., [3]. The main issues are the preservation of shift-invariance properties, and the correct pairing of the estimated path parameters using a coupled eigenvalue method.

Joint angle-delay estimation is covered in [4–8].

The IQML-2D method of [9] was originally developed for estimating the two-dimensional modes of sinusoids in Gaussian noise. As it is based on ML, it is expected to show high performance and convergence to the CRB for large number of samples. It can be used to determine angles and delays if both manifolds have Vandermonde structure.

Joint diagonalization problems such as encountered in this chapter have received wide interest in the 1990s. If eigenvalues are distinct, then already a single matrix allows to compute the separating beamformer. To achieve this situation, one line of approaches was to form linear combinations of the two matrices to ensure that the combination has distinct eigenvalues: see e.g., [3]. Several Jacobi-type algorithms have been proposed as well, although some of these assume that $\mathbf{T}$ is a unitary matrix [10–23].

Although these algorithms usually give good performance, the problem of joint diagonalization with non-hermitian matrices has not yet been optimally solved. It is very relevant to study such overdetermined eigenvalue problems. Indeed, a third matrix arises if we use a two-dimensional uniform antenna array, by which we can measure both azimuth and elevation, or any other array with multiple independent baselines. We will see several other examples of joint eigenvalue problems later in this book.

Bibliography

[1] M.D. Zoltowski and C.P. Mathews, "Real-time frequency and 2-D angle estimation with sub-Nyquist spatio-temporal sampling," *IEEE Trans. Signal Proc.*, vol. 42, pp. 2781–2794, October 1994.

[2] K.-B. Yu, "Recursive super-resolution algorithm for low-elevation target angle tracking in multipath," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 141, pp. 223–229, August 1994.

[3] M.D. Zoltowski, M. Haardt, and C.P. Mathews, "Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via Unitary ESPRIT," *IEEE Trans. Signal Proc.*, vol. 44, pp. 316–328, February 1996.

[4] Y. Ogawa, N. Hamaguchi, K. Ohshima, and K. Itoh, "High-resolution analysis of indoor multipath propagation structure," *IEICE Trans. Communications*, vol. E78-B, pp. 1450–1457, November 1995.

[5] J. Gunther and A.L. Swindlehurst, "Algorithms for blind equalization with multiple antennas based on frequency domain subspaces," in *Proc. IEEE ICASSP*, (Atlanta, GA), pp. 2421–2424, 1996.

[6] M. Wax and A. Leshem, "Joint estimation of directions-of-arrival and time-delays of multiple reflections of known signal," *IEEE Trans. Signal Proc.*, vol. 45, pp. 2477–2484, October 1997.

[7] M.C. Vanderveen, C.B. Papadias, and A. Paulraj, "Joint angle and delay estimation (JADE) for multipath signals arriving at an antenna array," *IEEE Communications Letters*, vol. 1, pp. 12–14, January 1997.

[8] A.J. van der Veen, M.C. Vanderveen, and A. Paulraj, "Joint angle and delay estimation using shift-invariance techniques," *IEEE Trans. Signal Proc.*, vol. 46, pp. 405–418, February 1998.

[9] M.P. Clark and L.L. Scharf, "Two-dimensional modal analysis based on maximum likelihood," *IEEE Trans. Signal Processing*, vol. 42, pp. 1443–52, June 1994.

[10] A.J. van der Veen, P.B. Ober, and E.F. Deprettere, "Azimuth and elevation computation in high resolution DOA estimation," *IEEE Trans. Signal Proc.*, vol. 40, pp. 1828–1832, July 1992.

[11] M. Haardt, *Efficient One-, Two-, and Multidimensional High-Resolution Array Signal Processing*. PhD thesis, TU München, Munich, Germany, 1997.

[12] Y. Hua, "Estimating two-dimensional frequencies by matrix enhancement and matrix pencil," *IEEE Trans. Signal Proc.*, vol. 40, pp. 2267–2280, September 1992.

[13] J.F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proc. F (Radar and Signal Processing)*, vol. 140, pp. 362–370, December 1993.

[14] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Proc.*, vol. 45, pp. 434–444, February 1997.

[15] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Independent component analysis based on higher-order statistics only," in *Proc. IEEE SP Workshop on Stat. Signal Array Proc.*, (Corfu, Greece), pp. 356–359, 1996.

[16] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*. PhD thesis, KU Leuven, Leuven, Belgium, 1997.

[17] A.J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, pp. 1136–1155, May 1996.

[18] P. Binding, "Simultaneous diagonalization of several Hermitian matrices," *SIAM J. Matrix Anal. Appl.*, vol. 4, no. 11, pp. 531–536, 1990.

[19] M.T. Chu, "A continuous Jacobi-like approach to the simultaneous reduction of real matrices," *Lin. Alg. Appl.*, vol. 147, pp. 75–96, 1991.

[20] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, "Numerical methods for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 4, pp. 927–949, 1993.

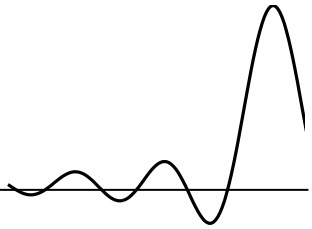
[21] B.D. Flury and B.E. Neuenschwander, "Simultaneous diagonalization algorithms with applications in multivariate statistics," in *Approximation and Computation* (R.V.M. Zahar, ed.), pp. 179–205, Basel: Birkhäuser, 1995.

[22] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 161–164, 1996.

[23] M. Wax and J. Sheinvald, "A least-squares approach to joint diagonalization," *IEEE Signal Proc. Letters*, vol. 4, pp. 52–53, February 1997.

Chapter **10**

FACTOR ANALYSIS



Contents

Many array signal processing algorithms are at some point based on the eigenvalue decomposition, which is used e.g., to make a distinction between the "signal subspace" and the "noise subspace". By using orthogonal projections, part of the noise is projected out and only the signal subspace remains. This can then be used for applications such as high-resolution direction-of-arrival estimation, blind source separation, etc. In these applications, it is commonly assumed that the noise is spatially white. However, this is valid only after suitable calibration.

Factor analysis considers covariance data models where the noise is uncorrelated but has unknown powers at each sensor, i.e., the noise covariance matrix is an arbitrary diagonal with positive real entries. In these cases the familiar eigenvalue decomposition (EVD) has to be replaced by a more general "Factor Analysis" decomposition (FAD), which then reveals all relevant information. It is a very relevant model for the early stages of data processing in radio astronomy, because at that point the instrument is not yet calibrated and the noise powers on the various antennas may be quite different.

As it turns out, this problem has been studied in the psychometrics, biometrics and statistics literature since the 1930s (but usually for real-valued matrices) [1, 2]. The problem has received much less attention in the signal processing literature. In this chapter, we describe the FAD, some applications, and some algorithms for computing it.

10.1 THE FACTOR ANALYSIS PROBLEM

10.1.1 Problem formulation

Assume as before that we have a set of Q narrow-band Gaussian signals impinging on an array of P sensors. The received signal can be described in complex envelope (baseband) form by

$$\mathbf{x}[n] \;=\; \sum_{q=1}^{Q} \mathbf{a}_q s_q[n] + \mathbf{n}[n] \;=\; \mathbf{A}\mathbf{s}[n] + \mathbf{n}[n] \tag{10.1}$$

where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_Q]$ contains the array response vectors. In this model, $\mathbf{A}$ is unknown, and the array response vectors are unstructured, i.e., we do not consider a directional model for them. The source vector $\mathbf{s}[n]$ and noise vector $\mathbf{n}[n]$ are considered zero mean i.i.d. complex Gaussian, i.e., the corresponding covariance matrices are diagonal.

The data model leads to a model for the data covariance matrix as

$$\mathbf{R} = \mathbf{A}\mathbf{\Sigma_s}\mathbf{A}^{\mathrm{H}} + \mathbf{\Sigma_n}\,,$$

where $\mathbf{\Sigma_s}$ is the (diagonal) source covariance matrix, and $\mathbf{\Sigma_n}$ is the (diagonal) noise covariance matrix.

For given $\mathbf{R}$, can we estimate $\mathbf{A}$, $\mathbf{\Sigma_s}$, and $\mathbf{\Sigma_n}$? If $\mathbf{A}$ has no special structure (such as imposed by a parametrically known array response vector), then we cannot distinguish $\mathbf{A}$ and $\mathbf{A}' = \mathbf{A}\mathbf{\Sigma}^{1/2}$: without loss of generality, we can scale the source signals such that the source covariance matrix $\mathbf{\Sigma_s}$ is identity.

Therefore, in this section we will consider a data covariance matrix of the form

$$\mathbf{R} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{D} \tag{10.2}$$

where $\mathbf{D}$ is the (diagonal) noise covariance matrix, and $\mathbf{A}$ has full column rank Q. We assume $Q < P$ so that $\mathbf{A}\mathbf{A}^{\mathrm{H}}$ is rank deficient. Many signal processing algorithms are based on computing an eigenvalue decomposition of $\mathbf{R}$ as $\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}}$, where $\mathbf{U}$ is unitary and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues in descending order.

- If $\mathbf{D} = \mathbf{0}$ (no noise), then $\mathbf{R}$ has rank Q and the eigenvalue decomposition specializes to

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}_0\mathbf{U}^{\mathrm{H}} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \mathbf{\Lambda}_s & \\ & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^{\mathrm{H}} \\ \mathbf{U}_n^{\mathrm{H}} \end{bmatrix}$$

  where $\mathbf{\Lambda}_s$ contains the Q nonzero eigenvalues and $\mathbf{U}_s$ the corresponding eigenvectors. The range of $\mathbf{U}_s$ is called the signal subspace, its orthogonal complement $\mathbf{U}_n$ the noise subspace.

  Since without noise $\mathbf{R} = \mathbf{A}\mathbf{A}^{\mathrm{H}}$, we see that the column span of $\mathbf{U}_s$ equals the column span of $\mathbf{A}$, i.e., $\mathrm{ran}(\mathbf{U}_s) = \mathrm{ran}(\mathbf{A})$.

- For spatially white noise, $\mathbf{D} = \sigma^2\mathbf{I}$, we can write $\mathbf{D} = \sigma^2\mathbf{U}\mathbf{U}^{\mathrm{H}}$, and the eigenvalue decomposition becomes

$$\mathbf{R} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{H}} = \mathbf{U}(\boldsymbol{\Lambda}_0 + \sigma^2\mathbf{I})\mathbf{U}^{\mathrm{H}} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \boldsymbol{\Lambda}_s + \sigma^2\mathbf{I} & \\ & \sigma^2\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^{\mathrm{H}} \\ \mathbf{U}_n^{\mathrm{H}} \end{bmatrix}. \tag{10.3}$$

  Hence, all eigenvalues are raised by σ^2, but the eigenvectors are unchanged. Algorithms based on $\mathbf{U}_s$ can thus proceed as if there was no noise, thus leading to the use of the EVD and related subspace estimation algorithms in many array signal processing applications. See e.g., Chap. 8.

- If the noise is not uniform, then $\mathbf{D}$ is an unknown diagonal matrix, and the EVD of $\mathbf{R}$ does not reveal the signal subspace $\mathbf{U}_s$.

In practice, we are given a finite number of samples $\mathbf{x}[n]$, $n = 0, \cdots, N-1$, and compute the sample covariance matrix

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{n=0}^{N-1} \mathbf{x}[n]\mathbf{x}[n]^{\mathrm{H}}.$$

For large enough N, this estimate is close, but not quite equal, to $\mathbf{R}$.

The objective for factor analysis is, for given $\hat{\mathbf{R}}$, to identify $\mathbf{A}$ and $\mathbf{D}$, as well as the factor dimension Q. This can be seen as an extension of the eigenvalue decomposition, to be used if the noise covariance is not $\sigma^2\mathbf{I}$ but an unknown diagonal.

It is clear that for an arbitrary Hermitian matrix $\mathbf{R}$, the factorization $\mathbf{R} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{D}$ can exist in its exact form only for $Q \geq P$, in which case we can set $\mathbf{D} = \mathbf{0}$, or any other value, which makes the factorization useless. Hence, for a noise-perturbed matrix, we wish to detect the smallest Q which gives a "reasonable fit", and we will assume that $Q < P$ is sufficiently small so that unique decompositions exist. What we consider reasonable depends on N, as de accuracy of $\hat{\mathbf{R}}$ (or: its covariance) scales with $1/N$.

10.1.2 Identifiability and uniqueness

It is immediately clear from (10.2) that the factors are not uniquely identifiable. E.g., $\mathbf{A}$ is not unique: The columns of $\mathbf{A}$ can be permuted and if $\mathbf{A}$ satisfies the model, then also $\mathbf{A}' = \mathbf{A}\mathbf{Q}$ is valid, for any unitary matrix $\mathbf{Q}$. However, the column span of $\mathbf{A}$ is invariant under these transformations, and thus these do not harm subspace estimation techniques.

To be accurate, if we denote $\mathbf{A}\mathbf{A}^{\mathrm{H}} = \mathbf{U}_s\boldsymbol{\Lambda}_s\mathbf{U}_s^{\mathrm{H}}$, we see we can can estimate a bit more than just the column span of $\mathbf{A}$ (given by $\mathrm{ran}(\mathbf{U}_s)$), because also the "signal eigenvalues" $\boldsymbol{\Lambda}_s$ tells us something more about $\mathbf{A}$. This might help to estimate the source covariance matrix $\boldsymbol{\Sigma}_\mathbf{s}$ which for the moment we assumed to be the identity.

More important is the uniqueness of $\mathbf{D}$. By counting numbers of observations (or equations) and numbers of unknowns, we see that the number of columns Q of $\mathbf{A}$ cannot be too large, in fact we need $Q < P - \sqrt{P}$ as can be determined as follows.

The number of available observations is equal to the number of (real) parameters in $\hat{\mathbf{R}}$, which is P (real) entries on the main diagonal and $P(P-1)$ (real) parameters for the off-diagonal (complex) entries, taking into account Hermitian symmetry. In total these are P^2 observations. The number of unknowns is $2PQ$ (real) parameters for $\mathbf{A}$, and P parameters for $\mathbf{D}$, minus the number of constraints to make $\mathbf{A}$ unique. A constraint which is often used is to make the columns of $\mathbf{A}' := \mathbf{D}^{-1/2}\mathbf{A}$ orthogonal, or equivalently, $\mathbf{A}^{\mathrm{H}}\mathbf{D}^{-1}\mathbf{A}$ is diagonal (this is motivated in Sec. 10.1.3 below). This gives $Q^2 - Q$ constraints on the parameters of $\mathbf{A}$. Further restricting the first row of $\mathbf{A}$ to be real gives another Q constraints. In total we have for the number of equations minus the number of unknowns

$$s = P + P(P-1) \; - \; (2PQ + P - (Q^2 - Q + Q)) = (P-Q)^2 - P\,. \tag{10.4}$$

This number is also called the *degree of freedom*, and plays a role in the asymptotic modeling of the likelihood. Requiring $s > 0$ leads to the condition $Q < P - \sqrt{P}$. This is an upper bound on the factor rank.

Even if we satisfy this constraint, $\mathbf{D}$ is not always unique, as seen from the following example. Consider $\mathbf{R} = \mathbf{A}_1\mathbf{A}_1^{\mathrm{H}} + \mathbf{D}_1$, where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}.$$

Then we also have $\mathbf{R} = \mathbf{A}_2\mathbf{A}_2^{\mathrm{H}} + \mathbf{D}_2$, where

$$\mathbf{A}_2 = \sqrt{2}\begin{bmatrix} 1/2 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad \mathbf{D}_2 = \mathbf{D}_1 + \frac{1}{2}\mathbf{e}_1\mathbf{e}_1^{\mathrm{T}}$$

and $\mathbf{e}_i$ is the ith column of the identity matrix. The problem in this case is caused by a submatrix of $\mathbf{A}_1$ being rank-deficient. This can be considered an uncommon technicality that can be detected after the factors have been estimated. Throughout the rest of the chapter, we assume that $\mathbf{D}$ can be identified uniquely.

10.1.3 Constraints on A

If $\mathbf{D}$ is identifiable, then $\mathbf{A}$ is unique up to a rotation $\mathbf{Q}$. We can make $\mathbf{A}$ unique by adding additional constraints. This essentially amounts to choosing a non-redundant parametrization. Not all algorithms require this, but it may be needed to avoid singularities during the computation of the Cramer-Rao Bound (CRB) or when we use Newton gradient descent techniques. For complex data, Q^2 constraint equations are needed. Common constraints are to force the columns of $\mathbf{A}$ to be orthogonal with respect to a certain weight matrix $\mathbf{W} > 0$, i.e. to require that $\mathbf{A}^{\mathrm{H}}\mathbf{W}\mathbf{A}$ is diagonal.

In more detail, suppose we have estimated $\mathbf{D}$, then we can whiten the noise covariance matrix in $\mathbf{R}$:

$$\tilde{\mathbf{R}} := \mathbf{D}^{-1/2}\mathbf{R}\mathbf{D}^{-1/2} = (\mathbf{D}^{-1/2}\mathbf{A})(\mathbf{A}^{\mathrm{H}}\mathbf{D}^{-1/2}) + \mathbf{I}\,.$$

At this point, we can introduce the usual eigenvalue decomposition of $\tilde{\mathbf{R}}$:

$$\tilde{\mathbf{R}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^{\mathrm{H}} = \tilde{\mathbf{U}}(\tilde{\mathbf{\Lambda}}_s + \mathbf{I})\tilde{\mathbf{U}}^{\mathrm{H}}\,,$$

and identify $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{V} = \tilde{\mathbf{U}}$, or $\mathbf{A} = \mathbf{D}^{1/2}\tilde{\mathbf{U}}\mathbf{V}^{\mathrm{H}}$, where $\mathbf{V}$ is an arbitrary unitary factor. If we choose $\mathbf{V} = \mathbf{I}$, we obtain $\mathbf{A}^{\mathrm{H}}\mathbf{D}^{-1}\mathbf{A} = \tilde{\mathbf{\Lambda}}_s$ is diagonal. We can use this as a constraint to obtain a more unique parametrization of $\mathbf{A}$. Note that $\mathbf{A}$ is not yet quite unique, because in the complex case each column of $\mathbf{A}$ can be scaled by an arbitrary complex phase, and the columns may be reordered as well.

If we compute a matrix $\mathbf{A}$ without satisfying constraints, the required transformation $\mathbf{Q}$ such that $\mathbf{A}' = \mathbf{A}\mathbf{Q}$ satisfies the constraints is easily determined afterwards. Hence, in most algorithms the constraints do not play a role. In the literature, constraints such as setting $\mathbf{A}^{\mathrm{H}}\mathbf{D}^{-1}\mathbf{A}$ diagonal have been introduced in an attempt to interprete the resulting "latent factors", but without prior structural information on $\mathbf{A}$, these attempts are often futile.

10.2 COMPUTING THE FACTOR ANALYSIS DECOMPOSITION

Factor analysis is a classical problem. It was introduced in 1904 [3] and over time, several algorithms were proposed [4–6], all for real data matrices (although readily extended to the complex case). In this section we briefly review some of these approaches.

Consider again the model

$$\mathbf{R} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{D}, \tag{10.5}$$

where $\mathbf{A}$ has Q columns, and $\mathbf{D}$ is a diagonal matrix with positive diagonal elements. As data, we are given a sample covariance matrix $\hat{\mathbf{R}}$ based on N samples,

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{n=0}^{N-1}\mathbf{x}[n]\mathbf{x}[n]^{\mathrm{H}}\,.$$

In Factor Analysis, there are two problems:

1. *Detection:* given $\hat{\mathbf{R}}$, estimate Q. The hypothesis that the factor rank is q is denoted by $\mathcal{H}_q$. We can formulate this as a likelihood ratio test.
2. *Identification:* given $\hat{\mathbf{R}}$ and Q, estimate $\mathbf{D}$ and $\mathbf{A}$, or $\mathbf{\Lambda}_s$ and $\mathbf{U}_s$ in the formulation $\mathbf{R} = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^{\mathrm{H}} + \mathbf{D}$.

We consider the latter problem first. Detection is studied in Sec. 10.3.

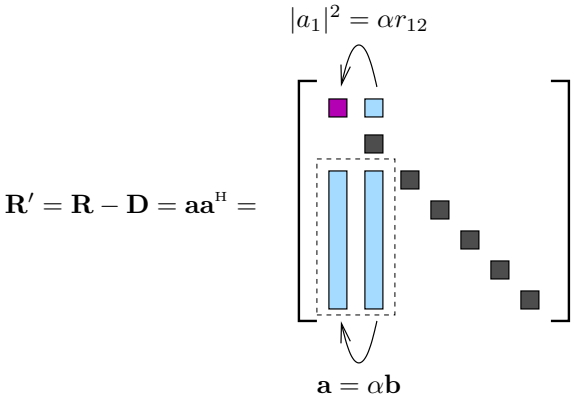


Figure 10.1. Ad hoc method: Away from the main diagonal, all submatrices have at most rank Q. This can be used to estimate the main diagonal. The figure shows how this is done for $Q = 1$.

If we separate detection from identification, then for the latter, the objective is to estimate the factors $\mathbf{A}$ and $\mathbf{D}$ from $\hat{\mathbf{R}}$, where the number of columns Q of $\mathbf{A}$ is known. We first present an ad-hoc algorithm, which gives some insight in the problem. Then, we look at Maximum Likelihood (ML)-type algorithms, and in particular consider a Weighted Least Squares (WLS) formulation that is minimized using fast-converging Gauss-Newton iterations.

10.2.1 Ad Hoc Method

If the rank Q is relatively small, it is possible to solve the FA problem in closed form. As an example, let $Q = 1$, assume we know $\mathbf{R}$ exactly, and consider $\mathbf{R}' = \mathbf{R} - \mathbf{D} = \mathbf{a}\mathbf{a}^{\mathrm{H}}$. Clearly, $\mathbf{R}'$ is rank 1, and this implies that each column in the matrix is a multipe of another column. Moreover, if $\mathbf{D}$ is unknown, then only the diagonal entries of $\mathbf{R}'$ are unknown. Each submatrix of $\mathbf{R}'$ that does not involve the main diagonal is completely known and will have rank 1. This can be used to fill in the diagonal entries of $\mathbf{R}'$ such that the entire matrix is of rank 1. Indeed, as shown in Fig. 10.1, the first column is α times the second column, and we can find α from this ratio. Then $r'_{11} = \alpha r_{12}$ is found immediately. Likewise, we can find the entire main diagonal.

This ad hoc estimation algorithm can be extended to higher ranks, but perhaps not to the maximal rank $P - \sqrt{P}$. Also, if we only have an estimate $\hat{\mathbf{R}}$, then the algorithm will not be optimal. However, it could be used to provide a good initial point for an iterative algorithm.

10.2.2 Alternating Least Squares

The estimation problem can also be approached as a two-stage minimization problem [2]. In this approach we minimize the LS cost function

$$\min_{\mathbf{A},\mathbf{D}} \|\hat{\mathbf{R}} - \mathbf{A}\mathbf{A}^{\mathrm{H}} - \mathbf{D}\|_{\mathrm{F}}^2 \tag{10.6}$$

by an alternating least-squares (ALS) approach, where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. First, for a given $\mathbf{A}$, (10.6) is minimized with respect to $\mathbf{D}$ and in the next stage, $\mathbf{D}$ is held constant and a new $\mathbf{A}$ is found. Both problems can be optimized in closed form.

Let the subscript (k) denote the iteration count. The iteration steps are

$$\mathbf{D}_{(k+1)} := \mathrm{diag}(\hat{\mathbf{R}} - \mathbf{A}_{(k)}\mathbf{A}_{(k)}^{\mathrm{H}}) \tag{10.7}$$

$$\mathbf{U}_{(k+1)}\mathbf{\Lambda}_{(k+1)}\mathbf{U}_{(k+1)}^{\mathrm{H}} := \hat{\mathbf{R}} - \mathbf{D}_{(k+1)} \quad [\mathrm{EVD}] \tag{10.8}$$

$$\mathbf{A}_{(k+1)} := \mathbf{U}_{s,(k+1)}\mathbf{\Lambda}_{s,(k+1)}^{1/2}, \tag{10.9}$$

where $\mathbf{U}_{(k+1)}$ and $\mathbf{\Lambda}_{(k+1)}$ follow from an eigenvalue decomposition, and $\mathbf{U}_{s,(k+1)}$ and $\mathbf{\Lambda}_{s,(k+1)}$ contain the Q dominant eigenvectors and corresponding eigenvalues. A Weighted Least Squares formulation could be considered instead of (10.6), leading to similar iterations, but involving the EVD of $\mathbf{D}^{-1/2}\hat{\mathbf{R}}\mathbf{D}^{-1/2}$, if we take $\mathbf{D}^{-1}$ as a weight.

The iteration is usually initialized by taking

$$\mathbf{D}_{(0)} = [\mathrm{diag}(\hat{\mathbf{R}}^{-1})]^{-1}.$$

As for most ALS approaches, the rate of convergence is slow (linear). An EVD is required at each iteration, which makes it prohibitive for large problems.

10.2.3 Maximum Likelihood Estimator

We now aim for more optimal techniques. The standard approach to tackle estimation problems is to consider the Maximum Likelihood estimator. We assume we know Q. The first step is to choose a suitable parametrization.

Parametrization Let us write the model as $\mathbf{R}(\boldsymbol{\theta}) = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{D}$, where the vector $\boldsymbol{\theta}$ represents the unknown parameters in the model. Since $\mathbf{A}$ is complex, a direct representation of its entries gives complex parameters. We could represent them as independent real and purely imaginary components, but a popular alternative is to represent them using Wirtinger operators [7, App.2], [8]: for an unknown complex parameter θ_i we consider its conjugate θ_i^* as an independent parameter while real parameters are represented only once. Using this method we define the parameter vector as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_{\mathbf{A}} \\ \boldsymbol{\theta}_{\mathbf{A}^*} \\ \boldsymbol{\theta}_{\mathbf{D}} \end{bmatrix} \tag{10.10}$$

where

$$\begin{aligned} \boldsymbol{\theta}_{\mathbf{A}} &= \mathrm{vec}(\mathbf{A}) \\ \boldsymbol{\theta}_{\mathbf{A}^*} &= \mathrm{vec}(\mathbf{A}^*) \\ \boldsymbol{\theta}_{\mathbf{D}} &= \mathrm{diag}(\mathbf{D}) = \mathbf{d}. \end{aligned}$$

This parametrization is redundant: it does not implement the Q^2 constraints we need to place on $\mathbf{A}$ to make it unique. However, it is more convenient to do this at a later stage.

Using this parameterization and properties of Kronecker products (5.7) and (5.8), we have

$$\begin{aligned} \mathbf{r} = \mathrm{vec}(\mathbf{R}) &= (\mathbf{A}^* \otimes \mathbf{I}_P)\mathrm{vec}(\mathbf{A}) + (\mathbf{I}_P \circ \mathbf{I}_P)\mathbf{d} \\ &= (\mathbf{A}^* \otimes \mathbf{I}_P)\boldsymbol{\theta}_{\mathbf{A}} + (\mathbf{I}_P \circ \mathbf{I}_P)\boldsymbol{\theta}_{\mathbf{D}}\,. \end{aligned} \tag{10.11}$$

To show how $\mathbf{r}$ depends on $\boldsymbol{\theta}_{\mathbf{A}^*}$, let $\mathbf{K}$ be the exchange matrix defined by $\mathrm{vec}(\mathbf{A}^{\mathrm{T}}) = \mathbf{K}\mathrm{vec}(\mathbf{A})$ (cf. (5.15)). Then we can also write $\mathbf{r}$ as

$$\begin{aligned} \mathbf{r} &= (\mathbf{I}_P \otimes \mathbf{A})\mathrm{vec}(\mathbf{A}^{\mathrm{T}}) + (\mathbf{I}_P \circ \mathbf{I}_P)\mathbf{d} \\ &= (\mathbf{I}_P \otimes \mathbf{A})\mathbf{K}\boldsymbol{\theta}_{\mathbf{A}^*} + (\mathbf{I}_P \circ \mathbf{I}_P)\boldsymbol{\theta}_{\mathbf{D}}\,. \end{aligned} \tag{10.12}$$

In the Wirtinger calculus, the derivative of a function to a complex parameter $z = x + jy$ is defined as [8]

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

Moreover, z and z^* are treated as independent variables in the differentiation.

Based on the parametrization of $\mathbf{R}(\boldsymbol{\theta})$, we can then derive its Jacobian $\mathbf{J}(\boldsymbol{\theta})$ as

$$\begin{aligned} \mathbf{J} &= \frac{\partial \mathrm{vec}(\mathbf{R})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} = \left[\frac{\partial \mathrm{vec}(\mathbf{R})}{\partial \boldsymbol{\theta}_{\mathbf{A}}^{\mathrm{T}}}, \frac{\partial \mathrm{vec}(\mathbf{R})}{\partial \boldsymbol{\theta}_{\mathbf{A}^*}^{\mathrm{T}}}, \frac{\partial \mathrm{vec}(\mathbf{R})}{\partial \boldsymbol{\theta}_{\mathbf{D}}^{\mathrm{T}}}\right] \\ &= [\mathbf{J}_{\mathbf{A}}, \mathbf{J}_{\mathbf{A}^*}, \mathbf{J}_{\mathbf{D}}]\,, \end{aligned} \tag{10.13}$$

where

$$\mathbf{J}_{\mathbf{A}} = \mathbf{A}^* \otimes \mathbf{I}_P, \quad \mathbf{J}_{\mathbf{A}^*} = (\mathbf{I}_P \otimes \mathbf{A})\mathbf{K}, \quad \mathbf{J}_{\mathbf{D}} = \mathbf{I}_P \circ \mathbf{I}_P\,. \tag{10.14}$$

ML cost and Fisher score If we assume that the samples $\mathbf{x}$ are generated by zero mean complex proper Gaussian sources, i.e.,

$$p(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{\pi^P \det(\mathbf{R})}\exp\left[-\mathbf{x}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{x}\right]\,,$$

then the complex log-likelihood function for N independent samples is given by

$$l(\boldsymbol{\theta}) = -N\left[P\log(\pi) + \log\det(\mathbf{R}(\boldsymbol{\theta})) + \mathrm{tr}(\mathbf{R}(\boldsymbol{\theta})^{-1}\hat{\mathbf{R}})\right]. \tag{10.15}$$

The maximum likelihood approach aims to find a $\boldsymbol{\theta}$ that maximizes this function. To this end, we find the gradient of the likelihood function (called the Fisher score) and set it equal to zero. For complex parameters, the Fisher score for a proper Gaussian distributed signal is defined as

$$\mathbf{g}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{g}_{\mathbf{A}} \\ \mathbf{g}_{\mathbf{A}^*} \\ \mathbf{g}_{\mathbf{D}} \end{bmatrix} = \left[\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{X};\boldsymbol{\theta})\right]^{\mathrm{H}} = \begin{bmatrix} \vdots \\ \frac{\partial}{\partial \theta_j^*} \log p(\mathbf{X};\boldsymbol{\theta}) \\ \vdots \end{bmatrix}.$$

Inserting (10.15), the jth entry of $\mathbf{g}(\boldsymbol{\theta})$ can be evaluated as

$$[\mathbf{g}(\boldsymbol{\theta})]_j = -N\frac{\partial}{\partial \theta_j^*} \log\det(\mathbf{R}) - N\frac{\partial}{\partial \theta_j^*} \mathrm{tr}(\mathbf{R}^{-1}\hat{\mathbf{R}}).$$

We need some results for matrix differentials [8, p.53]:

$$\begin{aligned} \partial \det(\mathbf{R}) &= \det(\mathbf{R})\mathrm{tr}(\mathbf{R}^{-1}\partial\mathbf{R}) \\ \partial\mathbf{R}^{-1} &= -\mathbf{R}^{-1}\,\partial\mathbf{R}\,\mathbf{R}^{-1}. \end{aligned}$$

This gives

$$[\mathbf{g}(\boldsymbol{\theta})]_j = -N\mathrm{tr}[\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\theta_j^*}] + N\mathrm{tr}[\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\theta_j^*}\mathbf{R}^{-1}\hat{\mathbf{R}}].$$

Next, we use some properties of Kronecker products (see Sec. 5.1.6):

$$\begin{aligned} \mathrm{tr}(\mathbf{AB}) &= \mathrm{vec}^{\mathrm{H}}(\mathbf{A}^{\mathrm{H}})\mathrm{vec}(\mathbf{B}) \\ \\ \mathrm{tr}(\mathbf{ABCD}) &= \mathrm{vec}^{\mathrm{H}}(\mathbf{A}^{\mathrm{H}})(\mathbf{D}^{\mathrm{T}}\otimes\mathbf{B})\mathrm{vec}(\mathbf{C}). \end{aligned}$$

This results in

$$\begin{aligned} [\mathbf{g}(\boldsymbol{\theta})]_j &= -N\mathrm{vec}^{\mathrm{H}}(\frac{\partial\mathbf{R}}{\partial\theta_j^*})\,\mathrm{vec}(\mathbf{R}^{-1}) + N\mathrm{vec}^{\mathrm{H}}(\frac{\partial\mathbf{R}}{\partial\theta_j^*})(\mathbf{R}^{-T}\otimes\mathbf{R}^{-1})\mathrm{vec}(\hat{\mathbf{R}}) \\ &= N(\frac{\partial\mathrm{vec}(\mathbf{R})}{\partial\theta_j^*})^{\mathrm{H}}(\mathbf{R}^{-T}\otimes\mathbf{R}^{-1})\mathrm{vec}(\hat{\mathbf{R}}-\mathbf{R}). \end{aligned}$$

Finally, stacking for all j and using (10.13), we find a compact expression for $\mathbf{g}(\boldsymbol{\theta})$ as

$$\mathbf{g}(\boldsymbol{\theta}) = N\mathbf{J}^{\mathrm{H}}\left(\mathbf{R}^{-T}\otimes\mathbf{R}^{-1}\right)\mathrm{vec}(\hat{\mathbf{R}}-\mathbf{R}). \tag{10.16}$$

This is a general expression. Let us now look at our specific parametrization for $\mathbf{R}$: inserting (10.13) into (10.16), the elements of the Fisher score $\mathbf{g}(\boldsymbol{\theta})$ become

$$\begin{aligned} \mathbf{g}_{\mathbf{A}} &= N(\mathbf{A}^{\mathrm{T}}\mathbf{R}^{-T}\otimes\mathbf{R}^{-1})\mathrm{vec}(\hat{\mathbf{R}}-\mathbf{R}) \\ &= N\mathrm{vec}\left[\mathbf{R}^{-1}(\hat{\mathbf{R}}-\mathbf{R})\mathbf{R}^{-1}\mathbf{A}\right] && (10.17) \\ \mathbf{g}_{\mathbf{A}^*} &= \mathbf{g}_{\mathbf{A}}^* && (10.18) \\ \mathbf{g}_{\mathbf{D}} &= N\mathrm{vecdiag}\left[\mathbf{R}^{-1}(\hat{\mathbf{R}}-\mathbf{R})\mathbf{R}^{-1}\right]. && (10.19) \end{aligned}$$

The ML technique requires us to set (10.17) and (10.19) equal to zero, but unfortunately this does not produce a closed-form solution. One approach to numerically compute the ML estimate is to consider Newton-Raphson-like algorithms, as these provide quadratic convergence. Besides the gradient, we will also need an expression for the Hessian.

The Scoring Method The scoring algorithm is a variant of the Newton-Raphson algorithm where the gradient is the Fisher score (10.16) and the Hessian is replaced by the Fisher information matrix [9]. The Fisher information matrix (FIM) is defined as

$$\mathbf{F} = -\mathrm{E}\left[\frac{\partial \mathbf{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}}\right]$$

where the expectation is over the data (i.e., $\hat{\mathbf{R}}$). Inserting (10.16), and realizing that after the expectation only the derivative of $\mathrm{vec}(\hat{\mathbf{R}} - \mathbf{R})$ results in a nonzero contribution, gives

$$\mathbf{F} = N\,\mathbf{J}^{\mathrm{H}}(\mathbf{R}^{-T} \otimes \mathbf{R}^{-1})\mathbf{J},$$

where $\mathbf{J}$ is given by (10.13). The resulting iterations in the scoring algorithm are

$$\boldsymbol{\theta}_{(k+1)} = \boldsymbol{\theta}_{(k)} + \mu_{(k)}\boldsymbol{\delta}, \tag{10.20}$$

where $\boldsymbol{\theta}_{(k)}$ is the current estimate of the parameters, $\mu_{(k)}$ is a step size, and

$$\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{A}} \\ \boldsymbol{\delta}_{\mathbf{A}^*} \\ \boldsymbol{\delta}_{\mathbf{D}} \end{bmatrix}$$

is the direction of descent. The latter follows from solving

$$\mathbf{F}_{(k)}\boldsymbol{\delta} = \mathbf{g}_{(k)}, \tag{10.21}$$

where $\mathbf{g}_{(k)} = \mathbf{g}(\boldsymbol{\theta}_{(k)})$ is the Fisher score and $\mathbf{F}_{(k)} = \mathbf{F}(\boldsymbol{\theta}_{(k)})$ is the FIM. Since without constraints the parametrization is redundant (see Sec. 10.1.2), the FIM is singular. However, this does not need to cause complications because (10.16) shows that $\mathbf{g}_{(k)}$ is in the column span of $\mathbf{F}_{(k)}$, so that the system of equations has a solution, and (taking the minimum-norm solution) standard convergence results for the scoring method follow.

A problem with the scoring method is that the matrix $\mathbf{F}$ quickly becomes large, as its dimension is equal to the number of unknown parameters. Solving (10.21) then becomes unattractive. Similarly, we also do not want to directly work with $\mathbf{R}^{-T} \otimes \mathbf{R}^{-1}$ as it is a matrix of size $P^2 \times P^2$. Another problem is that $\mathbf{R}$ changes each iteration cycle and its inverse has to be recomputed each time.

Covariance matching techniques We can view Factor Analysis as a special case of covariance matching, as studied in Chap. 7. In this approach, the ML problem is replaced by a Weighted Least Squares (WLS) fitting of the sample covariance. The large sample properties of the estimators are the same. Solving this nonlinear least squares problem using gradient descent techniques is closely connected to the scoring algorithm.

The corresponding Nonlinear Weighted Least Squares (NLWLS) problem is

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; \|\mathbf{W}^{1/2}[\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})]\|^2 = \arg\min_{\boldsymbol{\theta}} \; [\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})]^{\mathrm{H}}\mathbf{W}[\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})] \tag{10.22}$$

where $\mathbf{r} = \mathrm{vec}(\mathbf{R})$, $\hat{\mathbf{r}} = \mathrm{vec}(\hat{\mathbf{R}})$. The optimal weighting matrix $\mathbf{W}$ is the inverse of the covariance matrix of $\hat{\mathbf{r}}$. We derived in Sec. 3.2.2 that this covariance is equal to $\mathbf{C} = (1/N)(\mathbf{R}^* \otimes \mathbf{R})$. Because we only have access to the sample covariance matrices $\hat{\mathbf{R}}$, we use instead

$$\mathbf{W} = \hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1}\,, \tag{10.23}$$

and then $\hat{\boldsymbol{\theta}}$ asymptotically (for large N) converges to the optimal ML solution for a Gaussian distributed data matrix.

This is precisely in context of [10], and we can use one of the algorithms proposed there: Gauss-Newton iterations, the scoring algorithm, or sequential estimation algorithms. Here, we derive the Gauss-Newton iterations.

Gauss-Newton algorithm for solving NLWLS For the Gauss-Newton iteration, the Hessian is replaced by the Gramian of the Jacobians [11]. The updates are similar to the scoring method updates (10.20):

$$\boldsymbol{\theta}_{(k+1)} = \boldsymbol{\theta}_{(k)} + \mu_{(k)}\boldsymbol{\delta}, \tag{10.24}$$

where $\boldsymbol{\delta}$ is the direction of descent. To find $\boldsymbol{\delta}$ we need to solve

$$\mathbf{B}(\boldsymbol{\theta}_{(k)})\boldsymbol{\delta} = \mathbf{g}(\boldsymbol{\theta}_{(k)}), \tag{10.25}$$

where

$$\mathbf{g}(\boldsymbol{\theta}) = \mathbf{J}^{\mathrm{H}}(\boldsymbol{\theta})\mathbf{W}[\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})] \tag{10.26}$$

$$\mathbf{B}(\boldsymbol{\theta}) = \mathbf{J}^{\mathrm{H}}(\boldsymbol{\theta})\mathbf{W}\mathbf{J}(\boldsymbol{\theta})\,. \tag{10.27}$$

The weight $\mathbf{W}$ is given by (10.23) and the Jacobian $\mathbf{J}(\boldsymbol{\theta})$ by (10.13).

The iterations given by (10.24) are repeated until $\|\mathbf{g}(\boldsymbol{\theta}_{(k)})\|_2 < \epsilon$, where $\epsilon > 0$ depends on the desired accuracy. Clearly, the equations are very similar to the scoring method (10.20, except that the sample covariance matrices in $\mathbf{W}$ are constant and have to be inverted only once.

The key step in the Gauss-Newton iteration is solving the linear system (10.25). The matrix dimensions can become large. We propose an algorithm based on a symbolic inversion of $\mathbf{B}$.

Closed-form solution for direction of descent A complicated derivation [12] that we omit here shows how we can solve for $\boldsymbol{\delta}_\mathbf{D}$ inside $\boldsymbol{\delta}$ in closed form. Define

$$\begin{aligned}
\tilde{\mathbf{W}} &= = \hat{\mathbf{R}}^{-1} - \hat{\mathbf{R}}^{-1}\mathbf{A}(\mathbf{A}^{\mathrm{H}}\hat{\mathbf{R}}^{-1}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{H}}\hat{\mathbf{R}}^{-1} \\
\tilde{\mathbf{B}}_\mathbf{D} &= \mathbf{J}_\mathbf{D}^{\mathrm{H}}\left(\tilde{\mathbf{W}}^{\mathrm{T}} \otimes \tilde{\mathbf{W}}\right)\mathbf{J}_\mathbf{D} \quad = \tilde{\mathbf{W}}^{\mathrm{T}} \odot \tilde{\mathbf{W}} \qquad \text{[using } \mathbf{J}_\mathbf{D} = \mathbf{I} \circ \mathbf{I} \text{ and (5.5)]} \\
\tilde{\mathbf{g}}_\mathbf{D} &= \mathbf{J}_\mathbf{D}^{\mathrm{H}}\left(\tilde{\mathbf{W}}^{\mathrm{T}} \otimes \tilde{\mathbf{W}}\right)\mathrm{vec}[\hat{\mathbf{R}} - \mathbf{R}(\boldsymbol{\theta})]\,.
\end{aligned}$$

Note that $\tilde{\mathbf{W}}\mathbf{A} = \mathbf{0}$. Then the computation of

$$\boldsymbol{\delta} = \begin{bmatrix} \mathrm{vec}(\boldsymbol{\Delta}_\mathbf{A}) \\ \mathrm{vec}(\boldsymbol{\Delta}_{\mathbf{A}^*}) \\ \boldsymbol{\delta}_\mathbf{D} \end{bmatrix}$$

in (10.25) reduces to the computation of $\boldsymbol{\delta}_\mathbf{D}$ from $\tilde{\mathbf{B}}_\mathbf{D}\boldsymbol{\delta}_\mathbf{D} = \tilde{\mathbf{g}}_\mathbf{D}$, while

$$\boldsymbol{\Delta}_\mathbf{A} = \frac{1}{2}\left(\mathbf{I} + \hat{\mathbf{R}}\tilde{\mathbf{W}}\right)\left(\hat{\mathbf{R}} - \mathbf{R}(\boldsymbol{\theta}) - \mathrm{diag}(\boldsymbol{\delta}_\mathbf{D})\right)\hat{\mathbf{R}}^{-1}\mathbf{A}(\mathbf{A}^{\mathrm{H}}\hat{\mathbf{R}}^{-1}\mathbf{A})^{-1} \tag{10.28}$$

and $\boldsymbol{\Delta}_{\mathbf{A}^*} = \boldsymbol{\Delta}_\mathbf{A}^*$. Each of these computations requires us to handle matrices not larger than size $P \times P$.

Alternating Weighted Least Squares (AWLS) algorithm If we take step size $\mu = 1$, then the closed-form result simplifies to

$$\boldsymbol{\theta}_\mathbf{D}^{(k+1)} = \boldsymbol{\theta}_\mathbf{D}^{(k)} + \boldsymbol{\delta}_\mathbf{D}\,.$$

Premultiplying with $\tilde{\mathbf{B}}_\mathbf{D}$ gives

$$\begin{aligned}
\tilde{\mathbf{B}}_\mathbf{D}\boldsymbol{\theta}_\mathbf{D}^{(k+1)} &= \tilde{\mathbf{B}}_\mathbf{D}\boldsymbol{\theta}_\mathbf{D}^{(k)} + \tilde{\mathbf{g}}_\mathbf{D} \\
&= \tilde{\mathbf{B}}_\mathbf{D}\boldsymbol{\theta}_\mathbf{D}^{(k)} + \mathbf{J}_\mathbf{D}^{\mathrm{H}}\left(\tilde{\mathbf{W}}^{\mathrm{T}} \otimes \tilde{\mathbf{W}}\right)\mathrm{vec}(\hat{\mathbf{R}} - \mathbf{A}\mathbf{A}^{\mathrm{H}} - \mathbf{D}) \\
&= \mathbf{J}_\mathbf{D}^{\mathrm{H}}\left(\tilde{\mathbf{W}}^{\mathrm{T}} \otimes \tilde{\mathbf{W}}\right)\mathrm{vec}(\hat{\mathbf{R}} - \mathbf{A}\mathbf{A}^{\mathrm{H}})\,.
\end{aligned}$$

Here we used $\mathrm{vec}(\mathbf{D}) = \mathbf{J}_\mathbf{D}\boldsymbol{\theta}_\mathbf{D}^{(k)}$ and the definition of $\tilde{\mathbf{B}}_\mathbf{D}$, and in the notation dropped the dependency on k from $\tilde{\mathbf{B}}_\mathbf{D}$, $\tilde{\mathbf{W}}$, $\mathbf{A}$ and $\mathbf{D}$.

Since $\tilde{\mathbf{W}}\mathbf{A} = \mathbf{0}$ and $\mathbf{J}_\mathbf{D} = \mathbf{I} \circ \mathbf{I}$, this reduces to

$$\begin{aligned}
\tilde{\mathbf{B}}_\mathbf{D}\boldsymbol{\theta}_\mathbf{D}^{(k+1)} &= \mathbf{J}_\mathbf{D}^{\mathrm{H}}\left(\tilde{\mathbf{W}}^{\mathrm{T}} \otimes \tilde{\mathbf{W}}\right)\mathrm{vec}(\hat{\mathbf{R}}) \\
\Leftrightarrow \qquad \left[\tilde{\mathbf{W}}^{\mathrm{T}} \odot \tilde{\mathbf{W}}\right]\boldsymbol{\theta}_\mathbf{D}^{(k+1)} &= \mathrm{vecdiag}(\tilde{\mathbf{W}}\hat{\mathbf{R}}\tilde{\mathbf{W}}) \\
&= \mathrm{vecdiag}(\tilde{\mathbf{W}})\,.
\end{aligned}$$

$\tilde{\mathbf{W}}$ acting on $\hat{\mathbf{R}}$ can be interpreted as "projecting out" the contribution of the term $\mathbf{A}\mathbf{A}^{\mathrm{H}}$ in $\hat{\mathbf{R}}$ after which the remaining term $\mathbf{D}$ can be estimated. The final simplification used that $\tilde{\mathbf{W}}\hat{\mathbf{R}}\tilde{\mathbf{W}} = \tilde{\mathbf{W}}$.

The result can be formulated as the Alternating Weighted Least Squares (AWLS) algorithm [12]. First, for given $\mathbf{D}_{(k)}$, compute

$$\begin{aligned}\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}} &:= \mathbf{D}_{(k)}^{-1/2}\hat{\mathbf{R}}\mathbf{D}_{(k)}^{-1/2} \\ \mathbf{A}_{(k+1)} &:= \mathbf{D}_{(k)}^{1/2}\mathbf{U}_s(\mathbf{\Lambda}_s - \mathbf{I})^{1/2}\end{aligned}$$

where the first line represents an eigenvalue decomposition, and in the second line, $\mathbf{\Lambda}_s$ contains the largest Q eigenvalues of $\mathbf{\Lambda}$, and $\mathbf{U}_s$ the corresponding eigenvectors. This step is similar to the (prewhitened) alternating LS algorithm in Sec. 10.2.2. Alternatively, (10.28) could have been used.

Next, let $\mathbf{W} = \hat{\mathbf{R}}^{-1}$, and update the estimate of $\mathbf{D}$:

$$\begin{aligned}&\tilde{\mathbf{W}} := \mathbf{W} - \mathbf{W}\mathbf{A}_{(k+1)}(\mathbf{A}_{(k+1)}^{\mathrm{H}}\mathbf{W}\mathbf{A}_{(k+1)})^{-1}\mathbf{A}_{(k+1)}^{\mathrm{H}}\mathbf{W} \\ &\mathbf{d}_{(k+1)} := \left[\tilde{\mathbf{W}}^{\mathrm{T}} \odot \tilde{\mathbf{W}}\right]^{-1} \mathrm{vecdiag}(\tilde{\mathbf{W}}) \\ &\mathbf{D}_{(k+1)} := \mathrm{diag}(\mathbf{d}_{(k+1)})\,.\end{aligned}$$

These two steps are alternated until convergence. In this algorithm, all computations are on matrices of size $P \times P$, which makes the computational complexity of the same order as that of an EVD: $O(P^3)$.

10.2.4 Convergence

The following simulation experiment gives an indication on the convergence speed. We use $P = 100$ sensors, $N = 1000$ samples. The matrix $\mathbf{A}$ is chosen randomly with a standard complex Gaussian distribution (i.e. each element is distributed as $\mathcal{CN}(0, 1)$) and $\mathbf{D}$ is chosen randomly with a uniform distribution between 1 and 5.

Convergence is gauged by looking at the norm of the gradient.

AWLS is tested against a range of other algorithms which are described in [12]. In the graph, the "Ad Hoc" method is the ALS, "Joreskog" is an implementation of WLS using Fletcher-Powell iterations [13] as used in many standard toolboxes, while "CM" is the Constrained Maximization algorithm [14], which was derived from the EM algorithm, is straightforward to implement and shows quadratic convergence. "KLD/EM" is another representative of an EM algorithm with a straightforward implementation [15].

As seen in Fig. 10.2, the AWLS algorithm converges fastest (in 10-15 iterations), while the ALS and EM algorithms have slow convergence (over 1000 iterations for large Q).

10.3 RANK DETECTION

The detection problem is to estimate the factor rank Q (i.e., the number of columns of $\mathbf{A}$). In array processing, this relates to detecting the number of sources that the array is exposed to. An

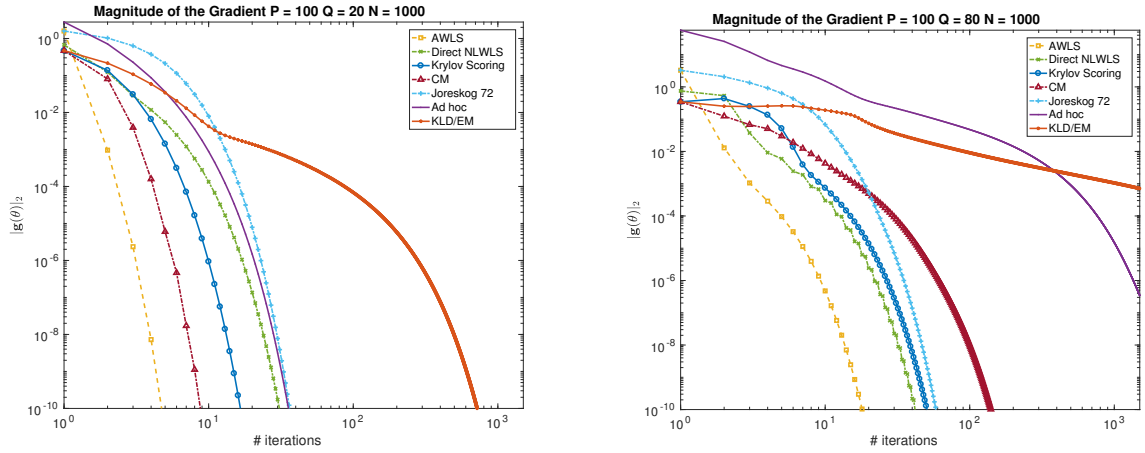


Figure 10.2. (*a*) $Q = 20$ sources; (*b*) $Q = 80$ sources.

extensive literature exists on this topic; here we limit the discussion to a general likelihood ratio test (GLRT) [16], which is used to decide whether the FA model fits a given sample covariance matrix. We can use the GLRT to design a constant false alarm ratio detector. In the special case where $Q = 0$, this test indicates whether there are any sources active during the measurement (we detect whether $\mathbf{R}$ is diagonal). The largest permissible value of Q is that for which the number of equations minus the number of unknown (real) parameters $s = (P-Q)^2 - P > 0$, or $Q_{\max} < P - \sqrt{P}$. For larger Q, there is no identifiability of $\mathbf{A}$ and $\mathbf{D}$: any sample covariance matrix $\hat{\mathbf{R}}$ can be fitted.

Let $\mathbf{R}_q$ denote the covariance matrix of the FA model with q sources,

$$\mathbf{R}_q = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{D}\,, \quad \text{where} \quad \mathbf{A}:\ P \times q\,, \quad \mathbf{D}\ \text{diagonal}\,,$$

and let $\mathcal{CN}(\mathbf{0}, \mathbf{R}_q)$ denote the zero-mean complex normal distribution with covariance $\mathbf{R}_q$. To find Q using the GLRT, we define a collection of hypotheses

$$\mathcal{H}_q: \quad \mathbf{x}(k) \ \sim\ \mathcal{CN}(\mathbf{0}, \mathbf{R}_q) \qquad q = 0, 1, 2, \cdots \tag{10.29}$$

which are tested in turn against the null hypothesis

$$\mathcal{H}': \quad \mathbf{x}(k) \ \sim\ \mathcal{CN}(\mathbf{0}, \mathbf{R}')\,.$$

$\mathcal{H}'$ corresponds to a default hypothesis of an arbitrary (unstructured) positive definite matrix $\mathbf{R}'$.

In the GLRT, we have to insert maximum likelihood estimates for each of the unknown parameters, under each of the hypotheses. For $\mathcal{H}_q$, we can use the estimation techniques from the previous section, resulting in an estimated model $\mathbf{R}_q$. For $\mathcal{H}'$, the ML estimate $\mathbf{R}'$ is equal to the sample covariance, $\mathbf{R}' = \hat{\mathbf{R}}$.

Under $\mathcal{H}_q$, respectively $\mathcal{H}'$, the maximum values of the log-likelihood are (dropping constants)

$$\begin{aligned} \log(L_q) &= -N \log\det(\mathbf{R}_q^{-1}) - N\mathrm{tr}(\mathbf{R}_q^{-1}\hat{\mathbf{R}}) \\ \log(L') &= -N \log\det(\hat{\mathbf{R}}^{-1}) - NP\,. \end{aligned}$$

The log-likelihood ratio is then

$$\log(\lambda) := \log\left(\frac{L'}{L_q}\right) = N\mathrm{tr}(\mathbf{R}_q^{-1}\hat{\mathbf{R}}) + N\log\det(\mathbf{R}_q^{-1}\hat{\mathbf{R}}) - NP\,. \tag{10.30}$$

Here, $\lambda = L'/L_q$ is the test statistic (likelihood ratio), and we will reject $\mathcal{H}_q$ and accept $\mathcal{H}'$ if $\lambda > \gamma$, where γ is a predetermined threshold. Typically, γ is determined such that we obtain an acceptable "false-alarm" rate (i.e., the probability that we accept $\mathcal{H}'$ instead of $\mathcal{H}_q$, while $\mathcal{H}_q$ is actually true). To establish γ, we need to know the statistics of λ under $\mathcal{H}_q$.

Generalizing the results from the real-valued case [1, 2], we obtain that for moderately large N (say $N > 50$), the test statistic $2\log(\lambda)$ has approximately a χ^2_s distribution, where s is equal to "the number of free parameters" under $\mathcal{H}_q$ (the number of equations minus the number of unknowns). For the complex case, we saw that this number is $s = (P-q)^2 - P$ degrees of freedom.

In view of results of Box and Bartlett, a better fit of the distribution of $2\log(\lambda)$ to a χ^2_s distribution is obtained by replacing N in (10.30) by [1, 2]

$$N' = N - \frac{1}{6}(2P+11) - \frac{2}{3}Q\,.$$

To detect Q, we start with $q = 0$, and apply the test for increasing values of q until it is accepted, or until $q > Q_{\max}$. In that case, the hypothesis $\mathcal{H}'$ is accepted, i.e., the given $\hat{\mathbf{R}}$ is an unstructured covariance matrix. A disadvantage of this process is that the model parameters for each q have to be estimated, which can become quite cumbersome if P is large.[1]

However, note that if the GLRT passes for a given estimate Q_0 it also passes for any $Q > Q_0$, and if it fails it also fails for any $Q < Q_0$. Therefore, instead of a linear search for Q we can use a binary search. The maximum number of possible sources for FA is given by $Q_{\max} < P - \sqrt{P}$. In a binary search, we split the entire interval into two segments, and test on the boundary to decide in which interval the solution must lie. Proceeding recursively in this way, the number of needed FA estimates is on average $\log_2(Q_{\max}) + 1$, which is reasonable even for large P.

10.4 EXTENSIONS OF THE CLASSICAL MODEL

We present two extensions of the classical model: joint and extended factor analysis.

[1] Also, as for any sequential hypothesis test, the actual false alarm rate that is achieved is unknown, because the tests are not independent.

10.4.1 Joint Factor Analysis Model

In some applications, the signal subspace (i.e. $\mathbf{A}$) is not stationary, while the noise covariance is stationary. Consider e.g., DOA estimation of moving sources and an uncalibrated array. An available dataset is then partitioned into M short subsets or "snapshots", each containing N samples. This leads to M sample covariance matrices $\hat{\mathbf{R}}_m$, $m = 1, \ldots, M$, with model

$$\mathbf{R}_m = \mathbf{A}_m \mathbf{A}_m^{\mathrm{H}} + \mathbf{D}, \quad m = 1, \ldots, M\,. \tag{10.31}$$

$\mathbf{A}_m$ is a low-rank matrix of size $P \times Q_m$ with $Q_m < P$ for all $m = 1, \ldots, M$, and $\mathbf{D}$ is a positive real diagonal matrix common among the M models. We call this model Joint Factor Analysis (JFA). The objective is to estimate $\mathbf{D}$ and $\{\mathbf{A}_m\}$ jointly, based on the available sample covariance matrices $\{\hat{\mathbf{R}}_m\}$. In many applications we are just interested in the column span of $\mathbf{A}_m$.

An example where this model could occur is in wideband processing in frequency domain, where m represents a frequency index and each $\mathbf{R}_m$ corresponds to a narrowband model. If the noise powers are frequency-independent, then $\mathbf{D}$ common among the various covariance matrices. A joint estimate will be more accurate than an algorithm where we first estimate the FAD for each m, and then average the $\mathbf{D}_m$.

10.4.2 Extended and Joint Extended FA Model

Another extension is to consider the noise covariance matrix to be more general than a diagonal matrix, say $\mathbf{R}_\mathbf{n} = \boldsymbol{\Psi}$, where $\boldsymbol{\Psi}$ has a certain structure, assumed to be known. Here we consider $\boldsymbol{\Psi}$ of the form

$$\boldsymbol{\Psi} = \mathbf{M} \odot \boldsymbol{\Psi},$$

where $\mathbf{M}$ is a symmetric matrix containing only ones and zeros and $\odot$ denotes the Hadamard or entrywise product. We call $\mathbf{M}$ a mask matrix; the main diagonal is assumed to be nonzero.

We can model various types of covariance matrices using this approach (for example: block-diagonal matrices, band matrices, sparse matrices, etc.). A further generalization of this is to model $\boldsymbol{\Psi}$ as a linear sum of known matrices, i.e.,

$$\mathrm{vec}(\boldsymbol{\Psi}) = \mathbf{G}\boldsymbol{\theta}_{\boldsymbol{\psi}},$$

where $\mathbf{G}$ is a fixed basis. For example, $\mathbf{G}$ could contain selected columns of a Fourier matrix to model spatially lowpass noise [10].

We assume $\mathbf{M}$ to be known based on the application. The Extended FA (EFA) model then becomes

$$\mathbf{R} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \boldsymbol{\Psi}\,. \tag{10.32}$$

Both generalizations can be combined into Joint Extended FA (JEFA), where we have

$$\mathbf{R}_m = \mathbf{A}_m \mathbf{A}_m^{\mathrm{H}} + \boldsymbol{\Psi}\,, \quad m = 1, \ldots, M\,. \tag{10.33}$$

Algorithms to find these decompositions are straightforward generalizations of the previously presented algorithms [12].

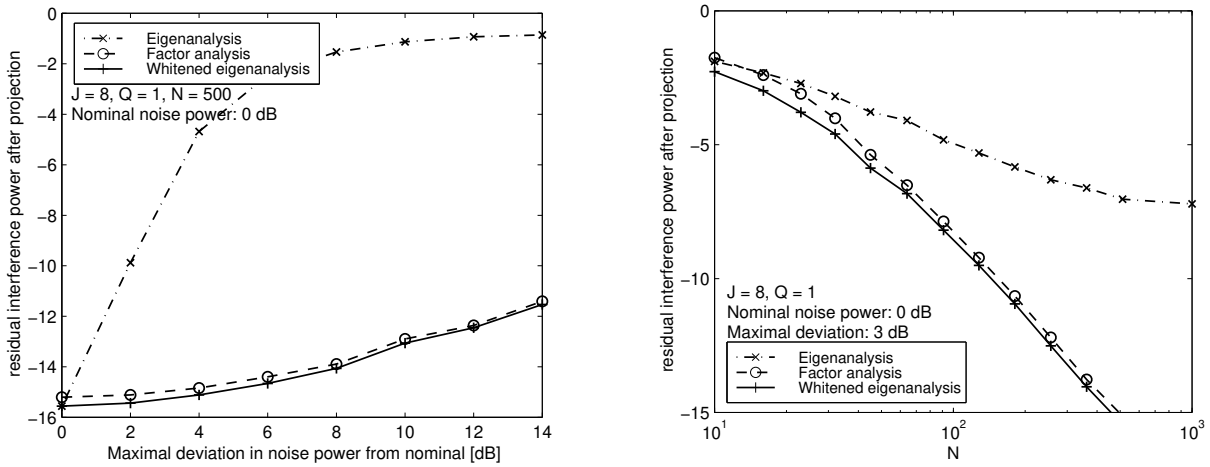


Figure 10.3. Residual interference power after projections.

10.5 APPLICATION TO INTERFERENCE CANCELLATION

10.5.1 Interference projection

In the context of radio astronomy, factor analysis shows up in interference cancellation. In general, this is a large topic with many aspects. Here, we consider a simple case where we take short integration intervals and an uncalibrated array. Since astronomical sources are weak and much below the background noise level, if we integrate only over short intervals, the noise is dominant. Therefore, in the absence of interference, the data covariance matrix $\mathbf{R}$ from a single short term integration interval could be modeled as a diagonal $\mathbf{D}$. Assuming Q independent interfering signals gives us a contribution $\mathbf{A}\mathbf{A}^{\mathrm{H}}$. The approach for interference cancellation using spatial filtering is to estimate ran($\mathbf{A}$), and to apply to $\mathbf{R}$ a projector $\mathbf{P}_{\mathbf{A}}^{\perp}$ onto the orthogonal complement of the span, i.e., $\mathbf{R}' = \mathbf{P}_{\mathbf{A}}^{\perp}\mathbf{R}\mathbf{P}_{\mathbf{A}}^{\perp}$. That should remove the interference. The filtered covariance matrices are further averaged over multiple integration intervals, and corrections need to be applied since also the astronomical data has been filtered. Details on this approach can be found in [17, 18].

Simulation Here, we describe only a limited-scope simulation on synthetic data, where we estimate a rank-1 subspace (*i*) using factor analysis, and for comparison (*ii*) using an eigendecomposition assuming that $\mathbf{D} = \sigma^2\mathbf{I}$, or (*iii*) using an eigendecomposition after whitening by $\mathbf{D}^{-1/2}$, assuming the true $\mathbf{D}$ is known from calibration. The correct rank is $Q = 1$, and we show the residual interference power after projection, i.e., $\|\mathbf{P}_{\hat{\mathbf{a}}}^{\perp}\mathbf{a}\|$ as a function of number of samples N, mean noise power, and deviation in noise power. The noise powers in $\mathbf{D}$ are randomly generated at the beginning of the simulation, uniformly in an interval. Legends in the graphs indicate the nominal noise power and the maximal deviation. All simulations use $P = 8$ sensors, and a nominal interference to noise ratio per channel of 0 dB.

The results are shown in Fig. 10.3. The left graph shows the residual interference power for varying maximal deviations, the right graph shows the residual for varying number of samples N, and a maximal deviation of 3 dB of the noise powers. The figures indicate that already for small deviations of the noise powers it is essential to take this into account, by using the FAD instead of the EVD. Furthermore, the estimates from the factor analysis are nearly as good as can be obtained via whitening with known noise powers.

10.5.2 Reference antenna array

In the previous paragraph, we projected out the interference dimension, and this effectively reduces the number of antennas (dishes) by the number of detected interferers. An alternative is to use a *reference antenna array*, with antennas that receive a good copy of the interfering signals, but have little gain towards the desired sky sources. So suppose we have a primary array with p_0 antennas, and a reference array with p_1 antennas. The received signal model is

$$\begin{array}{rcll} \mathbf{x}_0(t) & = & \mathbf{v}_0(t) + & \mathbf{A}_0(t)\mathbf{s}(t) + \mathbf{n}_0(t) \\ \mathbf{x}_1(t) & = & & \mathbf{A}_1(t)\mathbf{s}(t) + \mathbf{n}_1(t) \end{array}$$

where the subscripts 0 and 1 refer to the primary and reference array, respectively, $\mathbf{v}_0(t)$ contains the desired sky source signals, $\mathbf{s}(t)$ the q interfering signals, and $\mathbf{n}_i(t)$ the noise on each array. Collecting all antenna signals into a single vector $\mathbf{x}(t)$, we can write

$$\mathbf{x}(t) = \mathbf{v}(t) + \mathbf{A}(t)\mathbf{s}(t) + \mathbf{n}(t)$$

where $\mathbf{A} : p \times q$ has q columns corresponding to the q interferers. The covariance matrix of $\mathbf{x}(t)$ can be partitioned as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{00} & \mathbf{R}_{01} \\ \mathbf{R}_{10} & \mathbf{R}_{11} \end{bmatrix}.$$

According to the assumptions, $\mathbf{R}$ has model

$$\begin{aligned} \mathbf{R} &= \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{\Psi} \\ &= \left[\begin{array}{c|c} \mathbf{R}_{v,0} + \mathbf{A}_0\mathbf{A}_0^{\mathrm{H}} + \mathbf{\Sigma}_0 & \mathbf{A}_0\mathbf{A}_1^{\mathrm{H}} \\ \hline \mathbf{A}_1\mathbf{A}_0^{\mathrm{H}} & \mathbf{A}_1\mathbf{A}_1^{\mathrm{H}} + \mathbf{\Sigma}_1 \end{array}\right] \end{aligned} \tag{10.34}$$

where $\mathbf{\Psi} := \mathbf{R}_v + \mathbf{\Sigma}$ is the interference-free covariance matrix, $\mathbf{R}_v := \mathrm{bdiag}[\mathbf{R}_{v,0}, \mathbf{0}]$ contains the astronomical visibilities, and $\mathbf{\Sigma} := \mathrm{bdiag}[\mathbf{\Sigma}_0, \mathbf{\Sigma}_1]$ is the diagonal noise covariance matrix. The objective is to estimate the interference-free covariance submatrix $\mathbf{\Psi}_{00} := \mathbf{R}_{v,0} + \mathbf{\Sigma}_0$.

The data model (10.34) satisfies the Extended Factor Analysis (EFA) model. The covariance model (10.34) is

$$\mathbf{R} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \mathbf{\Psi} = \mathbf{A}\mathbf{A}^{\mathrm{H}} + \left[\begin{array}{c|c} \mathbf{\Psi}_{00} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{\Sigma}_1 \end{array}\right]. \tag{10.35}$$

where we are interested in estimating the unknown square matrix $\mathbf{\Psi}_{00}$ and, for an uncalibrated array, $\mathbf{\Sigma}_1$ is unknown. Thus, the appropriate masking matrix $\mathbf{M}$ such that $\mathbf{\Psi} = \mathbf{M} \odot \mathbf{\Psi}$ is

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{1}\mathbf{1}^{\mathrm{T}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right].$$

This is an EFA model and we can apply the corresponding algorithms for estimating $\mathbf{A}$ and $\mathbf{\Psi}$. Each $\hat{\mathbf{R}}$ will give us an estimate $\hat{\mathbf{\Psi}}$, and $\hat{\mathbf{\Psi}}_{00}$ is simply the upper left sub-block of this matrix.

A necessary condition for identification is that the degree of freedom $s > 0$. Compared to FA, we see that the p parameters of $\mathbf{\Sigma}$ are now replaced by the $p_0^2 + p_1$ (real) parameters in $\mathbf{\Psi}$. Thus, we require

$$s = p^2 + q^2 - 2pq - (p_0^2 + p_1)$$

to be larger than 0. With $p = p_0 + p_1$ and solving for the number of reference antennas p_1 we find

$$p_1 > q - (p_0 - \tfrac{1}{2}) + \sqrt{q + (p_0 - \tfrac{1}{2})^2}\,. \tag{10.36}$$

Thus, if p_0 is small, we need $p_1 > q + \sqrt{q}$, and if p_0 is large, we need $p_1 > q$.

If $\hat{\mathbf{R}}$ is based on a short-term interval (a snapshot estimate of the covariance) and we have multiple snapshots, then we can apply JEFA to estimate the varying interfering subspaces while exploiting that the sky covariance is constant and common among all snapshots.

We show two examples with experimental data, taken from [19].

Example 10.1. To test the algorithm on actual data, we have made a short observation of the strong astronomical source 3C48 contaminated by Afristar satellite signals. The primary array consists of $p_0 = 3$ of the 14 telescope dishes of the Westerbork Synthesis Radio Telescope (WSRT), located in The Netherlands. As reference signals we use $p_1 = 27$ of 52 elements of a focal–plane array that is mounted on another dish of WSRT which is set off-target (see Fig. 10.4) such that it has no dish gain towards the astronomical source nor to the interferer.

We recorded 13.4 seconds of data with 80 MS/s, and processed these offline. Using short-term windowed Fourier transforms, the data was first split into 8192 frequency bins (from which we used 1537), and subsequently correlated and averaged over $M = 4048$ samples to obtain $N = 64$ short-term covariance matrices.

Fig. 10.5(a) shows the autocorrelations and crosscorrelations on the primary antennas and Fig. 10.5(b) shows the autocorrelation of 6 reference antennas. The interference is clearly seen in the spectrum. The interference consists of a lower and higher frequency part. The low frequency part is stronger on the reference antenna and the higher part is stronger on the primary antenna. However, because of a relatively large number of reference antennas the total INR, as we will see, is high enough for the algorithms to be effective.

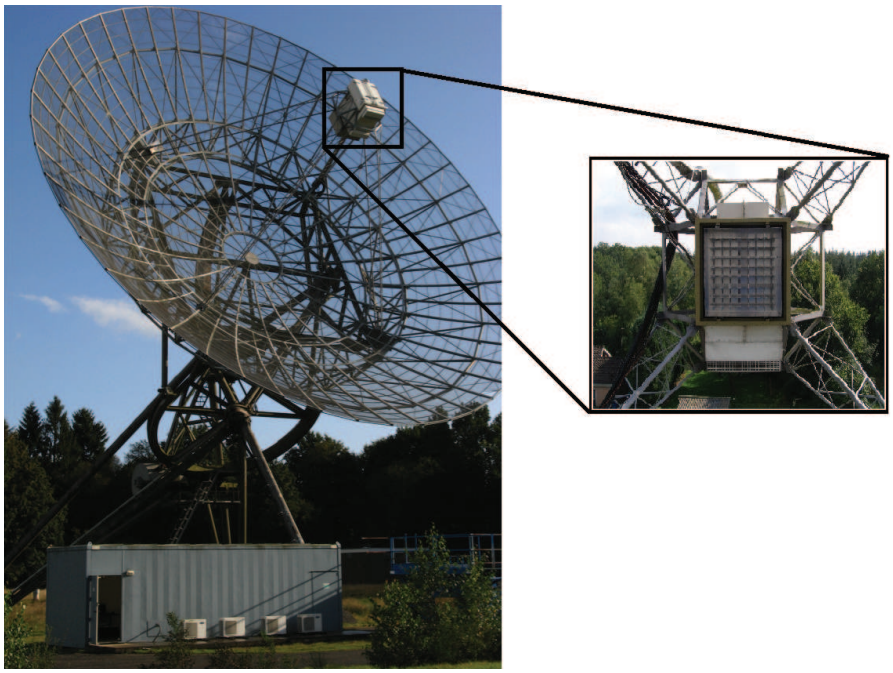


Figure 10.4. Reference focal-plane array mounted on a dish.

Because no calibration step has been performed we use a generalized likelihood ratio test (GLRT) [20] to detect if each frequency bin is contaminated with RFI and then we use EFA to estimate the noise powers and the signal spatial signature. The result of whitening the spectrum with the estimated result of EFA is shown in Fig. 10.6(a).

The resulting auto- and crosscorrelation spectra after filtering are shown in Fig. 10.6(b). The autocorrelation spectra are almost flat, and close to 1 (the whitened noise power). The cross-correlation spectra show that the spatial filtering with the reference antenna has removed the RFI within the sensitivity of the telescope. Also it shows the power of using EFA at this stage in the processing chain, as it is not required for the array to be calibrated.

Example 10.2. In a second experiment, we use raw data from LOFAR station RS409 (100-200 MHz). Data from 46 (out of 48) x-polarization receiving elements are sampled with a frequency of 200 MHz and correlated. Samples are then divided into 1024 subbands with the help of tapering and an FFT. From these samples we form $N = 4$ covariance matrices with an integration time of 19 ms ($M = 1862$) for each subband. No calibration was done on the resulting covariance matrices.

The LOFAR HBA has a hierarchy of antennas, where a single receiving element output is the result of analog beamforming on 16 antennas (4×4) in a tile. During

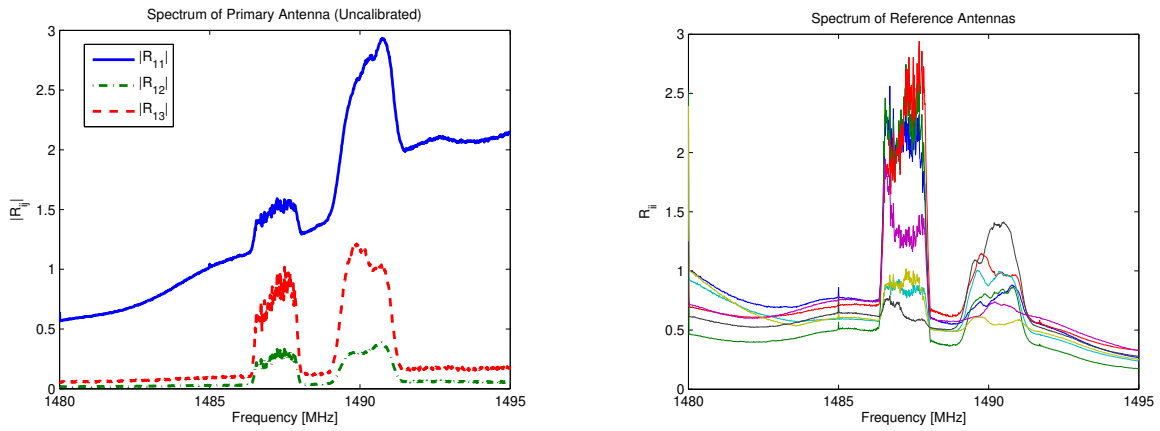


Figure 10.5. Observed spectrum from (a) the primary telescopes, (b) 6 of the reference antennas.

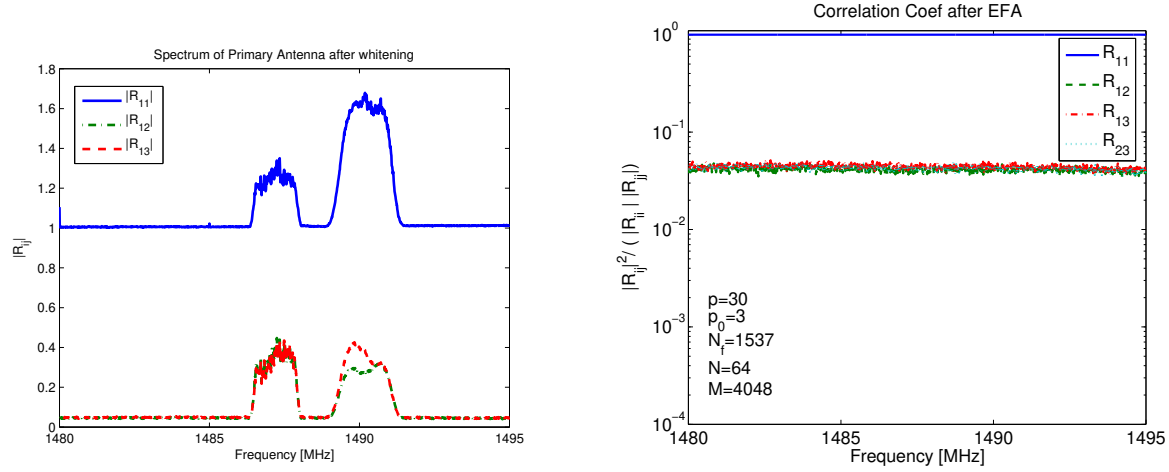


Figure 10.6. After EFA, the covariance matrices can be whitened: (a) Spectrum of primary antenna after whitening, (b) averaged normalized correlation coefficients after filtering.

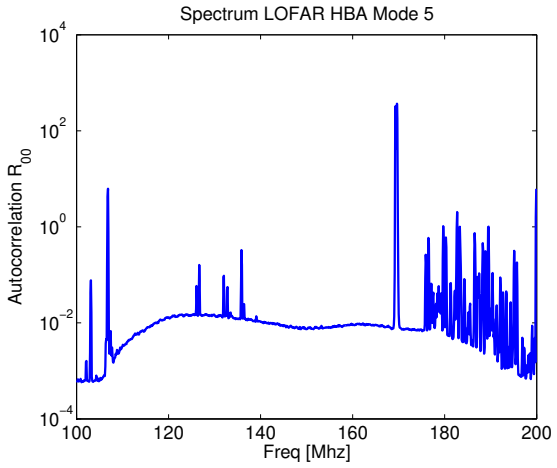


Figure 10.7. Spectrum received at a LOFAR HBA station

the measurements the analog beamformers were tracking the strong astronomical source Cygnus A.

The received spectrum is shown in Fig. 10.7. Above 174 MHz, the spectrum is heavily contaminated by wideband Digital Audio Broadcast (DAB) transmissions.

We have used 6 of the 46 receiving elements as reference array for our filtering techniques and the rest as primary array. Because we do not have dedicated reference antennas and because the data is already beamformed the assumption that the source is too weak at each short integration time (19 ms) is not completely valid. Also the assumption that the sky sources are much weaker on the reference antennas is not valid in this case because the reference array elements are also following Cygnus A. Finally, we have the same exposure to the RFI on the secondary array as we have on the primary so there is no additional RFI gain for the secondary array.

To illustrate the performance of the filtering technique we produce snapshot images of the sky (i.e., images based on a single covariance matrix). For an uncontaminated image, we have chosen subband 250 at 175.59 MHz, see Fig. 10.8(a), while for RFI-contaminated data we take subband 247 at 175.88 MHz, see Fig. 10.8(b). These two subbands have been chosen because they are close to each other (in frequency) and we expect that the astronomical images for these bands would be similar. Subband 247 is heavily contaminated and has a 10 dB flux increase on the auto-correlations and a 20 dB increase on the cross-correlations.

The repeated source visible in Fig. 10.8(a) is Cygnus A; the repetition is due to the spatial aliasing which occurs at these frequencies (the tiles are separated by more than half a wavelength). The contaminated image in Fig. 10.8(b) shows no trace of Cygnus A; note the different amplitude scale which has been increased by a factor

100.

Fig. 10.9 shows the image after using EFA. The image is very similar to the clean image in Fig. 10.8(a)).

10.6 APPLICATION TO ARRAY CALIBRATION

Before we can do any beamforming, we need to calibrate the array. Indeed, in the previous chapters, we assumed we fully knew the array response function, and in many cases, we even assumed omnidirectional antennas (i.e., the individual antennas have the same unit response in all directions). Before we are in this situation, we need to estimate these responses. Generally, this involves a single test source that we scan across the array, but this is not always practical once the array is out of the factory and deployed in the field. A particular example is radio astronomy, where the "antennas" are large dishes or beamformed stations, and the calibrator sources are strong celestial objects. Obviously we have no control over them and cannot switch them off, but on the other hand their positions and source powers are accurately known from tables. In this section, it is shown how factor analysis can be used to solve the problem of calibration.

The calibration problem does not only involve the antenna response functions, it also involves the receiver noise present on each antenna. In previous chapters, we usually assumed the noise was spatially white: independent and of equal power on each antenna. However, before calibration the receiver noise generally has different powers on each antenna. These also need to be estimated.

10.6.1 Non-ideal measurements

So far we ignored the beam shape of the individual elements (antennas or dishes) of the array. In fact, any antenna has its own directional response $b(\boldsymbol{\zeta})$, where $\boldsymbol{\zeta}$ denotes a unit-length source direction vector (see (2.6)). This function is called the primary beam. For simplicity, it is generally assumed that the primary beam is equal for all elements in the array, although this is also subject to calibration. With Q point sources, we will collect the resulting samples of the primary beam into a vector $\mathbf{b} = [b(\boldsymbol{\zeta}_1), \cdots, b(\boldsymbol{\zeta}_Q)]^{\mathrm{T}}$. These coefficients are seen as gains that (squared) will multiply the source powers σ_q^2. The general shape of the primary beam $b(\boldsymbol{\zeta})$ is known from electromagnetic modeling during the design of the antenna. If this is not sufficiently accurate, then it has to be calibrated.

We also have direction-independent differences in gains and phases among the antennas, e.g., due to differences in the receiver chains of each element in the array. Initially these are also unknown and have to be estimated. We thus have an unknown vector $\mathbf{g}$ (size $P \times 1$) with complex entries that each multiply the output signal of each antenna.

Also the noise powers of each element are unknown and generally unequal to each other. We will still assume that the noise is independent from element to element. We can thus model the

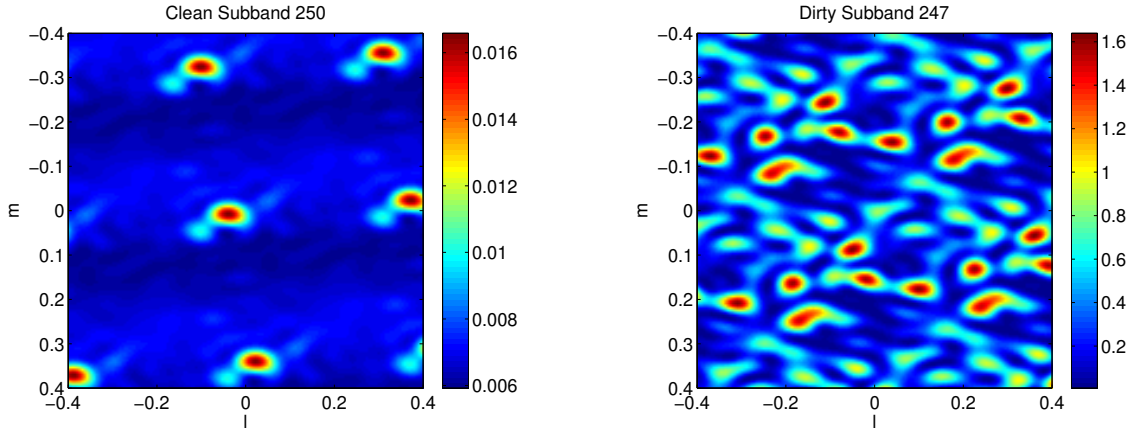


Figure 10.8. (a) Clean subband 250, (b) Contaminated subband 247

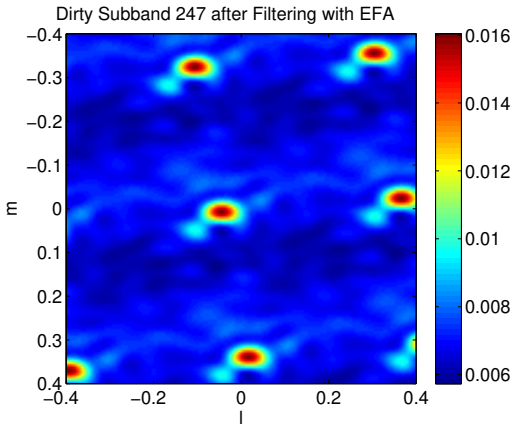


Figure 10.9. Result of filtering using EFA

noise covariance matrix by an (unknown) diagonal $\mathbf{\Sigma_n}$.

For a calibrated array, we used until now the covariance data model

$$\mathbf{R} = \mathbf{A}(\boldsymbol{\theta})\, \mathbf{\Sigma_s}\, \mathbf{A}(\boldsymbol{\theta})^{\mathrm{H}} \;+\; \mathbf{\Sigma_n}\,. \tag{10.37}$$

Here, the array response matrix $\mathbf{A}(\boldsymbol{\theta})$ is a known function of the source direction vectors $\{\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_Q\}$, suitably parametrized by the vector $\boldsymbol{\theta}$ (with typically two direction cosines per source).

The modified data model that captures the unknown gain/phase/noise effects and replaces (10.37) is then

$$\mathbf{R} = [\mathbf{\Gamma A}(\boldsymbol{\theta})\mathbf{B}]\, \mathbf{\Sigma_s}\, [\mathbf{B}^{\mathrm{H}}\mathbf{A}(\boldsymbol{\theta})^{\mathrm{H}}\mathbf{\Gamma}^{\mathrm{H}}] \;+\; \mathbf{\Sigma_n} \tag{10.38}$$

where $\mathbf{\Gamma} = \mathrm{diag}(\mathbf{g})$ is a diagonal with unknown receiver complex gains, and $\mathbf{B} = \mathrm{diag}(\mathbf{b})$ contains the samples of the primary beam (the directional response of each antenna). Usually, $\mathbf{\Gamma}$ and $\mathbf{B}$ are considered to vary only slowly with time and frequency, so that we can combine multiple covariance matrices $\mathbf{R}_{m,k}$ with the same $\mathbf{\Gamma}$ and $\mathbf{B}$.

In some cases, the source directions are disturbed as well, e.g. due to atmospheric effects (or due to ionospheric delays in radio astronomy). In first order, we can replace $\mathbf{A}(\boldsymbol{\theta})$ by $\mathbf{A}(\boldsymbol{\theta}')$, where $\boldsymbol{\theta}'$ differs from $\boldsymbol{\theta}$ due to the shift in apparent direction of each source. The modified data model that captures the above effects is thus

$$\mathbf{R} = [\mathbf{\Gamma A}(\boldsymbol{\theta}')\mathbf{B}]\, \mathbf{\Sigma_s}\, [\mathbf{B}^{\mathrm{H}}\mathbf{A}(\boldsymbol{\theta}')^{\mathrm{H}}\mathbf{\Gamma}^{\mathrm{H}}] \;+\; \mathbf{\Sigma_n}\,. \tag{10.39}$$

If we wish to be very general, we can write this as

$$\mathbf{R} = [\mathbf{G} \odot \mathbf{A}(\boldsymbol{\theta})]\, \mathbf{\Sigma_s}\, [\mathbf{G} \odot \mathbf{A}(\boldsymbol{\theta})]^{\mathrm{H}} + \mathbf{\Sigma_n} \tag{10.40}$$

where $\odot$ indicates an entrywise multiplication of two matrices (Schur-Hadamard product). Here, $\mathbf{G}$ is a full matrix that captures all non-linear measurement effects. Equation (10.38) is recovered if we write $\mathbf{G} = \mathbf{g}\mathbf{b}^{\mathrm{H}}$ (i.e., a rank-1 matrix), and equation (10.39) if we write $\mathbf{G} = \mathbf{g}\mathbf{b}^{\mathrm{H}} \odot \mathbf{A}'$, where $\mathbf{A}'$ is a matrix consisting of phase corrections such that $\mathbf{A}(\boldsymbol{\theta}') = \mathbf{A}(\boldsymbol{\theta}) \odot \mathbf{A}'$.

Calibration is the process of identifying the unknown parameters in $\mathbf{G}$, and subsequently correcting for $\mathbf{G}$ during the imaging step. The model (10.40) in its generality is not identifiable unless we make assumptions on the structure of $\mathbf{G}$ (in the form of a suitable parametrization) and describe how it varies with time and frequency, e.g., in the form of (stochastic) models for these variations.

In the next subsection, we will first describe how models of the form (10.38) or (10.39) can be identified. This step will serve as a stepping stone in the identification of a more general $\mathbf{G}$.

10.6.2 Calibration algorithms

Let us assume a model of the form (10.38), where there are Q dominant calibration sources within the field of view. For these sources, we assume that their positions and source powers are

known with sufficient accuracy, i.e., we assume that $\mathbf{A}$ and $\mathbf{\Sigma_s}$ are known. We can then write (10.38) as

$$\mathbf{R} = \mathbf{\Gamma A \Sigma A}^{\mathrm{H}}\mathbf{\Gamma}^{\mathrm{H}} + \mathbf{\Sigma_n} \tag{10.41}$$

where $\mathbf{\Sigma} = \mathbf{B\Sigma_s B}$ is a diagonal with apparent source powers. With $\mathbf{B}$ unknown, $\mathbf{\Sigma}$ is unknown, but estimating $\mathbf{\Sigma}$ is precisely the problem of estimating source powers in given directions: a problem we studied before. Thus, once we have estimated $\mathbf{\Sigma}$ and know $\mathbf{\Sigma_s}$, we can easily estimate the directional gains $\mathbf{B}$. The problem thus reduces to estimate the diagonal matrices $\mathbf{\Gamma}$, $\mathbf{\Sigma}$ and $\mathbf{\Sigma_n}$ from a model of the form (10.41).

Single calibrator source For some cases, e.g., radio telescope arrays where the elements are traditional telescope dishes, the field of view is quite narrow (degrees) and we may assume that there is only a single calibrator source in the observation. Then $\mathbf{\Sigma} = \sigma^2$ is a scalar and the problem reduces to

$$\mathbf{R} \;=\; \mathbf{g}\sigma^2\mathbf{g}^{\mathrm{H}} \;+\; \mathbf{\Sigma_n}$$

and since $\mathbf{g}$ is unknown, we could even absorb the unknown σ in $\mathbf{g}$ (it is not separately identifiable). The structure of $\mathbf{R}$ is a rank-1 matrix $\mathbf{g}\sigma^2\mathbf{g}^{\mathrm{H}}$ plus a diagonal $\mathbf{\Sigma_n}$. This is recognized as a "rank-1 factor analysis" model.

Example 10.3. In a radio astronomy experiment reported in [21], we observe a strong point source in the sky with the Westerbork Synthesis Radio Telescope (WSRT). The point source requirement is that the source angular size is much smaller than the telescope main beam power.

In the experiments, $p = 8$ of the 14 WSRT telescopes were used, in single linear polarisation mode, with a maximum distance (baseline) of 1 km. The telescopes tracked the strong astronomical point source "3C48" at a sky frequency of 1420.4 MHz with a receiver bandwidth of 1.25 MHz. The earth-rotation related phase drift was compensated for, which means that during the experiment the telescope–interferometer phase was constant. We split the data into 32 frequency bins, each with a bandwidth of 39 kHz, which fits the narrow band assumption reasonably well. Each bin had $N = 131\,072$ samples. The data was subsequently spatially cross-correlated resulting in complex covariance matrices for each of the frequency bins, to which the factor analysis algorithms are applied.

The data model in the experiment is $\mathbf{R} = \mathbf{g}\sigma_s^2\mathbf{g}^{\mathrm{H}} + \mathbf{\Sigma_n}$, where σ_s is the source flux (known from tables). The FA algorithm gives an estimate of $\mathbf{g}$ and $\mathbf{\Sigma_n}$. Figure 10.10 shows the resulting entries of these vectors as function of frequency. The figure confirms that the received SNR is about -13 dB for each antenna, which is expected from calibration tables for this source. A bump in the noise power curves at 1420.4 MHz corresponds to the spectral line of neutral hydrogen, and is caused by the galactic emission of our Milky Way. As the Milky Way is a spatially wide source of radiowaves, it is not resolved by the WSRT interferometers, and is therefore visible only in the noise estimates.

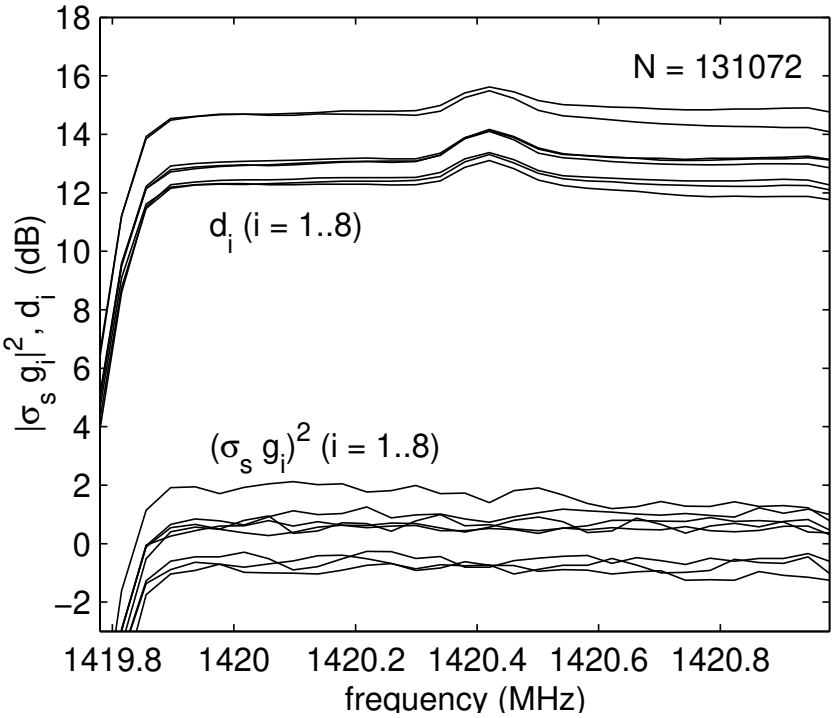


Figure 10.10. Gain magnitude and noise power estimates, as function of frequency, for an observation of the astronomical source 3C48.

In general, there are more calibrator sources (Q) in the field of view, and we have to solve (10.41). Using Factor Analysis, we can first solve for $\mathbf{A}'$ and $\mathbf{\Sigma_n}$ in

$$\mathbf{R} = \mathbf{A}'\mathbf{A}'^{\mathrm{H}} + \mathbf{\Sigma_n}\,.$$

Then, since $\mathbf{A}'$ is not unique, we can identify

$$\mathbf{A}' = \mathbf{\Gamma A \Sigma}^{1/2}\mathbf{Q}$$

where $\mathbf{Q}$ is an unknown unitary matrix. Equivalently, define $\mathbf{R}' = \mathbf{A}'\mathbf{A}^{\mathrm{H}} = \mathbf{R} - \mathbf{\Sigma_n}$, then

$$\mathbf{R}' = \mathbf{\Gamma A \Sigma A}^{\mathrm{H}}\mathbf{\Gamma}$$

where both $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ are diagonal and unknown. We may resort to an Alternating Least Squares approach. If $\mathbf{\Gamma}$ is considered known, then we can correct $\mathbf{R}_0$ for it, so that we have precisely the same problem as we considered before, (**??**), and we can solve for $\mathbf{\Sigma}$ using the techniques discussed in section 7.4. Alternatively, with $\mathbf{\Sigma}$ known, we can say we know a reference model $\mathbf{R}_0 = \mathbf{A\Sigma A}^{\mathrm{H}}$, and the problem is to identify the element gains $\mathbf{\Gamma} = \mathrm{diag}(\mathbf{g})$ from a model of the form

$$\mathbf{R}' = \mathbf{\Gamma R}_0\mathbf{\Gamma}^{\mathrm{H}}\,.$$

After applying the vec$(\cdot)$-operation, this is

$$\mathrm{vec}(\mathbf{R}') = \mathrm{diag}(\mathrm{vec}(\mathbf{R}_0))(\mathbf{g}^* \otimes \mathbf{g})\,.$$

This leads to the Least Squares problem

$$\hat{\mathbf{g}} \;=\; \arg\min_{\mathbf{g}} \; \|\mathrm{vec}(\hat{\mathbf{R}} - \mathbf{\Sigma_n}) - \mathrm{diag}(\mathrm{vec}(\mathbf{R}_0))(\mathbf{g}^* \otimes \mathbf{g})\|^2\,.$$

This problem cannot be solved in closed form. Alternatively, we can first solve an unstructured problem: define $\mathbf{x} = \mathbf{g}^* \otimes \mathbf{g}$ and solve

$$\hat{\mathbf{x}} = \mathrm{diag}(\mathrm{vec}(\mathbf{R}_0))^{-1}\mathrm{vec}(\hat{\mathbf{R}} - \mathbf{\Sigma_n})$$

or equivalently, if we define $\mathbf{X} = \mathbf{g}\mathbf{g}^{\mathrm{H}}$,

$$\hat{\mathbf{X}} = (\hat{\mathbf{R}} - \mathbf{\Sigma_n}) \oslash \mathbf{R}_0.$$

where $\oslash$ denotes an entrywise matrix division. After estimating the unstructured vector $\mathbf{X}$, we enforce the rank-1 structure $\mathbf{X} = \mathbf{g}\mathbf{g}^{\mathrm{H}}$, via a rank-1 approximation, and find an estimate for $\mathbf{g}$. The pointwise division can lead to noise enhancement; this is remediated by only using the result as an initial estimate for a Gauss-Newton iteration [22] or by formulating a *weighted* least squares problem instead [23, 24].

With $\mathbf{g}$ known, we can again estimate $\mathbf{\Sigma}$ and $\mathbf{\Sigma_n}$, and make an iteration. Overall we then obtain an alternating least squares solution.

The more general calibration problem (10.39) follows from (10.38) by writing $\mathbf{A} = \mathbf{A}(\boldsymbol{\theta}')$ where $\boldsymbol{\theta}'$ are the apparent source locations. In the alternating least squares framework, this problem can be solved in quite the same way: we solve for $\mathbf{g}$, $\boldsymbol{\theta}'$, $\boldsymbol{\sigma}_\mathbf{s}$ and $\boldsymbol{\sigma}_\mathbf{n}$ in turn, keeping the other parameters fixed at their previous estimates. After that, we can relate the apparent source locations to the (known) locations of the calibrator sources $\boldsymbol{\theta}$.

Estimating the general model In the more general case (10.40), viz.

$$\mathbf{R} = (\mathbf{G} \odot \mathbf{A})\mathbf{\Sigma_s}(\mathbf{G} \odot \mathbf{A})^{\mathrm{H}} + \mathbf{\Sigma_n}\,,$$

we have an unknown full matrix $\mathbf{G}$. We assume $\mathbf{A}$ and $\mathbf{\Sigma_s}$ known. Since $\mathbf{A}$ pointwise multiplies $\mathbf{G}$ and $\mathbf{G}$ is unknown, we might as well omit $\mathbf{A}$ from the equations without loss of generality. For the same reason also $\mathbf{\Sigma_s}$ can be omitted. This leads to a problem of the form

$$\mathbf{R} = \mathbf{G}\mathbf{G}^{\mathrm{H}} + \mathbf{\Sigma_n}\,,$$

where $\mathbf{G} : P \times Q$ and $\mathbf{\Sigma_n}$ (diagonal) are unknown. This problem is recognized as a rank-Q factor analysis problem. For reasonably small Q, as compared to the size P of $\mathbf{R}$, the factor $\mathbf{G}$ can be solved for, again using algorithms for covariance matching such as in [10].

It is important to note that $\mathbf{G}$ can be identified only up to a unitary factor $\mathbf{V}$ at the right: $\mathbf{G}' = \mathbf{G}\mathbf{V}$ would also be a solution. This factor makes the gains unidentifiable unless we introduce more structure to the problem.

10.7 NOTES

Material presented in this chapter was derived from [12, 25].

FA for real-valued matrices was first introduced by Spearman [3] in 1904 to find a quantitative measure for intelligence, given a series of test results. Between 1940 and 1970, Lawley, Anderson, Jöreskog and others developed FA as an established multivariate technique [1, 2, 5, 26, 27]. Currently, FA is an important and popular tool for latent variable analysis with many applications in various fields of science [28]. However, its application within the signal processing community has been surprisingly limited.

In the context of signal processing, the FA problem and several extensions can be regarded as a specific case of *covariance matching*, studied in detail in [10]. In there, the model (10.2) is presented more generically in terms of a parametric model $\mathbf{A}(\boldsymbol{\theta})$ and a linear parametric model for the noise covariance (not restricted to diagonal), and maximum likelihood algorithms are presented to estimate the parameters. This relates to the topic of sensor array parameter estimation (e.g., direction of arrival) in the presence of colored noise or spatially correlated noise, under a variety of possible model assumptions such as $\mathbf{D}$ being diagonal, block diagonal, or composed of a linear sum of known matrices [29–32].

Generally, algorithms for finding the model parameters in the FA model can be categorized into two groups. "Classical" approaches are based on Maximum Likelihood (ML) or related weighted least squares optimization. This results in large nonlinear optimization problems that are often implemented using Newton-Raphson or more efficient Fletcher-Powell iterations [13, 26, 33]. These algorithms are still very popular and standard toolboxes (Matlab, SPSS) use them. Unfortunately, they are relatively hard to implement and computationally rather complex due to the inversion of a large matrix containing the second-order derivatives, so that approximations

are necessary. Alternatively, the ML solution is found using Expectation-Maximization (EM) techniques, first proposed in [34], resulting in algorithms that are simpler to implement but often show slow convergence. The Conditional Maximization (CM) algorithm [14] has quadratic convergence and currently seems most competitive.

A second class of algorithms is inspired by the work of Ledermann in 1940 [4] and gained renewed momentum in recent years due to the popularity of convex optimization. The factors are found using the trace function as a convex relaxation of a minimum-rank constraint [35–37]. Recently, several new approaches for matrix completion have been proposed that involve low-rank plus sparse matrices [38, 39]. This leads to similar convex optimization algorithms, although not specifically designed with covariance matrices in mind.

In [12], we proposed new algorithms for FA (and extensions) that are of the ML type, resulting in particular in the attractive AWLS algorithm that is easy to implement and the fastest in convergence.

For the radio astronomy application, we applied FA to calibration and interference detection/filtering in [40–43]. These addressed the case where the noise covariance matrix is diagonal with unknown elements. For cases where the noise covariance matrix is no longer diagonal but has a known sparse structure, we later proposed the "extended FA" (EFA) model [12]. We also considered applications where the desired subspace changes rapidly while the noise remains stationary. In this case we can compute a series of short-term covariance matrices or "snapshots" (each with the classical FA model form (10.2) but with a common matrix $\mathbf{D}$), requiring an extension toward "joint FA" (JFA). Combined, this led to "joint extended FA" (JEFA) [12].

We recommend FA as an extension of the eigenvalue decomposition (EVD) to cases where the noise is not white. The simulations in [12] indicated that even if the noise is white, the performance penalty with respect to EVD is minor. Therefore, the more general structure of the extended FA data models enable their application in a wide range of signal processing applications.

Cramér-Rao Bounds for the presented models appear in [19].

The potential of FA and (J)EFA in practical scenarios was demonstrated for spatial filtering of RFI signals present in astronomical data in [19]. Calibration of the Westerbork radio telescope array ($P = 14$ dishes) using the Ad Hoc approach was shown in [40]. Calibration of one station of the LOFAR radio telescope array ($P = 96$ antennas) was reported in [43–45], and this application is run in daily practice of the array [46]. Using LOFAR data, EFA was demonstrated in [47] to suppress the Milky Way (broadband emission).

Bibliography

[1] Derrick Norman Lawley and A.E. Maxwell, *Factor analysis as a statistical method.* 2nd. ed., New York: Am. Elsevier Publ., 1971.

[2] K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis*. Academic Press, 1979.

[3] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, pp. 72–101, Jan 1904.

[4] Walter Ledermann, "On a problem concerning matrices with variable diagonal elements," *Proceedings of the Royal Society of Edinburgh*, vol. 60, pp. 1–17, 1 1940.

[5] K. G. Jöreskog, "A general approach to confirmatory maximum likelihood factor analysis," *Psychometrika*, vol. 34, no. 2, pp. 183–202, 1969.

[6] Sik Yum Lee, "The Gauss-Newton algorithm for the Weighted Least Squares factor analysis," *Journal of the Royal Statistical Society*, vol. 27, June 1978.

[7] Peter J. Schreier, *Statistical Signal Processing of Complex-Valued Data*. Cambridge University Press, 2010.

[8] Are Hjørungnes, *Complex-Valued Matrix Derivatives with Applications in Signal Processing and Communications*. Cambridge University Press, 2011.

[9] Steven M. Kay, *Fundamentals of Statistical Signal Processing, Estimation theory*, vol. Volume I. Prentice Hall, 1993.

[10] B. Ottersten, P. Stoica, and R. Roy, "Covariance matching estimation techniques for array signal proce ssing applications," *Digital Signal Processing, A Review Journal*, vol. 8, pp. 185–210, July 1998.

[11] P. Gill, W. Murray, and M.H. Wright, *Practical optimization*. London: Academic Press, 1981.

[12] A.M. Sardarabadi and A.J. van der Veen, "Complex factor analysis and extensions," *IEEE Tr. Signal Processing*, vol. 66, February 2018.

[13] Karl G. Jöreskog and Arthur S. Goldberger, "Factor analysis by generalized least squares," *Psychometrika*, vol. 37, pp. 243–260, Sep 1972.

[14] J.-H. Zhao, Philip Yu, and Qibao Jiang, "ML estimation for factor analysis: EM or non-EM?," *Statistics and Computing*, vol. 18, pp. 109–123, 2008. 10.1007/s11222-007-9042-y.

[15] A.-K. Seghouane, "An iterative projections algorithm for ML factor analysis," in *IEEE Workshop on Machine Learning for Signal Processing*, pp. 333–338, Oct. 2008.

[16] S.M. Kay, *Fundamentals of Statistical Signal Processing. Volume II: Detection Theory*. Upper Saddle River, NJ: Prentice Hall PTR, 1998.

[17] J. Raza, A-J Boonstra, and A-J. van der Veen, "Spatial filtering of RF interference in radio astronomy," *IEEE Signal Processing Letters*, vol. 9, Mar. 2002.

[18] S. van der Tol and A. J. van der Veen, "Performance analysis of spatial filtering of RF interference in radio astronomy," *IEEE Transactions on Signal Processing*, vol. 53, pp. 896–910, Mar. 2005.

[19] A. Mouri Sardarabadi, A.-J. van der Veen, and A.-J. Boonstra, "Spatial Filtering of RF Interference in Radio Astronomy Using a Reference Antenna Array," *IEEE Trans. Signal Process.*, vol. 64, pp. 432–447, Jan 2016.

[20] A. Leshem and A.-J. van der Veen, "Multichannel detection of Gaussian signals with uncalibrated receivers," *IEEE Signal Processing Letters*, vol. 8, no. 4, pp. 120–122, 2001.

[21] A. J. Boonstra and A. J. van der Veen, "Gain calibration methods for radio telescope arrays," *IEEE Trans. Signal Processing*, vol. 51, pp. 25–38, Jan. 2003.

[22] D. R. Fuhrmann, "Estimation of sensor gain and phase," *IEEE Trans. Signal Processing*, vol. 42, pp. 77–87, Jan. 1994.

[23] S. J. Wijnholds and A. J. Boonstra, "A multisource calibration method for phased array telescopes," in *Fourth IEEE Workshop on Sensor Array and Multi-channel Processing (SAM)*, (Waltham (Mass.), USA), July 2006.

[24] S. J. Wijnholds and A. J. van der Veen, "Multisource self-calibration for sensor arrays," *IEEE Tr. Signal Processing*, vol. 57, pp. 3512–3522, Sept. 2009.

[25] A.J. van der Veen, S.J. Wijnholds, and A.M. Sardarabadi, "Signal processing for radio astronomy," in *Handbook of Signal Processing Systems, 3rd ed.*, Springer, November 2018. ISBN 978-3-319-91734-4.

[26] Derrick N Lawley, "The estimation of factor loadings by the method of maximum likelihood.," *Proceedings of the Royal Society of Edinburgh*, vol. 60, no. 01, pp. 64–82, 1940.

[27] T. W. Anderson and H. Rubin, "Statistical inference in factor analysis," *In Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 5, pp. 111 – 150, 1956.

[28] David J. Bartholomew, Martin Knott, and Irini Moustaki, *Latent Variable Models and Factor Analysis: A Unified Approach.* John Wiley and Sons, 2011.

[29] M. Viberg, P. Stoica, and B. Ottersten, "Array processing in correlated noise fields based on instrumental variables and subspace fitting," *IEEE Trans. Signal Process.*, vol. 43, p. 1187–1199, Jan. 1995.

[30] V. Nagesha and S. M. Kay, "Maximum likelihood estimation for array processing in colored noise," *IEEE Trans. Signal Process.*, vol. 44, p. 169–180, Feb. 1996.

[31] P. Stoica, M. Viberg, K. M. Wong, and Q. Wu, "Maximum-likelihood bearing estimation with partly calibrated arrays in spatially correlated noise fields," *IEEE Trans. Signal Process.*, vol. 44, p. 888–899, Apr. 1996.

[32] M. Wax, J. Sheinvald, and A. J. Weiss, "Detection and localization in colored noise via generalized least squares," *IEEE Tr. Signal Process.*, vol. 44, pp. 1734–1743, July 1996.

[33] K. G. Jöreskog, "Some contributions to maximum likelihood factor analysis," *Psychometrika*, vol. 32, no. 4, pp. 433–482, 1967.

[34] Donald Rubin and Dorothy Thayer, "EM algorithms for ML factor analysis," *Psychometrika*, vol. 47, pp. 69–76, 1982. 10.1007/BF02293851.

[35] Alexander Shapiro, "Weighted minimum trace factor analysis," *Psychometrika*, vol. 47, no. 3, pp. 243–264, 1982.

[36] Alexander Shapiro, "Rank-reducibility of a symmetric matrix and sampling theory of minimum trace factor analysis," *Psychometrika*, vol. 47, no. 2, pp. 187–199, 1982.

[37] James Saunderson, Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky, "Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 4, pp. 1395–1416, 2012.

[38] E.J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *arXiv preprint arXiv:0912.3599*, 2009.

[39] Emmanuel J Candès and Benjamin Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[40] A.-J. Boonstra and A.-J. van der Veen, "Gain calibration methods for radio telescope arrays," *IEEE Tr. Signal Processing*, vol. 51, pp. 25–38, Jan. 2003.

[41] A-J. van der Veen, A. Leshem, and A-J. Boonstra, "Array signal processing for radio astronomy," *Experimental Astronomy (EXPA)*, vol. 17, no. 1-3, pp. 231–249, 2004. ISSN 0922-6435.

[42] A-J. van der Veen, A. Leshem, and A-J. Boonstra, "Array signal processing for radio astronomy," in *The Square Kilometre Array: An Engineering Perspective* (P.J. Hall, ed.), pp. 231–249, Dordrecht: Springer, 2005. ISBN 1-4020-3797-x. Reprinted from Experimental Astronomy, 17(1-3),2004.

[43] S.J. Wijnholds and A.-J. van der Veen, "Multisource self-calibration for sensor arrays," *Signal Processing, IEEE Transactions on*, vol. 57, pp. 3512–3522, Sept 2009.

[44] S.J. Wijnholds, S. van der Tol, R. Nijboer, and A.-J. van der Veen, "Calibration challenges for future radio telescopes," *IEEE Signal Processing Magazine*, vol. 27, pp. 30–42, Jan 2010.

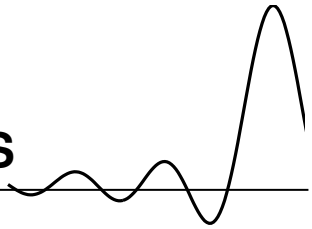
[45] A. Mouri Sardarabadi and A.-J. van der Veen, "Application of Krylov based methods in calibration for radio astronomy," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 153–156, June 2014.

[46] M. P. van Haarlem, M. W. Wise, A. W. Gunst, *et al.*, "LOFAR: The LOw-Frequency ARray," *Astronomy & Applications*, vol. 556, p. A2, 2013.

[47] A. Mouri Sardarabadi and A.-J. van der Veen, "Subspace estimation using factor analysis," in *2012 IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 477 –480, June 2012.

Chapter **11**

INDEPENDENT COMPONENT ANALYSIS



Contents

11.1 FOURTH-ORDER CUMULANTS

11.2 DATA MODEL

11.3 JADE

11.4 APPLICATION: ACMA