In Y.Huang and J. Benesty (eds.), Audio Signal Processing, Kluwer Academic Publishers, Boston, Feb. 2004

Chapter 10

BLIND SOURCE SEPARATION FOR CONVOLUTIVE MIXTURES: A UNIFIED TREATMENT

Herbert Buchner, Robert Aichner, Walter Kellermann
Telecommunications Laboratory
Multimedia Communications and Signal Processing
University of Erlangen–Nuremberg
buchner@LNT.de, aichner@LNT.de, wk@LNT.de

Abstract Blind source separation (BSS) algorithms for time series can exploit three properties of the source signals: nonwhiteness, nonstationarity, and nongaussianity. While methods utilizing the first two properties are usually based on second-order statistics (SOS), higher-order statistics (HOS) must be considered to exploit nongaussianity. In this chapter, we consider all three properties simultaneously to design BSS algorithms for convolutive mixtures within a new generic framework. This concept derives its generality from an appropriate matrix notation combined with the use of multivariate probability densities for considering the time-dependencies of the source signals. Based on a generalized cost function we rigorously derive the corresponding time-domain and frequency-domain broadband algorithms. Due to the broadband approach, time-domain constraints are obtained which provide a more detailed understanding of the internal permutation problem in traditional narrowband frequency-domain BSS. For both, the time-domain and the frequency-domain versions, we discuss links to well-known and also to novel algorithms that follow as special cases of the framework. Moreover, we use models for correlated spherically invariant random processes (SIRPs) which are well suited for a variety of source signals including speech to obtain efficient solutions in the HOS case. The concept provides a basis for off-line, on-line, and block-on-line algorithms by introducing a general weighting function, thereby allowing for tracking of time-varying real acoustic environments.

Keywords: Blind Source Separation, Convolutive Mixtures, Second-Order Statistics, Higher-Order Statistics, Time Domain, Frequency Domain, Broadband Approach, Spherically Invariant Random Processes.

1. INTRODUCTION

The problem of separating convolutive mixtures of unknown time series arises in several application domains, a prominent example being the so-called cocktail party problem, where we want to recover the speech signals of multiple speakers who are simultaneously talking in a room. The room will generally be reverberant due to reflections on the walls, i.e., the original source signals $s_q(n)$, $q = 1, \ldots, Q$ of our separation problem are filtered by a linear multiple input and multiple output (MIMO) system before they are picked up by the sensors. Most commonly used

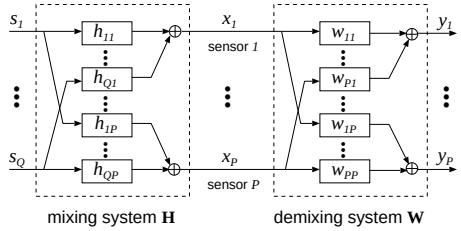


Figure 10.1 Linear MIMO model for BSS.

BSS algorithms are developed under the assumption that the number Q of source signals $s_q(n)$ equals the number P of sensor signals $x_p(n)$. However, the more general scenario with an arbitrary number of sources and sensors can always be reduced to the standard BSS model (Fig. 10.1). The case that the sensors outnumber the sources is termed *overdetermined* BSS ($P > Q$). The main approach to simplify the separation problem in this case is to apply principle component analysis (PCA) [1], extract the first P components and then use standard BSS algorithms. The more difficult case $P < Q$ is called *underdetermined* BSS or BSS with *overcomplete bases*. Mostly the sparseness of the sources in the time-frequency domain is used to determine clusters which correspond to the separated sources (e.g., [2]). Recent developments showed that the sparseness can be exploited to eliminate $Q - P$ sources, and then again standard BSS algorithms can be applied [3].

Throughout this chapter, we therefore regard the standard BSS model where the number Q of source signals $s_q(n)$ equals the number of sensor signals $x_p(n)$, $p = 1, \ldots, P$ (Fig. 10.1). An M-tap mixing system is thus described by

$$x_p(n) = \sum_{q=1}^{P} \sum_{\kappa=0}^{M-1} h_{qp}(\kappa) s_q(n - \kappa), \tag{10.1}$$

where $h_{qp}(\kappa)$, $\kappa = 0, \ldots, M-1$ denote the coefficients of the finite impulse response (FIR) filter model from the q-th source to the p-th sensor.

In BSS, we are interested in finding a corresponding demixing system according to Fig. 10.1, where the output signals $y_q(n)$, $q = 1, \ldots, P$ are described by

$$y_q(n) = \sum_{p=1}^{P} \sum_{\kappa=0}^{L-1} w_{pq}(\kappa) x_p(n-\kappa). \tag{10.2}$$

The separation of the mixtures obtained by the sensor signals $x_p(n)$ utilizes the fundamental assumption of statistical independence between the original source signals $s_q(n)$. It can be shown (see, e.g., [1]) that the MIMO demixing system coefficients $w_{pq}(\kappa)$ can in fact reconstruct the sources up to an unknown permutation of their order and an unknown filtering of the individual signals, where the demixing filter length L should be chosen at least equal to M. It should be stressed that the filtering ambiguity prevents a deconvolution of the sensor signals and therefore BSS achieves a mere separation of statistically independent signals.

From the description of the BSS model (see Fig. 10.1) it can be seen that this technique is closely related to adaptive beamforming. This relationship was first shown in [4] where BSS was also termed blind beamforming. Thus, as an inherent advantage of BSS, prior knowledge of the spatial position of the sensors and sources is not necessary and, therefore, BSS is robust against unknown array deformations or distortions of the wavefront. Another important difference is the optimization criterion in BSS which utilizes the statistical independence of the source signals. Thus, adaptation of the demixing system is possible even if all source signals are simultaneously active in contrast to adaptive beamforming where the distinction between target signal activity and interfering signal activity has to be made [5]. However, one drawback of most BSS algorithms is that currently the number of sources has to be known for estimating the demixing system.

In [6] it was shown that merely decorrelating the output signals $y_q(n)$ does not lead to a separation of the sources. This implies that we have to force the output signals to become statistically decoupled up to joint moments of a certain order by using additional conditions. This can be realized by using approaches to blindly estimate the P^2L MIMO coefficients $w_{pq}(\kappa)$ in (10.2) by exploiting one of the following source signal properties [1]:

(i) Nonwhiteness. Exploited by simultaneous diagonalization of output correlation matrices over multiple time-lags, e.g., [7, 8].

(ii) Nonstationarity. Exploited by simultaneous diagonalization of short-time output correlation matrices at different time instants, e.g., [6], [9]-[17].

(iii) Nongaussianity. Exploited by using higher order statistics for independent component analysis (ICA), e.g., [18]-[23].

While there are several algorithms for convolutive mixtures - both in the time domain and in the frequency domain - utilizing one of these properties, few algorithms explicitly exploit two properties [24, 25] and so far, none is known which simultaneously exploits all three properties. However, it has recently been shown that in practical scenarios, the combination of these criteria can lead to improved performance [24, 25].

Extending the work in [26, 27], we present in the following a rigorous derivation of a unified framework for convolutive mixtures exploiting all three signal properties by using HOS. This is made possible by introducing an appropriate matrix notation combined with the use of multivariate probability densities for considering the time-dependencies of the source signals. The approach is suitable for on-line and off-line algorithms as it uses a general weighting function, thereby allowing for tracking of time-varying environments [28]. The processing delay can be kept low by working with overlapping and/or partitioned signal blocks [29]. Having derived a generic time-domain algorithm, we introduce a model for spherically invariant random processes (SIRPs) [30] which are well suited, e.g., for speech to allow efficient realizations. Moreover, we discuss links to well-known SOS algorithms and we show that a previously presented algorithm [26] is the optimum second-order BSS approach in the sense of minimum mutual information known from information theory. Furthermore we introduce an equivalent broadband formulation in the frequency domain by extending the tools of [31] to unsupervised adaptive filtering. This will also give a detailed insight in the internal permutation problem of narrowband frequency-domain BSS. Again, links to well-known and extended HOS and SOS algorithms as special cases are discussed. Moreover, using the so-called generalized coherence [32], links between the time-domain and frequency-domain SOS algorithms can be established [26] showing that our cost function leads to an update equation with an inherent normalization. As shown by experimental results, this allows an efficient separation of real-world speech signals.

2. GENERIC BLOCK TIME-DOMAIN BSS ALGORITHM

In this section, we first introduce a general matrix formulation as a basis for a rigorous derivation of time-domain algorithms from a cost function which inherently takes into account all three fundamental signal properties (i)-(iii). We then consider the so-called equivariance property in the convolutive case for deriving the corresponding natural gradient update. From this formulation, several well-known and novel algorithms follow as special cases.

2.1 MATRIX NOTATION FOR CONVOLUTIVE MIXTURES

From Fig. 10.1, it can be seen that the output signals $y_q(n)$ are obtained by convolving the input signals $x_p(n)$ with the demixing filter coefficients w_{pq}. In addition to the filter length L and the number of channels P we need to introduce two more parameters for the following general formulation:

- the number of time-lags D taken into account for exploiting the nonwhiteness property of the input signals as shown below ($1 \leq D \leq L$), and
- the block length N as basis for averaging the estimates of the multivariate probability density functions (pdfs) as used below ($N > PD$ in general; $N > D$ for the natural gradient update discussed below).

To derive an algorithm for block processing of convolutive mixtures taking into account D time-lags, we first need to reformulate the convolution (10.2):

$$\mathbf{y}_q(m,j) = \sum_{p=1}^{P} \mathbf{x}_p(m,j)\mathbf{W}_{pq}, \tag{10.3}$$

where m denotes the block index, and $j = 0, \cdots, N-1$ is a time-shift index within a block of length N, and

$$\mathbf{x}_p(m,j) = [x_p(mL+j), \ldots, x_p(mL-2L+1+j)], \tag{10.4}$$
$$\mathbf{y}_q(m,j) = [y_q(mL+j), \ldots, y_q(mL-D+1+j)]. \tag{10.5}$$

The $2L \times D$ matrix $\mathbf{W}_{pq}$ exhibits a Sylvester structure that contains all L coefficients of the respective demixing filter in each column needed for

the matrix formulation of the linear convolution:

$$\mathbf{W}_{pq} = \begin{bmatrix} w_{pq,0} & 0 & \cdots & 0 \\ w_{pq,1} & w_{pq,0} & \ddots & \vdots \\ \vdots & w_{pq,1} & \ddots & 0 \\ w_{pq,L-1} & \vdots & \ddots & w_{pq,0} \\ 0 & w_{pq,L-1} & \ddots & w_{pq,1} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & w_{pq,L-1} \\ 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{10.6}$$

It can be seen that for the general case, $1 \leq D \leq L$, the last $L-D+1$ rows are padded with zeros to ensure compatibility with the length of $\mathbf{x}_p(m,j)$ with regard to a concise frequency-domain formulation in Sect. 3. Finally, to allow a convenient notation of the algorithm we combine all channels, and thus we can write (10.3) compactly as

$$\mathbf{y}(m,j) = \mathbf{x}(m,j)\mathbf{W}, \tag{10.7}$$

with

$$\begin{aligned} \mathbf{x}(m,j) &= [\mathbf{x}_1(m,j), \ldots, \mathbf{x}_P(m,j)], && (10.8)\\ \mathbf{y}(m,j) &= [\mathbf{y}_1(m,j), \ldots, \mathbf{y}_P(m,j)], && (10.9)\\ \mathbf{W} &= \begin{bmatrix} \mathbf{W}_{11} & \cdots & \mathbf{W}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{P1} & \cdots & \mathbf{W}_{PP} \end{bmatrix}. && (10.10) \end{aligned}$$

Also, with respect to the frequency-domain derivation in Sect. 3. we extend (10.7) by collecting all N vectors $\mathbf{x}_p$, $\mathbf{y}_q$, so that all output signal samples of the m-th block are captured:

$$\mathbf{Y}(m) = \mathbf{X}(m)\mathbf{W}, \tag{10.11}$$

with the matrices

$$\begin{aligned} \mathbf{Y}(m) &= [\mathbf{Y}_1(m), \cdots, \mathbf{Y}_P(m)], && (10.12)\\ \mathbf{X}(m) &= [\mathbf{X}_1(m), \cdots, \mathbf{X}_P(m)], && (10.13)\\ \mathbf{Y}_q(m) &= [\mathbf{y}_q^T(m,0), \ldots, \mathbf{y}_q^T(m,N-1)]^T, && (10.14)\\ \mathbf{X}_p(m) &= [\mathbf{x}_p^T(m,0), \ldots, \mathbf{x}_p^T(m,N-1)]^T. && (10.15) \end{aligned}$$

Superscript ^{T} denotes the transposition of a vector or a matrix. Obviously, $\mathbf{X}_p(m)$, $p = 1, \ldots, P$ in (10.15) are Toeplitz matrices of size $(N \times 2L)$ due to the shift of subsequent rows by one sample each:

$$\mathbf{X}_p(m) = \begin{bmatrix} x_p(mL) & \cdots & x_p(mL - 2L + 1) \\ x_p(mL + 1) & \ddots & x_p(mL - 2L + 2) \\ \vdots & \ddots & \vdots \\ x_p(mL + N - 1) & \cdots & x_p(mL - 2L + N) \end{bmatrix}. \quad (10.16)$$

Analogously to supervised block-based adaptive filtering [29, 31], the approach followed here can also be carried out with overlapping and/or partitioned data blocks to increase the convergence rate and to reduce the signal delay. Overlapping is done by simply replacing the time index mL in the equations by $m\frac{L}{\alpha}$ with the overlap factor $1 \leq \alpha \leq L$. For clarity, we will omit the overlap factor and will point to it when necessary.

2.2 COST FUNCTION AND ALGORITHM DERIVATION

A generic SOS algorithm for convolutive mixtures has been derived in [26] from a cost function that explicitly contains correlation matrices that include several time-lags (c.f. property (i)) under the assumption of short-time stationarity (c.f. property (ii)). Additionally, for exploiting property (iii), higher order statistics have to be considered. Higher-order approaches for BSS can be divided into three classes [1]: maximum likelihood (ML) estimation [21], minimization of the mutual information (MMI) among the output signals [33], and maximization of the entropy (ME/'infomax') [23]. Although all of these HOS approaches lead to similar update rules, MMI can be regarded as the most general one [33].

Based on a generalization of Shannon's mutual information [34], we now define the following cost function which simultaneously accounts for the three fundamental properties (i)-(iii):

$$\begin{aligned} \mathcal{J}(m) \;=\; & -\sum_{i=0}^{\infty} \beta(i,m) \frac{1}{N} \sum_{j=0}^{N-1} \{\log\left(\hat{p}_{1,D}(\mathbf{y}_1(i,j)) \cdot \ldots \cdot \hat{p}_{P,D}(\mathbf{y}_P(i,j))\right) \\ & - \log\left(\hat{p}_{PD}(\mathbf{y}_1(i,j), \ldots, \mathbf{y}_P(i,j))\right)\}, \end{aligned} \quad (10.17)$$

where $\hat{p}_{p,D}(\cdot)$ is the estimated or assumed *multivariate* probability density function (pdf) for channel p of dimension D and $\hat{p}_{PD}(\cdot)$ is the joint pdf of dimension PD over all channels. Furthermore, D is the memory length, i.e., the number of time-lags to model the nonwhiteness of

the P signals as above. Note also that the time series of these pdf estimates completely describes any multichannel stochastic process with the assumption of short-time stationarity over length-N blocks (this assumption is reasonable for many real-world signals such as speech). The expectation operator of the mutual information [34] is replaced in (10.17) by short-time averages within these blocks. β is a window function with finite support that is normalized according to $\sum_{i=0}^{\infty}\beta(i,m)=1$, and allows off-line, on-line, and block-online implementations of the algorithms. As an example, $\beta(i,m)=(1-\lambda)\lambda^{m-i}$ for $0\leq i\leq m$, and $\beta(i,m)=0$ else, leads to an efficient on-line version allowing for tracking in time-varying environments [28].

In this chapter, we consider algorithms based on first-order gradients. An extension to higher-order gradients would be straightforward but computationally more expensive. Moreover, to obtain general expressions allowing a smooth transition to the frequency domain, we consider complex signals for the derivative. In order to calculate the gradient [35, 36]

$$\nabla_{\mathbf{W}}\mathcal{J}(m)=2\frac{\partial\mathcal{J}(m)}{\partial\mathbf{W}^{*}}, \tag{10.18}$$

we need to express the cost function (10.17) in terms of the demixing matrix $\mathbf{W}$ which contains the coefficients of all channels. A common way to achieve this is to transform the output signal pdf $\hat{p}_{PD}(\mathbf{y})$ into the PD-dimensional input signal pdf using $\mathbf{W}$ which is considered as a mapping matrix for this linear transformation. This procedure is directly applied to the second term in the braces of (10.17), followed by differentiation w.r.t. $\mathbf{W}$. The derivative of the input signal pdf, which appears as an additive constant due to the logarithm, vanishes as it is independent of $\mathbf{W}$. The argument of the logarithm in the first term in the braces, however, is factorized among the channels. Therefore, we apply the chain rule in this case, rather than transforming the pdfs.

Finally, the generic HOS gradient for the coefficient update utilizing all three signal properties (i)-(iii) can be expressed as

$$\begin{aligned}\nabla_{\mathbf{W}}\mathcal{J}(m) \;=\; & \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\sum_{j=0}^{N-1}\Big\{\mathbf{x}^{H}(i,j)\mathbf{\Phi}(\mathbf{y}(i,j)) \\ & -\mathbf{V}_{2LP\times DP}^{1_D 0}\left(\mathbf{W}^{H}\mathbf{V}_{2LP\times DP}^{1_D 0}\right)^{-1}\Big\},\end{aligned} \tag{10.19}$$

with the *multivariate score function*

$$\mathbf{\Phi}(\mathbf{y}(i,j))=\left[-\frac{\frac{\partial\hat{p}_{1,D}(\mathbf{y}_1(i,j))}{\partial\mathbf{y}_1(i,j)}}{\hat{p}_{1,D}(\mathbf{y}_1(i,j))},\ldots,-\frac{\frac{\partial\hat{p}_{P,D}(\mathbf{y}_P(i,j))}{\partial\mathbf{y}_P(i,j)}}{\hat{p}_{P,D}(\mathbf{y}_P(i,j))}\right], \tag{10.20}$$

and the $2LP \times DP$ window matrix $\mathbf{V}^{1_D 0}_{2LP \times DP}$ defined as

$$\mathbf{V}^{1_D 0}_{2LP \times DP} = \text{bdiag}\left\{\mathbf{W}^{1_D 0}_{2L \times D}, \ldots, \mathbf{W}^{1_D 0}_{2L \times D}\right\}, \tag{10.21}$$

$$\mathbf{W}^{1_D 0}_{2L \times D} = \left[\mathbf{I}_{D \times D}, \mathbf{0}_{(2L-D) \times D}\right]. \tag{10.22}$$

The operator $\text{bdiag}\{\mathbf{A}_1, \ldots, \mathbf{A}_P\}$ denotes a block-diagonal matrix with submatrices $\mathbf{A}_1, \ldots, \mathbf{A}_P$ on its diagonal. For the description of window matrices (also appearing in the frequency-domain algorithms in Sect. 3.) we use the following conventions:

- The lower index of a matrix denotes its dimensions.
- P-channel matrices (as indicated by the size in the lower index) are partitioned into P single-channel window matrices.
- The upper index describes the positions of ones and zeros. Unity submatrices are always located at the upper left ('10') or lower right ('01') corners of the respective single-channel window matrix. The size of these clusters is indicated in subscript (e.g., '01_L').

The window matrix $\mathbf{V}^{1_D 0}_{2LP \times DP}$ appears due to the transformation of pdfs by the non-square Sylvester matrix $\mathbf{W}$ [37].

With an iterative optimization procedure, the current demixing matrix is obtained by the recursive update equation

$$\mathbf{W}(m) = \mathbf{W}(m-1) - \mu \Delta \mathbf{W}(m), \tag{10.23}$$

where μ is a stepsize parameter, and $\Delta \mathbf{W}(m)$ is the update which is set equal to $\nabla_{\mathbf{W}} \mathcal{J}(m)$ for gradient descent adaptation. Due to the adaptation process, the coefficient matrix becomes time-variant. For clarity we will generally omit the block index of $\mathbf{W}$ and will point to it when necessary. Note that the Sylvester structure (see Eqs. 10.6, 10.10) of the update in (10.23) has to be ensured. (The structure of the update might be disturbed by imprecision effects and also depends on the technique used for estimating the pdfs.) A simple remedy for this generic update is to pick the first column and replicate it. For special cases, and frequency-domain versions discussed later, we will give more specific solutions for enforcing this constraint.

2.3 EQUIVARIANCE PROPERTY AND NATURAL GRADIENT

It is known that stochastic gradient descent, i.e., $\Delta \mathbf{W}(m) = \nabla_{\mathbf{W}} \mathcal{J}(m)$ suffers from slow convergence in many practical problems due to statistical dependencies in the data being processed.

In the BSS application, we can show that the separation performance of the gradient update rule (10.19), (10.23) depends on the MIMO mixing system. The mixing process can be described analogously to (10.7) by $\mathbf{x}(m,j) = \mathbf{s}(m,j)\mathbf{H}$, where $\mathbf{s}(m,j)$ is the corresponding $1\times P(M+L-1)$ source signal row vector and $\mathbf{H}$ is the $P(M+L-1)\times 2PL$ mixing matrix in Sylvester structure. The dimensions result from the linearity condition of the convolution. Due to the inevitable filtering ambiguity in convolutive BSS (e.g., [1]), it is at best possible to obtain an arbitrary *block diagonal* matrix $\mathbf{C} = \mathbf{HW}$, i.e., $\mathbf{C} - \operatorname{bdiag}\mathbf{C} = \mathbf{0}$, where $\mathbf{C}$ combines mixing and unmixing coefficient matrices. This means the output signals can become mutually independent but the output signals are still arbitrarily filtered versions of the source signals. To see how (10.19) behaves, we pre-multiply both sides of (10.19) by $\mathbf{H}$. This way it can easily be shown that $\mathbf{C}(m)$ depends on the mixing system $\mathbf{H}$, and, therefore, on its conditioning.

Fortunately, a modification of the ordinary gradient, termed the *natural gradient* by Amari [20] and the *relative gradient* by Cardoso [21] (which is equivalent to the natural gradient in the BSS application) has been developed that largely removes all effects of an ill-conditioned mixing matrix $\mathbf{H}$ assuming an appropriate initialization of $\mathbf{W}$. The idea of the relative gradient is based on the equivariance property. Generally speaking, an estimator behaves equivariantly if it produces estimates that, under data transformation, are similarly transformed. A key property of equivariant estimators is that they exhibit uniform performance. In [26] the natural/relative gradient has been extended to the case of Sylvester matrices yielding

$$\nabla_{\mathbf{W}}^{\mathrm{NG}}\mathcal{J} = \mathbf{W}\mathbf{W}^H\nabla_{\mathbf{W}}\mathcal{J}. \tag{10.24}$$

Together with (10.19) this immediately leads to the following expression:

$$\nabla_{\mathbf{W}}^{\mathrm{NG}}\mathcal{J}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\sum_{j=0}^{N-1}\mathbf{W}\left\{\mathbf{y}^H(i,j)\mathbf{\Phi}(\mathbf{y}(i,j)) - \mathbf{I}\right\}, \tag{10.25}$$

which is then used as update $\Delta\mathbf{W}$ in (10.23).

In the derivation of the natural gradient for instantaneous mixtures, the fact that the demixing matrices form a so-called Lie group has played an important role [20]. However, the block-Sylvester matrices $\mathbf{W}$ after (10.6), (10.10) do not form a Lie group (as they are generally not invertible). To see that the above formulation of the natural gradient is indeed justified, we again pre-multiply the update (10.25) with $\mathbf{H}$, which

leads to

$$\Delta\mathbf{C}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\sum_{j=0}^{N-1}\mathbf{C}(i)\left\{\mathbf{y}^H(i,j)\mathbf{\Phi}(\mathbf{y}(i,j)) - \mathbf{I}\right\}. \quad (10.26)$$

Thus, the temporal evolution of $\mathbf{C} = \mathbf{C}(m)$ depends only on the estimated source signal vector sequence and the stepsize μ, and the dependency on the mixing matrix $\mathbf{H}$ has been absorbed as an initial condition into $\mathbf{C}(0) = \mathbf{H}\mathbf{W}(0)$ leading to the desired uniform performance of (10.25) proving the equivariance property of the natural gradient.

Another well-known advantage of using the natural gradient is a reduction of the computational complexity of the update as the inversion of the $PD \times PD$ matrix $\mathbf{W}^H\mathbf{V}_{2LP\times PD}^{1_D 0}$ in (10.19) need not be carried out in (10.25). Furthermore, it can be shown for specific pdfs (Sect. 2.4) that instead of $N > PD$ the condition $N > D$ is sufficient for the natural gradient update due to the smaller matrices to be inverted [26].

Moreover, noting that the products of Sylvester matrices $\mathbf{W}_{pq}$ and the remaining matrices in the update equation (10.25) can be described by linear convolutions, they can be efficiently implemented by a fast convolution.

The update in (10.25) represents a so-called holonomic algorithm as it imposes the constraint $\mathbf{y}^H(i,j)\mathbf{\Phi}(\mathbf{y}(i,j)) = \mathbf{I}$ on the magnitudes of the recovered signals. However, when the source signals are nonstationary, these constraints may force a rapid change in the magnitude of the demixing matrix leading to numerical instabilities in some cases (see, e.g., [19]). Replacing $\mathbf{I}$ in (10.25) by the term $\mathrm{bdiag}\{\mathbf{y}^H(i,j)\mathbf{\Phi}(\mathbf{y}(i,j))\}$ yields the nonholonomic natural gradient algorithm with improved convergence characteristics for nonstationary sources:

$$\begin{aligned}\nabla_{\mathbf{W}}^{\mathrm{NG}}\mathcal{J}(m) &= \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\sum_{j=0}^{N-1}\mathbf{W}\Big\{\mathbf{y}^H(i,j)\mathbf{\Phi}(\mathbf{y}(i,j)) \\ &\quad -\mathrm{bdiag}\{\mathbf{y}^H(i,j)\mathbf{\Phi}(\mathbf{y}(i,j))\}\Big\}. \quad (10.27)\end{aligned}$$

Here, the bdiag operator sets all channel-wise cross-terms to zero. Note that the nonholonomic property can also be directly taken into account in the cost function as shown in [27].

2.4 SPECIAL CASES AND LINKS TO KNOWN TIME-DOMAIN ALGORITHMS

The update rules (10.19) and (10.25) provide a very general basis for BSS of convolutive mixtures. However, to apply them in a real-world scenario, an appropriate multivariate score function (10.20) has to

be determined, i.e., we have to handle P high-dimensional multivariate pdfs $\hat{p}_{p,D}(\mathbf{y}_p(i,j))$, $p = 1,\ldots,P$. In general, this is a very challenging task, as it includes all corresponding higher-order cumulants (including time-lags which may be on the order of several hundred in real acoustic environments).

In the following we will present an efficient solution for these problems by assuming so-called spherically invariant random processes (SIRPs). Moreover we will show some links to SOS algorithms. Without loss of generality we consider now the case $P = Q = 2$ for simplicity.

2.4.1 Incorporating Spherically Invariant Random Processes (SIRPs) as Signal Model. The SIRP models are representative for a wide class of stochastic processes. It has been shown that speech signals in particular can very accurately be represented by SIRPs [30]. One of the great advantages arising from the SIRP model is that multivariate pdfs can be derived analytically from the corresponding univariate probability density function together with the correlation matrices including time-lags. The correlation matrices can be estimated from the data while for the univariate pdf, we can assume one of the well-known functions for speech signals, e.g., the Laplacian density, or we can estimate the univariate pdf as well, based on parameterized representations, such as the Gram-Charlier or Edgeworth expansions [18].

The general model of a correlated SIRP of D-th order for channel p is given with a properly chosen function $f_{p,D}(\cdot)$ by [30]

$$\hat{p}_{p,D}(\mathbf{y}_p(i,j)) = \frac{1}{\sqrt{\pi^D \det(\mathbf{R}_{\mathbf{y}_p\mathbf{y}_p}(i))}} f_{p,D}\left(\mathbf{y}_p(i,j)\mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p}(i)\mathbf{y}^H_p(i,j)\right) \tag{10.28}$$

with the $D \times D$ correlation matrix $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}$ defined as

$$\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i) = \frac{1}{N}\sum_{j=0}^{N-1} \mathbf{y}^H_p(i,j)\mathbf{y}_q(i,j) = \frac{1}{N}\mathbf{Y}^H_p(i)\mathbf{Y}_q(i). \tag{10.29}$$

As the best known example, the multivariate Gaussian can be viewed as a special case of the class of SIRPs. To calculate the score function for SIRPs in general, we employ the chain rule [36] to Eq. 10.28

$$\frac{\frac{\partial \hat{p}_{p,D}(\mathbf{y}_p(i,j))}{\partial \mathbf{y}_p(i,j)}}{\hat{p}_{p,D}(\mathbf{y}_p(i,j))} = \underbrace{\left[-\frac{1}{f_{p,D}(u_p)}\frac{\partial f_{p,D}(u_p)}{\partial u_p}\right]}_{:=\phi_{p,D}(u_p)} \mathbf{y}_p(i,j)\mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p}(i), \tag{10.30}$$

where $u_p = \mathbf{y}_p \mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p} \mathbf{y}_p^H$. For convenience, we call the scalar function $\phi_{p,D}(u_p)$ the *SIRP score* of channel p.

Having derived the multivariate score function for the SIRP model (10.30), we can now introduce it into the generic HOS natural gradient update equation (10.25) with its nonholonomic extension. In the 2-by-2 case, this leads to the following expression for the *nonholonomic* HOS-SIRP update:

$$\Delta\mathbf{W}(m) = 2\sum_{i=0}^{\infty} \beta(i,m)\mathbf{W} \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{R}}_{\mathbf{y}_1\mathbf{y}_2}(i)\mathbf{R}^{-1}_{\mathbf{y}_2\mathbf{y}_2}(i) \\ \tilde{\mathbf{R}}_{\mathbf{y}_2\mathbf{y}_1}(i)\mathbf{R}^{-1}_{\mathbf{y}_1\mathbf{y}_1}(i) & \mathbf{0} \end{bmatrix}, \tag{10.31}$$

where the modified matrices $\tilde{\mathbf{R}}_{\mathbf{y}_p\mathbf{y}_q}$, $p \neq q$ are given by

$$\tilde{\mathbf{R}}_{\mathbf{y}_p\mathbf{y}_q}(i) = \frac{1}{N}\sum_{j=0}^{N-1} \phi_{q,D}\left(\mathbf{y}_q(i,j)\mathbf{R}^{-1}_{\mathbf{y}_q\mathbf{y}_q}(i)\mathbf{y}_q^H(i,j)\right)\mathbf{y}_p^H(i,j)\mathbf{y}_q(i,j), \tag{10.32}$$

$$\phi_{q,D}(u_q) = -\frac{f'_{q,D}(u_q)}{f_{q,D}(u_q)}. \tag{10.33}$$

The SIRP score $\phi_{q,D}(u_q)$ of channel q in (10.32) is a scalar value function which causes a weighting of the correlation matrix.

From the update equation (10.31), we see that the SIRP model leads to an inherent normalization by the auto-correlation submatrices.

To derive a HOS-SIRP realization using (10.33) we need an analytical expression of the multivariate pdfs (10.28) for all channels. As noted above, for SIRPs, these expressions can actually be derived from the univariate pdfs [30]. Following the procedure in [30], we obtain, e.g., as the *optimum SIRP score for univariate Laplacian pdfs* [27]:

$$\phi_{q,D}(u_q) = -\frac{1}{D - \sqrt{2s}\frac{K_{D/2+1}(\sqrt{2u_q})}{K_{D/2}(\sqrt{2u_q})}}, \tag{10.34}$$

where $K_\nu(\cdot)$ denotes the ν-th order modified Bessel function of the second kind.

2.4.2 Generic BSS based on Second-Order Statistics. To see the link to second-order BSS algorithms we use the model of multivariate Gaussian pdfs in the general cost function (10.17). As for Gaussian pdfs the cost function reduces to SOS we only utilize the nonstationarity and the nonwhiteness of the source signals. We now insert the multivariate Gaussian pdf

$$\hat{p}_{p,D}(\mathbf{y}_p(i,j)) = \frac{1}{\sqrt{(2\pi)^D \det(\mathbf{R}_{\mathbf{y}_p\mathbf{y}_p}(i))}} e^{-\frac{1}{2}\mathbf{y}_p(i,j)\mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p}(i)\mathbf{y}_p^H(i,j)} \tag{10.35}$$

in the natural gradient update equation of the generic HOS BSS algorithm (10.25). Note that there are several different representations of real and complex Gaussian multivariate pdfs in the literature [37, 38]. The most important ones in practice being the real case for speech and audio applications, and the rotation-invariant complex case mostly used in communication theory. In both cases the elements of the score function $\mathbf{\Phi}(\mathbf{y}(i,j))$ for a Gaussian pdf reduce to

$$\frac{\frac{\partial \hat{p}_{p,D}(\mathbf{y}_p(i,j))}{\partial \mathbf{y}_p(i,j)}}{\hat{p}_{p,D}(\mathbf{y}_p(i,j))} = \mathbf{y}_p(i,j)\mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p}(i). \tag{10.36}$$

With (10.25) and (10.36) we finally obtain the natural gradient update of the generic SOS BSS algorithm originally introduced in [26]

$$\nabla^{\mathrm{NG}}_{\mathbf{W}}\mathcal{J}(m) = 2\sum_{i=0}^{\infty}\beta(i,m)\mathbf{W}\left\{\mathbf{R}_{\mathbf{yy}}(i) - \operatorname{bdiag}\mathbf{R}_{\mathbf{yy}}(i)\right\}\operatorname{bdiag}^{-1}\mathbf{R}_{\mathbf{yy}}(i) \tag{10.37}$$

with the $PD \times PD$ short-time correlation matrix $\mathbf{R}_{\mathbf{yy}}(i)$ defined as

$$\mathbf{R}_{\mathbf{yy}}(i) = \frac{1}{N}\sum_{j=0}^{N-1}\mathbf{y}^H(i,j)\mathbf{y}(i,j) = \frac{1}{N}\mathbf{Y}^H(i)\mathbf{Y}(i). \tag{10.38}$$

For the 2×2 case we can express (10.37) as

$$\nabla^{\mathrm{NG}}_{\mathbf{W}}\mathcal{J}(m) = 2\sum_{i=0}^{\infty}\beta(i,m)\mathbf{W}\begin{bmatrix} \mathbf{0} & \mathbf{R}_{\mathbf{y}_1\mathbf{y}_2}(i)\mathbf{R}^{-1}_{\mathbf{y}_2\mathbf{y}_2}(i) \\ \mathbf{R}_{\mathbf{y}_2\mathbf{y}_1}(i)\mathbf{R}^{-1}_{\mathbf{y}_1\mathbf{y}_1}(i) & \mathbf{0} \end{bmatrix}. \tag{10.39}$$

This generic SOS algorithm leads to very robust practical solutions even for a large number of filter taps (see below) due to an inherent normalization by the auto-correlation matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_p}$ as known from the recursive least-squares (RLS) algorithm in supervised adaptive filtering [35]. Again, it is important to note that the products of Sylvester matrices $\mathbf{W}_{pq}$ and the remaining matrices in the update equation (10.39) can be described by linear convolutions. Thus they can be efficiently implemented by a fast convolution as in [25].

Moreover, by comparing (10.39) to the HOS-SIRP update (10.31), it can be seen that due to the fact that only SOS are utilized we obtain the same update with the nonlinearity omitted, i.e., $\phi_{q,D}(u_q) = 1$, $q = 1, \ldots, P$.

The original derivation [26] of the generic SOS natural gradient update (10.37) was based on a generalization of the cost function of [10]:

$$\mathcal{J}(m) = \sum_{i=0}^{\infty}\beta(i,m)\left\{\log\det\operatorname{bdiag}\mathbf{R}_{\mathbf{yy}}(i) - \log\det\mathbf{R}_{\mathbf{yy}}(i)\right\}. \tag{10.40}$$

In Fig. 10.2 the mechanism of the SOS cost function (10.40) is illustrated. By minimizing $\mathcal{J}(m)$, all cross-correlations for D time-lags are reduced and will ideally vanish, while the auto-correlations are untouched. As both cost functions (10.17) and (10.40) lead to the same

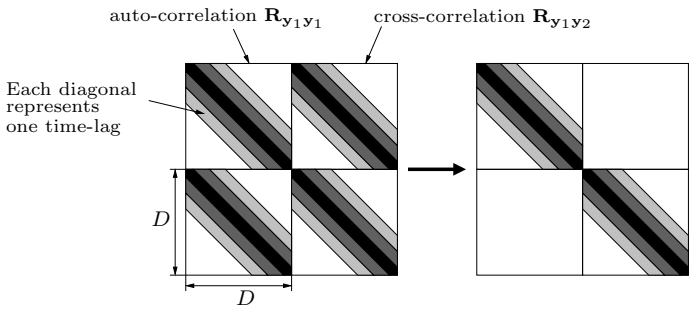


Figure 10.2 Illustration of the SOS cost function (10.40).

result in the SOS case, we may now conclude that the algorithm in [26] is in fact the optimum SOS algorithm for convolutive mixtures in the sense of minimum mutual information or ML, which also implies asymptotic Fisher-efficiency [1, 39].

Another interesting finding is that for both, the holonomic and nonholonomic versions of the HOS update (10.25), (10.27), the SOS BSS algorithm obtained by inserting the Gaussian pdf (10.35) turns out to be nonholonomic confirming its good performance for speech sources.

Note that in principle, there are two basic methods to estimate the output correlation matrices (10.38) for nonstationary output signals: the so-called correlation method, and the covariance method as they are known from linear prediction problems [40]. While the correlation method leads to a slightly lower computational complexity due to the Toeplitz structure of the matrices $\mathbf{R}_{\mathbf{yy}}$ (and to smaller matrices, when implemented in the frequency domain covered in Sect. 3.), we consider the more accurate covariance method in this chapter. Note also that (10.38) is full rank since in general we assume $N > PD$.

2.4.3 Approximations of the Generic BSS based on Second-Order Statistics. The generic update (10.37) is now analyzed and links to known algorithms (see Fig. 10.3) are presented. We highlight here two realizations.

For $D = 1$, the correlation matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i)$ become scalar values as only a single lag is considered for the correlations. Thus the resulting algorithm is only taking the nonstationarity property into account. This was first proposed by Kawamoto et al. in [11].

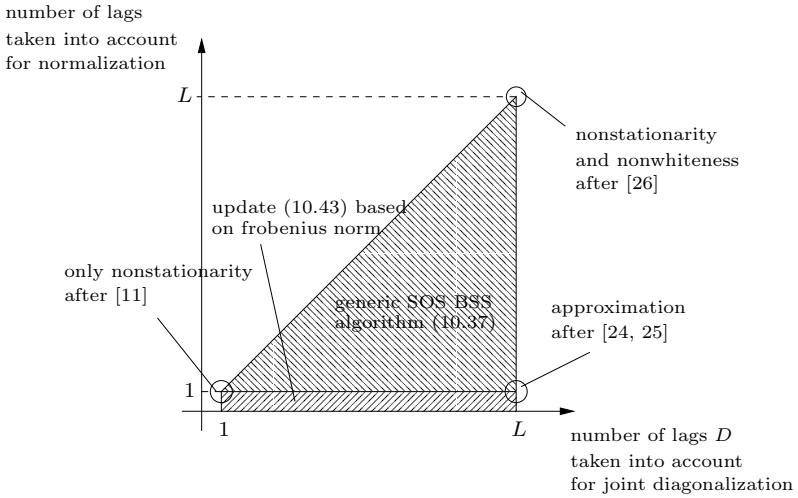


Figure 10.3 Overview of time-domain algorithms based on second-order statistics.

In [24, 25], a time-domain algorithm was presented that copes very well with reverberant acoustic environments. Although it was originally introduced as a heuristic extension of [11] incorporating several time-lags, this algorithm can be directly obtained from (10.39) for $D = L$ by approximating the auto-correlation matrices $\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}(i)$ by the output signal powers, i.e.,

$$\bar{\mathbf{R}}_{\mathbf{y}_q\mathbf{y}_q}(i) = \frac{1}{N}\overline{\mathbf{y}}_q^H(i)\overline{\mathbf{y}}_q(i)\mathbf{I}_{D\times D} \tag{10.41}$$

for $q = 1, \ldots, P$, where $\overline{\mathbf{y}}_q(\cdot)$ denotes the first column of $\mathbf{Y}_q(\cdot)$. Thus, this approximation is comparable to the well-known normalized least mean squares (NLMS) algorithm in supervised adaptive filtering approximating the RLS algorithm [35]. In addition to the reduced computational complexity, we can ensure the Sylvester structure of the update by using the correlation method [40] for calculation of the short-time correlation matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i)$ resulting in Toeplitz matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i)$. The remaining products of Sylvester matrices and Toeplitz matrices in the update equation (10.39) can again be efficiently implemented by a (fast) convolution as was done in [25].

Another very popular subclass of second-order BSS algorithms, particularly for instantaneous mixtures, is based on a cost function using the Frobenius norm $\|\mathbf{A}\|_{\mathrm{F}}^2 = \sum_{i,j} a_{ij}^2$ of a matrix $\mathbf{A} = (a_{ij})$, e.g., [1, 7], [12]-[15]. Analogously to (10.40), this approach may be generalized for

convolutive mixtures to

$$\mathcal{J}_{\mathrm{F}}(m) = \sum_{i=0}^{\infty} \beta(i,m) \left\| \mathbf{Y}^H(i)\mathbf{Y}(i) - \mathrm{bdiag}\, \mathbf{Y}^H(i)\mathbf{Y}(i) \right\|_{\mathrm{F}}^2, \qquad (10.42)$$

which leads (after taking the natural gradient w.r.t. $\mathbf{W}$ in a similar way as in [26]) to the following update equation:

$$\nabla_{\mathbf{W}}^{\mathrm{NG}} \mathcal{J}(m) = 2 \sum_{i=0}^{\infty} \beta(i,m) \mathbf{W} \mathbf{R}_{\mathbf{yy}}(i) \begin{bmatrix} \mathbf{0} & \mathbf{R}_{\mathbf{y}_1\mathbf{y}_2}(i) \\ \mathbf{R}_{\mathbf{y}_2\mathbf{y}_1}(i) & \mathbf{0} \end{bmatrix}. \qquad (10.43)$$

We see that this update equation differs from the more general equation (10.39) mainly in the inherent normalization expressed by the inverse matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_p}^{-1}$. Thus, (10.43) can be regarded as an analogon to the least mean square (LMS) algorithm [35] in supervised adaptive filtering. However, many simulation results have shown that for large filter lengths L, (10.43) is prone to instability, while (10.39) shows a very robust convergence behaviour (see Sect. 5.) even for hundreds or thousands of filter coefficients in BSS for real acoustic environments.

3. GENERIC FREQUENCY-DOMAIN BSS ALGORITHM

Frequency-domain BSS is very popular for convolutive BSS since all techniques originally developed for instantaneous BSS can be applied independently in each frequency bin in the discrete Fourier transform (DFT) domain. Furthermore, the fast Fourier transform (FFT) can be used for an efficient implementation. Such narrowband approaches can be found, e.g., in [1], [3], [6], [9], [12]-[17], [22]. Unfortunately, the permutation problem, which is inherent in BSS (e.g., [1]), may then also appear independently in each frequency bin so that extra measures have to be taken to avoid this *internal* permutation. Additionally, as discussed in Sections 2.3 and 2.4 the products involving Sylvester matrices in the time-domain update equations correspond to linear convolutions. Thus, in the narrowband frequency-domain approach these convolutions become circular ones. The resulting wrap-around effects may limit the separation performance. Based on the above matrix formulation in the time domain, the following derivation of broadband frequency-domain algorithms shows explicitly the relation between time-domain and traditional frequency-domain algorithms, as well as some extensions. In contrast to the narrowband approach this inherently resolves the permutation ambiguity and prevents circular convolution effects in the update equation. Moreover, as in the time-domain, (10.17) also leads to the very desirable property of an inherent stepsize normalization in the

frequency domain which also becomes clear by a link with [17] for the SOS case. As pointed out in the previous section, the conditions for the paramters L, N, and D for the natural gradient adaptation are given by the relations $N > D$ and $1 \leq D \leq L$. Therefore, we may assume $N = L$ without loss of generality for the following derivation.

3.1 GENERAL FREQUENCY-DOMAIN FORMULATION

The matrix formulation (10.11) introduced for the time-domain in Sect. 2. allows a rigorous derivation of the corresponding frequency-domain BSS algorithms. In the frequency domain, the structure of the algorithm depends on the method chosen for estimating the correlation matrices. Here, we consider again the more accurate covariance method [40] (see Sect. 2.4.2). The matrices $\mathbf{X}_p(m)$ and $\mathbf{W}_{pq}$, introduced in Sect. 2.1 are now diagonalized in two steps to obtain frequency-domain representations. In the following, we mark frequency-domain quantities by an underbar. This does, however, not imply that they are simply DFTs of the corresponding time-domain quantities. Each quantity has to be transformed individually. We first consider the $L \times 2L$ Toeplitz matrices $\mathbf{X}_p(m)$.

Step 1: Transformation of Toeplitz matrices into circulant matrices. Any Toeplitz matrix $\mathbf{X}_p$ (10.16) can be transformed, by doubling its size, to a circulant matrix $\mathbf{C}_{X_p}(m)$ [31]. In our case we define the $4L \times 4L$ circulant matrix by taking into account (10.16) by

$$\mathbf{C}_{X_p}(m) = \begin{bmatrix} \mathbf{X}'_p(m-3) & \mathbf{X}_p(m-1) \\ \mathbf{X}_p(m-2) & \mathbf{X}_p(m) \\ \mathbf{X}_p(m-1) & \mathbf{X}'_p(m-3) \\ \mathbf{X}_p(m) & \mathbf{X}_p(m-2) \end{bmatrix}, \tag{10.44}$$

where $\mathbf{X}'_p(m-3)$ is a properly chosen extension ensuring a circular shift of the $4L$ input values in the first column. It follows

$$\mathbf{X}_p(m) = \mathbf{W}^{01_L}_{L\times 4L} \mathbf{C}_{X_p}(m) \mathbf{W}^{1_{2L}0}_{4L\times 2L}, \tag{10.45}$$

where we introduced the windowing matrices

$$\begin{aligned} \mathbf{W}^{01_L}_{L\times 4L} &= [\mathbf{0}_{L\times 3L}, \mathbf{I}_{L\times L}], \\ \mathbf{W}^{1_{2L}0}_{4L\times 2L} &= [\mathbf{I}_{2L\times 2L}, \mathbf{0}_{2L\times 2L}]^T. \end{aligned}$$

This notation follows the conventions listed in Sect. 2.2.

Step 2: Transformation of the circulant matrices into diagonal matrices.

Using the $4L \times 4L$ DFT matrix $\mathbf{F}_{4L\times 4L}$, the circulant matrices are diagonalized as follows:

$$\mathbf{C}_{X_p}(m) = \mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{X}}_p(m)\mathbf{F}_{4L\times 4L}, \tag{10.46}$$

where the diagonal matrices $\underline{\mathbf{X}}_p(m)$ representing the frequency-domain versions of $\mathbf{X}_p(m)$, can be expressed by the first columns of $\mathbf{C}_{X_p}(m)$,

$$\begin{aligned}\underline{\mathbf{X}}_p(m) &= \operatorname{diag}\{\mathbf{F}_{4L\times 4L}[x_p(mL-3L),\ldots,x_p(mL-1),\\ &\quad x_p(mL),x_p(mL+1),\ldots,x_p(mL+L-1)]^T\},\end{aligned} \tag{10.47}$$

i.e., to obtain $\underline{\mathbf{X}}_p(m)$, we transform the concatenated vectors of the current block and three previous blocks of the input signals $x_p(n)$. Here, $\operatorname{diag}\{\mathbf{a}\}$ denotes a square matrix with the elements of vector $\mathbf{a}$ on its main diagonal. Now, (10.45) can be rewritten equivalently as

$$\mathbf{X}_p(m) = \mathbf{W}^{01_L}_{L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{X}}_p(m)\mathbf{F}_{4L\times 4L}\mathbf{W}^{1_{2L}0}_{4L\times 2L}. \tag{10.48}$$

Equations (10.47) and (10.48) exhibit a form that is structurally similar to that of the corresponding counterparts of the well-known (supervised) frequency-domain adaptive filters [31]. However, the major difference here is that we need a transformation length of at least $4L$ instead of $2L$ for an accurate broadband formulation. This should come as no surprise, since in BSS using the covariance method, both convolution and correlation is carried out where both operations double the transformation length.

We now transform the matrices $\mathbf{W}_{pq}$ in the same way as shown above for $\mathbf{X}_p$. Thereby, we obtain

$$\mathbf{W}_{pq} = \mathbf{W}^{1_{2L}0}_{2L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{W}}_{pq}\mathbf{F}_{4L\times 4L}\mathbf{W}^{1_D0}_{4L\times D}, \tag{10.49}$$

where

$$\begin{aligned}\mathbf{W}^{1_D0}_{4L\times D} &= [\mathbf{I}_{D\times D},\mathbf{0}_{D\times(4L-D)}]^T,\\ \mathbf{W}^{1_{2L}0}_{2L\times 4L} &= [\mathbf{I}_{2L\times 2L},\mathbf{0}_{2L\times 2L}] = \left(\mathbf{W}^{1_{2L}0}_{4L\times 2L}\right)^T,\end{aligned}$$

and the frequency-domain representation of the demixing matrix

$$\underline{\mathbf{W}}_{pq} = \operatorname{diag}\{\mathbf{F}_{4L\times 4L}[w_{pq,0},\ldots,w_{pq,L-1},0,\ldots,0]^T\}. \tag{10.50}$$

Equation (10.49) is illustrated in Fig. 10.4. Note that the column vector in (10.50) corresponds to the first column of the $4L \times 4L$ matrix $\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{W}}_{pq}\mathbf{F}_{4L\times 4L}$ in Fig. 10.4. Moreover, it can be seen that the premultiplied transformation $\mathbf{W}^{1_{2L}0}_{2L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}$ in (10.49) is related to the

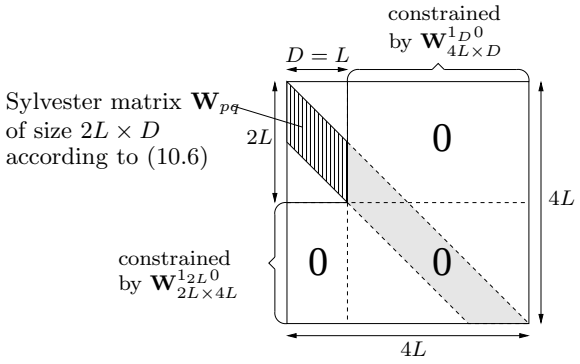


Figure 10.4 Illustration of equation (10.49).

demixing filter taps in the first column of $\mathbf{W}_{pq}$, while the post-multiplied transformation in (10.49), which we denote by

$$\mathbf{L}^{1_D 0}_{4L\times D} = \mathbf{F}_{4L\times 4L}\mathbf{W}^{1_D 0}_{4L\times D}, \tag{10.51}$$

is related to the introduction of D time-lags (see also Sect. 3.3.1). Combining all channels, we obtain from (10.48) and (10.49)

$$\begin{aligned}\mathbf{X}(m) &= \mathbf{W}^{01_L}_{L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{X}}(m) \\ &\quad \cdot\mathrm{bdiag}\{\mathbf{F}_{4L\times 4L}\mathbf{W}^{1_{2L}0}_{4L\times 2L},\ldots,\mathbf{F}_{4L\times 4L}\mathbf{W}^{1_{2L}0}_{4L\times 2L}\}, \quad (10.52)\\ \mathbf{W} &= \mathrm{bdiag}\{\mathbf{W}^{1_{2L}0}_{2L\times 4L}\mathbf{F}^{-1}_{4L\times 4L},\ldots,\mathbf{W}^{1_{2L}0}_{2L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\}\underline{\mathbf{W}}\mathbf{L}, \quad (10.53)\end{aligned}$$

where $\underline{\mathbf{X}}(m)$ and $\underline{\mathbf{W}}$ are defined analogously to (10.13) and (10.10), respectively. $\mathbf{L}$ denotes the $4LP \times DP$ matrix

$$\mathbf{L} = \mathrm{bdiag}\{\mathbf{L}^{1_D 0}_{4L\times D},\ldots,\mathbf{L}^{1_D 0}_{4L\times D}\}. \tag{10.54}$$

From (10.11),(10.52), and (10.53) we further obtain

$$\mathbf{Y}(m) = \mathbf{W}^{01_L}_{L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{Y}}(m)\mathbf{L}, \tag{10.55}$$

with

$$\underline{\mathbf{Y}}(m) = \underline{\mathbf{X}}(m)\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}, \tag{10.56}$$

and the time-domain constraints

$$\begin{aligned}\mathbf{G}^{1_{2L}0}_{4LP\times 4LP} &= \mathrm{bdiag}\left\{\mathbf{G}^{1_{2L}0}_{4L\times 4L},\ldots,\mathbf{G}^{1_{2L}0}_{4L\times 4L}\right\}, \\ \mathbf{G}^{1_{2L}0}_{4L\times 4L} &= \mathbf{F}_{4L\times 4L}\mathbf{W}^{1_{2L}0}_{4L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}, \\ \mathbf{W}^{1_{2L}0}_{4L\times 4L} &= \mathbf{W}^{1_{2L}0}_{4L\times 2L}\mathbf{W}^{1_{2L}0}_{2L\times 4L} \\ &= \begin{bmatrix}\mathbf{I}_{2L\times 2L} & \mathbf{0}_{2L\times 2L}\\ \mathbf{0}_{2L\times 2L} & \mathbf{0}_{2L\times 2L}\end{bmatrix}.\end{aligned}$$

To formulate the cost function (10.17) in the frequency domain, we first need to express it *equivalently* using matrices $\mathbf{Y}_p$, $p = 1, \ldots, P$. This inevitably leads to the introduction of pdfs which depend on matrices in their arguments. In general, such pdfs are determined by a fourth-order tensor which contains all cross-relations between the matrix elements. However, due to the Toeplitz structure of the matrices $\mathbf{Y}_p$ a redundancy is introduced which neither appears in the cost function (10.17) nor leads to any improved results compared to (10.17).

Thus, we can replace the tensor by a matrix containing only the desired information on the cross-relations between the D time-lags. This yields the following equivalent representation of (10.17):

$$\begin{aligned}\mathcal{J}(m) &= -\sum_{i=0}^{\infty}\beta(i,m)\frac{1}{N}\left\{\log\left(\tilde{p}_{1,N\times D}(\mathbf{Y}_1(i))\cdot\ldots\cdot\tilde{p}_{P,N\times D}(\mathbf{Y}_P(i))\right)\right. \\ &\quad \left.-\log\left(\tilde{p}_{N\times PD}(\mathbf{Y}_1(i),\ldots,\mathbf{Y}_P(i))\right)\right\}, \qquad (10.57)\end{aligned}$$

with the *auxiliary pdfs* which we define here by

$$\tilde{p}_{p,N\times D}(\mathbf{Y}_p(i)) = \prod_{j=0}^{N-1}\hat{p}_{p,D}(\mathbf{y}_p(i,j)), \qquad (10.58)$$

$$\tilde{p}_{N\times PD}(\mathbf{Y}_1(i),\ldots,\mathbf{Y}_P(i)) = \prod_{j=0}^{N-1}\hat{p}_{PD}(\mathbf{y}_1(i,j),\ldots,\mathbf{y}_P(i,j)), \qquad (10.59)$$

showing the relation to the multivariate pdfs. The equivalence to (10.17) can easily be verified by inserting (10.58) and (10.59) in (10.57). The advantage of introducing such auxiliary pdfs is that they can formally be handled like standard pdfs where the rows of the matrix in their argument are mutually statistically independent. This allows a compact representation of the following equations.

To proceed with the derivation, we take the gradient of (10.57) w.r.t. the frequency-domain coefficient matrix $\underline{\mathbf{W}}$. This is done analogously to the time-domain derivation of (10.19). However, (10.55) and (10.56) have to be taken into account by using the chain rule for matrices [41]. This finally leads to the following gradient for the frequency-domain update:

$$\begin{aligned}\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m) &= \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\left\{\mathbf{G}_{4LP\times 4LP}^{1_{2L}0}\underline{\mathbf{X}}^H(i)\underline{\mathbf{\Phi}}(\underline{\mathbf{Y}}(i))\right. \\ &\quad \left.-\mathbf{L}\left(\mathbf{L}^H\underline{\mathbf{W}}^H\mathbf{L}\right)^{-1}\mathbf{L}^H\right\} \qquad (10.60)\end{aligned}$$

with the frequency-domain score function

$$\underline{\mathbf{\Phi}}(\underline{\mathbf{Y}}(i)) = \left[-\frac{\frac{\partial \tilde{\underline{p}}_{1,4L\times 4L}(\underline{\mathbf{Y}}_1(i))}{\partial \underline{\mathbf{Y}}_1(i)}}{\tilde{\underline{p}}_{1,4L\times 4L}(\underline{\mathbf{Y}}_1(i))}, \ldots, -\frac{\frac{\partial \tilde{\underline{p}}_{P,4L\times 4L}(\underline{\mathbf{Y}}_P(i))}{\partial \underline{\mathbf{Y}}_P(i)}}{\tilde{\underline{p}}_{P,4L\times 4L}(\underline{\mathbf{Y}}_P(i))} \right]. \tag{10.61}$$

Note that the pdf $\tilde{\underline{p}}_{q,4L\times 4L}(\underline{\mathbf{Y}}_q(i))$ of the frequency-domain matrix $\underline{\mathbf{Y}}_q(i)$ is obtained by transforming the pdf $\tilde{p}_{q,N\times D}(\mathbf{Y}_q(i))$ of time-domain variables using (10.55). We will go into the precise formulation of $\tilde{\underline{p}}_{q,4L\times 4L}(\underline{\mathbf{Y}}_q(i))$ within the scope of the special cases treated in Sect. 3.3. Equations (10.60) and (10.61) are the generic frequency-domain counterparts of (10.19) and (10.20), respectively, and may be equivalently used for coefficient adaptation.

As in the time-domain, we need not calculate the entire coefficient matrix $\underline{\mathbf{W}}$ explicitly due to the redundancy introduced by the Sylvester structure in the time domain, and the diagonal structure of the submatrices $\underline{\mathbf{W}}_{pq}$ in the frequency domain, respectively. While the structure of matrix $\underline{\mathbf{W}}$ is independent of D, matrix $\mathbf{L}$ introduces the number of time-lags taken into account by the cost function, as shown by (10.51) and (10.55) (see also Fig. 10.4). To calculate the separated output signals, given a demixing matrix $\underline{\mathbf{W}}$, we need to pick the first column of $\mathbf{Y}$ in (10.55) (the other columns were introduced in (10.11) for including multiple time-lags in the cost function). This is done by using $\mathbf{L} = \mathbf{L}_\mathrm{I} = \mathrm{bdiag}\{\mathbf{1}_{4L\times 1}, \cdots, \mathbf{1}_{4L\times 1}\}$ in (10.55). Then, $\underline{\mathbf{W}}\mathbf{L}$ in that equation becomes a $4LP \times P$ matrix $\underline{\mathbf{W}}'$ whose columns correspond to the diagonals of $\underline{\mathbf{W}}$. As a general rule,

$$\underline{\mathbf{W}}' = \underline{\mathbf{W}}\mathbf{L}_\mathrm{I}, \tag{10.62}$$

and building diagonal submatrices $\underline{\mathbf{W}}_{pq}$ of $\underline{\mathbf{W}}$ using the entries of $\underline{\mathbf{W}}'$, transforms the two equivalent representations into each other. Thus, to formally obtain the update of $\underline{\mathbf{W}}'$ needed for the output signal calculation, we post-multiply (10.60) by $\mathbf{L}_\mathrm{I}$, both simplifying the calculation of (10.60), and enforcing the diagonal structure of $\underline{\mathbf{W}}_{pq}$ during the adaptation. This simplification results from the fact that we only have to operate with vectors rather than matrices for each channel when constructing the update equation from the right to the left in a practical realization.

In addition to the diagonal structure of $\underline{\mathbf{W}}$, we have to ensure the Sylvester structure in the time domain as noted previously. As can be seen in Fig. 10.4, (10.50) determines the first column, and thus the whole $4L \times 4L$ Sylvester matrix. In other words, we have to ensure that the time-domain column vector in (10.50) contains only L filter coefficients

and $3L$ zeros. Therefore, the gradient (10.60) has to be constrained by $\mathbf{G}^{1_L 0}_{4LP\times 4LP}$. Together with (10.62) this leads to

$$\Delta\underline{\mathbf{W}}'(m) = \mathbf{G}^{1_L 0}_{4LP\times 4LP}\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m)\mathbf{L}_{\mathrm{I}}, \tag{10.63}$$

which may again be implemented efficiently from the right to the left. Then the constraint $\mathbf{G}^{1_L 0}_{4LP\times 4LP}$ reduces to channel-wise inverse FFT, windowing (see also Sect. 3.3.2), and FFT operations.

3.2 NATURAL GRADIENT IN THE FREQUENCY DOMAIN

In Sect. 2.3, it has been shown that the natural gradient for convolutive mixtures introduced there for the time domain yields equivariant adaptation algorithms, i.e., the evolutionary behaviour of

$$\mathbf{C}(m) = \mathbf{H}\mathbf{W}(m) \tag{10.64}$$

and $\Delta\mathbf{C}(m) = \mathbf{H}\Delta\mathbf{W}(m)$ does not explicitly depend on $\mathbf{H}$ in (10.26).

In this section, we investigate how this formulation of the natural gradient transforms into the frequency domain. To begin with, we start by the following approach containing arbitrary matrices $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, and $\mathbf{A}_4$ of proper size:

$$\nabla^{\mathrm{NG}}_{\underline{\mathbf{W}}}\mathcal{J} = \mathbf{A}_1\underline{\mathbf{W}}\mathbf{A}_2\mathbf{A}_3\underline{\mathbf{W}}^H\mathbf{A}_4\nabla_{\underline{\mathbf{W}}}\mathcal{J}. \tag{10.65}$$

Now, our task is to determine the four matrices $\mathbf{A}_i$ such that the resulting coefficient update exhibits desired properties.

As a first condition, matrix $\mathbf{A}_1\underline{\mathbf{W}}\mathbf{A}_2\mathbf{A}_3\underline{\mathbf{W}}^H\mathbf{A}_4$ in (10.65) must be positive definite, i.e., all its eigenvalues must be positive to ensure convergence [39]. This determines matrices $\mathbf{A}_3$, and $\mathbf{A}_4$ up to a positive scalar constant, which can be absorbed in the stepsize, so that we obtain

$$\nabla^{\mathrm{NG}}_{\underline{\mathbf{W}}}\mathcal{J} = \mathbf{A}_1\underline{\mathbf{W}}\mathbf{A}_2\mathbf{A}_2^H\underline{\mathbf{W}}^H\mathbf{A}_1^H\nabla_{\underline{\mathbf{W}}}\mathcal{J}. \tag{10.66}$$

As the second, and most important condition, it is required that the equivariance property is fulfilled. Combining (10.64) with (10.53), we obtain a relation between $\mathbf{C}$ and the frequency-domain coefficients $\underline{\mathbf{W}}$,

$$\mathbf{C} = \mathbf{H}\,\mathrm{bdiag}\{\cdots\}\underline{\mathbf{W}}\mathbf{L}, \tag{10.67}$$

and analogously

$$\begin{aligned}\Delta\mathbf{C} &= \mathbf{H}\,\mathrm{bdiag}\{\cdots\}\Delta\underline{\mathbf{W}}\mathbf{L}\\ &= \mathbf{H}\,\mathrm{bdiag}\{\cdots\}\nabla^{\mathrm{NG}}_{\underline{\mathbf{W}}}\mathcal{J}\mathbf{L}.\end{aligned} \tag{10.68}$$

As in the time domain (see (10.26)), it is required that (10.68) in combination with the natural gradient (10.66) can be expressed by $\mathbf{C}$ defined in (10.67), and therefore does not explicitly depend on $\mathbf{H}$. This leads to the claim

$$\Delta\mathbf{C} = \underbrace{\mathbf{H}\,\mathrm{bdiag}\{\cdots\}\mathbf{A}_1\underline{\mathbf{W}}\mathbf{A}_2}_{=\mathbf{C}}\,\mathbf{A}_2^H\underline{\mathbf{W}}^H\mathbf{A}_1^H\nabla_{\underline{\mathbf{W}}}\mathcal{J}\mathbf{L}, \tag{10.69}$$

and a comparison of (10.69) with (10.67) yields the matrices

$$\mathbf{A}_1 = \mathbf{G}^{1_{2L}0}_{4LP\times 4LP},\ \mathbf{A}_2 = \mathbf{L}. \tag{10.70}$$

Note that $\mathbf{A}_1 = \mathbf{I}$ is not the general solution. This can be verified by inserting (10.70) in (10.69), and considering the argument of $\mathrm{bdiag}\{\cdot\}$ acording to (10.53).

Finally, we obtain the natural gradient

$$\begin{aligned}\nabla^{\mathrm{NG}}_{\underline{\mathbf{W}}}\mathcal{J} &= \mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}\mathbf{L}\mathbf{L}^H\underline{\mathbf{W}}^H\left(\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\right)^H\nabla_{\underline{\mathbf{W}}}\mathcal{J}\\ &= \mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}\mathbf{L}\mathbf{L}^H\underline{\mathbf{W}}^H\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\nabla_{\underline{\mathbf{W}}}\mathcal{J},\end{aligned} \tag{10.71}$$

and together with (10.60) it follows the coefficient update

$$\nabla^{\mathrm{NG}}_{\underline{\mathbf{W}}}\mathcal{J}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}\mathbf{L}\mathbf{L}^H\left\{\underline{\mathbf{Y}}^H(i)\underline{\mathbf{\Phi}}(\underline{\mathbf{Y}}(i)) - \mathbf{I}\right\}. \tag{10.72}$$

Note that in this equation the natural gradient shows again the convenient property of avoiding one matrix inversion. Formally, as in Sect. 3.1, (10.63) can be used to obtain $\Delta\underline{\mathbf{W}}'$.

3.3 SPECIAL CASES AND LINKS TO KNOWN FREQUENCY-DOMAIN ALGORITHMS

The generic gradient (10.60) and generic natural gradient (10.72), respectively, exhibit three types of quantities that fully specify practical realizations which follow as special cases. These quantities can be related to the three fundamental signal properties, as shown in Table 10.1.

3.3.1 The Constraints and the Internal Permutation Problem in Frequency-Domain BSS. Two types of constraints appear in the gradient (10.60) and in the natural gradient update (10.72):

- The matrices $\mathbf{G}^{\cdots}_{\cdots}$ in (10.56) and in the update equations are mainly responsible for preventing decoupling of the individual frequency

quantity	related to	examples in
constraints $\mathbf{G}_{\cdots}^{\cdots}$, $\mathbf{L}$	nonwhiteness	Sect. 3.3.1, 3.3.2
score function $\underline{\mathbf{\Phi}}(\cdot)$	nongaussianity	Sect. 3.3.1, 3.3.3
weighting function $\beta(\cdot)$	nonstationarity	Sect. 4.

Table 10.1 Quantities defining a certain frequency-domain algorithm.

components, and thus avoiding the internal permutation among the different frequency bins and circular convolution effects.

- Matrix $\mathbf{L}$ has two different functions: on the one hand, it allows joint diagonalization over D time-lags, and on the other hand, it acts as time-domain constraint similar to the matrices $\mathbf{G}_{\cdots}^{\cdots}$ (see Fig. 10.4).

Note that the constraints $\mathbf{G}_{\cdots}^{\cdots}$ and $\mathbf{L}$ also appear in the score function $\underline{\mathbf{\Phi}}(\cdot)$ as can be seen later (e.g., Sect. 3.3.3) in more detail.

Concerning matrix $\mathbf{L}$ we can distinguish between four different cases:

a) $D < L$: As in the time domain, this choice allows the exploitation of the nonwhiteness property with up to D time-lags.

b) $D = L$: This is the optimum case as in the time domain.

c) $D > L$: This choice is not meaningful in the time domain. In the frequency domain, however, we can choose D up to the transformation length $4L$ due to the introduced circulant matrix, as shown in Fig. 10.4. For $D > L$ the time-domain constraint is relaxed, which may also lead to a suboptimum solution.

d) $D = 4L$: According to Fig. 10.4 this corresponds to the traditional narrowband approximation (apart from constraints $\mathbf{G}_{\cdots}^{\cdots}$) so that all matrices $\mathbf{L}$ cancel out in the update equations, which can also be verified using (10.51).

Case d), i.e., neglecting matrix $\mathbf{L}$ in (10.60) yields a simplified gradient

$$\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\left\{\mathbf{G}_{4LP\times 4LP}^{1_{2L}0}\underline{\mathbf{X}}^{H}(i)\underline{\mathbf{\Phi}}(\underline{\mathbf{Y}}(i)) - \underline{\mathbf{W}}^{-H}\right\}, \tag{10.73}$$

where $^{-H}$ denotes the inverse of a conjugate transpose of a matrix, and from (10.72), we obtain a simplified natural gradient

$$\nabla_{\underline{\mathbf{W}}}^{\mathrm{NG}}\mathcal{J}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\mathbf{G}_{4LP\times 4LP}^{1_{2L}0}\underline{\mathbf{W}}\left\{\underline{\mathbf{Y}}^{H}(i)\underline{\mathbf{\Phi}}(\underline{\mathbf{Y}}(i)) - \mathbf{I}\right\}. \tag{10.74}$$

Note that these expressions still largely avoid the well-known internal permutation problem of frequency-domain BSS using the constraints $\mathbf{G}_{\cdots}^{\cdots}$ in the calculation of $\underline{\mathbf{Y}}$ in (10.56) and in the update equations obtained from inserting (10.73) or (10.74) in (10.63).

By additionally approximating $\mathbf{G}_{\cdots}^{\cdots}$ as scaled identity matrices [31] in the gradients, the submatrices $\underline{\mathbf{Y}}_q$ of $\underline{\mathbf{Y}}$ in (10.73) and (10.74) also become diagonal, as illustrated in Fig. 10.5. Moreover, the frequency-domain multivariate score function $\underline{\mathbf{\Phi}}(\cdot)$ can be decomposed to frequency bin selective score functions $\underline{\mathbf{\Phi}}^{(\nu)}(\cdot)$ containing only univariate pdfs $\underline{\tilde{p}}_{p,1}^{(\nu)}(\cdot)$ for channel p, i.e.,

$$\underline{\mathbf{\Phi}}^{(\nu)}(\underline{\mathbf{Y}}^{(\nu)}(i)) = \left[-\frac{\frac{\partial \underline{\tilde{p}}_{1,1}^{(\nu)}(\underline{Y}_1^{(\nu)}(i))}{\partial \underline{Y}_1^{(\nu)}(i)}}{\underline{\tilde{p}}_{1,1}^{(\nu)}(\underline{Y}_1^{(\nu)}(i))}, \ldots, -\frac{\frac{\partial \underline{\tilde{p}}_{P,1}^{(\nu)}(\underline{Y}_P^{(\nu)}(i))}{\partial \underline{Y}_P^{(\nu)}(i)}}{\underline{\tilde{p}}_{P,1}^{(\nu)}(\underline{Y}_P^{(\nu)}(i))} \right], \tag{10.75}$$

where $\nu = 0, \ldots, 4L-1$ denotes the frequency bin index. This approxi-

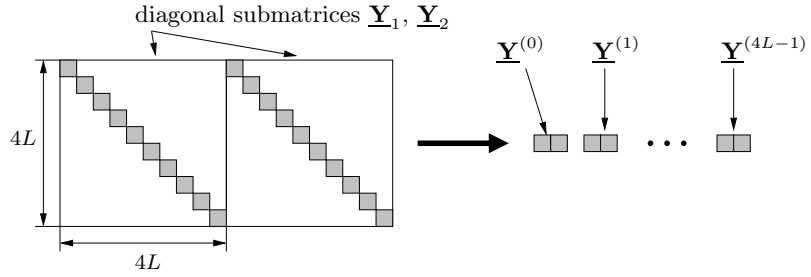


Figure 10.5 Illustration of bin-wise decomposition for the 2-channel case.

mation combined with $D = 4L$ (case d) from above) corresponds to the traditional narrowband approach. Only in this case both equations can be decomposed into its frequency components, i.e., we can equivalently write

$$\nabla_{\underline{\mathbf{W}}} \mathcal{J}^{(\nu)}(m) = \frac{2}{N} \sum_{i=0}^{\infty} \beta(i,m) \left\{ \left(\underline{\mathbf{X}}^{(\nu)}(i) \right)^H \underline{\mathbf{\Phi}}^{(\nu)}(\underline{\mathbf{Y}}^{(\nu)}(i)) - \left(\underline{\mathbf{W}}^{(\nu)} \right)^{-H} \right\}, \tag{10.76}$$

and

$$\nabla_{\underline{\mathbf{W}}}^{\mathrm{NG}} \mathcal{J}^{(\nu)}(m) = \frac{2}{N} \sum_{i=0}^{\infty} \beta(i,m) \underline{\mathbf{W}}^{(\nu)} \left\{ \left(\underline{\mathbf{Y}}^{(\nu)}(i) \right)^H \underline{\mathbf{\Phi}}^{(\nu)}(\underline{\mathbf{Y}}^{(\nu)}(i)) - \mathbf{I} \right\}, \tag{10.77}$$

respectively. In contrast to $\underline{\mathbf{W}}$ and $\underline{\mathbf{Y}}$ in (10.60), (10.72) which are $4LP \times 4LP$ and $4L \times 4LP$ matrices, respectively, the corresponding

matrices $\underline{\mathbf{W}}^{(\nu)}$ and $\underline{\mathbf{Y}}^{(\nu)}$ in (10.76), (10.77) are only of dimensions $P \times P$ and $1 \times P$, respectively.

The approximation (10.77) of the natural gradient corresponds to the ICA narrowband approach originally proposed by Smaragdis [22] as an extension of the information maximization approach [23].

Note that the nonholonomic version of the natural gradient (10.77) can be obtained similarly to the time domain by replacing matrix $\mathbf{I}$ with $\operatorname{diag}\left\{\left(\underline{\mathbf{Y}}^{(\nu)}(i)\right)^H \underline{\mathbf{\Phi}}^{(\nu)}\left(\underline{\mathbf{Y}}^{(\nu)}(i)\right)\right\}$.

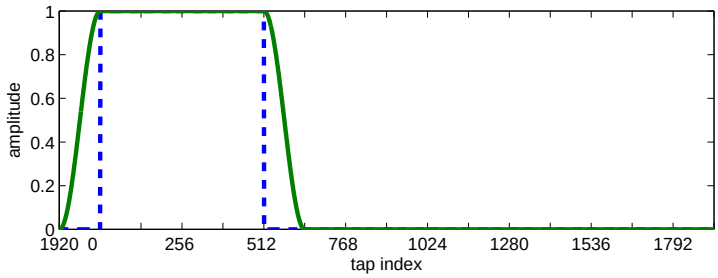
To derive the update equations from the approximated gradients, we apply again (10.63) which contains another constraint $\mathbf{G}_{4LP\times 4LP}^{1_L 0}$ transforming the filter coefficients back into the time domain, zeroing the last $3L$ values, and transforming the result back to the frequency domain. Thus, even if (10.76) and (10.77) can be efficiently computed in a bin-selective manner, this constraint prevents a complete decoupling of the frequency-components in the update equations. This procedure appears similarly in the well-known "constrained frequency-domain adaptive filtering" in the supervised case [35],[31]. In BSS, this theoretically founded mechanism largely eliminates the internal permutation problem in a simple way. It was first heuristically introduced in [22], and also in [14]. A more detailed experimental examination on this constraint was reported in [42] confirming that the ratio between filter length L and transformation length $4L$ - as obtained here analytically - yields optimum separation performance. However, due to the omission of the other constraints in the approximated gradients we will not perfectly remove the permutation ambiguity as observed experimentally in [42]. Traditional narrowband approaches also neglecting the time-domain constraint in (10.63) need additional measures for solving the permutation problem (e.g., [13], [43]).

3.3.2 Alternative Approximations of the Constraints. The generic algorithm (10.72) with its constraint matrices $\mathbf{G}_{\ldots}^{\ldots}$ suggests alternative efficient approximations to allow improved tradeoffs between the exact broadband approach (large computational complexity) and the narrowband approach (internal permutation ambiguity) by choosing certain efficient approximations of the constraints.

Generally, we can distinguish between approximations depending on the block index and approximations within each block. One example for the former class is to simply apply the constraints periodically for a reduced number of blocks which has also been proposed for the supervised case [44].

The other class is based on efficient approximations of the rectangular window appearing in the constraints. This is done by smoothing the rectangular window (Fig 10.6a) so that its frequency-domain representation can be well-described by a small number of coefficients (Fig 10.6b). Having such a representation, it is often more efficient to directly apply

a) Time-domain window function:



b) Frequency-domain representation:

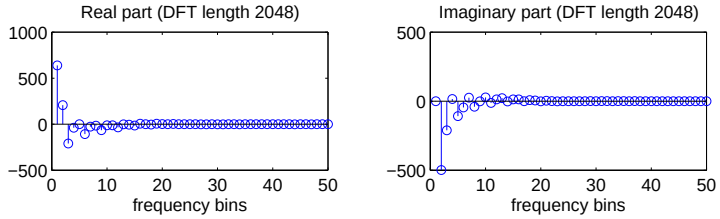


Figure 10.6 Illustration of a smoothed window function for $L = 512$, i.e., transformation length 2048. Note that the window functions are circular.

the convolution operation in the frequency-domain instead of going back and forth between the time domain and frequency domain. This general idea has been discussed earlier for supervised adaptive filtering [45], especially after the introduction of the supervised generic frequency-domain framework [46, 31], see, e.g., [47, 48]. There are several variations possible to design the smoothed window (see also filter design techniques) [49]. However, the smoothed window has to be flat within the length L (e.g., Tukey window [49]). Otherwise compensation terms are necessary [48]. In BSS, a similar windowing has been proposed heuristically in [50].

3.3.3 Generic Frequency-Domain BSS based on SOS. As shown for the time domain, we derive a generic SOS algorithm by considering Gaussian pdfs. The corresponding Gaussian auxiliary pdf for

matrices in the sense described above is obtained using (10.58). It follows

$$\tilde{p}_{p,N\times D}(\mathbf{Y}_p(i)) = \frac{1}{\sqrt{((2\pi)^D \det \mathrm{R}_{\mathbf{y}_p\mathbf{y}_p}(i))^N}} e^{-\frac{1}{2}\mathrm{tr}\left\{\mathbf{Y}_p(i)\mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p}(i)\mathbf{Y}_p^H(i)\right\}}. \tag{10.78}$$

Transforming this Gaussian pdf into the pdf for the corresponding frequency domain variables $\underline{\mathbf{Y}}_p$ gives again a Gaussian. Using (10.55) and (10.56) we obtain

$$\begin{aligned}\underline{\tilde{p}}_{p,4L\times 4L}(\underline{\mathbf{Y}}_p(i)) \;\propto\; \exp\Bigg\{-\frac{1}{2}\mathrm{tr}\Big\{&\mathbf{W}^{01_L}_{L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{Y}}_p(i)\mathbf{L}^{1_D 0}_{L\times D}\mathbf{R}^{-1}_{\mathbf{y}_p\mathbf{y}_p}(i)\\ &\cdot(\mathbf{L}^{1_D 0}_{L\times D})^H\underline{\mathbf{Y}}_p^H(i)\mathbf{F}_{4L\times 4L}\mathbf{W}^{01_L}_{4L\times L}\Big\}\Bigg\},\end{aligned} \tag{10.79}$$

where

$$\begin{aligned}
\mathbf{R}_{\mathbf{yy}}(i) &= \frac{1}{N}\mathbf{Y}^H(i)\mathbf{Y}(i) = \frac{1}{N}\mathbf{L}^H\mathbf{S}_{\mathbf{yy}}(i)\mathbf{L},\\
\mathbf{S}_{\mathbf{yy}}(i) &= \underline{\mathbf{Y}}^H(i)\mathbf{G}^{01_L}_{4L\times 4L}\underline{\mathbf{Y}}(i)\\
&= \underline{\mathbf{W}}^H\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\mathbf{S}_{\mathbf{xx}}(i)\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}, \qquad (10.80)\\
\mathbf{S}_{\mathbf{xx}}(i) &= \underline{\mathbf{X}}^H(i)\mathbf{G}^{01_L}_{4L\times 4L}\underline{\mathbf{X}}(i), \qquad (10.81)\\
\mathbf{G}^{01_L}_{4L\times 4L} &= \mathbf{F}_{4L\times 4L}\mathbf{W}^{01_L}_{4L\times 4L}\mathbf{F}^{-1}_{4L\times 4L},\\
\mathbf{W}^{01_L}_{4L\times 4L} &= \mathbf{W}^{01_L}_{4L\times L}\mathbf{W}^{01_L}_{L\times 4L}\\
&= \begin{bmatrix}\mathbf{0}_{3L\times 3L} & \mathbf{0}_{3L\times L}\\ \mathbf{0}_{L\times 3L} & \mathbf{I}_{L\times L}\end{bmatrix}.
\end{aligned}$$

The resulting score function (10.61) reads

$$\underline{\mathbf{\Phi}}(\underline{\mathbf{Y}}(i)) = -\mathbf{G}^{01_L}_{4L\times 4L}\underline{\mathbf{Y}}(i)\mathbf{L}\cdot\mathrm{bdiag}^{-1}\left(\mathbf{L}^H\mathbf{S}_{\mathbf{yy}}(i)\mathbf{L}\right)\mathbf{L}^H. \tag{10.82}$$

This leads to

$$\begin{aligned}\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m) &= \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\mathbf{S}_{\mathbf{xy}}\mathbf{L}\\ &\quad\cdot(\mathbf{L}^H\mathbf{S}_{\mathbf{yy}}\mathbf{L})^{-1}\mathbf{L}^H\left\{\mathbf{S}_{\mathbf{yy}} - \mathrm{bdiag}\,\mathbf{S}_{\mathbf{yy}}\right\}\mathbf{L}\\ &\quad\cdot\mathrm{bdiag}^{-1}\left(\mathbf{L}^H\mathbf{S}_{\mathbf{yy}}\mathbf{L}\right)\cdot\mathbf{L}^H,\end{aligned} \tag{10.83}$$

where

$$\mathbf{S}_{\mathbf{xy}}(i) = \mathbf{S}_{\mathbf{xx}}(i)\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}. \tag{10.84}$$

Finally with (10.71), we obtain the natural gradient

$$\begin{aligned}\nabla^{\mathrm{NG}}_{\underline{\mathbf{W}}}\mathcal{J}(m) &= \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\mathbf{G}^{1_{2L}0}_{4LP\times 4LP}\underline{\mathbf{W}}\mathbf{L}\mathbf{L}^H\left\{\mathbf{S}_{\mathbf{yy}} - \mathrm{bdiag}\,\mathbf{S}_{\mathbf{yy}}\right\}\mathbf{L}\\ &\quad\cdot\mathrm{bdiag}^{-1}\left(\mathbf{L}^H\mathbf{S}_{\mathbf{yy}}\mathbf{L}\right)\mathbf{L}^H.\end{aligned} \tag{10.85}$$

Equations (10.83) and (10.85) are the SOS analoga to (10.60) and (10.72). In the same way as shown here for the Gaussian case, we could also analogously define auxiliary pdfs for SIRPs (see Sect. 2.4). Note that in (10.85) the natural gradient shows again the convenient property of avoiding one matrix inversion. Formally, as in Sect. 3.1, (10.63) can be used to obtain $\Delta\underline{\mathbf{W}}'$.

3.3.4 Approximation of the Generic Frequency-Domain BSS Based on SOS. In the SOS case we can apply the same approximation steps as discussed for the HOS case in Sect. 3.3.1. By analogously neglecting matrix $\mathbf{L}$ in (10.83) and (10.85) we obtain a simplified gradient and a simplified natural gradient, respectively, which still largely avoid the internal permutation problem of frequency-domain BSS.

The narrowband approach is obtained by additionally approximating $\mathbf{G}_{\cdots}^{\cdots}$ as scaled identity matrices [31] yielding gradients which can be decomposed in its frequency components, i.e.,

$$\nabla_{\underline{\mathbf{W}}}\mathcal{J}^{(\nu)}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\mathbf{S}_{\mathbf{xy}}^{(\nu)}\left(\mathbf{S}_{\mathbf{yy}}^{(\nu)}\right)^{-1}\left\{\mathbf{S}_{\mathbf{yy}}^{(\nu)} - \operatorname{diag}\mathbf{S}_{\mathbf{yy}}^{(\nu)}\right\}\operatorname{diag}^{-1}\mathbf{S}_{\mathbf{yy}}^{(\nu)} \tag{10.86}$$

and

$$\nabla_{\underline{\mathbf{W}}}^{\mathrm{NG}}\mathcal{J}^{(\nu)}(m) = \frac{2}{N}\sum_{i=0}^{\infty}\beta(i,m)\underline{\mathbf{W}}^{(\nu)}\left\{\mathbf{S}_{\mathbf{yy}}^{(\nu)} - \operatorname{diag}\mathbf{S}_{\mathbf{yy}}^{(\nu)}\right\}\operatorname{diag}^{-1}\mathbf{S}_{\mathbf{yy}}^{(\nu)}, \tag{10.87}$$

respectively, where $\nu = 0,\ldots,4L-1$ denotes the frequency bins. In contrast to $\mathbf{S}_{\mathbf{xy}}$, $\mathbf{S}_{\mathbf{yy}}$, and $\underline{\mathbf{W}}$ in (10.83), (10.85) which are $4LP \times 4LP$ matrices each, the corresponding matrices $\mathbf{S}_{\mathbf{xy}}^{(\nu)}$, $\mathbf{S}_{\mathbf{yy}}^{(\nu)}$, and $\underline{\mathbf{W}}^{(\nu)}$ in (10.86), (10.87) are only of dimension $P \times P$.

To obtain the update equations from the approximated gradients, we apply again (10.63) preventing the complete decoupling by the constraint $\mathbf{G}_{4LP\times 4LP}^{1_L 0}$.

The approximated coefficient update (10.86) is directly related to some well-known frequency-domain BSS algorithms. In [16], an algorithm that is similar to (10.86) was derived by directly optimizing a cost function similar to the one in [10] in a bin-wise manner. More recently, Fancourt and Parra proposed in [17] to apply the magnitude-squared coherence

$$|\gamma_{y_p y_q}^{(\nu)}(m)|^2 = \frac{|S_{y_p y_q}^{(\nu)}(m)|^2}{S_{y_p y_p}^{(\nu)}(m) S_{y_q y_q}^{(\nu)}(m)}, \tag{10.88}$$

$p,q \in \{1,2\}$ as a cost function for frequency-domain BSS, where $S_{y_p y_q}^{(\nu)}(m)$ denotes the (p,q)-th element of $\mathbf{S}_{\mathbf{yy}}^{(\nu)}(m)$, i.e., the power spec-

tral density in the ν-th bin and block m. The coherence (10.88) has the very desirable property that

$$0 \leq |\gamma_{y_p y_q}^{(\nu)}(m)|^2 \leq 1, \tag{10.89}$$

which directly translates into an inherent stepsize normalization of the corresponding update equation [17]. In particular, $|\gamma_{y_1 y_2}^{(\nu)}(m)|^2 = 0$ if $\mathbf{y}_1$ and $\mathbf{y}_2$ are orthogonal, and $|\gamma_{y_1 y_2}^{(\nu)}(m)|^2 = 1$ when $\mathbf{y}_1 = a\mathbf{y}_2$ for any non-zero complex number a.

Comparing the update equation (10.86) with that derived in [17], we see that an additional approximation of $\left(\mathbf{S}_{\mathbf{yy}}^{(\nu)}\right)^{-1}$ as a diagonal matrix was used in [17], which results in

$$\begin{aligned} \nabla_{\underline{\mathbf{W}}} \mathcal{J}^{(\nu)}(m) &= \frac{2}{N} \sum_{i=0}^{\infty} \beta(i,m) \mathbf{S}_{\mathbf{xy}}^{(\nu)} \operatorname{diag}^{-1} \mathbf{S}_{\mathbf{yy}}^{(\nu)} \\ &\quad \cdot \left\{ \mathbf{S}_{\mathbf{yy}}^{(\nu)} - \operatorname{diag} \mathbf{S}_{\mathbf{yy}}^{(\nu)} \right\} \operatorname{diag}^{-1} \mathbf{S}_{\mathbf{yy}}^{(\nu)}. \end{aligned} \tag{10.90}$$

The coherence function (10.88) applied in [17] can be extended to the case $P > 2$ by using the so-called generalized coherence [32]. In [26] a link between the SOS cost function (10.40) and the generalized coherence was established. This relationship allows a geometric interpretation of (10.40) and shows that this cost function leads to an inherent stepsize normalization for the coefficient updates.

4. WEIGHTING FUNCTION

In the generalized cost functions (10.17) and (10.40) a weighting function $\beta(i,m)$ was introduced with the block time indices i, m to allow different realizations of the algorithms. Based on the cost function we previously derived stochastic and natural gradient update equations in the time domain and frequency domain. Due to the similar structure of these equations, we will now consider only the time domain for simplicity. There, we can express the coefficient update as

$$\Delta \mathbf{W}(m) = \sum_{i=0}^{\infty} \beta(i,m) \mathcal{Q}(i), \tag{10.91}$$

where $\mathcal{Q}(i)$ denotes the term originating from the i-th block. In the following we distinguish three different types of weighting functions $\beta(i,m)$ for off-line, on-line, and block-on-line realizations [28]. The weighting functions have a finite support, and are normalized such that $\sum_{i=0}^{\infty} \beta(i,m) = 1$.

4.1 OFF-LINE IMPLEMENTATION

When realizing the algorithm as an off-line or so-called batch algorithm, then $\beta(i,m)$ corresponds to a rectangular window (Fig. 10.7), which is described by $\beta(i,m) = \frac{1}{K_{\text{sig}}}\epsilon_{0,(K_{\text{sig}}-1)}(i)$, where $\epsilon_{a,b}(i) = 1$ for $a \leq i \leq b$, and $\epsilon_{a,b}(i) = 0$ else. The entire signal is segmented into K_{sig}

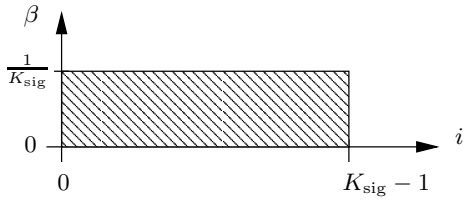


Figure 10.7 Weighting function $\beta(i,m)$ for off-line implementation.

blocks, and then the entire signal is processed to estimate the demixing matrix $\mathbf{W}^{\ell}$ where the superscript ℓ denotes the current iteration of the coefficient update

$$\mathbf{W}^{\ell} = \mathbf{W}^{\ell-1} - \frac{\mu}{K_{\text{sig}}} \sum_{i=0}^{K_{\text{sig}}-1} \mathcal{Q}(i). \tag{10.92}$$

Hence, the algorithm is generally visiting the signal data repeatedly for each iteration ℓ and therefore it usually achieves a better performance compared to its on-line counterpart.

4.2 ON-LINE IMPLEMENTATION

In time-varying environments an on-line implementation of (10.91) is required. An efficient realization can be achieved by using a weighting function with an exponential forgetting factor λ (Fig. 10.8). It is defined by

$$\beta(i,m) = (1-\lambda)\lambda^{m-i}\epsilon_{0,m}(i), \tag{10.93}$$

where $0 \leq \lambda < 1$. Thus (10.91) reads

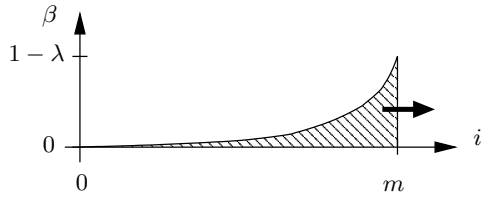


Figure 10.8 Weighting function $\beta(i,m)$ for on-line implementation.

$$\Delta \mathbf{W}(m) = (1-\lambda) \sum_{i=0}^{m} \lambda^{m-i} \mathcal{Q}(i), \tag{10.94}$$

where m denotes the current block. Additionally, (10.94) can be formulated recursively to reduce computational complexity and memory requirements since only the preceding demixing matrix has to be saved for the update. This leads to the following coefficient update to be used in (10.23):

$$\Delta \mathbf{W}(m) = \lambda \Delta \mathbf{W}(m-1) + (1-\lambda)\mathcal{Q}(m). \tag{10.95}$$

For the special case $\lambda = 0$ we have

$$\mathbf{W}(m) = \mathbf{W}(m-1) - \mu \mathcal{Q}(m), \tag{10.96}$$

which corresponds to $\beta(i,m) = \delta(i-m)$.

4.3 BLOCK-ON-LINE IMPLEMENTATION

The on-line and off-line approaches can be combined in a so-called block-on-line method (Fig. 10.9) which has been applied for BSS, e.g., in [51]. After obtaining K blocks of length N we process an off-line

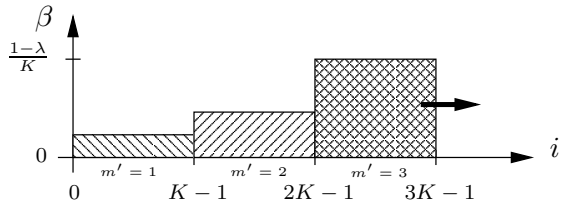


Figure 10.9 Weighting function $\beta(i,m)$ for block-on-line implementation. Note that $m' = \frac{m}{K}$ denotes the new block index.

algorithm with $\ell_{\max}$ iterations. The demixing filter matrix $\mathbf{W}(m')$ of the current block m' is then used as initial value for the off-line algorithm of the next block. This block-on-line approach allows a tradeoff between computational complexity on the one hand and separation performance and speed of convergence on the other hand by adjusting the maximum number of iterations $\ell_{\max}$ as we will see in Sect. 5..

5. EXPERIMENTS AND RESULTS

Experiments have been conducted using speech data convolved with impulse responses of a real room (580cm × 590cm × 310cm) with a reverberation time $T_{60} = 150$ms and a sampling frequency of 16 kHz. A two-element microphone array with an inter-element spacing of 16 cm was used. The speech signals arrived from two different directions, -45^o

and 45^o. The distance between the speakers and the microphones was 2.0m. The length of the source signals (two male speakers from the TIMIT speech corpus [52]) was 10 seconds. The performance was evaluated by means of the signal-to-interference ratio (SIR), defined as the ratio of the signal power of the target signal to the signal power from the jammer signal. For off-line implementations the SIR was calculated over the entire signal length, whereas for on-line implementations it was continuously calculated for each block. In the following the SIR is averaged over both channels.

In our experiments we compared off-line and on-line realizations and we examined the effect of taking into account different numbers of time-lags D for the computation of the correlation function in (10.29) and (10.38). In all experiments the unmixing filter length was set to $L = 512$, the number of lags to $D = 512$, and the block length to $N = 1024$, respectively. Note that the stepsizes of all algorithms have been maximized up to the stability margins.

The framework developed here also allows a better understanding of the initialization of $\mathbf{W}$. It can be shown using (10.6) and (10.25) that the first coefficient of each filter $\mathbf{W}_{pp}$ must be nonzero. This is ensured by using unit impulses for the first filter tap in each $\mathbf{W}_{pp}$. The filters $\mathbf{W}_{pq}$, $p \neq q$ are set to zero.

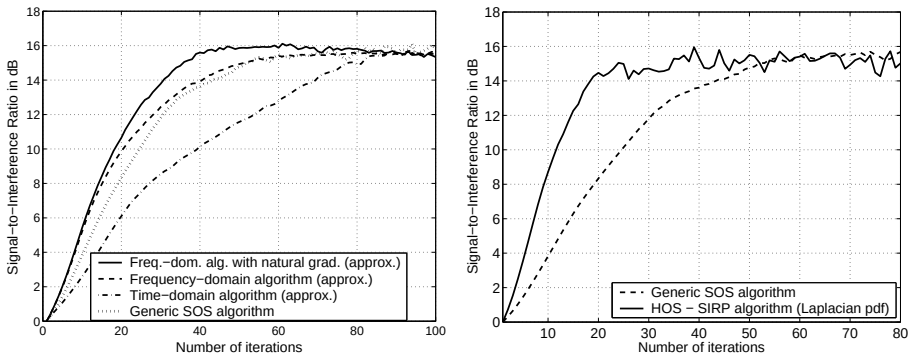


Figure 10.10 Comparison of different off-line realizations (left: SOS algorithms, right: SOS vs. HOS).

In the left plot of Fig. 10.10 different off-line SOS algorithms are shown. It can be seen that the approximated gradient (10.86) (dashed) and natural gradient (10.87) (solid) versions of the generic SOS frequency-domain algorithm exhibit the best convergence. This is mainly due to the decomposition of the update equation in its frequency components and hence we have an independent update in each frequency

bin. The complete decoupling and therefore also the internal permutation problem is prevented by considering the constraint $\mathbf{G}_{4LP\times 4LP}^{1_L 0}$ (10.63) (see Sect. 3.1).

It should be pointed out that the generic SOS time-domain algorithm (10.37) (dotted) achieves almost the same convergence as the frequency-domain algorithms. This shows that also time-domain algorithms can exhibit a stable and robust convergence behaviour for long unmixing filters. However, in the generic SOS time-domain algorithm this comes with an increased computational cost, as an inversion of a large matrix is required due to the RLS-like normalization (see Sect. 2.4.2). The approximated version of the generic SOS time-domain algorithm (dash-dotted) according to (10.41) shows a slower convergence as the RLS-like normalization is replaced by a diagonal matrix which corresponds to an NLMS-like normalization. Moreover it can be seen that all curves converge to the same maximum SIR value which does not depend on the choice of adaptation algorithm.

In the right plot of Fig. 10.10 we compared the generic SOS algorithm in the time-domain and the generic HOS algorithm with the SIRP model from the Laplacian pdf (10.34). Note that the argument of the modified Bessel functions $K_{D/2+1}(\cdot)$ in (10.34) has to be properly regularized. The additional gain in convergence speed of HOS over SOS is due to the additional exploitation of nongaussianity.

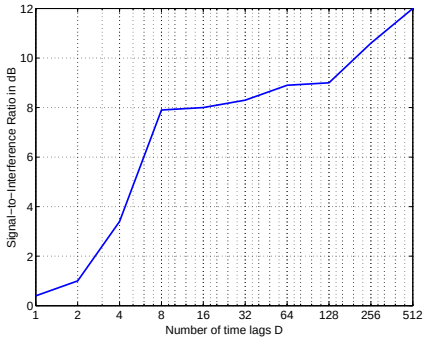


Figure 10.11 Effect of exploiting nonwhiteness by taking into account different numbers of lags D. ($L = 512$)

In Fig. 10.11 the dependency of the SIR on the number of lags D used for the computation of the correlation function $\mathbf{R}_{\mathbf{yy}}$ in the SOS algorithms is illustrated. An off-line version of the approximated time-domain algorithm (10.41) was evaluated after 50 iterations. We observe a steep increase of the achievable separation performance for up to 8 lags. This can be explained by the fact that speech is strongly correlated

within the first lags. By considering these temporal correlations, i.e., nonwhiteness, additional information about the mixtures is taken into account for the simultaneous diagonalization of $\mathbf{R}_{\mathbf{yy}}$. A further increase of D still improves the SIR slightly as the temporal correlation of the room impulse response is considered in the adaptation.

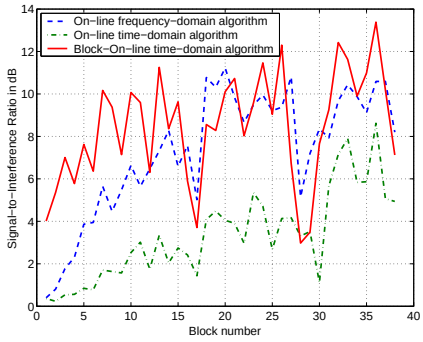


Figure 10.12 Comparison of different on-line realizations.

Various on-line realizations of SOS algorithms are shown in Fig. 10.12. Obviously, the frequency-domain algorithm (dashed) exhibits superior convergence compared to the time-domain algorithm (dash-dotted) due to the NLMS-like approximation of the normalization in the time domain. However, it can also be seen that this effect can be mitigated by using a block-on-line adaptation (see 4.3) (solid) with $K = 8$, $N = 512$, and $\ell_{\max} = 10$ iterations. This leads to improved convergence and separation performance at the expense of increased computational cost.

6. CONCLUSIONS

We presented a unified treatment of BSS algorithms for convolutive mixtures. This framework contains two main principles: Firstly, three fundamental signal properties, nonwhiteness, nonstationarity, and nongaussianity are explicitly taken into account in the generic cost function. Secondly, the framework is based on a general broadband formulation and optimization of this cost function. Due to this approach, rigorous derivations of both known and novel algorithms in the time and frequency domain became possible. Moreover, the introduced matrix formulation with the resulting constraints provides a deeper understanding of the internal permutation ambiguity appearing in traditional narrowband frequency-domain BSS. Experimental results confirm the theoretical findings and demonstrate that this approach allows BSS in both, time and frequency domains for reverberant acoustic environments.

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley & Sons, Inc., New York, 2001.

[2] M. Zibulevsky and B.A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, pp. 863-882, 2001.

[3] S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada, "Blind Separation of More Speech than Sensors with Less Distortion by Combining Sparseness and ICA," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, Sep. 2003, pp. 271-274.

[4] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non gaussian signals," *IEE Proceedings-F*, vol 140, no. 6, pp. 362-370, Dec. 1993.

[5] W. Herbordt and W. Kellermann, "Adaptive beamforming for audio signal acquisition," in *Adaptive signal processing: Application to real-world problems*, J. Benesty and Y. Huang, Eds., pp. 155-194, Springer, Berlin, Jan. 2003.

[6] E. Weinstein, M. Feder, and A. Oppenheim, "Multi-channel signal separation by decorrelation," *IEEE Trans. on Speech and Audio Processing*, vol 1, no. 4, pp. 405-413, Oct. 1993.

[7] L. Molgedey and H.G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review Letters,* vol. 72, pp. 3634-3636, 1994.

[8] L. Tong, R.-W. Liu, V.C. Soon, and Y.-F. Huang, "Indeterminacy and identifiability of blind identification," *IEEE Trans. on Circuits and Systems,* vol. 38, pp. 499-509, 1991.

[9] S. Van Gerven and D. Van Compernolle,"Signal separation by symmetric adaptive decorrelation: stability, convergence, and uniqueness," *IEEE Trans. Signal Processing,* vol. 43, no. 7, pp. 1602-1612, 1995.

[10] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," *Neural Networks,* vol. 8, no. 3, pp. 411-419, 1995.

[11] M. Kawamoto, K. Matsuoka, and N. Ohnishi, "A method of blind separation for convolved non-stationary signals," *Neurocomputing*, vol. 22, pp. 157-171, 1998.

[12] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Mat. Anal. Appl.,* vol. 17, no. 1, pp. 161-164, Jan. 1996.

[13] S. Ikeda and N. Murata, "An approach to blind source separation of speech signals," *Proc. Int. Symposium on Nonlinear Theory and its Applications,* Crans-Montana, Switzerland, 1998.

[14] L. Parra and C. Spence, "Convolutive blind source separation of non-stationary sources," *IEEE Trans. Speech and Audio Processing*, pp. 320-327, May 2000.

[15] D.W.E. Schobben and P.C.W. Sommen, "A frequency-domain blind signal separation method based on decorrelation," *IEEE Trans on Signal Processing,* vol. 50, no. 8, pp. 1855-1865, Aug. 2002.

[16] H.-C. Wu and J.C. Principe,"Simultaneous diagonalization in the frequency domain (SDIF) for source separation," in *Proc. IEEE Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, 1999, pp. 245-250.

[17] C.L. Fancourt and L. Parra, "The coherence function in blind source separation of convolutive mixtures of non-stationary signals," in Proc. *Int. Workshop on Neural Networks for Signal Processing (NNSP)*, 2001.

[18] P. Comon, "Independent component analysis, a new concept? " *Signal Processing,* vol. 36, no. 3, pp. 287-314, Apr. 1994.

[19] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*, Wiley & Sons, Ltd., Chichester, UK, 2002.

[20] S. Amari, A. Cichocki, and H.H. Yang, "A new learning algorithm for blind signal separation," in *Advances in neural information processing systems*, 8, Cambridge, MA, MIT Press, 1996, pp. 757-763.

[21] J.-F. Cardoso, "Blind signal separation: Statistical principles," *Proc. IEEE*, vol. 86, pp. 2009-2025, Oct. 1998.

[22] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, pp. 21-34, July 1998.

[23] A.J. Bell and T.J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129-1159, 1995.

[24] T. Nishikawa, H. Saruwatari, and K. Shikano, "Comparison of time-domain ICA, frequency-domain ICA and multistage ICA for blind source separation," in *Proc. European Signal Processing Conference (EUSIPCO)*, Sep. 2002, vol. 2, pp. 15-18.

[25] R. Aichner, S. Araki, S. Makino, T. Nishikawa, and H. Saruwatari, "Time-domain blind source separation of non-stationary convolved signals with utilization of geometric beamforming," in Proc. *Int. Workshop on Neural Networks for Signal Processing (NNSP)*, Martigny, Switzerland, 2002, pp. 445-454.

[26] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of a class of blind source separation algorithms for convolutive mixtures," *Proc. IEEE Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 945-950.

[27] H. Buchner, R. Aichner, and W. Kellermann, "Blind Source Separation for Convolutive Mixtures Exploiting Nongaussianity, Nonwhiteness, and Nonstationarity," *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, September 2003.

[28] R. Aichner, H. Buchner, S. Araki, and S. Makino, "On-line time-domain blind source separation of nonstationary convolved signals," *Proc. IEEE Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 987-992.

[29] E. Moulines, O. Ait Amrane, and Y. Grenier, "The generalized multidelay adaptive filter: structure and convergence analysis," *IEEE Trans. Signal Processing,* vol. 43, pp. 14-28, Jan. 1995.

[30] H. Brehm and W. Stammler, "Description and generation of spherically invariant speech-model signals," *Signal Processing* vol. 12, pp. 119-141, 1987.

[31] H. Buchner, J. Benesty, and W. Kellermann, "Multichannel Frequency-Domain Adaptive Algorithms with Application to Acoustic Echo Cancellation," in J.Benesty and Y.Huang (eds.), *Adaptive signal processing: Application to real-world problems*, Springer-Verlag, Berlin/Heidelberg, Jan. 2003.

[32] H. Gish and D. Cochran, "Generalized Coherence," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, New York, NY, USA, 1988, pp. 2745-2748.

[33] H.H. Yang and S. Amari, "Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information," *Neural Computation*, vol. 9, pp. 1457-1482, 1997.

[34] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley & Sons, New York, 1991.

[35] S. Haykin, *Adaptive Filter Theory,* 3rd ed., Prentice Hall., Englewood Cliffs, NJ, 1996.

[36] D.H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *Proc. IEE*, vol. 130, Pts. F and H, pp. 11-16, Feb. 1983.

[37] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, New York, 1991.

[38] F.D. Neeser and J.L. Massey, "Proper Complex Random Processes with Applications to Information Theory," *IEEE Trans. on Information Theory*, vol. 39, no. 4, pp. 1293-1302, July 1993.

[39] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation,* vol. 10, pp. 251-276, 1998.

[40] J.D. Markel and A.H. Gray, *Linear Prediction of Speech*, Springer-Verlag, Berlin, 1976.

[41] J.W. Brewer, "Kronecker Products and Matrix Calculus in System Theory," *IEEE Trans. Circuits and Systems,* vol. 25, no. 9, pp. 772-781, Sep. 1978.

[42] M.Z. Ikram and D.R. Morgan, "Exploring permutation inconsistency in blind separation of speech signals in a reverberant environment," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000, vol. 2, pp. 1041-1044.

[43] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Robust and precise method for solving the permutation problem of frequency-domain blind source separation," *Proc. IEEE Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 505-510.

[44] J.-S. Soo and K.K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-38, pp. 373-376, Feb. 1990.

[45] P.C.W. Sommen, P.J. Van Gerwen, H.J. Kotmans, and A.J.E.M. Janssen, "Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function," *IEEE Trans. Circuits and Systems,* vol. 34, no. 7, pp. 788-798, July 1987.

[46] J. Benesty, A. Gilloire, and Y. Grenier, "A frequency-domain stereophonic acoustic echo canceller exploiting the coherence between the channels," *J. Acoust. Soc. Am.,* vol. 106, pp. L30-L35, Sept. 1999.

[47] G. Enzner and P. Vary, "A soft-partitioned frequency-domain adaptive filter for acoustic echo cancellation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, China, April 2003, vol. 5, pp. 393-396.

[48] R.M.M. Derkx, G.P.M. Egelmeers, and P.C.W. Sommen, "New constraining method for partitioned block frequency-domain adaptive filters," *IEEE Trans. Signal Processing*, vol. 50, no. 9, pp. 2177-2186, Sept. 2002.

[49] F.J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, pp. 51-83, Jan. 1978.

[50] S. Sawada, R. Mukai, S. de la Kethulle de Ryhove, S. Araki, and S. Makino, "Spectral smoothing for frequency-domain blind source separation," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, Sep. 2003, pp. 311-314.

[51] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Real-Time Blind Source Separation for Moving Speakers using Blockwise ICA and Residual Crosstalk Subtraction," *Proc. IEEE Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 975-980.

[52] J.S. Garofolo et al., "TIMIT acoustic-phonetic continuous speech corpus," National Institute of Standards and Technology, 1993.