DIGITAL BEAMFORMING EMPLOYING SUBBAND TECHNIQUES

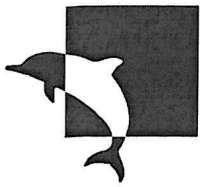
PHD THESIS

BY

WEI LIU

AUGUST 2003

UNIVERSITY OF SOUTHAMPTON



University of Southampton
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

Digital Beamforming Employing Subband Techniques

by

Wei LIU
B.Sc, L.L.B, M.Phil

A doctoral thesis submitted in partial fulfilment
of the requirements for the award of
Doctor of Philosophy

August 2003

Supervisors:
Dr. Stephan Weiss
and
Prof. Lajos Hanzo

This thesis is dedicated to
my mother Su Yan Wang,
as a memory of her, forever!

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Digital Beamforming Employing Subband Techniques

by Wei Liu

In this thesis, we propose some different methods to reduce the computational complexity of a broadband beamformer based on the generalised sidelobe canceller (GSC) structure.

A GSC is an alternative implementation of the linearly constrained minimum variance beamformer and relys on well-known adaptive filtering algorithms to perform unconstrained adaptive optimisation. Due to the low computational complexity of subband adaptive filtering techniques, we embed the subband adaptive algorithms into the adaptive part of the GSC and propose a new subband adaptive GSC structure. In addition to its lower computational complexity than those previously suggested subband beamforming structures, it achieves a faster convergence rate than the traditional fullband adaptive GSC due to its pre-whitening effect.

By studying the input-output relationship of the blocking matrix of a GSC, we also propose a specific construction of the blocking matrix, in which the impulse responses hosted by its column vectors constitute those of a series of bandpass filters. These filters select signals with specific directions of arrival and frequencies and result in bandlimited spectra of its outputs. This spatially/temporally subband-selective blocking matrix can be applied to the subband adaptive GSC or the transform-domain GSC to reduce their computational complexities due to the discarding of some of the subbands or frequency bin outputs prior to the following adaptation. Since an overlap and finite transition band between the bandpass filters in the blocking matrix have to be permitted, a better design quality can be attained by reducing the output dimension of the blocking matrix, which yields a partially adaptive beamformer with further reduction of the computational complexity. Because of its combined spatial/temporal decorrelation effect, a faster convergence speed is also achieved, as demonstrated in the corresponding simulations.

Moreover, based on one of the traditional blocking matrices, we propose a spatially/temporally subband-selective transformation matrix, which can be regarded as another implementation of the subband-selective blocking matrix, because it has the same effect as the subband-selective blocking matrix, when combined with the traditional blocking matrix. The advantage of this method is that it simplifies the design of the subband-selective blocking matrix into a general filter design problem and can be implemented efficiently by cosine modulation.

Contents

List of Publications

1. W. Liu, S. Weiss and L. Hanzo, "A Subband-selective Broadband GSC with Cosine-modulated Blocking Matrix", IEEE Trans. Antennas and Propagation, to appear in April, 2003.

2. W. Liu, S. Weiss and L. Hanzo, "A Novel Method for Partially Adaptive Broadband Beamforming", Journal of VLSI Signal Processing, vol. 33, issue 3, March, 2003, pp.337–344.

3. Wei Liu, S. Weiss, and L. Hanzo, "A subband-selective Transform-domain GSC with low-computational complexity", Proceedings of the 2003 Postgraduate Research Conference in Electronics, Photonics, Communications and Software (PREP), Exeter University, April 14-16, 2003, pp.31-32.

4. S. Weiss, R.W. Stewart and W. Liu, "A Broadband Adaptive Beamformer in Subbands with Scaled Aperture", Proceedings of the 36th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, November 2002, pp.1298-1302.

5. W. Liu, S. Weiss and L. Hanzo, "Low-complexity Frequency-domain GSC for Broadband Beamforming", Proceedings of the 2002 International Conference on Signal Processing (ICSP), Beijing, China, August, 2002, pp.386–389.

6. W. Liu, S. Weiss and L. Hanzo, "Partially Adaptive Broadband Beamforming with a Subband-selective Tranformation Matrix", Proceedings of the 2002 IEEE Workshop on Sensor Array and Multichannel Signal Processing (SAM), Virginia, USA, August, 2002, pp.43–47.

7. W. Liu, S. Weiss, and L. Hanzo, "Subband-selective Partially Adaptive Broadband Beamforming with Cosine-modulated Blocking Matrix", Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Orlando, Florida, USA, Vol. 3, May, 2002, pp.2913–1916.

8. W. Liu, S. Weiss, and L. Hanzo, "A Novel Method for Partially Adaptive Broadband Beamforming", Proceedings of the 2001 Workshop on Signal Processing Systems (SIPS), Antwerp, Belgium, September, 2001, pp.361–372.

9. W. Liu, S. Weiss, and L. Hanzo, "Subband Adaptive Generalized Sidelobe Canceller for Broadband Beamforming", Proceedings of the 11th IEEE Workshop on Statistical Signal Processing (SSP), Singapore, August, 2001, pp.591–594.

10. W. Liu, S. Weiss, and L. Hanzo, "A Novel Subband Adaptive Beamforming Architecture Based on the Generalized Sidelobe Canceller", Proceedings of the 2001 Postgraduate Research Conference in Electronics, Photonics, Communications and Software (PREP), University of Keele, April, 2001, pp.23–24.

11. S. Weiss, W. Liu, R. W. Stewart, and I. K. Proudler, "A Generalised Sidelobe Canceller Architecture Based on Oversampled Subband Decompositions", Proceedings of the Fifth International Conference on Mathematics in Signal Processing (IMA), Warwick, December, 2000.

List of Figures

List of Tables

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to my supervisors Dr. Stephan Weiss and Prof. Lajos Hanzo for their persistent support and guidance, and the freedom they gave me to develop and follow my research interests.

I am very grateful to Dr. Sheng Chen for his thoughtful comments and suggestions. Thanks also go to our group secretary Mrs. Denise Harvey and all my colleagues in the Communications Research Group, for their support and help during the past years.

Finally, I would like to thank my family in China and my wife Yan for their love and support during my study here, which is beyond any word. Especially, I would like to dedicate this thesis to my mother Su-Yan Wang, as a memory of her, forever!

Chapter 1

Introduction

1.1 Introduction

Beamforming has found many applications in various areas ranging from sonar and radar to wireless communications [5–7]. It is a signal processing technique to form beams in order to receive signals illuminating an array from specific directions, whilst attenuating signals from other directions. In the presence of interfering signals, linear temporal filtering cannot be used to separate a desired signal occupying the same frequency band. However, these signal components usually originate from different spatial locations. This spatial separation can be exploited to separate the desired signals from interfering signals by an array of sensors. These sensors can be positioned in space according to different patterns, e.g. along a line, around a circle, or on a plane. Such arrangements lead to linear arrays, circular arrays and planar arrays [7–9], respectively. For the time being, our discussion is limited to the family of uniformly spaced linear arrays, which are shown in Fig. 1.1. Specifically, each sampled sensor signal $x_m[n]$ is processed by a dedicated filter $\tilde{\mathbf{w}}_m$, $m = 0, 1, \cdots, M-1$. The wavefront of signals impinging

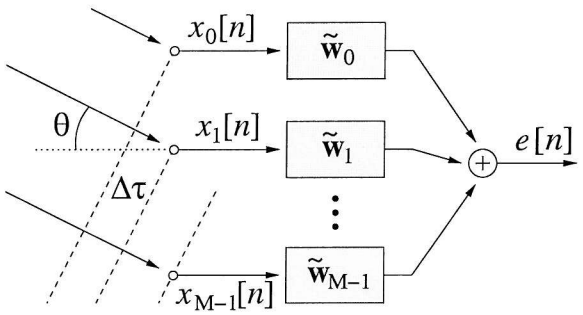


Fig. 1.1: A uniformly spaced linear array.

from a direction of arrival (DOA) θ arrives delayed by integer multiples of $\Delta\tau$ at these M

sensors. The impulse response (IR) duration of each attached filter varies according to the specific applications concerned.

To process narrowband signals [10], we can sample the propagating wave field in space and linearly combine the data of M sensors, i.e. we only need one IR coefficient for each filter $\tilde{\mathbf{w}}_m$ [11]. For beamforming of broadband signals [12], we sample the wave field in both temporal and spatial domains, and typically a higher number of coefficients are employed for $\tilde{\mathbf{w}}_m$. The filter coefficients can be designed for maintaining a fixed specified response for all signal/interference scenarios, which leads to the concept of a data independent beamformer [13]. Alternatively, they can be chosen based on the statistics of the array data for optimising the array's response, which forms a statistically optimum beamformer [13]. Since the statistics of the array data are often not known or may change over time, adaptive algorithms may be used to determine these coefficients.

To perform adaptive broadband beamforming with high interference rejection and angular resolution, arrays with a large number of sensors and filter coefficients have to be employed. Reducing the resultant high computational complexity has stimulated intensive research in the community. Numerous solutions have been suggested. In partially adaptive beamforming [14–18], only a fraction of available degrees of freedom is exploited at the expense of a reduced performance. In dynamic selective beamforming, the multi-beam outputs, as found in beamspace beamforming [18–20], or frequency bin outputs, as found in transform-domain or frequency-domain beamforming [21–23], are monitored and the specific outputs having a power below some threshold will be omitted from subsequent processing. In subband beamforming [1,3,4,24–28], the received sensor signals are first split into decimated frequency bands ("subbands"), prior to applying an independent beamformer to each subband.

In this thesis, based on the generalized sidelobe canceller (GSC) structure [29, 30], which is an alternative, but efficient implementation of the linearly constrained minimum variance (LCMV) beamformer [31], a range of novel methods is proposed for reducing the computational complexity of broadband beamforming. In addition to their lower computational complexity, all these methods promise a convergence speed faster than traditional implementations of least mean square (LMS) type adaptive algorithms. In the next section, we will list a number of efficient adaptive broadband beamforming realisations, which constitute the original contribution of this thesis.

1.2 Original Contributions

To combat the computational complexity of high-resolution broadband beamforming, we propose several methods employing subband techniques:

- **A Subband Adaptive Generalized Sidelobe Canceller [32, 33]**

  In this structure, we employ the subband adaptive filtering technique to decompose the input signals to the adaptive part of the GSC into subbands and perform unconstrained adaptive minimisation in each subband. This new subband adaptive system is capable of achieving a lower computational complexity than previously suggested subband beamforming structures. Further, we can reason and demonstrate that it can achieve a faster convergence rate than the traditional fullband adaptive GSC due to its pre-whitening effect.

- **A Spatially/Temporally Subband-selective Blocking Matrix**

  By studying the input-output relationship of the blocking matrix of a GSC, we propose a specific construction of this matrix, in which the impulse responses hosted by its column vectors constitute a series of bandpass filters. These filters select signals with specific DOAs and frequencies and result in the blocking matrix outputs having bandlimited spectra. Since an overlap and finite transition band between the bandpass filters in the blocking matrix have to be permitted, a better design quality can be attained by reducing the output dimension of the blocking matrix, yielding a partially adaptive beamformer. This scheme finds applications in two areas:

  - **application to subband adaptive GSC [34–37]**

    In this case, the subband decomposition of the bandlimited spectra will result in some subbands containing signals of almost zero energy, which can then be discarded in the following subband adaptation. By partial adaptivity, subband discarding and subband adaptation, the computational complexity of the system can be significantly reduced.

  - **application to transform-domain GSC [38]**

    Here, we apply the discrete Fourier transform (DFT) to the outputs of the blocking matrix. Because of the bandlimited spectra of these outputs, some of the frequency-bin outputs of the DFT will be approximately zero and can be omitted from the following adaptive processing. To enhance the bandlimitation, a window function needs to be applied before performing the DFT.

- **A Spatially/Temporally Subband-selective Transformation Matrix [39, 40]**

  This can be regarded as an alternative implementation of the subband-selective blocking matrix, because it has the same effect as the subband-selective blocking matrix, when combined with one of the traditional blocking matrices. The advantage of this method is that it simplifies the design of the subband-selective blocking matrix into a general filter design problem and can be efficiently implemented by a cosine modulation.

1.3 Thesis Outline

The outline of this thesis is as follows:

In Chapter 2, we first give an introduction to the basic ideas of beamforming and then focus on the LCMV beamforming technique and the GSC structure, which transforms the LCMV's constrained optimization problem into an unconstrained one. Various problems related to LCMV beamforming and GSC are briefly discussed, such as the formulation of constraints, the optimum mean square error solution, Frost's algorithm, and the blocking matrix design. Especially, the simplified GSC with tapped-delay lines is described. Adaptive algorithms commonly used for the optimisation of the beamformer's parameters are introduced, with an emphasis on their respective computational complexities.

In Chapter 3, we give a brief overview on the fundamentals of filter banks and the subband adaptive filtering (SAF) technique. Employing the SAF technique to the fullband adaptive part of a GSC, we propose a novel subband adaptive GSC. We will show that compared with traditional fullband adaptive GSC and previously proposed subband beamforming structures, this new subband adaptive system can attain a lower computational complexity. In addition, it achieves a convergence speed faster than the fullband adaptive GSC due to its pre-whitening effect, as demonstrated in simulations based on different signal environments and different formations of the blocking matrix.

In Chapter 4, the architecture of a partially adaptive GSC structure is reviewed. Based on this we propose a GSC with a novel spatially/temporally subband-selective blocking matrix. Two design methods are suggested in order to construct this matrix, comprising a constrained full design and a cosine-modulated design based on the decomposition of the prototype vector. Moreover, we develop a subband-selective transformation matrix which has the same effect as the subband-selective blocking matrix, but offers simplifications in design and implementation. The joint application of such blocking and transformation matrices to the subband adaptive GSC and the transform-domain GSC are then studied to reduce the computational complexity of the system. The effectiveness and benefits of the subband-selective systems are shown by an extensive number of simulations.

Finally, conclusions and an outlook on possible future work are given in Chapter 5.

Chapter 2

Beamforming

This section reviews the basic ideas of beamforming and focuses on the linearly constrained minimum variance beamforming technique as well as on its alternative implementation – the generalised sidelobe canceller, which transforms the constrained optimisation problem of the beamformer's parameters into an unconstrained one. Subsequently, we review various widely used standard adaptive algorithms, which can be employed in the resulting unconstrained optimisation.

2.1 Wave Propagation

Whenever a driving force is coupled to an open medium, travelling waves are generated. They propagate from the source of the excitation transporting energy in a specific form, e.g. acoustic or electro-magnetic, which depends on the physical nature of the driving force. In the far field, namely at a large distance away from the source, the waves become essentially planar. In our study, we assume that the signals impinging on the sensor array are always plane waves.

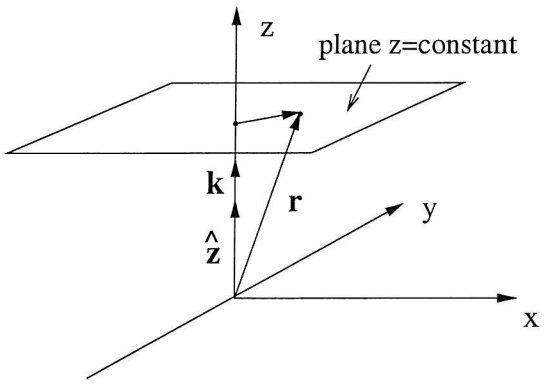


Fig. 2.1: A Cartesian coordinate system.

Consider a plane wave propagating in the direction of the unit vector $\hat{\mathbf{z}}$, along the $z-$axis of the Cartesian coordinate system as shown in Fig. 2.1. At the plane defined by $z = constant$, the wave function can be expressed as

$$s(t, z) = A\cos(2\pi ft - kz) \,, \tag{2.1}$$

where A is the amplitude, f the frequency, and t the time. The parameter k is referred to as the wavenumber [11] and defined as

$$k = \frac{\omega}{c} = \frac{2\pi}{\lambda} \,, \tag{2.2}$$

where ω is the (temporal) angular frequency, c denotes the speed of propagation in the specific medium and λ is the wavelength. Similar to ω, which means that in a temporal interval t the phase accumulates to ωt, the physical interpretation of the wavenumber k is that over a distance z, measured along the propagation direction $\hat{\mathbf{z}}$, the phase accumulates to kz radians and we can hence refer to k as the spatial frequency [41]. A specific difference between the spatial and temporal frequencies k and ω is their dimension. As time has only one dimension, the corresponding frequency is also one-dimensional. However, spatial quantities like k stretch, in general, over all three spatial dimensions, denoted either by the set of Cartesian variables $[x, y, z]$ or, in polar notation, by $[\theta, \phi, \rho]$ using the azimuth θ, elevation ϕ, and radius ρ. It is important to point out that k is characteristic of a monochromatic plane wave, in other words, the spatial and temporal frequencies are coupled and cannot be chosen independently.

Let the vector $\mathbf{r}$ denote a point in space, which is specified in terms of its coordinates with respect to the origin of the Cartesian coordinate system. Then the plane $z = constant$ is described by $z = \hat{\mathbf{z}}^T \cdot \mathbf{r} = constant$, which is shown in Fig. 2.1. Thus we can express the quantity kz in (2.1) as

$$kz = k(\hat{\mathbf{z}}^T \cdot \mathbf{r}) = \mathbf{k}^T \cdot \mathbf{r} \,, \tag{2.3}$$

where

$$\mathbf{k} = k(\hat{\mathbf{z}}) \,. \tag{2.4}$$

We refer to $\mathbf{k}$ as the wavenumber vector, which points in the direction of propagation $\hat{\mathbf{z}}$.

The argument of the sinusoidal wave function in (2.1) is referred to as the phase $\phi(t, z)$. Using this new notation, the phase can be expressed as

$$\phi(t, \mathbf{r}) = 2\pi ft - \mathbf{k}^T \cdot \mathbf{r} \,. \tag{2.5}$$

At a fixed time instant t, the points associated with the same phase ϕ are referred to as a

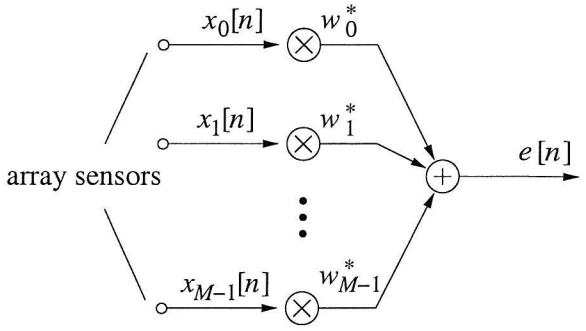


Fig. 2.2: A general structure for narrowband beamforming.

wavefront and are defined by

$$d\phi = 0 = 2\pi f\ dt - \mathbf{k}^T \cdot d\mathbf{r} \tag{2.6}$$

$$= 2\pi\ 0 - \mathbf{k}^T \cdot d\mathbf{r} \tag{2.7}$$

$$= -\mathbf{k}^T \cdot d\mathbf{r}\ . \tag{2.8}$$

In tangible physical terms, at a fixed time instance t, the phase has the same value at all points reached by adding up the vectors $d\mathbf{r}$ that are perpendicular to the wavenumber vector $\mathbf{k}$. All these points constitute the plane associated with z =constant. It is for this reason that such a wave is referred to as a plane wave.

2.2 General Beamforming

2.2.1 Beamforming Structures

In beamforming, we estimate the signal arriving from desired directions in the presence of noise and interfering signals with the aid of an array of sensors in order to provide a versatile form of spatial filtering. These sensors are located at different spatial positions and sample the propagating wave in space. The collected spatial samples are then processed in order to spatially extract the desired signal. Fig. 2.2 and Fig. 2.3 show two different beamforming structures.

Specifically, in Fig. 2.2 the M linearly equispaced sensors sample the wave field spatially and the output $e[n]$ at time n is given by a linear combination of these spatial samples as

$$e[n] = \sum_{m=0}^{M-1} x_m[n] \cdot w_m^*\ . \tag{2.9}$$

The beamformer associated with this structure is only useful for sinusoidal or narrowband

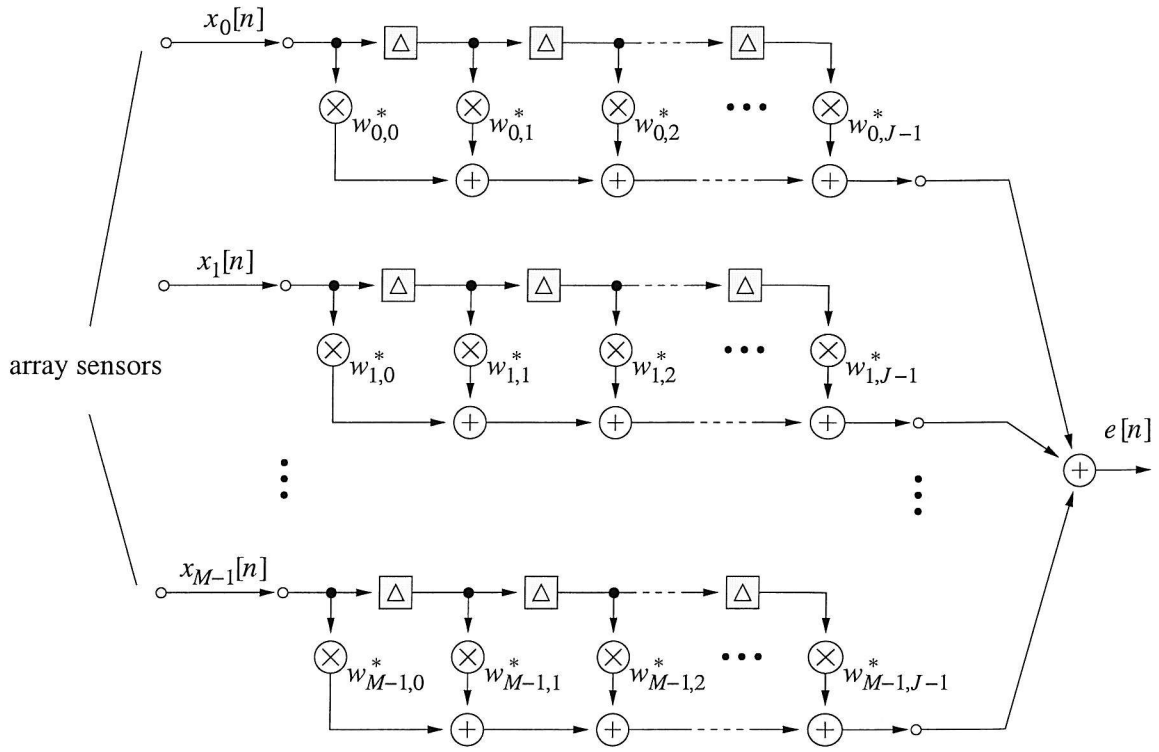


Fig. 2.3: A general structure for broadband beamforming.

signals, where "there is essentially no decorrelation between signals received on opposite ends of the array" [42] and hence it is termed a narrowband beamformer.

When signals with significant frequency extent, i.e. broadband signals, are of interest, rather than applying a single coefficient w_m^* as in the case of narrowband beamforming, we apply a tapped-delay line at the output of each sensor [12] as shown in Fig. 2.3. These tapped-delay lines form a finite impulse response (FIR) filter at the output of each sensor. The beamformer obeying this architecture samples the propagating wave field in both space and time and it is suitable for broadband beamforming. The output of such a broadband beamformer can be expressed as

$$e[n] = \sum_{m=0}^{M-1} \sum_{j=0}^{J-1} x_m[n-j] \cdot w_{m,j}^* \,, \tag{2.10}$$

where $J-1$ is the number of delay elements associated with each of the M sensor channels in Fig. 2.3.

For convenience, we develop notations for treating both kinds of beamformers. Let us consider a system having M sensors and J taps per sensor, where $J = 1$ for narrowband beamforming. We rewrite the array output $e[n]$ in Fig. 2.2 and Fig. 2.3 as

$$e[n] = \mathbf{w}^H \mathbf{x} \,. \tag{2.11}$$

The weight vector $\mathbf{w} \in \mathbb{C}^{MJ}$ holds all MJ sensor coefficients with

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_{J-1} \end{bmatrix}, \tag{2.12}$$

where each vector $\mathbf{w}_j$, $j = 0, 1, \cdots, J-1$, contains the M *complex conjugate* coefficients found at the jth tap position of the M filters, which is expressed as

$$\mathbf{w}_j = [w_{0,j}\ w_{1,j}\ \cdots\ w_{M-1,j}]^T . \tag{2.13}$$

Similarly, the input data are also accumulated in a vectorial form $\mathbf{x}$ as follows

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}[n] \\ \mathbf{x}[n-1] \\ \vdots \\ \mathbf{x}[n-J+1] \end{bmatrix}, \tag{2.14}$$

where $\mathbf{x}[n-j]$, $j = 0, 1, \cdots, J-1$, holds the jth data slice corresponding to the jth coefficient vector $\mathbf{w}_j$

$$\mathbf{x}[n-j] = \left[x_0[n-j]\ \ x_1[n-j]\ \ \cdots\ \ x_{M-1}[n-j]\right]^T . \tag{2.15}$$

Recall that this notation incorporates the narrowband beamformer with the special case of $J = 1$.

In our notation, we generally use lowercase bold letters for vector valued quantities, while uppercase bold letters symbolise a matrix. The operators $\{\cdot\}^T$ and $\{\cdot\}^H$ represent transpose and Hermitian transpose operations, respectively.

2.2.2 Beamformer Response and Beampattern

Let us now analyse the array's response to an impinging complex plane wave $e^{j\omega t}$ having a frequency ω and direction of arrival (DOA) θ, where the angle θ is measured with respect to the broadside of the linear array. For convenience, let the phase be zero at the first sensor, which implies $x_0[n] = e^{j\omega t}$ and $x_m[n] = e^{j\omega(t-m\tau(\theta))}$, $m = 0, 1, \cdots, M-1$, where $\tau(\theta) = 2\pi d \sin\theta/(\omega\lambda) = d\sin\theta/c$ with wave speed c and the distance d between adjacent sensors. Assuming furthermore that the tap delay period is T seconds, then $x_m[n-i] = e^{j\omega(t-m\tau(\theta)-iT)}$, $m = 0, \cdots, M-1$, $i = 0, \cdots, J-1$. Therefore the array output is given by

$$\begin{aligned} e[n] &= e^{j\omega t} \sum_{m=0}^{M-1} \sum_{i=0}^{J-1} e^{-j\omega(m\tau(\theta)+iT)} \cdot w_{m,i}^* \\ &= e^{j\omega t} \cdot r(\theta, \omega) , \end{aligned} \tag{2.16}$$

where $r(\theta, \omega)$ is the beamformer's angle and frequency-dependent response, which can be expressed in vectorial form as

$$r(\theta, \omega) = \mathbf{w}^H \mathbf{d}(\theta, \omega) \, . \tag{2.17}$$

The elements of $\mathbf{d}(\theta, \omega)$ correspond to the complex exponentials $e^{-j\omega(m\tau(\theta)+iT)}$ and in general we have

$$\mathbf{d}(\theta, \omega) = [e^{-j\omega\tau_0(\theta)} \;\; e^{-j\omega\tau_1(\theta)} \;\; \cdots \;\; e^{-j\omega\tau_{MJ-1}(\theta)}]^T \, , \tag{2.18}$$

where $\tau_0(\theta) = 0$ and $\tau_n(\theta)$, $n = 1, 2, \cdots, MJ-1$, are the time delays due to wave propagation and tap delays spanning the interval measured from the zero phase reference to the point at which the nth coefficient is applied. We refer to $\mathbf{d}(\theta, \omega)$ as the array response vector, which is also known as the steering vector or direction vector [13].

Based on the response vector, we briefly discuss the spatial aliasing problem encountered in array processing [13]. In conventional temporal sampling, aliasing implies ambiguity in temporal frequencies, i.e. signals with different frequencies have the same discrete-time sample series. Similarly, spatial aliasing corresponds to an ambiguity in spatial locations of the source signals. The implication of this spatial ambiguity is that sources at different locations have the same array response vector. For signals having the same frequency ω_0, but different DOAs θ_1 and θ_2 satisfying the condition of $\theta_1, \theta_2 \in [-\pi/2, \pi/2]$, aliasing implies that we have $\mathbf{d}(\theta_1, \omega_0) = \mathbf{d}(\theta_2, \omega_0)$, i.e.

$$\begin{aligned} e^{-j\omega_0\tau(\theta_1)} &= e^{-j\omega_0\tau(\theta_2)}, \\ e^{-j2\pi(\sin\theta_1)d/\lambda_0} &= e^{-j2\pi(\sin\theta_2)d/\lambda_0} \, . \end{aligned} \tag{2.19}$$

In order to be able to avoid aliasing, the condition of $|2\pi(\sin\theta)d/\lambda_0|_{\theta=\theta_1,\theta_2} < \pi$ has to be satisfied. Then we have $|d/\lambda_0 \sin\theta| < 1/2$. Since $|\sin\theta| \leqslant 1$, this requires that the array distance d should be less than $\lambda_0/2$. In the following, we will always set $d = \lambda_0/2$, with the assumption that the signal frequencies fulfil $\omega \in [0;\ \omega_0]$.

To describe the sensitivity of the array with respect to signals arriving from different spatial directions and with different frequencies, we use its beampattern, which is the amplitude response $|r(\theta, \omega)|$ of the beamformer with respect to both the angle of incidence and frequency of an impinging waveform. Suppose the array spacing d is set to be $\lambda_0/2$, where λ_0 is the wavelength of the maximum frequency component, and the temporal sampling frequency is twice the maximum frequency. Then the sampling period becomes $T = \lambda_0/2c = d/c$ and thus we have $\tau(\theta) = T\sin\theta$. Substituting $\Omega = \omega T$, where Ω is the normalised angular frequency,

from (2.16), we obtain

$$\begin{aligned} r(\theta,\omega) &= \sum_{m=0}^{M-1}\sum_{i=0}^{J-1} e^{-j\Omega(m\sin\theta+i)}\cdot w_{m,i}^{*} \\ r(\theta,\Omega) &= \sum_{m=0}^{M-1} e^{-jm\Omega\sin\theta}\sum_{i=0}^{J-1} e^{-ji\Omega}\cdot w_{m,i}^{*} \\ &= \sum_{m=0}^{M-1} e^{-jm\Omega\sin\theta}\cdot W_m(e^{j\Omega}) \ , \end{aligned} \tag{2.20}$$

where $W_m(e^{j\Omega})$ is the Fourier transform of the tapped-delay line $\tilde{\mathbf{w}}_m$ attached to the m-th sensor.

To draw the three-dimensional graph of the response with respect to frequency and DOA, we need to sample the continuous function $|r(\theta,\Omega)|$ according to different discrete values of θ and Ω. To calculate the beampattern for N_θ number of discrete DOA values and for N_Ω discrete temporal frequencies, an $N_\theta \times N_\Omega$ matrix is obtained holding the response samples on the defined DOA/frequency coordinates. As a simple example, consider an array having M=5 sensors and a tapped-delay line length of J=3. Suppose the weight vector is given as

$$\mathbf{w} = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T \ . \tag{2.21}$$

The beampattern of such an array is shown in Fig. 2.4 for $N_\Omega = 50$ and $N_\theta = 60$, where the gain is displayed on a logarithmic scale as $20\log_{10}|r(\theta,\Omega)|$.

In the example of Fig. 2.4, the weight coefficients were given fixed values and the resultant beamformer will maintain a fixed response independent of the signal/interference scenarios. In statistically optimum beamforming, the weight coefficients need to be updated based on the statistics of the array data. When the data statistics are unknown or time varying, adaptive optimisation is required. According to different signal environments and applicational requirements, different beamforming techniques may be employed such as the linearly constrained minimum variance beamformer [31] and the multiple sidelobe canceller [43], or we may invoke a reference signal [44] and the SNR maximisation [45] at the beamformer's output for adjusting the array weights. In this study the LCMV beamforming technique is of particular interest and will be further discussed in the next section.

2.3 Linearly Constrained Minimum Variance Beamforming

In linearly constrained minimum variance (LCMV) beamforming [31], the basic idea is to constrain the response of the beamformer such that the desired signals impinging on the array

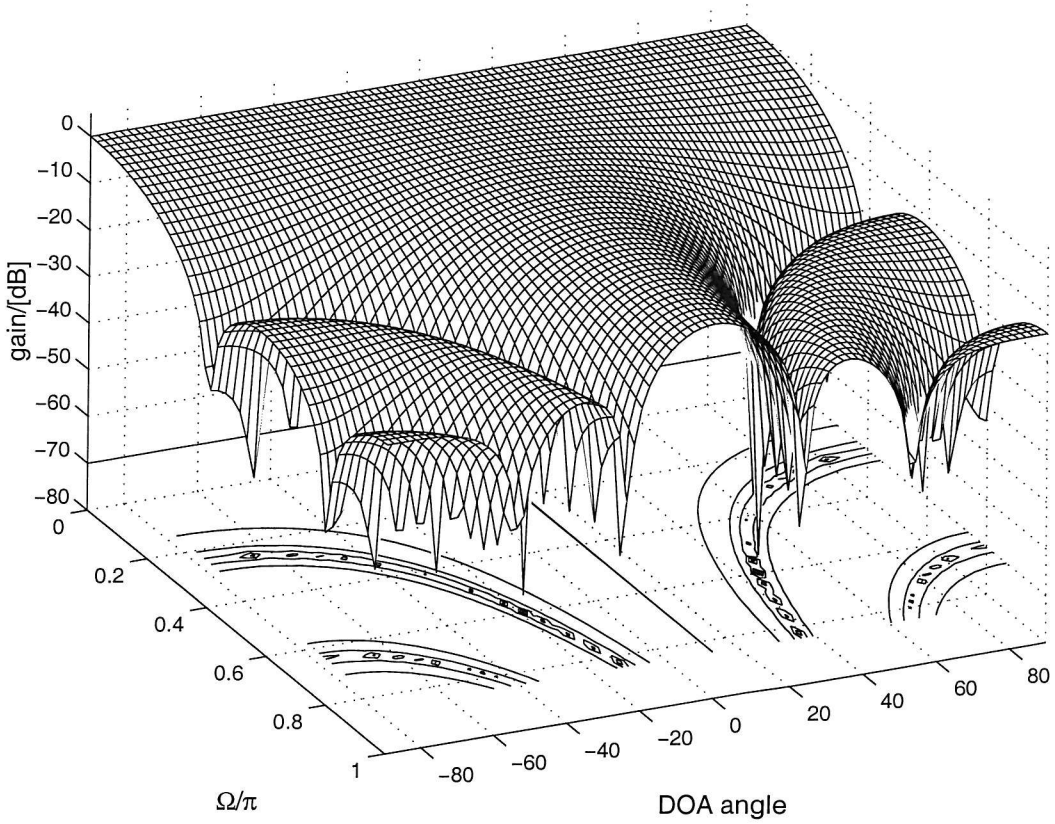


Fig. 2.4: A directivity pattern example of a linear array with M=5 and J=3.

from a specific direction pass with a specified gain and phase, while the variance or power of the output signal is minimised. As a result, the desired signals are preserved subject to a specified controlled response, while the contributions due to interfering signals and noise arriving from other directions are attenuated.

We have seen that the beamformer's response to a signal having a frequency ω and DOA θ can be expressed as in (2.17). In order to ensure that any signal having a frequency ω_0 and DOA θ_0 passes the beamformer with a specified response G, where G is a complex constant, we set this constraint to $\mathbf{w}^H\mathbf{d}(\theta_0, \omega_0) = G$. Noting that the expected value of the output power or variance is given by [31]

$$\mathcal{E}\{|e[n]|^2\} = \mathbf{w}^H\mathbf{R}_{xx}\mathbf{w} \ , \tag{2.22}$$

where $\mathcal{E}\{\cdot\}$ is the expectation operator and $\mathbf{R}_{xx}$ is the observed array data's correlation matrix, which is assumed to be positive definite and given in the form of

$$\mathbf{R}_{xx} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\} \ , \tag{2.23}$$

the LCMV problem of optimising the array weights can be formulated as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \mathbf{w}^H\mathbf{R}_{xx}\mathbf{w} \qquad \text{subject to} \qquad \mathbf{d}^H(\theta_0, \omega_0)\mathbf{w} = G^* \ . \tag{2.24}$$

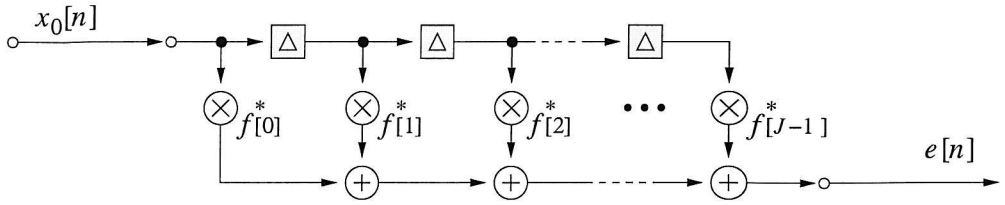


Fig. 2.5: The equivalent processor for a signal arriving from broadside.

The problem formulated in (2.24) obeys a single constraint. However, it can be generalised to multiple linear constraints for attaining an enhanced control over the beamformer's response, for example, by specifying more DOA angles and frequencies. If there are $r < MJ$ number of linearly independent constraints imposed on $\mathbf{w}$, we can formulate the constraints in matrix form as

$$\mathbf{C}^H\mathbf{w} = \mathbf{f}\,, \tag{2.25}$$

where the $MJ \times r$ dimensional matrix $\mathbf{C}$ is termed as the constraint matrix, while the r dimensional vector $\mathbf{f}$ is the so called response vector. In the next section we will discuss a special class of constraints in the broadband beamforming case, which is assumed throughout this thesis.

2.3.1 Formulation of Constraints

The constraints imposed on the LCMV beamformer ensure that the beamformer has the required response to signals arriving from specified angles and at given frequencies, no matter what values are assigned to the weights. In different applications, there are different constraints, one of which is that for a prescribed direction, the response of the array is maintained constant. The resultant beamformer is referred to as the minimum variance distortionless response (MVDR) beamformer [46,47]. All our simulations throughout this thesis are based on this type of beamformer. The application of such constraints is based on a simple relation between the response in look direction and the weights in the array. Based on Fig. 2.3, in the following we will briefly introduce this approach.

Assume that the signal of interest arrives from broadside, $\theta = 0$, i.e. the signal of interest impinges perpendicular to the line of sensors. If this is not the case, the array can be steered either mechanically or electrically by imposing appropriate time delays, or phase shifts in the narrowband beamforming scenario, immediately after each sensor output, such that the signals incident on the array from directions of interest other than broadside appear as identical replicas of one another at the outputs of the steering delay elements. With this pre-steering, the signal of interest can be treated as if it had arrived from broadside. Thus, identical signal

components appear at the sensors simultaneously and pass in parallel through the tapped-delay lines following the sensors. Hence, the FIR filters seen in Fig. 2.3 appear to be driven by a common input. As far as the signal of interest is concerned, the array processor is equivalent to a single tapped-delay line in which each weight is equal to the sum of weights in the corresponding vertical column, as indicated in Fig. 2.5 [31], where we have

$$f^*[j] = \sum_{m=0}^{M-1} w_{m,j}^* \tag{2.26}$$

with $j = 0, 1, \cdots, J-1$.

These summed weights in the equivalent tapped-delay line form a temporal filter, specifying the frequency response of the beamformer to the signal incident from the broadside and must be selected appropriately to give the desired response characteristic in the look direction. In the MVDR beamforming case, this response is a pure integer delay, i.e. one of the taps $f^*[j]$, $j = 0, 1, \cdots, J-1$ will be 1 and all the others are zero.

Thus, the broadside constraint can be formulated in the following way

$$\mathbf{C}^H \mathbf{w} = \mathbf{f}\,, \tag{2.27}$$

where

$$\mathbf{f} = \begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[J-1] \end{bmatrix} \tag{2.28}$$

and

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_0 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{c}_0 \end{bmatrix} \in \mathbb{C}^{MJ \times J} \tag{2.29}$$

with

$$\mathbf{c}_0 = [1 \;\; 1 \;\; \cdots \;\; 1]^T \in \mathbb{C}^{M \times 1}\,. \tag{2.30}$$

Note that the response vector $\mathbf{f}$ is defined to contain the complex conjugate of a desired gain.

Adaptive arrays obeying this broadside constraint can efficiently suppress sources of interference from other directions and achieve the maximum attainable signal-to-interference-plus-noise ratio (SINR) [42]. If, however, the desired signal does not appear exactly in phase at the outputs of the steering delays due to an error in the steering angle, the array will tend to null out the desired signal as if it were an interfering signal. A remedy to this problem is to impose

derivative constraints on the main beam of the array [30, 43, 48, 49]. Derivative constraints set the derivatives of the array's response with respect to the DOA in the desired direction to zero. The higher the order of the derivative constraints, the broader will be the beam pointing to the desired direction.

According to [48], the first-order derivative constraint formulated for a uniformly spaced linear array can be expressed as

$$\mathbf{c}_1^H \mathbf{w}_j = 0, \quad j = 0, 1, \cdots, J-1 \,, \tag{2.31}$$

where $\mathbf{c}_1$ is the M-dimensional vector given by

$$\mathbf{c}_1 = [c_1[0] \;\; c_1[1] \;\; \cdots \;\; c_1[M-1]]^T \in \mathbb{C}^{M\times 1} \tag{2.32}$$

with

$$c_1[m] = m - n_0, \quad m = 0, 1, \cdots, M-1 \,, \tag{2.33}$$

where n_0 is the phase origin point. If we choose the middle point along the array line as the reference point and set the signal phase at this point to be zero, then we have $n_0 = \frac{M-1}{2}$.

The $(S-1)$-order derivative constraint is

$$\mathbf{c}_{S-1}^H \mathbf{w}_j = 0, \quad j = 0, 1, \cdots, J-1 \,, \tag{2.34}$$

where $\mathbf{c}_{S-1}$ is given by

$$\mathbf{c}_{S-1} = [c_{S-1}[0] \;\; c_{S-1}[1] \;\; \cdots \;\; c_{S-1}[M-1]]^T \in \mathbb{C}^{M\times 1} \tag{2.35}$$

with

$$c_{S-1}[m] = (m - n_0)^{S-1}, \quad m = 0, 1, \cdots, M-1 \,. \tag{2.36}$$

In this context, we can consider the constraint in (2.27) to (2.30) as zero-order derivative constraint. Combining (2.30), (2.32) and (2.35) together, we obtain the general formulation in the following when we impose derivative constraints of zero until $S-1$ orders on the beamformer

$$\mathbf{C}^H \mathbf{w} = \mathbf{f} \,, \tag{2.37}$$

where

$$\mathbf{C} = \left[\hat{\mathbf{C}}_0 \ldots \hat{\mathbf{C}}_{S-1}\right] \in \mathbb{C}^{MJ\times SJ} \qquad \text{with} \quad \hat{\mathbf{C}}_i = \begin{bmatrix} \mathbf{c}_i & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{c}_i \end{bmatrix} \in \mathbb{C}^{MJ\times J} \tag{2.38}$$

and

$$\mathbf{f} = [f[0] \;\; f[1] \;\cdots\; f[J-1] \;\; 0 \;\; 0 \;\cdots\; 0]^T \in \mathbb{C}^{SJ\times 1} \,. \tag{2.39}$$

2.3.2 Optimum Solution to the LCMV Problem

The solution to the general LCMV problem in (2.24) and (2.25) can be obtained by the method of Lagrange multipliers [11], which is outlined here.

The Lagrangian is formed by the objective function $\mathcal{E}\{|e[n]|^2\} = \mathbf{w}^H\mathbf{R}_{xx}\mathbf{w}$, plus the real part of the constraint function of $\mathbf{C}^H\mathbf{w} - \mathbf{f}$, weighted elementwise by the r-dimensional vector of undetermined Lagrange multipliers $\boldsymbol{\lambda}$, which is given by

$$\mathbf{w}^H\mathbf{R}_{xx}\mathbf{w} + \boldsymbol{\lambda}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \boldsymbol{\lambda}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*) \,. \tag{2.40}$$

Note that the gradient of the constraint function constituted by the second and third terms of (2.40) must be linearly independent of each other for the Lagrange multipliers to hold, i.e. the columns of $\mathbf{C}$ must have full rank [50]. Differentiating the function in (2.40) with respect to $\mathbf{w}^*$, we have

$$\mathbf{R}_{xx}\mathbf{w} + \mathbf{C}\boldsymbol{\lambda} \,. \tag{2.41}$$

Setting this result equal to zero, we obtain the optimal weight vector $\mathbf{w}_{\text{opt}}$ in terms of the Lagrange multipliers as follows

$$\mathbf{w}_{\text{opt}} = -\mathbf{R}_{xx}^{-1}\mathbf{C}\boldsymbol{\lambda} \,. \tag{2.42}$$

Since the optimal weight vector must satisfy (2.25), we have

$$-\mathbf{C}^H\mathbf{R}_{xx}^{-1}\mathbf{C}\boldsymbol{\lambda} = \mathbf{f} \,. \tag{2.43}$$

Solving this equation for $\boldsymbol{\lambda}$ and finally substituting $\boldsymbol{\lambda}$ into (2.42) yields

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}_{xx}^{-1}\mathbf{C})^{-1}\mathbf{f} \,, \tag{2.44}$$

which represents the solution to the constrained optimisation problem in (2.24) and (2.25) [11].

2.3.3 Frost's Algorithm for LCMV Beamforming

From (2.44), we know that for the LCMV beamformer the optimum solution $\mathbf{w}_{\text{opt}}$ is based on the statistics of the array data. However, in numerous applications the second order statistics of the array data required in the correlation matrix (2.23) are unknown or may change over time. In this case, constrained adaptive algorithms can be employed for determining the coefficients in $\mathbf{w}$. One such approach is given by Frost's algorithm as proposed in [31], which we will briefly review in the following.

At the beginning, we set the weight vector to $\mathbf{w}[0] = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{f}$ for initialisation, which satisfies the constraint in (2.25). At each iteration, the vector $\mathbf{w}$ is updated in the direction of

the negative gradient expressed in (2.41) by a step proportional to a scaling factor μ according to

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu\big(\mathbf{R}_{xx}\mathbf{w}[n] + \mathbf{C}\boldsymbol{\lambda}[n]\big) \ . \tag{2.45}$$

Since $\mathbf{w}[n+1]$ must satisfy the constraint in (2.25), we can substitute (2.45) into (2.25) and solve for the Lagrange multipliers $\boldsymbol{\lambda}[n]$. Then we substitute $\boldsymbol{\lambda}[n]$ into the iteration equation (2.45) and arrive at

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu\big(\mathbf{I} - \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H\big)\mathbf{R}_{xx}\mathbf{w}[n] + \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\big(\mathbf{f} - \mathbf{C}^H\mathbf{w}[n]\big) \ . \tag{2.46}$$

Upon defining the short-hand of $\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H$, the algorithm in (2.46) can be rewritten as

$$\mathbf{w}[n+1] = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{f} + \mathbf{P}\big(\mathbf{w}[n] - \mu\mathbf{R}_{xx}\mathbf{w}[n]\big) \ . \tag{2.47}$$

Not knowing the true second order statistics $\mathbf{R}_{xx}$, the correlation matrix can be replaced by its simple approximation $\tilde{\mathbf{R}}_{xx} = \mathbf{x}\mathbf{x}^H$. This results in the minimisation of the instantaneous square error rather than the mean square error, and leads to the so-called stochastic constrained algorithm

$$\mathbf{w}[n+1] = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{f} + \mathbf{P}\big(\mathbf{w}[n] - \mu e^*[n]\mathbf{x}[n]\big) \ , \tag{2.48}$$

which is also known as the Frost's algorithm.

Instead of using the constrained adaptive algorithm of (2.48), Griffith [29, 30] proposed an alternative, but efficient implementation of the LCMV problem, which is referred to as the generalised sidelobe canceller (GSC) [29, 30]. The GSC employs an unconstrained adaptive algorithm and is presented in the next section.

2.4 Generalised Sidelobe Canceller

2.4.1 GSC Structure

The GSC can be viewed as a scheme designed for transforming the constrained minimisation problem of (2.24) into an unconstrained form. The evolution of the GSC structure from the LCMV problem is illustrated in Fig. 2.6(a)-(c). The resultant unconstrained problem can then be readily solved using well-known standard adaptive algorithms, such as the least mean square or recursive least squares algorithms [51] .

The basic philosophy of the GSC is to decompose the array weight vector $\mathbf{w}$, which is the quantity to be optimised in the LCMV problem of Fig. 2.6(a), into two orthogonal components

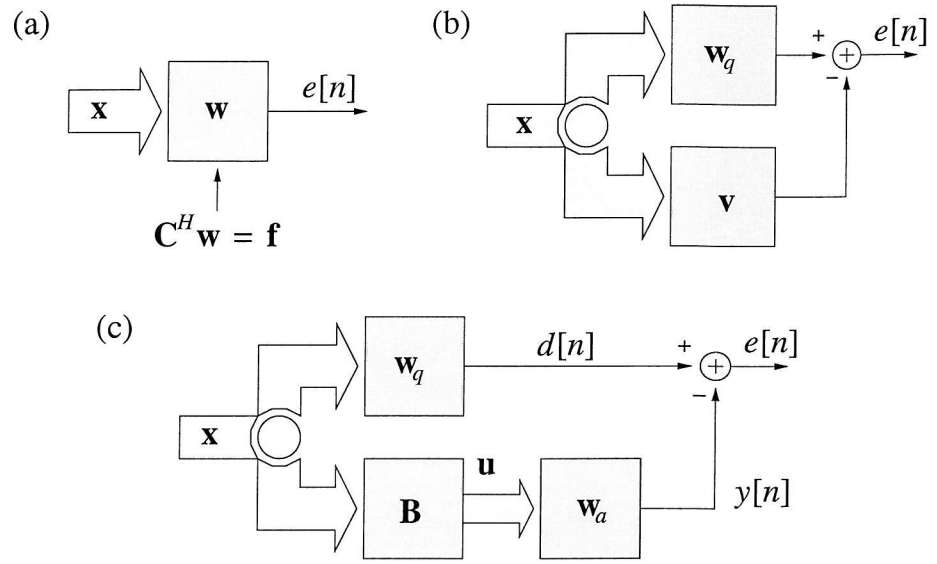


Fig. 2.6: From the LCMV problem to the unconstrained GSC beamformer: (a) LCMV beamformer; (b) separation of constraint output; (c) unconstrained GSC problem by projection of the data into the nullspace of the Hermitian constraint matrix [1].

$\mathbf{w}_q$ and $-\mathbf{v}$ defined in the context of $\mathbf{w} = \mathbf{w}_q - \mathbf{v}$, as seen in Fig. 2.6(b). The vector $\mathbf{w}_q$ lies in the range of the matrix $\mathbf{C}^H$, while the component $\mathbf{v}$ is contained in the null space of $\mathbf{C}^H$, i.e. the space of all $\mathbf{v}$ fulfilling $\mathbf{C}^H\mathbf{v} = 0$. Together the range and null space of a matrix span the entire space [50], so this decomposition can be used to represent any $\mathbf{w}$. The vector $\mathbf{w}_q$ is obtained by solving the constraint equation in (2.25) using the pseudo-inverse of $\mathbf{C}^H$ [52,53] as follows

$$\mathbf{w}_q = (\mathbf{C}^H)^\dagger \mathbf{f} = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{f}\,, \tag{2.49}$$

where $\{\cdot\}^\dagger$ indicates the pseudo-inverse. The resultant structure is shown in Fig. 2.6(b). In a quiet environment (quiescent condition), for which the received signal consists of white noise only, $\mathbf{w}_q$ will be the optimum solution of (2.44) as the correlation matrix $\mathbf{R}_{xx} = \sigma^2\mathbf{I}$, where σ^2 is the noise variance and $\mathbf{I}$ is the identity matrix. For this reason $\mathbf{w}_q$ has been named "quiescent vector".

The vector $\mathbf{v}$ can be expressed as a linear combination of basis vectors of the null space of $\mathbf{C}^H$. If the columns of a matrix $\mathbf{B} \in \mathbb{C}^{MJ\times MJ-r}$ form such a basis, i.e., if we have

$$\mathbf{C}^H\mathbf{B} = \mathbf{0}\,, \tag{2.50}$$

then we can write $\mathbf{v} = \mathbf{B}\mathbf{w}_a$, where r is the number of linearly independent constraints in $\mathbf{C}$ as mentioned in Sec. 2.3 and $\mathbf{w}_a$ is the vector to linearly combine the basis vectors in $\mathbf{B}$ to form $\mathbf{v}$. The matrix $\mathbf{B}$ can be obtained from $\mathbf{C}$ using orthogonalisation methods such as the QR decomposition [52,54]. We note that the most widely used methods in the context of the GSC

are the cascaded columns of difference (CCD) method [55] and the singular value decomposition (SVD) method [30, 54], which will be reviewed in Sec. 2.4.3. The structure with the described factorization of $\mathbf{v}$ is given in Fig. 2.6(c), where

$$y[n] = \mathbf{w}_a^H \mathbf{u} \, . \tag{2.51}$$

Since $\mathbf{w}_q$ is determined by the constrained specifications, the LCMV problem reduces to that of finding the weights $\mathbf{w}_a$. The choice for $\mathbf{w}_q$ and $\mathbf{B}$ implies that the constraints are satisfied independently of $\mathbf{w}_a$. Note that we can substitute $\mathbf{w} = \mathbf{w}_q - \mathbf{B}\mathbf{w}_a$ into (2.24). Since the constraint has been fulfilled by this factorization, the optimisation is not subject to the constraints any more. Thus, a modified LCMV formulation is obtained as

$$\mathbf{w}_a = \arg\min_{\mathbf{w}_a} [\mathbf{w}_q - \mathbf{B}\mathbf{w}_a]^H \mathbf{R}_{xx} [\mathbf{w}_q - \mathbf{B}\mathbf{w}_a] \, . \tag{2.52}$$

The solution to (2.52) can be obtained by (2.44). As $\mathbf{w}_{\text{opt}} = \mathbf{w}_q - \mathbf{B}\mathbf{w}_{a,\text{opt}}$, we have

$$\mathbf{B}\mathbf{w}_{a,\text{opt}} = \mathbf{w}_q - \mathbf{R}_{xx}^{-1}\mathbf{C}(\mathbf{C}^H \mathbf{R}_{xx}^{-1}\mathbf{C})^{-1}\mathbf{f} \, . \tag{2.53}$$

Multiplying the two sides of (2.53) by $\mathbf{B}^H\mathbf{R}_{xx}$, respectively and also noting that $\mathbf{B}^H\mathbf{C} = \mathbf{0}$, we can get

$$\mathbf{B}^H \mathbf{R}_{xx} \mathbf{B}\mathbf{w}_{a,\text{opt}} = \mathbf{B}^H \mathbf{R}_{xx} \mathbf{w}_q - \mathbf{0} \, . \tag{2.54}$$

Further multiplying the two sides of (2.54) by the inverse of $\mathbf{B}^H\mathbf{R}_{xx}\mathbf{B}$, the final solution to (2.52) [11, 13] is given by

$$\mathbf{w}_{a,\text{opt}} = (\mathbf{B}^H \mathbf{R}_{xx} \mathbf{B})^{-1} \mathbf{B}^H \mathbf{R}_{xx} \mathbf{w}_q \, . \tag{2.55}$$

If the constraints are designed to present a specified response to signals impinging on the antenna array from a set of directions and at different frequencies, then the columns of $\mathbf{B}$ will block those directions and frequencies. This concept leads to the term "blocking matrix" often used for describing $\mathbf{B}$. Those signals are only processed by $\mathbf{w}_q$ in the upper branch of the GSC in Fig. 2.6(c). Since $\mathbf{w}_q$ is designed to satisfy the specified constraints, the desired signals will pass through the beamformer with a desired gain and phase independent of $\mathbf{w}_a$, while the interfering signals and noise pass through the upper branch of the GSC with a response determined by $\mathbf{w}_q$. In the lower branch, since the desired signals are blocked, only the interfering signals and the noise can pass. When adapting $\mathbf{w}_a$, the scheme will tend to cancel the interference and noise component in the upper path, while minimising the variance or power of the output signal $e[n]$.

In Fig. 2.6(c), the upper branch output $d[n]$ is obtained by $d[n] = \mathbf{w}_q^H \mathbf{x}$. For the lower branch, $\mathbf{u} = \mathbf{B}^H\mathbf{x}$ and $y[n] = \mathbf{w}_a^H\mathbf{u}$. To calculate the blocking matrix output $\mathbf{u}$, $MJ(MJ - r)$ multiplications are required every sampling period, which represents a considerable burden for

large values of M and J. Based on the constraint formulations in Sec. 2.3.1, we will circumvent this potential problem by simplification of the blocking matrix and introducing the GSC with tapped-delay lines in Sec. 2.4.2.

2.4.2 GSC with Tapped-Delay Lines

Recall from Sec. 2.3.1, that when imposing derivative constraints of zero up to an order of $S-1$ on the beamformer, the constraint matrix and the corresponding response vector can be expressed as

$$\mathbf{C} = \left[\hat{\mathbf{C}}_0 \dots \hat{\mathbf{C}}_{S-1}\right] \in \mathbb{C}^{MJ \times SJ} \qquad \text{with} \quad \hat{\mathbf{C}}_i = \begin{bmatrix} \mathbf{c}_i & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{c}_i \end{bmatrix} \in \mathbb{C}^{MJ \times J} \tag{2.56}$$

and

$$\mathbf{f} = [f[0] \;\; f[1] \;\cdots\; f[J-1] \;\; 0 \;\; 0 \;\cdots\; 0]^T \in \mathbb{C}^{SJ \times 1} \,. \tag{2.57}$$

As the blocking matrix $\mathbf{B}$ is composed of the basis vectors of the null space of $\mathbf{C}$, we have $\mathbf{B} \in \mathbb{C}^{MJ \times (M-S)J}$. Assume that $\mathbf{B}$ has the following block diagonal form:

$$\mathbf{B} = \begin{bmatrix} \tilde{\mathbf{B}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{B}} & \dots & \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{B}} \end{bmatrix}, \tag{2.58}$$

where $\tilde{\mathbf{B}}$ is an $M \times (M-S)$-dimensional matrix. The condition for the blocking matrix $\mathbf{B}$ in (2.50) can then be expressed as

$$\mathbf{C}^H\mathbf{B} = \begin{bmatrix} \hat{\mathbf{C}}_0^H\mathbf{B} \\ \hat{\mathbf{C}}_1^H\mathbf{B} \\ \vdots \\ \hat{\mathbf{C}}_{S-1}^H\mathbf{B} \end{bmatrix} = \mathbf{0} \,, \tag{2.59}$$

where the rows $\hat{\mathbf{C}}_i^H\mathbf{B}$, $i = 0, 1, \cdots, S-1$, simplify to

$$\hat{\mathbf{C}}_i^H\mathbf{B} = \begin{bmatrix} \mathbf{c}_i^H\tilde{\mathbf{B}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_i^H\tilde{\mathbf{B}} & \dots & \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{c}_i^H\tilde{\mathbf{B}} \end{bmatrix}. \tag{2.60}$$

Then, as long as $\tilde{\mathbf{B}}$ fullfills

$$\tilde{\mathbf{C}}^H\tilde{\mathbf{B}} = \mathbf{0} \qquad \text{where} \quad \tilde{\mathbf{C}} = [\mathbf{c}_0 \ \cdots \mathbf{c}_{S-1}] \ , \tag{2.61}$$

the original blocking matrix $\mathbf{B}$ will automatically satisfy (2.50).

Under this construction, the blocking matrix output $\mathbf{u} = \mathbf{B}^H\mathbf{x}$ becomes

$$\mathbf{u} = \begin{bmatrix} \tilde{\mathbf{B}}^H\mathbf{x}[n] \\ \tilde{\mathbf{B}}^H\mathbf{x}[n-1] \\ \vdots \\ \tilde{\mathbf{B}}^H\mathbf{x}[n-J+1] \end{bmatrix} . \tag{2.62}$$

Assume that we have $\mathbf{u}[n] = [u_0[n], \ldots, u_{M-S-1}[n]]^T = \tilde{\mathbf{B}}^H\mathbf{x}[n]$. Then we see that the input signal $\mathbf{u}$ to the adaptive block $\mathbf{w}_a$ is a series of tapped-delays of the vectorial signal $\mathbf{u}[n]$, i.e.

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}[n] \\ \mathbf{u}[n-1] \\ \vdots \\ \mathbf{u}[n-J+1] \end{bmatrix} . \tag{2.63}$$

Thus, it is possible to apply a smaller blocking matrix $\tilde{\mathbf{B}}$ straight to the sensor signal vector $\mathbf{x}[n]$, rather than applying the fullsize matrix $\mathbf{B}$. As a result, the output vector $\mathbf{u}[n]$ from $\tilde{\mathbf{B}}$ is fed into $M - S$ tapped-delay lines of length J, which will be shown later as a block diagram in Fig. 2.7.

Considering the quiescent vector, the sparse nature of $\mathbf{C}$ allows us to rearrange (2.49) and obtain a simplified form for $\mathbf{w}_q$ [30] in the following:

$$\mathbf{w}_q = \hat{\mathbf{f}} \bigotimes \tilde{\mathbf{w}}_q \ , \tag{2.64}$$

where the operator $\bigotimes$ denotes the Kronecker product operator [56], and

$$\hat{\mathbf{f}} = \begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[J-1] \end{bmatrix} , \tag{2.65}$$

$$\tilde{\mathbf{w}}_q = \tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H\tilde{\mathbf{C}})^{-1}\mathbf{e} \tag{2.66}$$

with $\mathbf{e} = [1, 0, \ldots, 0]^T$ being an $S \times 1$-dimensional vector. We derive the formulation for the

quiescent vector $\mathbf{w}_q$ in (2.64) by first permutating $\mathbf{C}$ to obtain a new constraint matrix

$$\bar{\mathbf{C}} = \begin{bmatrix} \tilde{\mathbf{C}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}} & \dots & \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{C}} \end{bmatrix}_{MJ \times SJ} . \tag{2.67}$$

Correspondingly, the constraint of Equation (2.37) becomes

$$\bar{\mathbf{C}}^H \mathbf{w} = \hat{\mathbf{f}} \bigotimes \mathbf{e} \tag{2.68}$$

for the sake of keeping the $(S-1)$ order derivative constraints unchanged.

From (2.49), we have

$$\begin{aligned} \mathbf{w}_q &= \bar{\mathbf{C}}(\bar{\mathbf{C}}^H \bar{\mathbf{C}})^{-1}(\hat{\mathbf{f}} \bigotimes \mathbf{e}) \\ &= \bar{\mathbf{C}} \cdot \begin{bmatrix} \tilde{\mathbf{C}}^H \tilde{\mathbf{C}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}^H \tilde{\mathbf{C}} & \dots & \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{C}}^H \tilde{\mathbf{C}} \end{bmatrix}^{-1} \cdot (\hat{\mathbf{f}} \bigotimes \mathbf{e}) \\ &= \begin{bmatrix} \tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H \tilde{\mathbf{C}})^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H \tilde{\mathbf{C}})^{-1} & \dots & \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H \tilde{\mathbf{C}})^{-1} \end{bmatrix} \cdot (\hat{\mathbf{f}} \bigotimes \mathbf{e}) . \end{aligned} \tag{2.69}$$

Now we arrive at

$$\mathbf{w}_q = \begin{bmatrix} f[0]\tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H \tilde{\mathbf{C}})^{-1}\mathbf{e} \\ f[1]\tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H \tilde{\mathbf{C}})^{-1}\mathbf{e} \\ \vdots \\ f[J-1]\tilde{\mathbf{C}}(\tilde{\mathbf{C}}^H \tilde{\mathbf{C}})^{-1}\mathbf{e} \end{bmatrix} = \begin{bmatrix} f[0]\tilde{\mathbf{w}}_q \\ f[1]\tilde{\mathbf{w}}_q \\ \vdots \\ f[J-1]\tilde{\mathbf{w}}_q \end{bmatrix} , \tag{2.70}$$

which leads to (2.64). As we have $d[n] = \mathbf{w}_q^H \mathbf{x}$, with this simplification of $\mathbf{w}_q$ we can express $d[n]$ as

$$d[n] = \hat{\mathbf{f}}^H \cdot \begin{bmatrix} \tilde{\mathbf{w}}_q^H \mathbf{x}[n] \\ \tilde{\mathbf{w}}_q^H \mathbf{x}[n-1] \\ \vdots \\ \tilde{\mathbf{w}}_q^H \mathbf{x}[n-J+1] \end{bmatrix} . \tag{2.71}$$

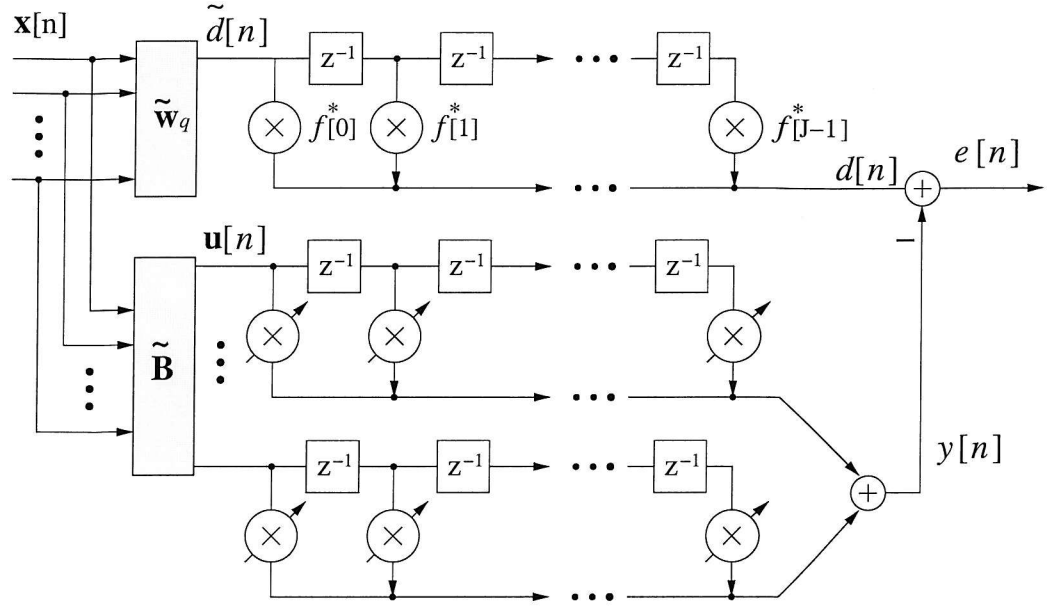


Fig. 2.7: A generalised sidelobe canceller with tapped-delay lines.

if we use the representation $\tilde{d}[n] = \tilde{\mathbf{w}}_q^H \mathbf{x}[n]$, Equation (2.71) can be further simplified, yielding

$$\begin{aligned} d[n] &= \hat{\mathbf{f}}^H \cdot \begin{bmatrix} \tilde{d}[n] \\ \tilde{d}[n-1] \\ \vdots \\ \tilde{d}[n-J+1] \end{bmatrix} \\ &= f^*[0] \cdot \tilde{d}[n] + f^*[1] \cdot \tilde{d}[n-1] + \cdots + f^*[J-1] \cdot \tilde{d}[n-J+1] \,. \end{aligned} \tag{2.72}$$

Now we can introduce the simplified GSC structure with tapped-delay lines as shown in Fig. 2.7. The output of the vector $\tilde{\mathbf{w}}_q$ is processed by an FIR filter with coefficients held in $\hat{\mathbf{f}}$, while the adaptive part of the GSC after $\tilde{\mathbf{B}}$ becomes a multi-channel adaptive filtering (MCAF) system, which will form the basis for our subband adaptive GSC proposed in Chapter 3. In this simplified structure, we still refer to $\tilde{\mathbf{w}}_q$ and $\tilde{\mathbf{B}}$ as the quiescent vector and blocking matrix, respectively.

For MVDR beamformers, only one coefficient of $\hat{\mathbf{f}}$ is unity and all the others are zero (see Sec. 2.3.1); thus the FIR filter in the upper branch of Fig. 2.7 becomes a pure delay line as shown in Fig. 2.8. Without loss of generality, we can set $\mathbf{f} = [1 \;\; 0 \;\; 0 \;\; \cdots \;\; 0]^T$. In this case the delay in Fig. 2.8 will be zero.

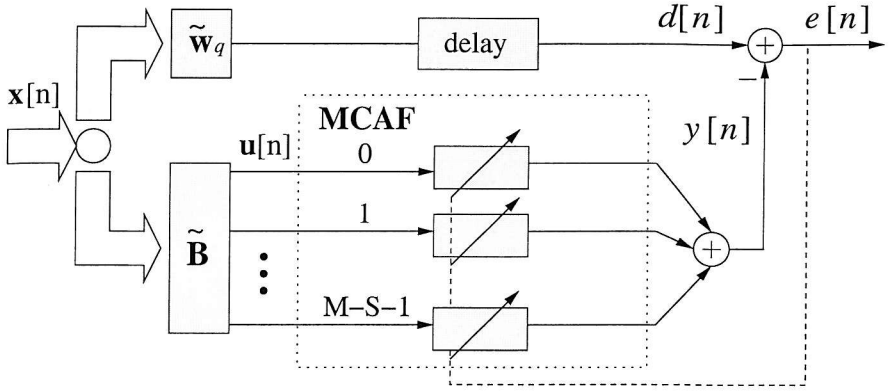


Fig. 2.8: A GSC for MVDR beamforming.

2.4.3 Blocking Matrix Design

In the GSC structure, we have to find the proper blocking matrix $\mathbf{B}$, which can fulfil the requirement formulated in (2.50). By the rearrangement outlined in Sec. 2.4.2, this problem is reduced to a suitable blocking matrix $\tilde{\mathbf{B}}$. As mentioned in Sec. 2.4.1, such a blocking matrix can be obtained by invoking CCD [55] or SVD methods [30, 54], which we briefly review under the constraints specified in Sec. 2.3.1. We will find later in Chap. 4 of this thesis, that the blocking matrix could be constructed quite differently from the approach of these two methods with the aim of satisfying certain specific characteristics, which can be exploited for reducing the complexity of the subband adaptive GSC [32, 33] and the transform-domain GSC [21].

Cascaded Columns of Differencing

The CCD method was first proposed in [29] and then systematically derived in [55]. The blocking matrix obtained by this method is formed by S cascaded columns of differencing to fulfil the derivative constraints in (2.37) as shown in Fig. 2.9.

In matrix form, the blocking matrix can be formulated as [55]

$$\tilde{\mathbf{B}} = \mathbf{B}_M \cdot \mathbf{B}_{M-1} \cdots \mathbf{B}_{M-S+1} \,, \tag{2.73}$$

where we have

$$\mathbf{B}_i = \begin{bmatrix} 1 & -1 & & & \mathbf{0} \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ \mathbf{0} & & & 1 & -1 \end{bmatrix}^T \in \mathbb{C}^{i \times i-1} \tag{2.74}$$

with $i = M, M-1, \cdots, M-S+1$. It is intuitively clear from Fig. 2.9 that no signal from

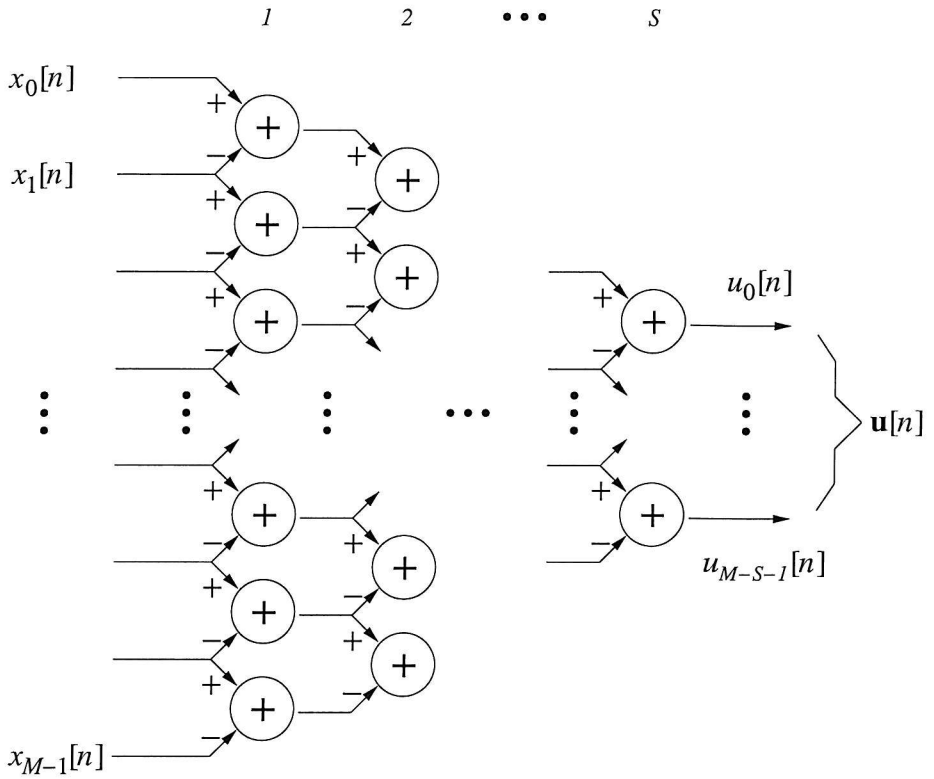


Fig. 2.9: Blocking matrix obatained by S cascaded columns of differencing.

broadside can pass through the blocking matrix for $S = 1$. For $S > 1$, the width of the mainlobe, centered at broadside, will be widened with increasing S.

Singular Value Decomposition

The singular value decomposition theorem [50, 51, 53] states that, given a matrix $\mathbf{A}$, there exist two unitary matrices $\mathbf{U}$ and $\mathbf{V}$, such that we have

$$\mathbf{U}^H \mathbf{A} \mathbf{V} = \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.75}$$

or

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^H , \tag{2.76}$$

where $\mathbf{\Sigma}$ is an $r \times r$ diagonal matrix containing the ordered positive definite singular values of $\mathbf{A}$. The variable r is the rank of $\mathbf{A}$ and represents the number of linearly independent columns in this matrix $\mathbf{A}$.

Let us separate matrix $\mathbf{U}$ into two parts as follows

$$\mathbf{U} = \left[\mathbf{U}_r \tilde{\mathbf{U}}_r \right] , \tag{2.77}$$

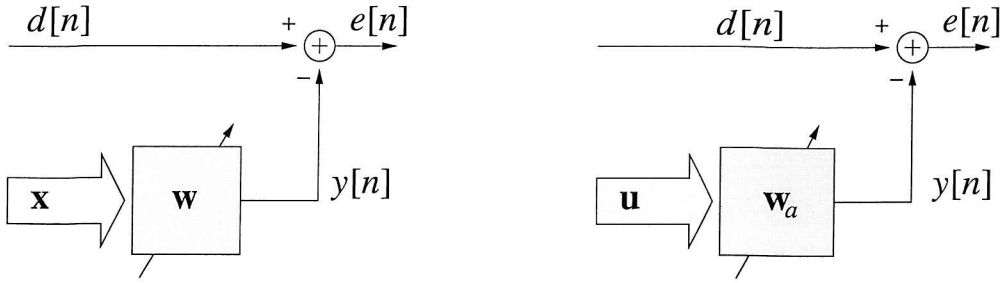


Fig. 2.10: (a) Standard generic adaptive filter setup and (b) the adaptive part of a GSC.

where $\mathbf{U}_r$ holds the first r columns of the matrix $\mathbf{U}$, whereas $\tilde{\mathbf{U}}_r$ holds the remaining columns of $\mathbf{U}$, then it is easy to see that

$$\tilde{\mathbf{U}}_r^H \mathbf{A} = \mathbf{0}\,, \tag{2.78}$$

i.e. $\tilde{\mathbf{U}}_r$ forms a basis for the null space of $\mathbf{A}^H$ [50, 53]. If we replace the matrix $\mathbf{A}$ by the constraint matrix $\mathbf{C}$ or $\tilde{\mathbf{C}}$ in the SVD decomposition, the resultant $\tilde{\mathbf{U}}_r$ will be our desired blocking matrix $\mathbf{B}$ or $\tilde{\mathbf{B}}$.

Compared to the CCD method [55], the SVD approach [30, 54] constitutes a more general way of obtaining the blocking matrix, because it can be applied not only to the broadside constraint of (2.37), but also to the general constraint of (2.25).

2.5 Adaptive Algorithms for GSC Structure

As mentioned in Sec. 2.4, the GSC structure enables us to apply standard adaptive filtering algorithms to the optimisation of the weight vector. In Fig. 2.10, the adaptive part of the GSC is compared to the generic setup of an adaptive filter [51,57], where the weight vector $\mathbf{w}_a$ of the GSC corresponds to the weight vector $\mathbf{w}$ of the general adaptive filter, and the blocking matrix output $\mathbf{u}$ in (2.63) formed by the tapped delays of the multichannel signal $\mathbf{u}[n]$ corresponds to the general adaptive filter's input vector $\mathbf{x}$. The output signal $y[n] = \mathbf{w}^H\mathbf{x}$ or $\mathbf{w}_a^H\mathbf{u}$ is compared to the desired signal $d[n]$, resulting in an error signal $e[n] = d[n] - y[n]$, which can be used for adjusting the weights $\mathbf{w}$ according to some criterion. Usually, this criterion is to minimise the error in a mean square or weighted sum of squares sense [51].

In the following, based on the standard generic adaptive filter setup in Fig. 2.10, the least mean square (LMS), the normalised LMS (NLMS) as well as the recursive least squares (RLS) algorithms will be briefly reviewed. Note that here $\mathbf{x}$ is a general input signal vector and $\mathbf{w}$ a general weight vector not to be confused with the definitions of $\mathbf{x}$ and $\mathbf{w}$ in (2.11) as the array

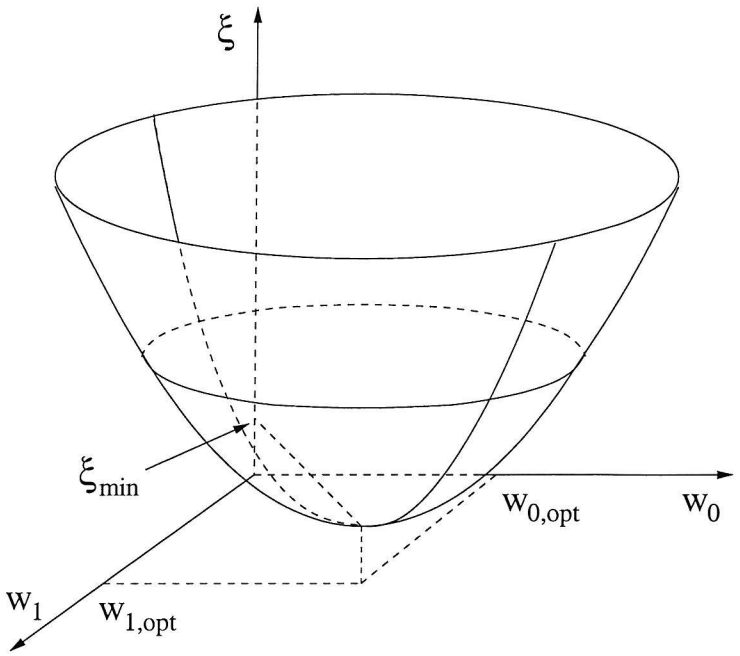


Fig. 2.11: Mean square error cost function ξ for the case of a weight vector $\mathbf{w}$ with 2 coefficients.

input signal and beamformer weights, respectively. We will include the time index n and use $\mathbf{x}[n]$ and $\mathbf{w}[n]$ to indicate their values at the time instant n in our following discussions.

2.5.1 Least Mean Square Algorithm

The LMS algorithm is a stochastic gradient technique based on the particular shape of the cost function employed. This cost function ξ, which is constituted by the mean square error (MSE), can be formulated as

$$\begin{aligned}\xi &= \mathcal{E}\{e[n] \cdot e^*[n]\} \\ &= \mathcal{E}\Big\{\big(d[n] - \mathbf{w}[n]^H\mathbf{x}[n]\big) \cdot \big(d[n] - \mathbf{w}[n]^H\mathbf{x}[n]\big)^H\Big\} \\ &= \sigma_{dd}^2 - \mathbf{w}[n]^H\mathbf{p} - \mathbf{p}^H\mathbf{w}[n] + \mathbf{w}[n]^H\mathbf{R}_{xx}\mathbf{w}[n]\ ,\end{aligned}$$

where $\sigma_{dd}^2 = \mathcal{E}\{|d[n]|^2\}$, $\mathbf{p} = \mathcal{E}\{\mathbf{x}[n]d[n]^*\}$ and $\mathbf{R}_{xx} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n]\}$. The operator $\{\cdot\}^*$ denotes complex conjugation.

Method of Steepest Descent

The MSE $\xi(\mathbf{w})$ is dependent on the elements of the tap-weight vector and — provided that the covariance matrix $\mathbf{R}_{xx}$ has full rank — has the shape of a hyperparabola with a unique global minimum ξ_{min} , as illustrated for the 2-dimensional case in Fig.2.11. Hence, gradient techniques can be successfully employed in order to search for this minimum. This can be achieved by an

update rule involving successive corrections of the tap-weight vector $\mathbf{w}[n]$ from an initial vector in the direction of the negative gradient of the MSE, which can be expressed as

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu\nabla(\xi[n]) \; . \tag{2.79}$$

The factor μ is a positive real-valued constant weighting the amount of innovation applied at each update. The variable $\nabla(\xi[n])$ denotes the value of the gradient vector at time n, which is formulated as

$$\nabla\xi[n] = \frac{\partial\xi[n]}{\partial\mathbf{w}} \; . \tag{2.80}$$

Using Wirtinger's calculus [58], the gradient vector can be evaluated as

$$\nabla(\xi[n]) = -\mathbf{p} + \mathbf{R}_{xx}\mathbf{w}[n] \; . \tag{2.81}$$

Thus, the updated value of the tap-weight vector for the standard filter configuration can be computed as

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu\big(\mathbf{p} - \mathbf{R}_{xx}\mathbf{w}[n]\big) \; . \tag{2.82}$$

The parameter μ controls the size of the correction at each iteration step and is referred to as the step-size parameter.

Stochastic Gradient Technique

A simplification of the method of steepest descent in (2.82) is to replace the expectation values, as used for the covariance matrix $\mathbf{R}_{xx}$ and the cross-correlation vector $\mathbf{p}$, by appropriate estimates. The use of single sample estimates leads directly to the least mean square algorithm, where $\mathbf{R}_{xx}$ and $\mathbf{p}$ are replaced by

$$\hat{\mathbf{R}}_{xx}[n] = \mathbf{x}[n]\mathbf{x}^H[n] \qquad \text{and} \qquad \hat{\mathbf{p}} = \mathbf{x}[n]d^*[n] \; . \tag{2.83}$$

These are instantaneous estimates based on the input vector $\mathbf{x}[n]$ and the desired signal $d[n]$. Substituting the estimate in the steepest-descent algorithm of (2.82) results in the simplified formula of

$$\begin{aligned} \mathbf{w}[n+1] &= \mathbf{w}[n] + \mu\big(d^*[n] - \mathbf{x}^H[n]\mathbf{w}[n]\big)\mathbf{x}[n] \\ &= \mathbf{w}[n] + \mu e^*[n]\mathbf{x}[n] \end{aligned} \tag{2.84}$$

with the relation $e[n] = d[n] - \mathbf{w}^H[n]\mathbf{x}[n]$. Equation (2.84) is the well-known LMS update.

The convergence and stability of the LMS algorithm depend on the correct choice of the step size parameter μ [51,57]. A large step-size parameter μ results in a fast convergence speed but also in a large excess mean-squared error after adaptation, i.e. the algorithm is not very

precise in reaching and staying at the exact minimum of the cost function ξ. If μ is chosen small, the adaptation is slow but the excess mean-squared error after adaptation is small, thus a trade-off exists and usually a compromise has to be made. Standard analysis evaluating the algorithm's behaviour in terms of the mean and the mean-square (i.e. checking for bias and consistency of the achieved solution) requires for μ to fulfil

$$0 \; < \; \mu \; < \; \frac{1}{\lambda_{\max}} , \tag{2.85}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the correlation matrix $\mathbf{R}_{xx}$. Since $\mathbf{R}_{xx}$ has a Toeplitz structure and hence is positive semidefinite, an approximation gives

$$\lambda_{\max} \leq \sum_{i=0}^{l_a-1} \lambda_i = \mathrm{tr}\{\mathbf{R}_{xx}\} = l_a \cdot \sigma_{xx}^2 , \tag{2.86}$$

where l_a is the dimension of $\mathbf{x}$, λ_i the eigenvalues of $\mathbf{R}_{xx}$, $\mathrm{tr}\{\cdot\}$ the trace of the matrix argument, and σ_{xx}^2 is the variance of input signal. This yields

$$\mu \; < \; \frac{1}{l_a \cdot \sigma_{xx}^2} , \tag{2.87}$$

as an upper convergence limit for μ. When exceeding this limit, μ is likely to cause the LMS algorithm to be unstable.

2.5.2 Normalized Least Mean Square Algorithm

In the LMS algorithm, the convergence coefficient has to be selected such that the step size factor μ never exceeds its upper bound in (2.87). In non-stationary environments or where σ_{xx}^2 is not known a priori, the worst case has to be assumed, which means that at most times a rather slow convergence may arise. Therefore, a *normalization* of the step size can ensure, that an approximately constant rate of adaptation is achieved at all times. Based on an estimate of (2.86) by

$$l_a \cdot \sigma_{xx}^2 \; \approx \; \mathbf{x}[n]^H \cdot \mathbf{x}[n] , \tag{2.88}$$

the normalization of the step size is given by

$$\mu = \frac{\mu_0}{\mathbf{x}[n]^H \cdot \mathbf{x}[n]} . \tag{2.89}$$

Substituting (2.89) into the LMS update yields a constant convergence speed independent of the power of the input signal $\mathbf{x}$.

2.5.3 Recursive Least Squares Algorithm

The recursive least squares algorithm differs from the LMS-type algorithms reviewed in Sec. 2.5.1 and 2.5.2 by trying to minimize a sum of squared error values as cost function:

$$\xi_{\mathrm{LS}}[n] = \sum_{\nu=0}^{n} \beta^{\nu} |e[n]|^2 = \sum_{\nu=0}^{n} \beta^{\nu} |d[n] - \mathbf{w}^H[n-\nu]\mathbf{x}[n-\nu]|^2 \ . \tag{2.90}$$

The factor β, $(0 < \beta \leq 1)$ is called forgetting factor and ensures that recent data is given higher consideration and past errors are "forgotten" according to an exponential weighting.

The minimization of the cost function is performed by solving

$$\nabla(\xi_{\mathrm{LS}}[n]) = \mathbf{0} \ , \tag{2.91}$$

which in some analogy to (2.81) leads to

$$\mathbf{R}_{xx}[n]\mathbf{w}[n] = \mathbf{p}[n] \ . \tag{2.92}$$

This step is based on the generic adaptive filter structure in Fig. 2.10(a), where the quantities $\mathbf{R}_{xx}[n]$ and $\mathbf{p}[n]$ are now defined as

$$\mathbf{R}_{xx}[n] = \sum_{\nu=0}^{n} \beta^{\nu} \mathbf{x}[n-\nu]\mathbf{x}^H[n-\nu] \tag{2.93}$$

$$\mathbf{p}[n] = \sum_{\nu=0}^{n} \beta^{\nu} d^*[n-\nu]\mathbf{x}[n-\nu] \ . \tag{2.94}$$

A recursive formulation for the quantities (2.93) and (2.94) is given by

$$\mathbf{R}_{xx}[n] = \beta\mathbf{R}_{xx}[n-1] + \mathbf{x}[n]\mathbf{x}^H[n] \tag{2.95}$$

$$\mathbf{p}[n] = \beta\mathbf{p}[n-1] + d^*[n]\mathbf{x}[n] \ . \tag{2.96}$$

Based on these recursions, the update equation for the tap weights $\mathbf{w}$ could be calculated by solving (2.92) at each time instance n, involving a matrix inversion of $\mathbf{R}_{xx}[n]$. In practice, such an inversion operation can be quite time consuming, especially for a large value of l_a.

However, by exploiting the matrix inversion lemma [51]

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1} \tag{2.97}$$

and identifying $\mathbf{A} = \beta\mathbf{R}_{xx}[n-1]$, $\mathbf{B} = \mathbf{x}[n]$, $\mathbf{C} = 1$, and $\mathbf{D} = \mathbf{x}^H[n]$, the inversion can be circumvented. Assuming that initial conditions have been chosen to ensure that $\mathbf{R}_{xx}[0]$ is not singular and denoting $\mathbf{S}[n] = \mathbf{R}_{xx}^{-1}[n]$, this results in

$$\mathbf{S}[n] = \frac{1}{\beta}\left(\mathbf{S}[n-1] - \mathbf{G}[n]\mathbf{x}[n]\mathbf{x}^H[n]\mathbf{S}[n-1]\right) \ , \tag{2.98}$$

with

$$\mathbf{G}[n] = \frac{\mathbf{S}[n-1]}{\beta + \mathbf{x}^H[n]\mathbf{S}[n-1]\mathbf{x}[n]} . \tag{2.99}$$

By rearranging (2.99), we can find that $\mathbf{G}[n] = \mathbf{S}[n]$. Inserting (2.95) and (2.98) into $\mathbf{w}[n+1] = \mathbf{R}_{xx}^{-1}[n]\mathbf{p}[n]$ leads to

$$\begin{aligned} \mathbf{w}[n+1] &= \mathbf{w}[n] + \mathbf{S}[n]\mathbf{x}[n]\big[d[n] - \mathbf{w}^H[n]\mathbf{x}[n]\big]^* \\ &= \mathbf{w}[n] + \mathbf{S}[n]\mathbf{x}[n]e^*[n] . \end{aligned} \tag{2.100}$$

The new tap weight vector is computed by updating its old value by the product of the estimation error $e^*[n]$, the input vector $\mathbf{x}[n]$ and the inverse of the correlation matrix of $\mathbf{x}[n]$, which is also updated at each step.

From (2.100), we can see that the main difference between the LMS and RLS algorithm is in replacing the step size μ in the LMS algorithm by $\mathbf{S}[n]$ in the RLS case. By this modification, the convergence speed of the RLS algorithm is independent of the eigenvalue distribution of the correlation matrix and typically an order of magnitude faster than that of the LMS algorithm. For a detailed discussion of the RLS's properties and its recent developments, please refer to [51, 58, 59].

2.5.4 Comparison of Computational Complexities

In our context, the computational complexity of the adaptive algorithm employed in the GSC is of great concern and our primary aim is to find low-complexity implementations of the adaptive algorithm in the GSC environment. As the computational complexity of an algorithm is also dependent on the processors we use, e.g. fix-point processors or floating-point processors, we here only consider the number of multiplications in each step as an indication of the computational complexity of an adaptive algorithm. Since the computational complexities for real-valued and complex-valued signals are different, we will discuss them separately. Moreover, for single-channel and multi-channel adaptive filters, as in the GSC case, the input signal vector $\mathbf{x}$ has different signal structures, which can also affect the computational complexity. In the following discussion, l_a is the total number of adaptive weights for both the single and multi-channel cases.

2.5.4.1 Real Input Signal

For the LMS algorithm, l_a multiplications are required to calculate the output $e[n]$, one multiplication for the product of $e[n]$ and μ, and l_a multiplications for the final multiplication with $\mathbf{x}[n]$, totaling to $2l_a + 1$ multiplications.

The NLMS needs 2 additional multiplications to update the value in (2.89) for a single-channel system and $L+1$ multiplications for an L-channel system. In total this yields $2l_a+3$ and $2l_a+L+2$ multiplications for single and multichannel realisations, respectively.

For the RLS algorithm, l_a multiplications are required to calculate the output $e[n]$, $2l_a^2+l_a$ multiplications to calculate $\mathbf{S}[n]$ according to (2.99), and $l_a^2+l_a$ multiplications for the weight update in (2.100). In total, its computational complexity will be $3l_a^2+3l_a$ multiplications.

For convenience, we summarize these results in Tab. 2.1.

2.5.4.2 Complex Input Signal

For the LMS algorithm, we needs $4l_a$ real multiplications to calculate the output $e[n]$, two real multiplications for the product between $e[n]$ and μ, and $4l_a$ real multiplications for the final multiplication with $\mathbf{x}[n]$. So, it totals to $8l_a+2$ real multiplications.

The NLMS needs an additional three real multiplications in the single-channel case and $2L+1$ in the L-channel case to update μ in (2.89), and the total computational complexity will be $8l_a+5$ or $8l_a+2L+3$ real multiplications.

For the RLS algorithm, it requires $4l_a$ real multiplications to calculate the output $e[n]$, $4l_a^2+4l_a$ real multiplications to calculate the complex result of $\beta+\mathbf{x}^H[n]\mathbf{S}[n-1]\mathbf{x}[n]$ of (2.99), additional $4l_a^2+4$ real multiplications to get $\mathbf{S}[n]$, and $4l_a^2+4l_a$ real multiplications for the weight update in (2.100). In total, its computational complexity is $12l_a^2+12l_a+4$ multiplications.

These complexities are also given in Tab. 2.2.

As shown in Tabs. 2.1 and 2.2, the RLS algorithm has a much higher computational complexity than the LMS-type algorithms. Although recently a fast stable RLS algorithm has been introduced with a computational complexity of order $\mathcal{O}(l_a)$ [59], it is still more costly than LMS-type algorithms. Therefore, in our simulations we will only use the LMS-type algorithms, especially the normalised LMS algorithm.

Tab. 2.1: Computational complexities for real input signal:

adaptive algorithms	real multiplications (single-channel)	real multiplications (L-channel)
LMS	$2l_a+1$	$2l_a+1$
NLMS	$2l_a+3$	$2l_a+L+2$
RLS	$3l_a^2+3l_a$	$3l_a^2+3l_a$

Tab. 2.2: Computational complexities for complex input signal:

adaptive algorithms	real multiplications (single-channel)	real multiplications (L-channel)
LMS	$8l_a + 2$	$8l_a + 2$
NLMS	$8l_a + 5$	$8l_a + 2L + 3$
RLS	$12l_a^2 + 12l_a + 4$	$12l_a^2 + 12l_a + 4$

2.5.5 Frequency-domain and Subband Adaptive Algorithms

Tab. 2.1 and Tab. 2.2 show that the computational complexities of an adaptive algorithm are in general of the order of $\mathcal{O}(l_a)$ for LMS-type or $\mathcal{O}(l_a^2)$ for RLS-type algorithms , which increases dramatically with the length of the adaptive filter. To reduce the large computational complexity imposed by using long adaptive filters, we can use frequency-domain adaptive filtering (FDAF) algorithms [60,61], which are based on block by block updating strategies whereby the filter convolution and the gradient correlation can be performed efficiently using fast Fourier transformation (FFT) algorithms [62]. Since the output and weight update are computed only after a large block of data has been accumulated, the computational complexity can be significantly reduced. Moreover, due to the approximately uncorrelated output signals of the FFT, different stepsizes can be used for different frequency bin outputs. As a result, the convergence speed of the algorithm may be improved.

The DFT, as employed in frequency-domain adaptive algorithms, can be viewed as a filter bank with maximal decimation [63, 64]. Because of its relatively poor frequency resolution, the DFT filter bank has a large degree of spectral overlap between the adjacent frequency band, which can lead to severe aliasing distortion and may cause a problem when the input data do not exactly lie on a frequency bin [65]. As an alternative, subband adaptive filtering algorithms have been developed [58,61,66–69], which are capable of achieving a reduced spectral overlap. We will use subband adaptive filtering algorithms in our GSC structure for reducing its computational complexity. The corresponding details will be discussed in the next chapter.

2.6 Summary

In this chapter, we have provided an introduction to the basic ideas of beamforming, specifically focusing on the GSC structure, which constitutes an alternative implementation of the linearly constrained minimum variance beamforming. The GSC transforms the constrained optimization problem into an unconstrained one, which can be solved by standard adaptive al-

gorithms. Various problems relating to the LCMV beamformer and the GSC have been briefly discussed, such as the formulation of the constraints, the optimum solution, and the blocking matrix design. Especially, a simplified GSC with tapped-delay lines after the quiescent vector and blocking matrix has been detailed. Commonly used adaptive algorithms for optimising the GSC's coefficients are also introduced, and their computational complexity is analysed . To reduce the computational complexity of a broadband GSC system, we will employ subband adaptive algorithms in its adaptive part in the next chapter, which will lead to the proposition of a novel subband adaptive GSC.

Chapter 3

Subband Adaptive Generalised Sidelobe Canceller

In this chapter, based on the generalised sidelobe canceller structure in Fig. 2.7, we employ subband adaptive filtering techniques in the unconstrained adaptive part and propose a novel subband adaptive GSC structure for broadband beamforming [32, 33]. Amongst other advantages, the proposed scheme aims to be computationally less complex than the algorithms outlined in Section 2.5. First, we give a brief overview of the fundamentals of filter banks in Section 3.1 and subband adaptive filtering techniques in Section 3.2. In Section 3.3 the structure of our proposed subband adaptive GSC is presented. Simulation results will be discussed in Section 3.4 to demonstrate the benefits of the proposed method.

3.1 Fundamentals of Filter Banks

Filter banks [58, 63, 64, 70, 71] constitute a set of filters designed for signal decomposition, reconstruction and processing. Since typically different sampling rates are employed in different parts of the system, they are sometimes referred to as multirate filter banks. The basic idea behind it is to split the signal spectrum into reduced-width subbands, which can be sampled at a lower rate due to the reduced bandwidth. Individual bands then may be treated separately during further processing such as audio coding [72–76], image coding [77–80] and adaptation [58, 61,66,67,81,82]. After processing, these subband signals can be reconstructed using a synthesis filter bank to obtain a system output at the original sampling rate. Fig 3.1 shows the general structure of a K-channel filter bank using a decimation factor N, where the input signal $x[n]$ is decomposed into K subbands by an analysis filter bank and hence each subband is decimated by a factor of $N \leq K$. After upsampling, these subbands are recombined by a synthesis filter bank to yield the fullband output signal $\hat{x}[n]$. In general, we consider systems that are of perfect

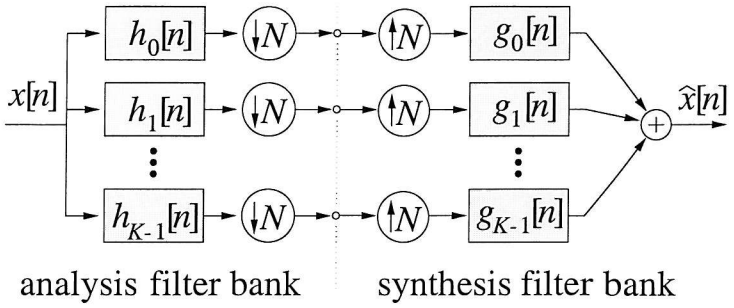


Fig. 3.1: General structure of a K-channel filter bank with a decimation factor of N.

reconstruction (PR) [83, 84], where the output signal $\hat{x}[n]$ will be identical to the input $x[n]$ except for a time delay.

In the following, we will first give an introduction to the basic operations in multirate signal processing and then analyse the PR conditions of filter banks. Finally, we will briefly discuss the design and implementation of a family of oversampled generalised DFT (GDFT) [85] filter banks [58], which will be used in our subband adaptive GSC.

3.1.1 Basic Multirate Operations

3.1.1.1 Decimation and Interpolation

Decimation and interpolation [63, 64] are the operations used to alter the sampling rate in a system and are represented as shown in Fig. 3.2 and Fig. 3.3, respectively.

Decimation is the process of reducing the sampling rate of a signal by an integer factor N, where the fullband signal $\tilde{x}[n]$ is first passed through an anti-aliasing filter $h[n]$, typically with a low-pass characteristic, and then downsampled to a lower sampling rate.

A downsampler is also referred to as a subsampler and represented by a circle with $\downarrow N$ inside. This operation retains only every N-th sample of its input and then relabels the index axis. An example of a downsampling process by $N = 3$ is shown in Fig. 3.2(b).

In time domain, the downsampling operation can be expressed as

$$y[n] = x[Nn] \; . \tag{3.1}$$

Its frequency-domain equivalence is

$$Y(e^{j\Omega}) = \frac{1}{N} \sum_{n=0}^{N-1} X(e^{j(\frac{\Omega - 2n\pi}{N})}) \; , \tag{3.2}$$

where $X(e^{j\Omega})$ and $Y(e^{j\Omega})$ are the Fourier transforms of the input and output signals of the downsampler, respectively.

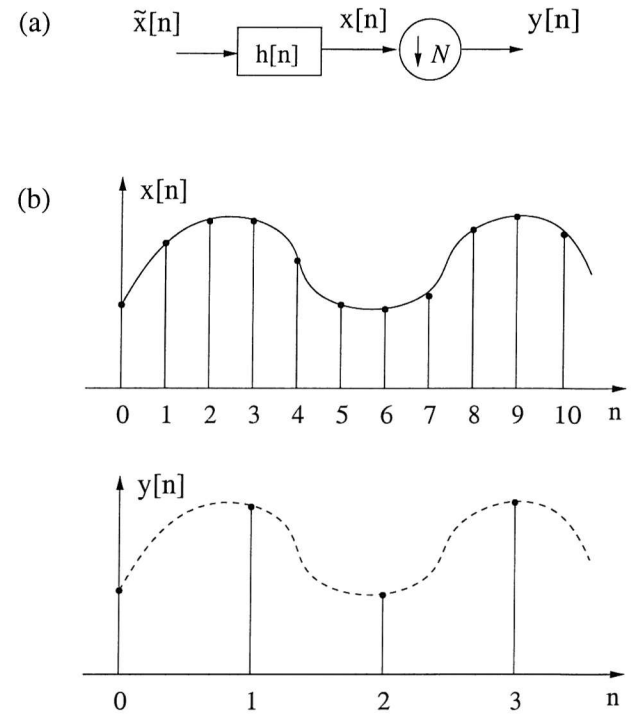


Fig. 3.2: Decimation operation by a factor of N: (a) anti-aliasing filter and downsampler; (b) example for a downsampling operation by $N = 3$.

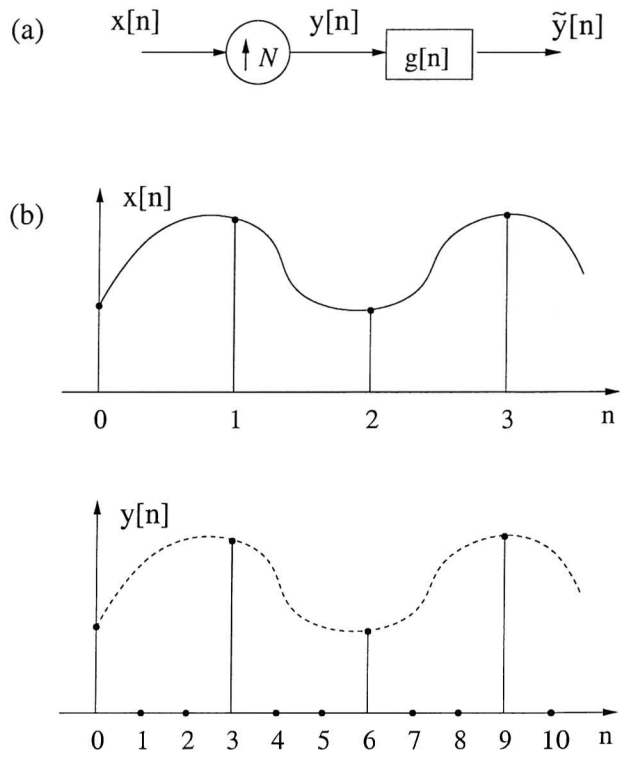


Fig. 3.3: Interpolation operation by a factor of N: (a) upsampler and interpolation filter; (b) example for a upsampling operation by $N = 3$.

It can be seen from (3.2) that downsampling creates $N-1$ aliased terms, $X(e^{j(\frac{\Omega-2n\pi}{N})})$, $n \neq 0$ in its output $Y(e^{j\Omega})$. Therefore, it can lead to loss of information, known as aliasing, if aliased terms overlap in frequency. To avoid aliasing, the bandwidth of the input signal has to be appropriately limited, e.g. to a lowpass signal having a bandwidth lower than π/N. For this purpose, an anti-aliasing filter $h[n]$ is often employed prior to downsampling.

Interpolation is the process of increasing the sampling rate of a signal, and is achieved by the combination of an upsampler and a lowpass filter as shown in Fig. 3.3 (a). An upsampler is also referred to as an expander, and upsampling by an integer factor of N is achieved by inserting $N-1$ zeros in between successive samples of the original signal. In the time domain, upsampling is represented by

$$y[n] = \begin{cases} x\left[\frac{n}{N}\right] & : \quad n = 0, \pm N, \pm 2N, \cdots \\ 0 & : \quad \text{otherwise} \end{cases} . \tag{3.3}$$

An example for an upsampling process by a factor of $N = 3$ is shown in Fig. 3.3 (b).

In frequency domain, upsampling is denoted as

$$Y(e^{j\Omega}) = X(e^{j\Omega N}) , \tag{3.4}$$

which means that the output spectrum $Y(e^{j\Omega})$ is related to the input spectrum by a compression by a factor of N.

Finally, the time-domain representations for decimation and interpolation operations, which include the filtering in Fig. 3.2 (a) and Fig. 3.3 (a), are given by

$$\begin{aligned} &\text{Decimation:} && y[n] = \textstyle\sum_k h[Nn-k]\ \tilde{x}[k] \ ; \\ &\text{Interpolation:} && \tilde{y}[n] = \textstyle\sum_k g[n-Nk]\ x[k] \ . \end{aligned} \tag{3.5}$$

3.1.1.2 Multirate Identities

There are several multirate identities, also known as Noble identities [63, 64], which are equivalent structures of multirate building blocks. In particular, Noble identities can be applied to simplify the derivation of the perfect reconstruction condition of filter banks and the manipulation of multirate building blocks. Fig. 3.4 shows two useful multirate identities that are commonly used [63, 64].

3.1.1.3 Polyphase Decomposition

Polyphase decomposition [86, 87] is an important tool in multirate signal processing and can greatly simplify the analysis and implementation of filter banks. In order to derive the PR

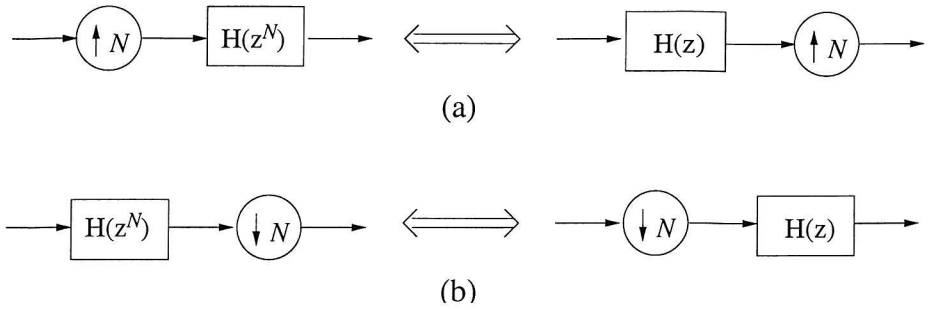


Fig. 3.4: Equivalent structures for (a) upsampling and (b) downsampling, where $H(z)$ is the z-transform of a filter.

condition of filter banks in matrix form, we need to decompose the analysis and synthesis filters into their polyphase components. In the following, we review two different types of polyphase decompositions.

For any sequence $h[n]$, we can always define a series of subsequences $h_i[n]$ as follows

$$h_i[n] = h[Nn + i], \quad i = 0, 1, \cdots, N-1 \; . \tag{3.6}$$

The z-transform $H_i(z)$ of a subsequence $h_i[n]$ is given by

$$H_i(z) = h[i] + h[i+N]z^{-1} + h[i+2N]z^{-2} + \cdots \; . \tag{3.7}$$

Then the z-transform of $h[n]$, namely $H(z)$, can be expressed as

$$H(z) = \sum_{i=0}^{N-1} z^{-i} H_i(z^N) \; . \tag{3.8}$$

We refer to (3.8) as the Type-I polyphase decomposition of $H(z)$ and $H_i(z)$, $i = 0, 1, \cdots, N-1$, are the N Type-I polyphase components of $H(z)$.

A similar decomposition to (3.8), referred to as the Type-II decomposition, is given by

$$\tilde{h}_i[n] = h[Nn + N - 1 - i], \quad i = 0, 1, \cdots, N-1 \; , \tag{3.9}$$

and

$$H(z) = \sum_{i=0}^{N-1} z^{-(N-1-i)} \tilde{H}_i(z^N) \; , \tag{3.10}$$

where $\tilde{H}_i(z)$ is the z-transform of $\tilde{h}_i[n]$. Therefore, the polyphase components $H_i(z)$ and $\tilde{H}_i(z)$ are related by

$$\tilde{H}_i(z) = H_{N-1-i}(z) \; . \tag{3.11}$$

3.1.2 Perfect Reconstruction Condition for Filter Banks

Let us now study the PR condition of the K-channel filter banks with a decimation factor of N as shown in Fig. 3.1. When $N = K$, the corresponding scheme is referred to as a critically decimated filter bank, whereas when $N < K$, it becomes an oversampled system. The PR condition discussed here is applicable to both the oversampled and critically decimated systems.

First we express the z-transforms of the analysis filters $h_k[n]$ and synthesis filters $g_k[n]$, $k = 0, 1, \cdots, K-1$, in vectorial form as follows

$$\mathbf{h}(z) = [H_0(z) \;\; H_1(z) \;\; \cdots \;\; H_{K-1}(z)]^T , \tag{3.12}$$

$$\mathbf{g}(z) = [G_0(z) \;\; G_1(z) \;\; \cdots \;\; G_{K-1}(z)]^T . \tag{3.13}$$

Then we decompose the analysis filters into their type-I polyphase components

$$H_k(z) = \sum_{n=0}^{N-1} z^{-n} H_{k,n}(z^N) , \tag{3.14}$$

and the synthesis filters into their type-II polyphase components

$$G_k(z) = \sum_{n=0}^{N-1} z^{-(N-1-n)} G_{k,n}(z^N) , \tag{3.15}$$

where $H_{k,n}(z)$ and $G_{k,n}(z)$ are the n-th polyphase components of the k-th analysis filter and synthesis filter, respectively. Now the vectors $\mathbf{h}(z)$ and $\mathbf{g}(z)$ can be expressed in their polyphase form as

$$\mathbf{h}(z) = \underbrace{\begin{bmatrix} H_{0,0}(z^N) & H_{0,1}(z^N) & \cdots & H_{0,N-1}(z^N) \\ H_{1,0}(z^N) & H_{1,1}(z^N) & \cdots & H_{1,N-1}(z^N) \\ \vdots & & \ddots & \vdots \\ H_{K-1,0}(z^N) & H_{K-1,1}(z^N) & \cdots & H_{K-1,N-1}(z^N) \end{bmatrix}}_{\mathbf{E}(z^N)} \cdot \underbrace{\begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(N-1)} \end{bmatrix}}_{\mathbf{e}_N} = \mathbf{E}(z^N) \cdot \mathbf{e}_N , \tag{3.16}$$

and

$$\begin{aligned} \mathbf{g}^T(z) &= \underbrace{\begin{bmatrix} z^{-(N-1)} & z^{-(N-2)} & \cdots & 1 \end{bmatrix}}_{\tilde{\mathbf{e}}_N} \cdot \underbrace{\begin{bmatrix} G_{0,0}(z^N) & G_{1,0}(z^N) & \cdots & G_{K-1,0}(z^N) \\ G_{0,1}(z^N) & G_{1,1}(z^N) & \cdots & G_{K-1,1}(z^N) \\ \vdots & & \ddots & \vdots \\ G_{0,N-1}(z^N) & G_{1,N-1}(z^N) & \cdots & G_{K-1,N-1}(z^N) \end{bmatrix}}_{\mathbf{R}(z^N)} \\ &= \tilde{\mathbf{e}}_N \cdot \mathbf{R}(z^N) . \end{aligned} \tag{3.17}$$

The matrix $\mathbf{E}(z)$ in (3.16) is referred to as polyphase analysis matrix, while $\mathbf{R}(z)$ in (3.17) is its dual, a so called polyphase synthesis matrix.

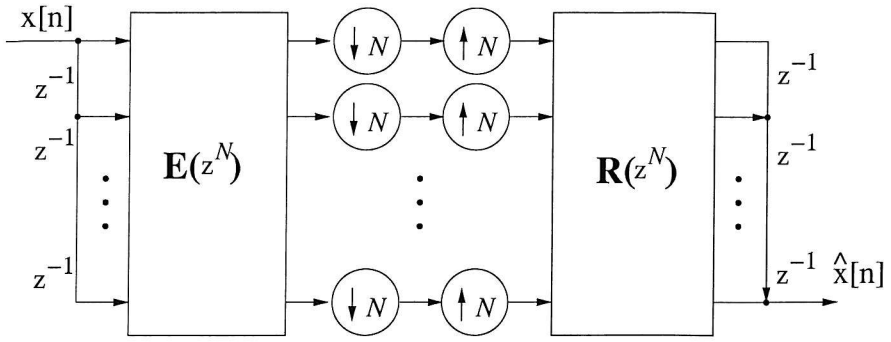


Fig. 3.5: Polyphase representation of Fig. 3.1.

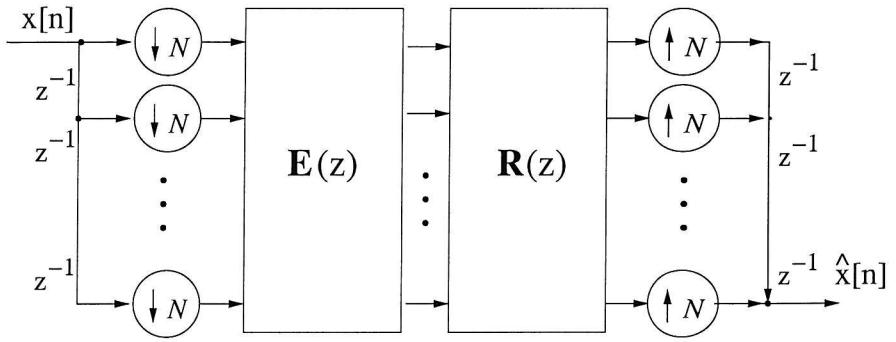


Fig. 3.6: Polyphase representation using multirate identities.

Using (3.16) and (3.17), we can redraw Fig 3.1 in its polyphase representation, as shown in Fig 3.5. Using Noble identities as introduced in Sec. 3.1.1.2, we can shift the downsamplers to the left-hand side of the analysis polyphase matrix and replace z^N by z in the argument of $\mathbf{E}(z^N)$. Similarly, we can shift the upsamplers to the right-hand side of the synthesis polyphase matrix and obtain the structure shown in Fig 3.6, which can be further simplified to the form shown in Fig. 3.7 with $\mathbf{P}(z) = \mathbf{R}(z) \cdot \mathbf{E}(z)$.

It is shown in [88] that the general PR condition is given by

$$\mathbf{P}(z) = cz^{-\alpha} \cdot \begin{bmatrix} \mathbf{0} & \mathbf{I}_{N-r} \\ z^{-1}\mathbf{I}_r & \mathbf{0} \end{bmatrix}_{N \times N} , \tag{3.18}$$

where c is a nonzero constant, and α and r are integers with $0 \leq r \leq (N-1)$. Under this condition the input-output relationship becomes

$$\hat{x}[n] = cx[n - n_0] , \tag{3.19}$$

where $n_0 = N\alpha + r + N - 1$ is the delay of the overall system.

A useful sufficient condition is

$$\mathbf{P}(z) = z^{-\alpha} \cdot \mathbf{I}_N , \tag{3.20}$$

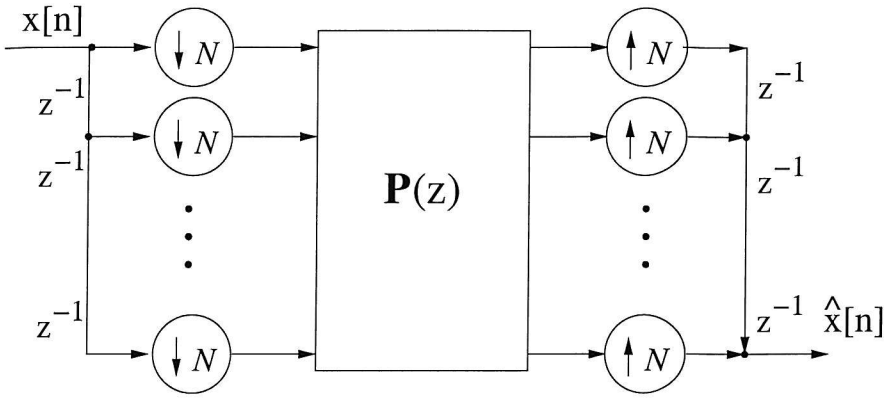


Fig. 3.7: Equivalent structure of Fig. 3.6 with $\mathbf{P}(z) = \mathbf{R}(z) \cdot \mathbf{E}(z)$.

which corresponds to the case where $r = 0$. The equivalent form in the time domain for (3.20) is given by [89]

$$\sum_n h_i[n] g_j[-lN - n] = \delta[l]\delta[i - j], \quad 0 \leq i, j \leq K - 1 \,, \tag{3.21}$$

where $\delta[l]$ is the Kronecker delta function. By this condition, the system in Fig. 3.6 reduces to a multiplexer and a demultiplexer in the analysis and synthesis banks, and we get

$$\hat{x}[n] = x[n - n_0] \text{ with } n_0 = N\alpha + N - 1 \,. \tag{3.22}$$

3.1.3 Oversampled Modulated Filter Banks

In the K-channel filter banks shown in Fig. 3.1, the filters $h_k[n]$ and $g_k[n]$, $k = 0, 1, \cdots, K-1$, may differ from each other and need to be correctly designed to satisfy the PR condition. In order to reduce the design and implementation complexity, we can use modulated filter banks, where both the analysis filters $h_k[n]$ and the synthesis filters $g_k[n]$ are derived from a prototype lowpass filter by some suitable modulation. Popular modulations for this purpose include cosine modulation [90], extended lapped transforms [91] and the discrete Fourier transform (DFT) [85, 92]. By modulation, the design of the PR filter banks is reduced to the design of the prototype filter and the implementation can be realized by the coefficients of prototype filter and the modulation block. Here we focus on a class of oversampled modulated filter banks with $N < K$, which are referred to as generalised DFT filter banks [85]. In such filter banks, the analysis filters are derived from a real valued lowpass prototype FIR filter $p[n]$ of length l_p by a generalised discrete Fourier transform (GDFT) [85] according to

$$h_k[n] = e^{j\frac{2\pi}{K}(k+k_0)(n+n_0)} \cdot p[n] \text{ with } k = 0, 1, \cdots, K-1 \text{ and } n = 0, 1, \cdots, l_p - 1 \,. \tag{3.23}$$

The term GDFT as defined in [85] stems from the offsets k_0 and n_0 introduced into the frequency and time indices of the DFT, which will be justified below. According to (3.23), the spectrum

of the resultant analysis filters $h_k[n]$ are spectrally shifted versions of the prototype filter $p[n]$ along the frequency axis by $2\pi(k + k_0)/K$. Specifically, in conjunction with $k_0 = 1/2$, for a real-valued input signal $x[n]$ it is sufficient to only process the first $K/2$ subbands covering the frequency interval $[0; \pi]$ as the remaining subbands are the complex conjugate versions of these subbands. For the full effect of the frequency offset k_0, please refer to [85]. In some applications such as image coding [77–80], maintaining a linear phase is required, which can be fulfilled by choosing

$$n_0 = -\frac{l_p - 1}{2} \tag{3.24}$$

along with a real-valued linear phase prototype filter $p[n]$ of even length.

If the prototype filter is designed such that the polyphase analysis matrix $\mathbf{E}(z)$ is paraunitary, i.e.

$$\mathbf{E}^H(z^{-1})\mathbf{E}(z) = c\mathbf{I}\,, \tag{3.25}$$

then we can simply choose the polyphase synthesis matrix as $\mathbf{R}(z) = z^{-\alpha}\mathbf{E}^H(z^{-1})$ in order to comply with the PR requirement of (3.20). In this case, the impulse responses of the analysis and synthesis filters are time-reversed, complex conjugate versions of each other

$$g_k[n] = h_k^*[l_p - 1 - n]\,. \tag{3.26}$$

If the analysis filters are linear in phase, (3.26) will become

$$g_k[n] = h_k[n], \quad n = 0, 1, \cdots, l_p - 1\,. \tag{3.27}$$

The modulation approach allows for both low memory consumption for storing filter coefficients and an efficient polyphase implementation. The latter even works for non-integer oversampling ratios K/N, and allows a factorisation of the filter bank into a real-valued polyphase network only depending on the prototype filter [58, 93]. The output of this network is rotated by a GDFT transform [85] , which can be mainly implemented using an FFT. According to [94], the number of real multiplications required to implement the GDFT filter banks is

$$C_{GDFT}^{real} = \frac{1}{N}\left(l_p + 4K\log_2 K + 4K\right) \tag{3.28}$$

for a real-valued input signal and

$$C_{GDFT}^{complex} = \frac{1}{N}\left(2l_p + 4K\log_2 K + 8K\right) \tag{3.29}$$

for a complex-valued input signal.

Furthermore, the filter bank design reduces to an appropriate choice of the prototype filter, which has to satisfy two criteria. Firstly, its attenuation in the stopband, $\Omega \in [\pi/N; \pi]$, has to

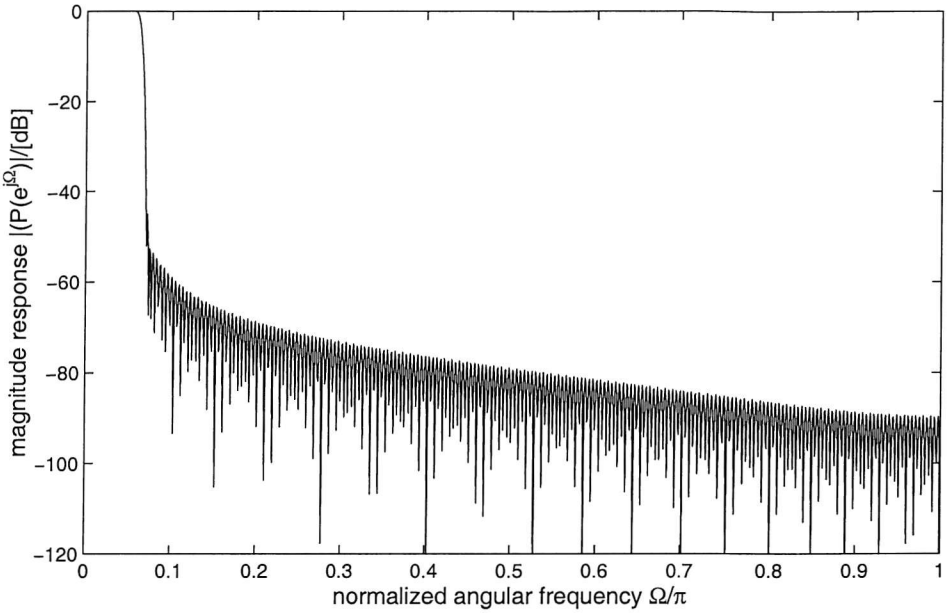


Fig. 3.8: Frequency response of a prototype filter with length $l_p = 448$ designed by the iterative least-squares method in [2] ($K = 16$, $N = 14$).

be sufficiently large. Every frequency of the input signal $x[n]$ lying within the interval $[\pi/N; \pi]$ will be aliased into the baseband after filtering and downsampling and hence will cause a distortion of the subband signal. A second constraint on the design is the PR condition. If the stopband attenuation of the prototype filter is sufficiently high for suppressing aliasing, this condition reduces to the consideration of inaccuracies in power complementarity [64],

$$\sum_{k=0}^{K-1} |H_k(e^{j\Omega})|^2 = 1 \;, \forall\, \Omega \;. \tag{3.30}$$

A prototype filter approximating these constraints can be constructed by, for example, iterative least-squares methods [2,95] or dyadically iterated halfband filters [58,96,97]. Fig. 3.8 shows a prototype filter with length of $l_p = 448$ designed by an iterative least-squares method according to [2] for a $K = 16$ channel GDFT filter bank with a decimation factor of $N = 14$.

3.2 Subband Adaptive Filtering

The subband adaptive filtering (SAF) technique [58, 61, 66, 68, 69] has been widely applied to problems such as acoustic echo cancellation (AEC) [67, 81, 98], identification of room acoustics [99], or equalisation [100–102], where a large number of adaptive parameters have to be adjusted and as a result, the computational complexity can be very high and the convergence rate of the adaptive filter can be slow using standard techniques. Similar problems arise in

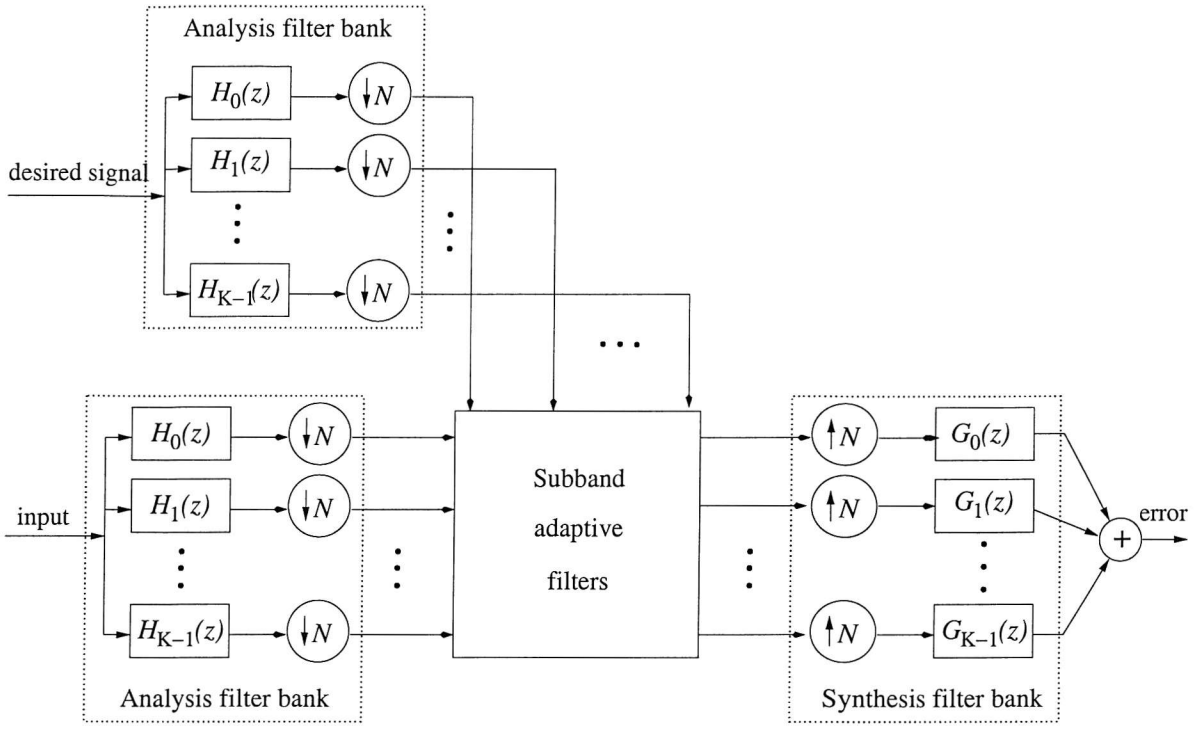


Fig. 3.9: A general SAF setup; the subband splitting and fullband error reconstruction is performed by filter banks.

broadband beamforming where arrays with a large number of sensors and filter coefficients have to be employed to perform beamforming with high interference rejection and resolution. Therefore general fullband adaptive systems, and in particular beamforming algorithms, can be computationally very costly to implement. A reduction in complexity can be achieved by processing in decimated subbands. Furthermore, the separation into frequency bands can bring additional advantages such as parallelisation of processing tasks and reduced spectral dynamics [67].

A general SAF system is shown in Fig. 3.9. There, both the input signal and the desired signal are split into decimated subbands by analysis filter banks, then the subband adaptive filters, which run at a lower, decimated sampling rate compared to the original fullband system as illustrated in Fig. 2.10(a), are employed to estimate the subband desired signals using the subband input signals. The resultant subband error signals are then reconstructed into a fullband error signal by a synthesis filter bank.

Depending on the filter banks employed, the subband adaptive filters can have different structures. In the case of critically decimated filter banks, where the decimation ratio N equals the number of uniform subbands K, we need either cross-terms at least between adjacent frequency bands [67], which compensate for the information loss in the region of spectral overlap, or gap filter banks [68,103], which introduce spectral loss to avoid problems with aliasing. The

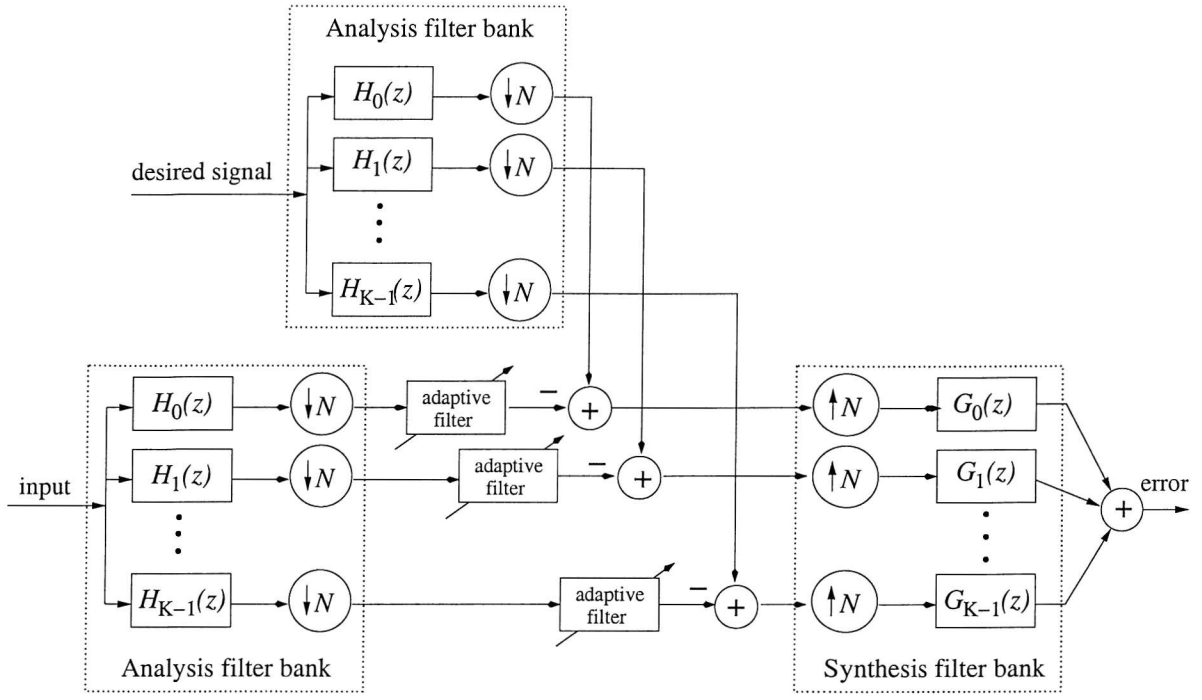


Fig. 3.10: An oversampled SAF system with adaptive filters working independently in K decimated subbands.

drawbacks are, that the inclusion of cross-terms requires multichannel adaptive algorithms with generally slower convergence and increased computational cost, while the distortion incurred through the application of gap-filter banks may not be acceptable.

Oversampled SAF systems with a decimation ratio $N < K$ are designed such that after decimation the alias level within the subbands is kept sufficiently low. Therefore, an independent subband adaptive filter can be operated in each of the corresponding subbands as shown in Fig. 3.10. Differences arise for the decimation of complex or real valued frequency bands. The decimation of real valued bandpass signals is generally complicated, and real valued signals have to be either modulated into the baseband prior to decimation by, for example, single sideband modulation (SSB) [85,95], or their bandwidth and decimation ratio have to be chosen in accordance with the bandpass sampling theorem [104], leading to non-uniform filter banks [105–107]. In contrast, the decimation of complex valued bandpass signals with any integer factor $N < K$ is straightforward. Therefore, in our following work employing subband adaptive filtering techniques, we focus on SAF systems that are based on oversampled GDFT filter banks, performing a particular type of complex-valued subband decomposition. In general, complex-valued filter banks can be shown to be at least as efficient to implement as their real-valued counterparts [58].

In SAF, the adaptive filter length in each subband can be selected shorter compared to a fullband adaptive filter in accordance with the sampling rate reduction by a factor of $N < K$.

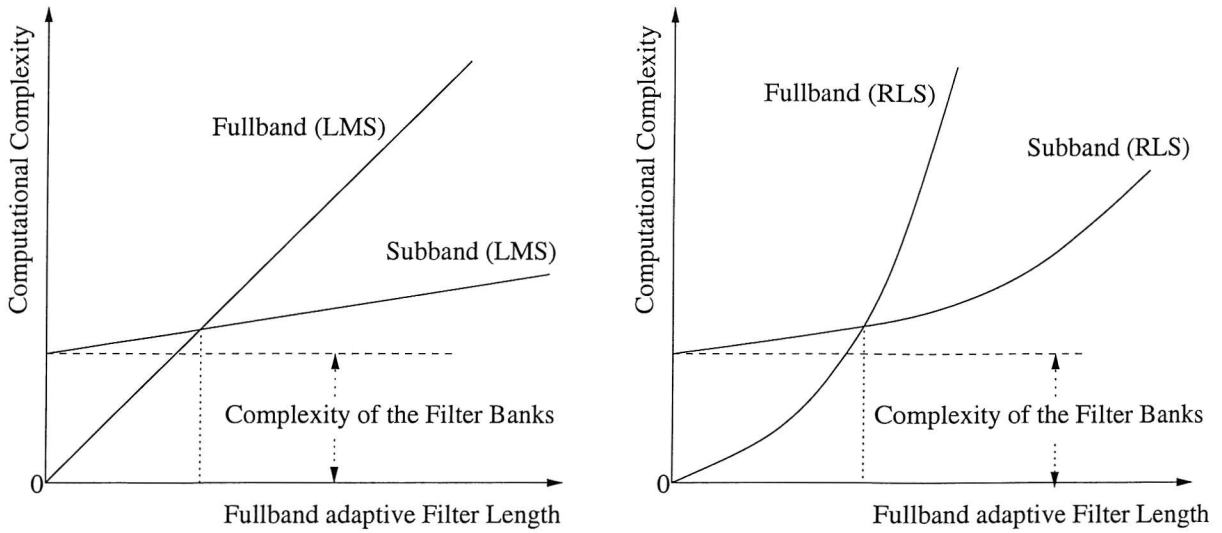


Fig. 3.11: A qualitative description of the computational complexities of fullband and subband adaptive methods for (a) LMS and (b) RLS adaptation.

The precise adaptive filter length l_{sub} in each subband with similar modelling capabilities as a fullband adaptive filter of length l_{full} is however not only determined by the decimation rate N, but also requires an offset term introduced by transients caused by the filter banks [58,67,108]. There is dispute about the exact length of l_{sub} and we here only give the heuristic but still conservative result [58]

$$l_{sub} = \frac{l_{full} + l_p}{N}, \tag{3.31}$$

where l_p is the length of the prototype filter used in the analysis filter bank. If we consider a large adaptive system, in general $l_p \ll l_{full}$ and hence the approximation $l_{sub} = \frac{l_{full}}{N}$ arises. For (3.31) being conservative [58] and for sake of simplicity, we will always assume $l_{sub} = \frac{l_{full}}{N}$.

Performing adaptive filtering in decimated subbands has three main advantages, which are detailed below:

1. **Reduction in Computational Complexity**
   Firstly, as mentioned, the adaptive filter length required in each subband can be shortened approximately by a factor of $N < K$ compared to a fullband adaptive filter. Secondly, the updating of the adaptive filters is carried out at a lower sampling rate. Following the approach of [95,109], we only consider the computational complexity of the subband adaptive part, i.e. do not include the extra computational overhead involved in the subband decomposition operation carried out by filter banks. Under this assumption, performing adaptation in the subbands yields a reduction in the computational complexity. Specifically, the complexity is reduced by a factor of $\mathcal{O}(K/N^2)$ for LMS-type algorithms and

$\mathcal{O}(K/N^3)$ for RLS algorithms. Therefore, it is advantageous in terms of computational cost to keep the oversampling ratio K/N close to unity, i.e. choose a non-integer oversampling ratio. However, considering the subband decomposition and adaptation as a whole, the SAF method does not necessarily save computations. If the adaptive filter is of a relatively short length, the extra computational load of the filter banks is likely to exceed any reduction of computational complexity achieved in the subband adaptive part. In this case, the SAF method may need a higher number of arithmetic operations than its fullband equivalent. With the increase of the adaptive filter length, the reduction of the computational complexity achieved by the subband adaptive part will become more substantial. Since the complexity of the filter banks is fixed, the overall SAF system will have a lower computational complexity than the fullband adaptive system. When the length of the adaptive filter is sufficiently high, the computations involved in the filter banks can be ignored and the reduction ratio for the whole subband adaptive system will be approximately $\mathcal{O}(K/N^2)$ for LMS-type algorithms and $\mathcal{O}(K/N^3)$ for RLS algorithms. A qualitative discription of the computational complexity of both the subband and fullband methods is provided in Fig. 3.11.

2. **Spectral Prewhitening**

   For LMS-type algorithms, the convergence speed is governed by the eigenvalue spread, which is linked to the spectral dynamics of the input to the adaptive filter [51]. As a result, coloured input signals with a non-uniform spectral distribution, such as speech signals, cause slow convergence for such fullband algorithms. The separation of the spectrum into spectral intervals as performed by the subband decomposition results in reduced spectral dynamics in each of the subbands with respect to the fullband and therefore can enable faster adaptation.

3. **Parallelisation**

   The structure of the subband adaptive system of Fig. 3.10 presents a natural parallelisation of the fullband processing task, that can be exploited by processing separate subband tasks on different processors [67]. This can be useful in cases where the subband approach alone does not give the required complexity reduction in order to implement a given problem in real-time on a given processor.

In the next section, we will applying the subband adaptive filtering techniques to broadband digital beamforming and propose a subband adaptive generalised sidelobe canceller.

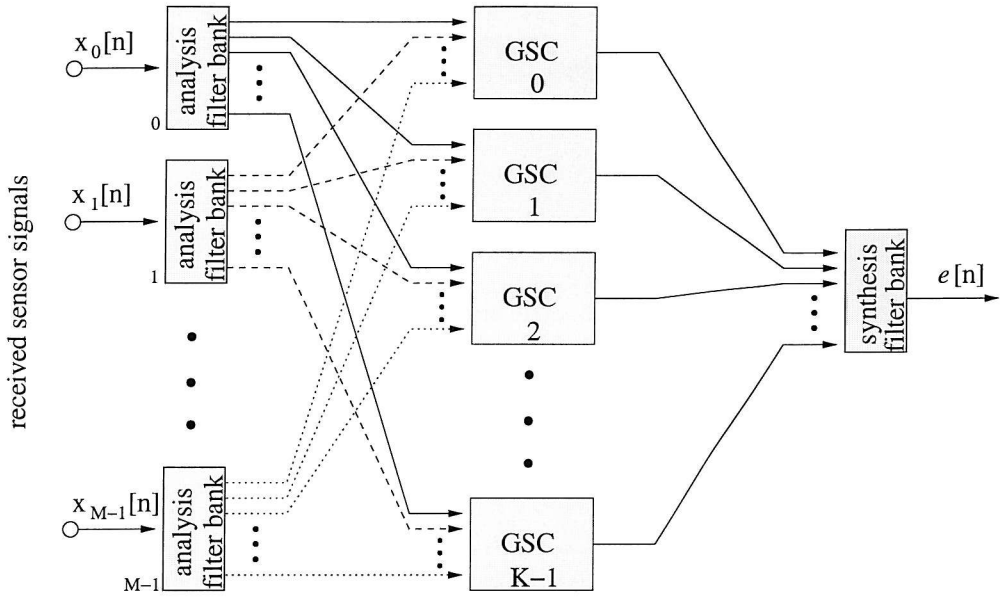


Fig. 3.12: A subband beamforming structure based on the GSC as proposed in [1, 3, 4].

3.3 Subband Adaptive Generalised Sidelobe Canceller

In the past, the application of subband methods to adaptive beamforming such as microphone or antenna arrays has been studied by a number of researchers [1, 3, 4, 24–26, 110–112]. The basic idea is to first split each of the received sensor signals into subbands and then operate an independent beamformer in each of the subbands, whereby the subband beamformer is selected according to the specific applications. In LCMV beamforming, subband based methods are employed to reduce the computational complexity and improve the convergence speed of a broadband beamformer [1, 3, 4], whereby the received sensor signals are split into decimated frequency bands ("subbands"). An independent beamformer, such as a generalised sidelobe canceller [29] can be applied to each of these subbands as shown in Fig. 3.12, where M is the number of sensors and K is the number of channels in the employed filter banks. In this method, we are restricted to use the same number of analysis filter banks as the sensor number M, and also the same number of GSCs as the subband number K, because we have to split each of the sensor signals into subbands and apply a GSC to each of the corresponding subbands. When the number of sensors and subbands is high, these operations will impose a high computational load on the system. Moreover, in this method, the fullband constraints of the beamformer and in particular the response vector $\mathbf{f}$, have to be decomposed into subband-based constraints in order to construct the GSC for each of the subbands [3]. This projection can incur inaccuracies because of the non-perfect reconstruction property of the filter banks and the limited number of weights to represent the constraints in each of the subbands.

To overcome these problems, we propose a different subband adaptive structure based on

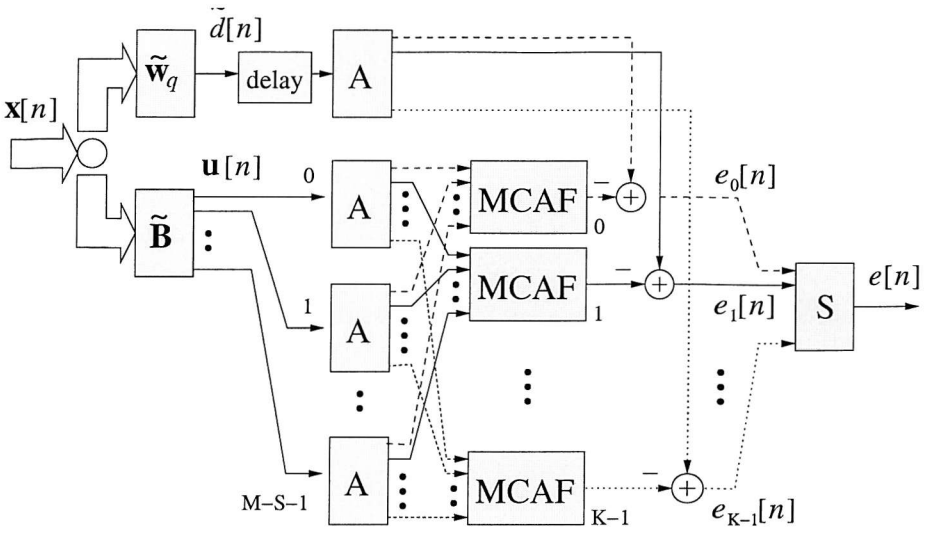


Fig. 3.13: Subband adaptive GSC structure with analysis (A) and synthesis (S) filter banks.

the GSC [33]. In contrast to Fig. 3.12, the sensor signals are directly fed into the GSC, and the outputs of the GSC's blocking matrix are split into subbands, in which independent adaptation is performed. By this approach, we use a single GSC and apply it to the fullband sensor signals directly, thus bypassing any decomposition of constraints. In addition, the number of analysis filter banks used is the same as the number of blocking matrix outputs, which can be considerably lower than the sensor number, especially for partially adaptive GSCs [14, 15]. Therefore, the computational complexity of our method is reduced further than in the previous methods reviewed above. Starting from the fullband beamformer in Fig. 2.7, we first introduce our proposed subband adaptive GSC in Sec. 3.3.1 and its computational complexity in Sec. 3.3.2. Sec. 3.3.3 discusses how an equivalent fullband beamformer is derived from the subband adaptive GSC in order to e.g. calculate a beam pattern. Simulations and results are given in Sec. 3.4.

3.3.1 Structure

Considering the multichannel characteristics of the GSC adaptation, when applying SAF techniques to the MCAF in the GSC structure in Fig. 2.7, the subband setup as shown in Fig. 3.13 arises, where the blocks labelled 'A' are analysis filter banks, splitting each of their inputs into K decimated frequency bands. Both the blocking matrix output signals $u_m[n]$, $m = 0, 1, \cdots, M-S-1$ and the upper path output $\tilde{d}[n] = \tilde{\mathbf{w}}_q^H \cdot \mathbf{x}[n]$ are split into subbands by analysis filter banks and in each corresponding subband, an MCAF system is set up to perform an unconstrained optimisation based on the subband error signal $e_k[n]$, $k = 0, 1, \cdots, K-1$. These subband error signals are then reconstructed to a fullband beamformer output $e[n]$ by

the synthesis filter bank, labelled 'S' in Fig. 3.13.

The above adaptation operating in subbands requires considerably less coefficients in the temporal dimension than a fullband beamformer. As a rule of thumb, the reduction in filter length is approximately N, where N is the decimation factor employed in the filter bank. This is particularly useful if the fullband beamformer in Fig. 2.3 has long filters $J \gg 1$, as is required for high resolution broadband beamforming.

In the structure of Fig. 3.12, each GSC has an MCAF block, which will be the same as the one in Fig. 3.13 as long as the constraints are decomposed into each subband without any error. In MVDR beamforming, when the delay imposed by the response vector $\mathbf{f}$, i.e. the position of the element 1 in $\mathbf{f}$, is an integer multiple of the decimation ratio N of the filter banks, such decomposition will be perfect [67, 108]. Since $\mathbf{f}$ is set to be $\mathbf{f} = [1 \ 0 \ 0 \cdots 0]^T$ in our simulations, it can be expected that almost no performance difference will be observed in this case between the methods in Fig. 3.12 and Fig. 3.13, which will be shown in our simulation.

3.3.2 Analysis of the Computational Complexity

In Chap. 2, we have shown that the dimension of the blocking matrix $\tilde{\mathbf{B}}$ is $M \times (M - S)$ when $S-1$ order derivative constraints are imposed. However, we will see later in Chap. 4 that $\tilde{\mathbf{B}}$ can be an $M \times L$-dimensional matrix with $L \leq M - S$, where the case of $L < M - S$ corresponds to a partially adaptive GSC [14, 15], whereas $L = M - S$ is the fully adaptive scenario. To make our discussion general, we will assume the dimension of $\tilde{\mathbf{B}}$ to be $M \times L$, and its L output signals will be $u_m[n]$, $m = 0, 1, \cdots, L - 1$.

For the fullband GSC, its adaptive part is an L-channel adaptive system with a complexity as given in Tab. 2.1 and Tab. 2.2, where the total number of the adaptive filter coefficients is $l_a = L \cdot J$. Moreover, because both $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{w}}_q$ are real-valued, they will need a total of $(ML + M)$ or $2(ML + M)$ real multiplications for real or complex input signals, respectively.

For our proposed subband adaptive GSC, the quiescent vector and the blocking matrix require the same number of multiplications as the fullband GSC. The subband adaptive part of the structure is composed of the filter banks and the adaptive part, which will be analysed separately in the following.

For the filter banks, there are a total of $L + 1$ analysis filter banks and one synthesis filter bank. Let us assume that we employ oversampled GDFT filter banks with a prototype filter of length l_p. Then for each filter bank $\frac{1}{N}(l_p + 4K \log_2 K + 4K)$ real multiplications are necessitated for a real-valued input and $\frac{1}{N}(2l_p + 4K \log_2 K + 8K)$ real multiplications are required for a complex-valued input, as given in Equations (3.28) and (3.29). Thus, the total number of real multiplications required for the filter banks is $\frac{L+2}{N}(l_p + 4K \log_2 K + 4K)$ or

$\frac{L+2}{N}(2l_p + 4K\log_2 K + 8K)$ for real or complex-valued input, respectively.

For the adaptive part, in general, it yields a reduction in computational complexity by a factor of $\mathcal{O}(K/N^2)$ for LMS-type algorithms and $\mathcal{O}(K/N^3)$ for RLS algorithms, as mentioned earlier. More specifically, in the real-valued input case, it requires $\frac{K}{2N}(8LJ/N + 2)$ (LMS), $\frac{K}{2N}(8LJ/N + 2L + 3)$ (NLMS) or $\frac{K}{2N}(12(LJ/N)^2 + 12LJ/N + 4)$ (RLS) real multiplications, because the number of adaptive filter coefficients required for each of the subband MCAF is LJ/N and only $K/2$ number of complex subbands decimated by $N < K$ have to be processed. If the array input signals are complex-valued, then the computational complexity will be $\frac{K}{N}(8LJ/N+2)$ (LMS), $\frac{K}{N}(8LJ/N+2L+3)$ (NLMS) or $\frac{K}{N}(12(LJ/N)^2+12LJ/N+4)$ (RLS) because we have to process all of the K subband MCAFs.

The above computational complexities are summarised in Tab. 3.1. To compare our method with the one in [1, 3], Tab. 3.1 also includes the computational complexity of the system in Fig. 3.12. As its adaptive part is the same as that in our proposed method, it has the same computational complexity for that part. However, there are a total of $M+1$ filter banks involved, which have a complexity of $\frac{M+1}{N}(l_p + 4K\log_2 K + 4K)$ and $\frac{M+1}{N}(2l_p + 4K\log_2 K + 8K)$ real multiplications for real-valued and complex-valued sensor signals, respectively. For the quiescent vectors and blocking matrices, $\frac{K}{2} \cdot 2(ML + M)$ or $K \cdot 2(ML + M)$ real multiplications are required in the context of real or complex sensor signals.

Example. In order to characterise the achievable computational complexity reduction, we give an example for a beamformer having $M = 12$ sensors, each with a variable number of J coefficients. The first-order derivative constraint ($S = 2$) is applied, hence we have $L{=}M-2$. By employing the oversampled GDFT filter banks [94] having $K = 8$ channels as characterised in Fig. 3.14 with near perfect reconstruction and a decimation factor of $N = 6$, J/N number of coefficients are used in each channel of the subband MCAF block. The prototype filter for the GDFT filter banks has a length of $l_p = 96$. Then for a real input sensor signal, the number of real multiplications required for the fullband adaptive GSC, for the subband GSC of Fig. 3.12 and for our proposed subband adaptive GSC expressed as a function of the adaptive filter length J using the NLMS algorithm can be shown in Fig. 3.15.

Note that the results shown in Tab. 3.1 are based on the most general case of the blocking matrix. If $\tilde{\mathbf{B}}$ is formed by the CCD method, then there will be no multiplications in the blocking matrix, and the term $(ML+M)$ seen in Tab. 3.1 accounting for the operations in the quiescent vector and the blocking matrix is reduced to M.

Tab. 3.1: Computational complexities for fullband and subband GSCs:

GSC structure	real multiplications (real-valued sensor signal)
fullband GSC	
LMS	$(ML+M)+(2LJ+1)$
NLMS	$(ML+M)+(2LJ+2L+2)$
RLS	$(ML+M)+(3L^2J^2+3LJ)$
GSC (Fig. 3.12)	
LMS	$K(ML+M)+\frac{M+1}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(8\frac{LJ}{N}+2)$
NLMS	$K(ML+M)+\frac{M+1}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(8\frac{LJ}{N}+2L+3)$
RLS	$K(ML+M)+\frac{M+1}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(12\frac{L^2J^2}{N^2}+12\frac{LJ}{N}+4)$
proposed GSC	
LMS	$(ML+M)+\frac{L+2}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(8\frac{LJ}{N}+2)$
NLMS	$(ML+M)+\frac{L+2}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(8\frac{LJ}{N}+2L+3)$
RLS	$(ML+M)+\frac{L+2}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(12\frac{L^2J^2}{N^2}+12\frac{LJ}{N}+4)$

GSC structure	real multiplications (complex-valued sensor signal)
fullband GSC	
LMS	$2(ML+M)+(8LJ+2)$
NLMS	$2(ML+M)+(8LJ+2L+3)$
RLS	$2(ML+M)+(12L^2J^2+12LJ+4)$
GSC (Fig. 3.12)	
LMS	$2K(ML+M)+\frac{M+1}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(8\frac{LJ}{N}+2)$
NLMS	$2K(ML+M)+\frac{M+1}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(8\frac{LJ}{N}+2L+3)$
RLS	$2K(ML+M)+\frac{M+1}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(12\frac{L^2J^2}{N^2}+12\frac{LJ}{N}+4)$
proposed GSC	
LMS	$2(ML+M)+\frac{L+2}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(8\frac{LJ}{N}+2)$
NLMS	$2(ML+M)+\frac{L+2}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(8\frac{LJ}{N}+2L+3)$
RLS	$2(ML+M)+\frac{L+2}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(12\frac{L^2J^2}{N^2}+12\frac{LJ}{N}+4)$

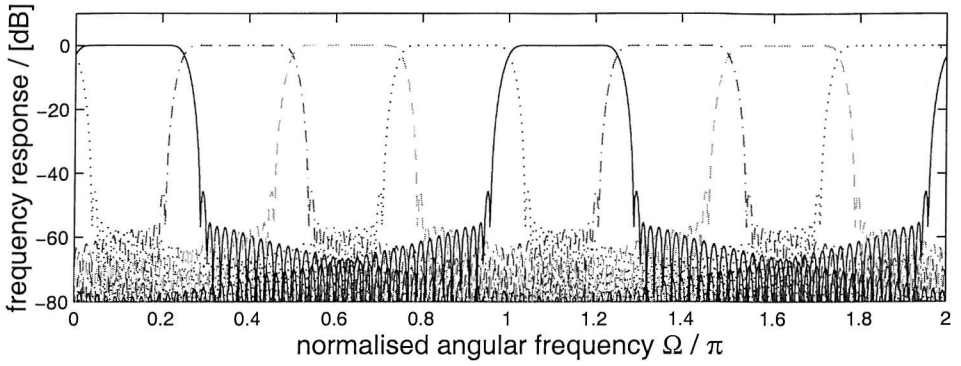


Fig. 3.14: Frequency response of a $K = 8$ channel filter bank decimated by $N = 6$.

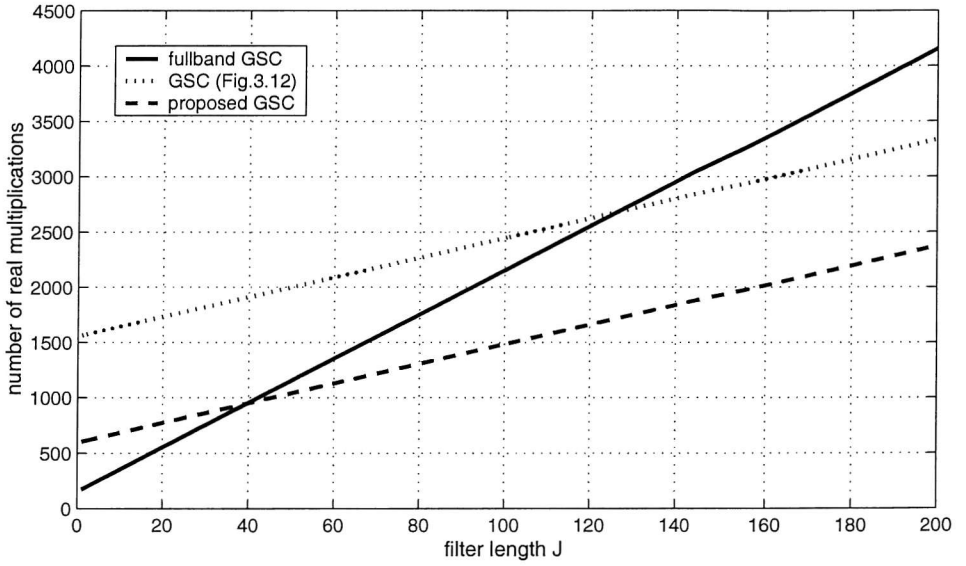


Fig. 3.15: An example of the computational complexities for different GSC structures.

3.3.3 Reconstruction of Fullband Beamformer

When the subband adaptive system reaches its steady state, we may want to calculate the fullband beamformer equivalent to the subband-structured beamformer in order to operate the beamformer in the fullband or to calculate its directivity pattern. Here, we use a procedure similar to [1] to determine the equivalent fullband model of our subband adaptive GSC. By this, each of the M branches of the beamforming structure in Fig. 2.3 has to be reconstructed separately according to the flow graph shown in Fig. 3.16. To yield the mth branch filter, the mth sensor signal is excited by the Kronecker delta function $\delta[n]$, while all other $M-1$ sensors receive a zero input. Assuming perfect adaptation of the subband components, the measured impulse response at the output of the subband adaptive GSC is a convolution of the mth filter of the equivalent fullband beamformer $\tilde{\mathbf{w}}_m$ as shown in Fig. 1.1 and the distortion function $t[n]$.

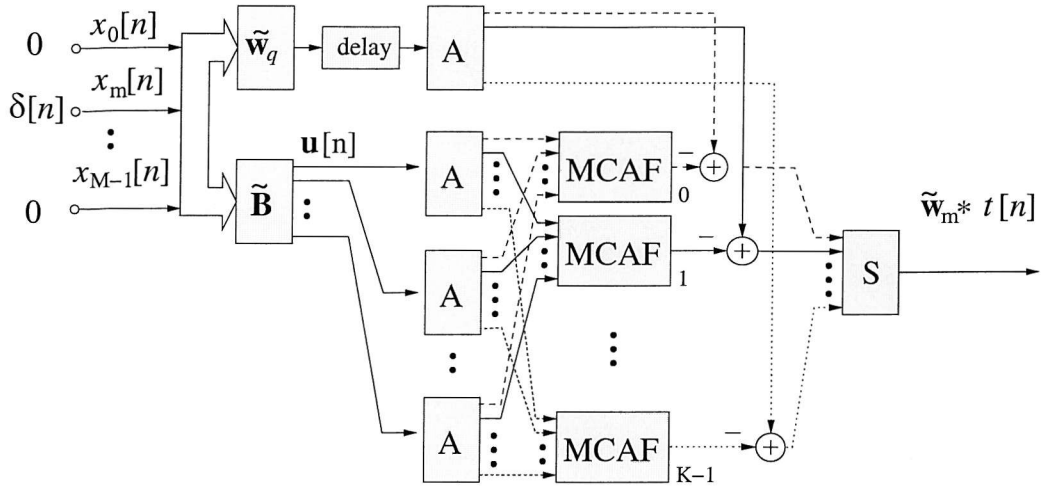


Fig. 3.16: Reconstruction of the mth equivalent fullband beamformer branch $\tilde{\mathbf{w}}_m$ from the subband adaptive structure.

This procedure is equivalent to the method described for the determination of an equivalent fullband model for a general subband adaptive system [58].

The distortion function $t[n]$ ideally only represents a delay. However, if the filter banks possess only a near perfect reconstruction property, $t[n]$ generally carries a small amplitude and phase distortion. As in the case for general SAF systems [58], this limitation can be quantified *a priori* from the filter bank prototype by measuring its deviation from power complementarity.

3.4 Simulations and Results

In this part, we will first make a comparison of the performances between our proposed subband adaptive GSC and the subband GSC structure in Fig. 3.12 by a set of simulations to show their similarity in performance. To demonstrate the effectiveness of our subband-based method in relation to the fullband approach, another two sets of simulations are provided, whereby narrowband and broadband interfering signals are to be suppressed.

3.4.1 Proposed Method versus the Method of Fig. 3.12

We employ the same array as in the example specified in Sec. 3.3.2 with the subband decomposition as given in Fig. 3.14. The filter length is set to $J = 96$. Our aim is to receive a white Gaussian signal of interest from broadside ($\theta = 0°$) and to adaptively suppress a broadband interfering signal covering the frequency interval $\Omega = [0.2\pi; 0.8\pi]$ arriving from a DOA of $\theta = 30°$ at a signal-to-interference ratio (SIR) of -20 dB. Additionally, the sensor signals are corrupted

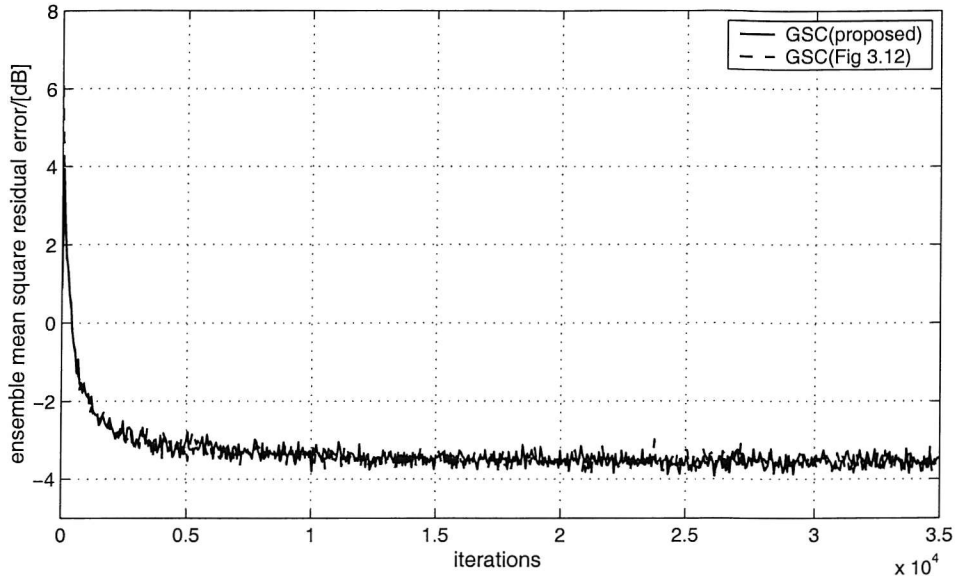


Fig. 3.17: Learning curves of the two subband methods based on CCD (S=1).

by independent white noise resulting in a signal-to-noise ratio (SNR) of 20 dB. A normalised LMS algorithm with a step size of 0.3 is used in all cases.

To assess the convergence of the two different beamformers, we consider the ensemble mean square value of the residual error, which is defined as the difference between the array output $e[n]$ and the appropriately delayed desired signal from broadside. Fig. 3.17 and Fig. 3.19 compare the ensemble mean square residual errors of these two subband beamformers based on the CCD method employed for forming the blocking matrix, whereas Figs. 3.18 and 3.20 are based on the SVD method. In either case, these two subband beamformers have a fairly similar performance in terms of convergence speed and steady state error, as we have pointed out earlier in Sec. 3.3.1, while the computational complexities of the proposed scheme are only about 90% and 60% of those for the structure in Fig. 3.12 based on the CCD and SVD approaches, respectively.

3.4.2 Proposed Method versus Fullband Method

In the following, the proposed subband method is compared to the fullband method in narrowband and broadband scenarios, respectively.

Narrowband Interference

In this simulation, we use a beamformer with $M = 10$ sensors and $J = 60$ coefficients for each attached filter. The system is constrained to receive a signal of interest from broadside, which is white Gaussian with unit variance and corrupted by additive Gaussian noise at an SNR of 20 dB. The beamformer aims to adaptively suppress an interfering signal coming from a DOA

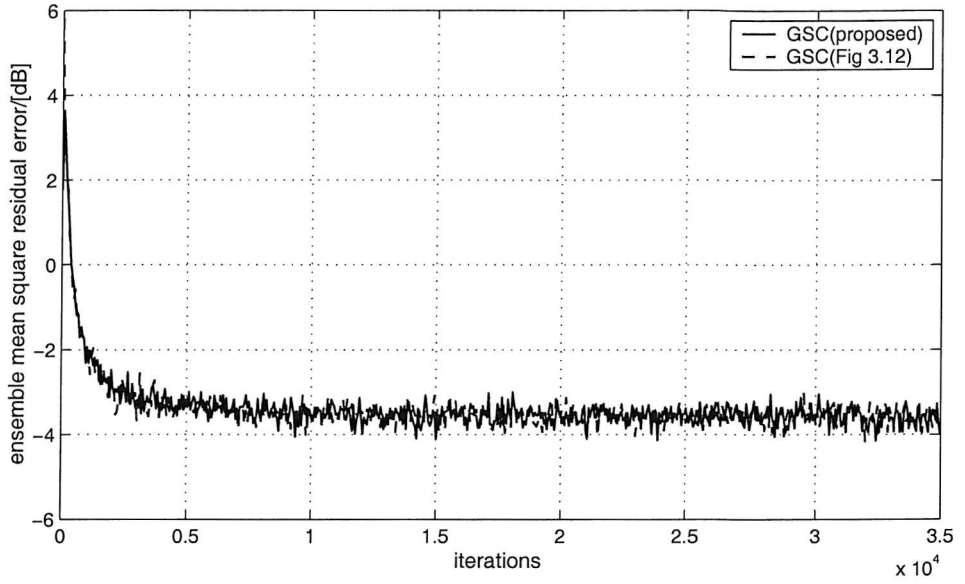


Fig. 3.18: Learning curves of the two subband methods based on SVD (S=1).

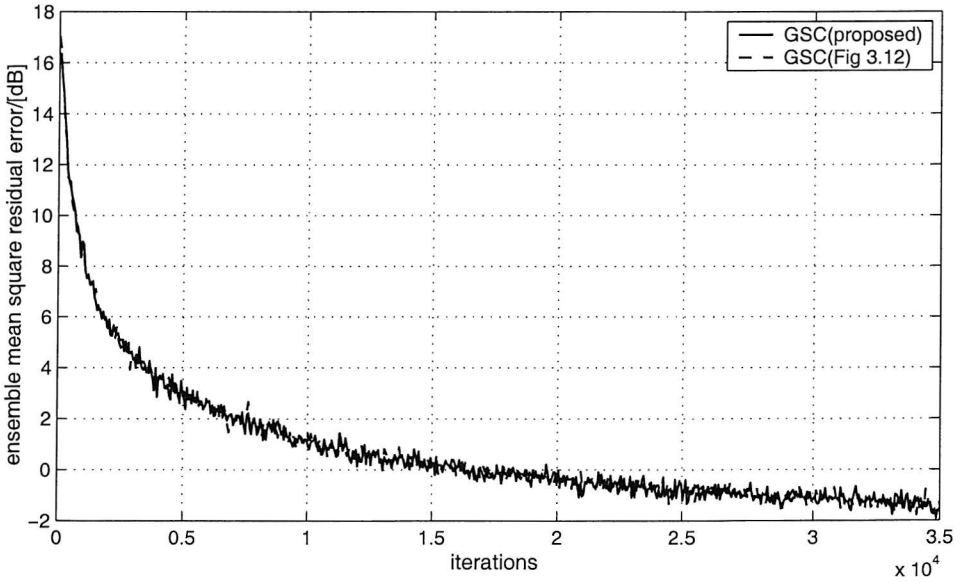


Fig. 3.19: Learning curves of the two subband methods based on CCD (S=2).

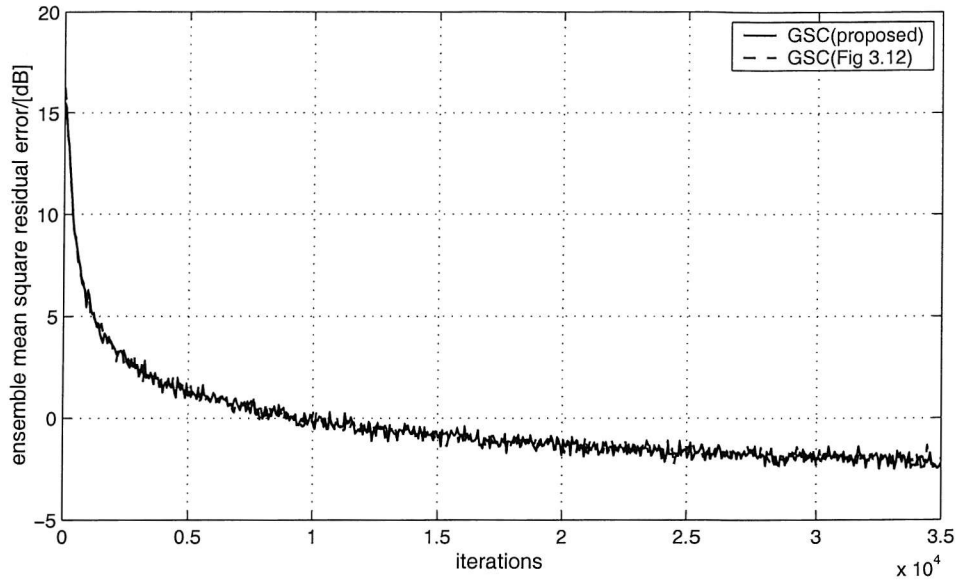


Fig. 3.20: Learning curves of the two subband methods based on SVD (S=2).

of $\theta = 20°$, which is composed of three sinusoidal signals with normalised frequencies of 0.1π, 0.5π and 0.9π, respectively. The SIR is -24 dB and no derivative constraints are imposed ($S = 1$). The subband GSC is based on the same 8-channel filter banks as in Sec. 3.4.1 and $J/N = 10$ coefficients are used for each subband MCAF channel. The MCAFs for both the fullband and subband cases are updated by a normalised LMS algorithm.

Because the interference is narrowband, in principle, we do not really need a broadband beamformer with tapped-delay lines to suppress it. The aim of this set of simulation is mainly to show the significant improvement in convergence speed by the subband method when the input signal is highly colored. Four simulation results with different step sizes and different forms of the blocking matrix are shown in Figs. 3.21 to 3.24. Comparing Fig. 3.21 with Fig. 3.22 and Fig. 3.23 with Fig. 3.24, we can see that with increasing step size, both the fullband and subband adaptive beamformers converge faster, at the expense of an increased steady-state error. This conforms with standard results on adaptive filtering [51,57], however the impact of the stepsize on the system's performance is particularly severe in the GSC case. One possible explanation is that the error signal employed for adaptation in the GSC includes the signal of interest, which could be much stronger than the error signal found in a standard adaptive filter. Thus, the impact of the step size is amplified in the GSC case. Moreover, all the results clearly show that the subband method converges much faster than the fullband method because of its pre-whitening effect.

Broadband Interference

In this simulation, the beamformer has $M = 12$ sensors, each with $J = 96$ coefficients for the

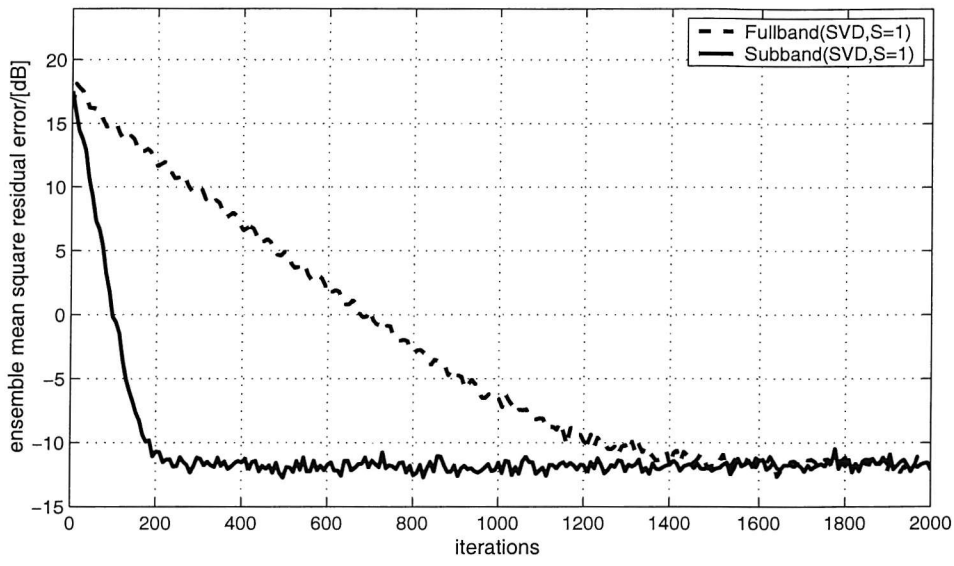


Fig. 3.21: Comparison of learning curves based on SVD method (step size μ_0=0.06).

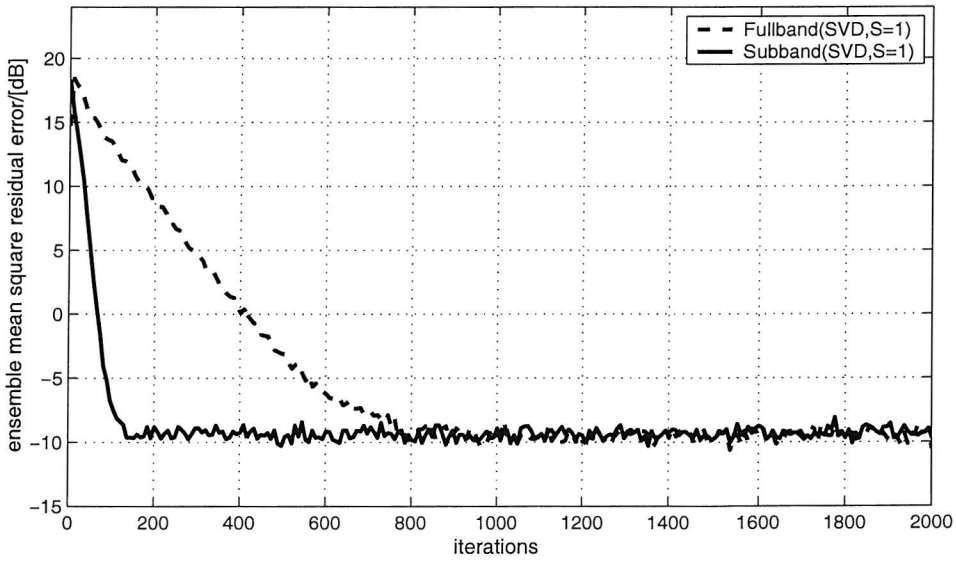


Fig. 3.22: Comparison of learning curves based on SVD method (step size μ_0=0.10).

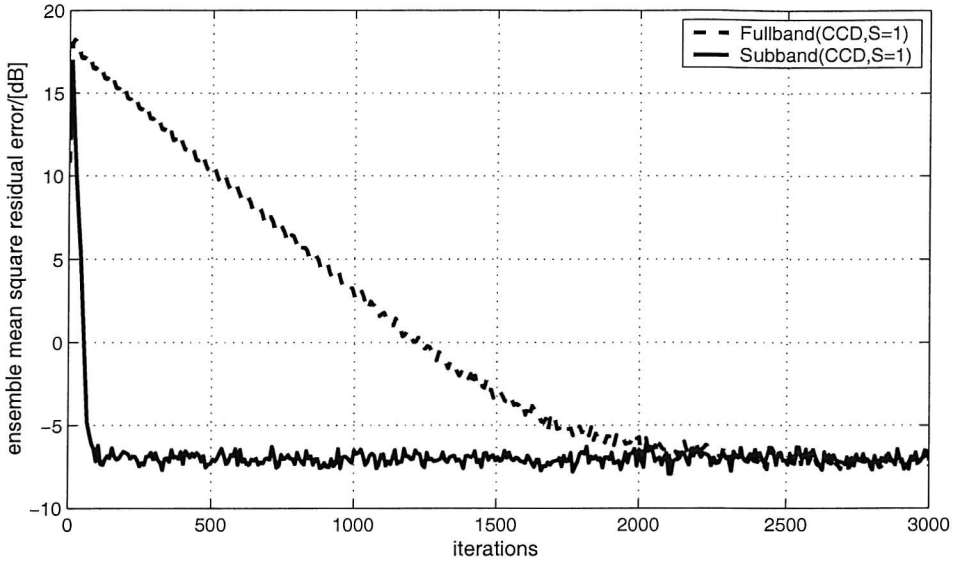


Fig. 3.23: Comparison of learning curves based on CCD method (step size μ_0=0.15).

fullband case. No derivative constraints are imposed ($S = 1$). The beamformer is intended to receive the same signal of interest from the broadside ($\theta = 0°$) as in the narrowband case, and to adaptively suppress two broadband interfering signals covering the frequency intervals $\Omega = [0.2\pi; 0.5\pi]$ and $\Omega = [0.5\pi; 0.8\pi]$, respectively, with DOAs of $\theta = 30°$ and $\theta = 60°$. The SIR and SNR are respectively -24 dB and 20 dB. In the subband adaptive GSC, we use the same filter banks as in the previous simulation and $J/N = 16$ coefficients are employed for each subband MCAF channel.

Figs. 3.25 and 3.26 show two representative simulation results, where the subband adaptive method exhibits a clear advantage in terms of a faster convergence speed due to the pre-whitening effect. Moreover, the subband method reaches a lower steady-state error at the specified stepsizes. We can expect that the fullband method will be even slower if it has the same steady-state error as the subband method. As for the computational complexity, the number of real multiplications required in the subband system is only about 69% of that of the fullband system in the SVD case and 67% in the CCD case. With its lower computational complexity and faster convergence, the proposed subband adaptive GSC clearly outperforms the traditional fullband implementation.

3.5 Summary

In this chapter, we have briefly reviewed the subband adaptive filtering techniques and the related theory of filter banks, which are employed for the subband decomposition of both the

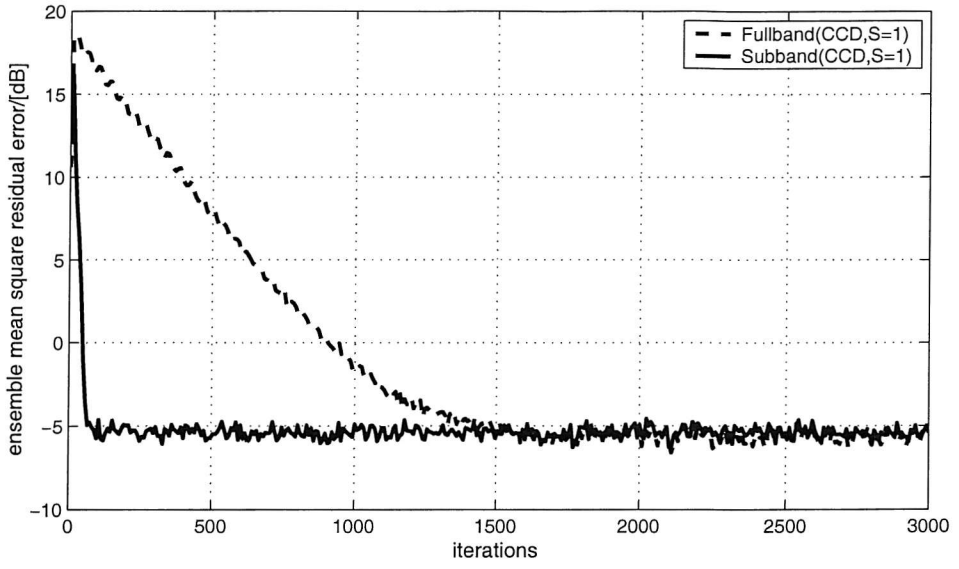


Fig. 3.24: Comparison of learning curves based on CCD method (step size μ_0=0.20).

desired and input signals in an adaptive filter. Its advantages of low computational complexity and increased convergence speed for the LMS-type adaptive algorithms motivate the application of SAF techniques to broadband adaptive beamforming. By combining SAF techniques with the multichannel adaptive filter in a GSC with derivative constraints, we have obtained a novel subband adaptive GSC structure. Compared with the traditional fullband adaptive GSC and a previously proposed subband beamforming structure, this new subband adaptive beamformer outperforms those methods with a lower computational complexity. Additionally, it also achieves a faster convergence speed than the fullband adaptive GSC owing to its pre-whitening effect, as demonstrated by our simulation results based on different signal environments and different formations of the blocking matrix. In the next chapter, we will see that the computational complexity of such a subband adaptive GSC can be reduced even further by contriving a specific arrangement for the blocking matrix.

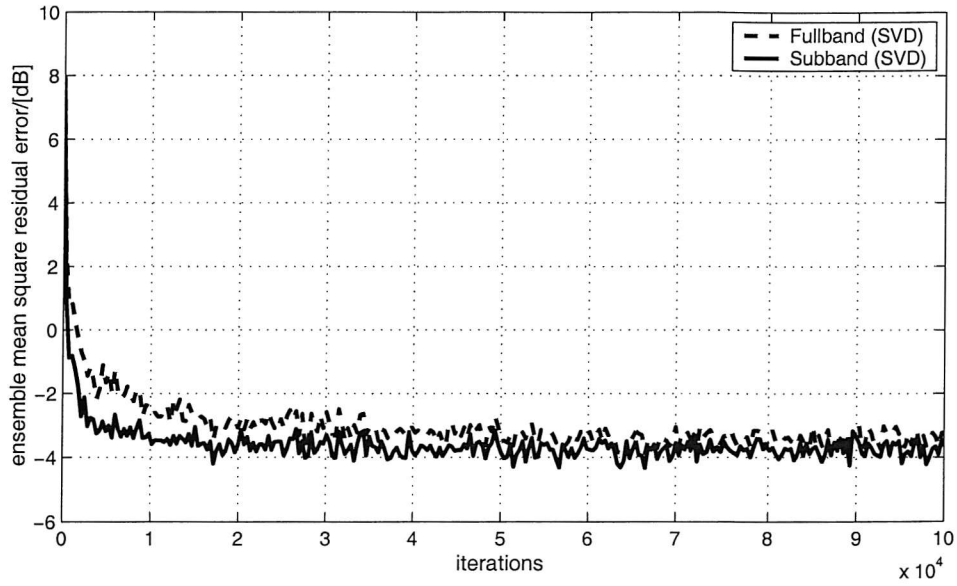


Fig. 3.25: Comparison of learning curves based on SVD method (step size μ_0=0.3).

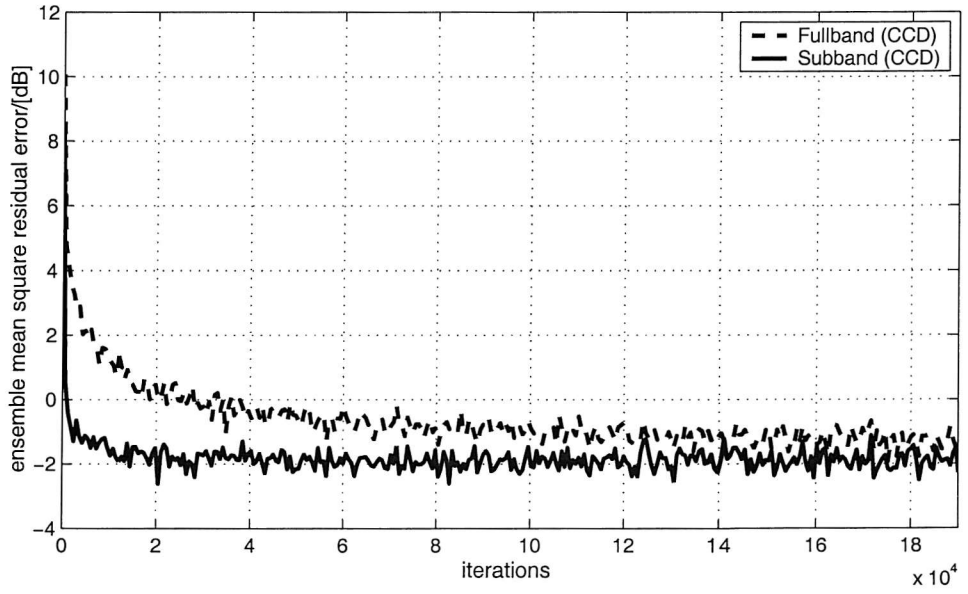


Fig. 3.26: Comparison of learning curves based on CCD method (step size μ_0=0.4)

Chapter 4

GSC Employing a Subband-selective Blocking Matrix

In this chapter, we propose a class of GSCs employing a novel spatially/temporally subband-selective blocking matrix for partially adaptive beamforming, in which the blocking matrix is constructed such that the impulse responses hosted by its columns constitute a series of bandpass filters. These filters select signals with specific DOAs and frequencies and result in bandlimited spectra of the blocking matrix outputs. When we apply such a blocking matrix to the subband adaptive GSC proposed in Chap. 3 or the transform-domain GSC as proposed in [21], the system complexity may be reduced. Moreover, due to its combined decorrelation in both spatial and temporal domains, a faster convergence rate can also be achieved.

In Sec. 4.1 we will give a brief introduction to partially adaptive beamforming based on the GSC structure. Thereafter we focus on the role of the blocking matrix and its proposed construction exploiting spatial/temporal filtering properties in Sec. 4.2. In Sec. 4.3, an alternative formation of this blocking matrix by a subband-selective transformation matrix is provided. The application of our novel blocking matrix to the subband adaptive GSC and transform-domain GSC is given in Secs. 4.4 and 4.5, respectively, where simulations underlining the benefits of the proposed method are discussed.

4.1 Partially Adaptive GSC

As mentioned in Sec. 1.1, to perform beamforming with high interference rejection and resolution, arrays with a large number of sensors and filter coefficients have to be employed and the computational burden of a fully adaptive processor thus becomes considerable. One way to reduce the computational complexity is partially adaptive beamforming, which employs only a

subset of available degrees of freedom (DOFs) in the weight update process at the expense of a somewhat reduced performance [14].

Partially adaptive techniques have been studied widely in the past, especially in the narrowband beamforming area. Chapman [14] performed one of the earliest works, which reduces the number of adaptive channels by means of a transformation. In [15], a fixed transformation was applied to achieve a weight reduction by minimising the output power over a set of interference scenarios. A "power-space method", which uses a singular value decomposition to obtain the rank-reducing transformation was developed by Yang *et al* [17]. In broadband beamforming, the wavelet-based beamformer introduced by Wang *et al* [18,113] reduces the dimension of the blocking matrix by utilising a set of wavelet filters, which are considered as a series of spatial filters. Thereafter, a dynamic selection of the blocking matrix outputs is performed by a prescribed statistical hypothesis test. The main problem with [18,113] is that the design of wavelet filters with good band-selection is difficult and such filters will sacrifice too many DOFs and tend to make their application in a beamformer rather unpractical.

For a brief description of partially adaptive beamforming based on the GSC structure, we consider the simplified system shown in Figs. 2.7 and 2.8. There the only constraint imposed on the blocking matrix $\tilde{\mathbf{B}}$ is equation (2.61). For convenience, we recall

$$\tilde{\mathbf{C}}^H\tilde{\mathbf{B}} = \mathbf{0} \qquad \text{where} \quad \tilde{\mathbf{C}} = [\mathbf{c}_0 \ \cdots \mathbf{c}_{S-1}] \ , \tag{4.1}$$

with $\tilde{\mathbf{C}} \in \mathbb{C}^{M\times S}$ and $\tilde{\mathbf{B}} \in \mathbb{C}^{M\times M-S}$. However, the column dimension of $\tilde{\mathbf{B}}$ is not restricted to be $L = M - S$, which is only the maximum possible value. Assume $\tilde{\mathbf{B}} \in \mathbb{C}^{M\times L}$ with

$$\tilde{\mathbf{B}} = [\mathbf{b}_0 \ \mathbf{b}_1 \ \ldots \ \mathbf{b}_{L-1}] \qquad \text{and} \tag{4.2}$$

$$\mathbf{b}_l = [b_l[0] \ b_l[1] \ \ldots \ b_l[M-1]]^H \ , \tag{4.3}$$

where $l = 0, 1, \cdots, L-1$. With this new definition, we redraw the simplified GSC for MVDR beamforming in Fig. 4.1, where the output signal of the blocking matrix $\mathbf{u}[n]$ is obtained by $\mathbf{u}[n] = \tilde{\mathbf{B}}^H\mathbf{x}[n]$ with

$$\begin{aligned} \mathbf{u}[n] &= [u_0[n] \ u_1[n] \ \cdots \ u_{L-1}[n]]^T \ , \\ \mathbf{x}[n] &= [x_0[n] \ x_1[n] \ \cdots \ x_{M-1}[n]]^T \ . \end{aligned} \tag{4.4}$$

Since $\tilde{\mathbf{C}}$ is an $M \times S$ matrix, the dimension L can be selected arbitrarily with $L \leq M - S$. The maximum value $M - S$ corresponds to the fully adaptive GSC. When a large number of sensors are employed, we can take a smaller value for L, i.e. $L < M - S$, resulting in a partially adaptive GSC [15,113]. By partial adaptivity, the number of DOFs, i.e. the number of adaptive weights, is reduced and it offers reduced complexity traded off against a potentially somewhat inferior performance. In the next section, we will trade the loss of DOFs against a specific design of the blocking matrix.

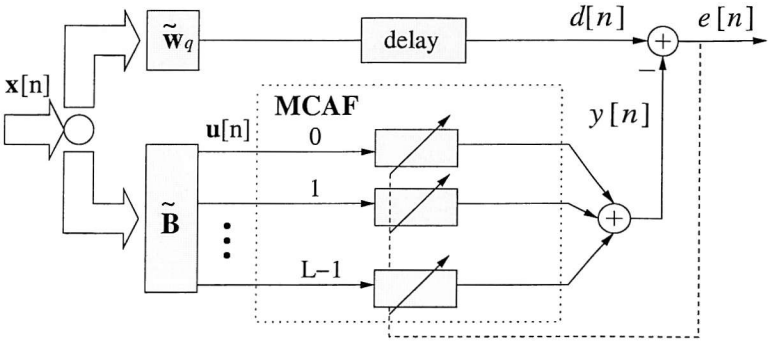


Fig. 4.1: A general structure of partially adaptive GSC.

4.2 Spatially/Temporally Subband-selective Blocking Matrix

4.2.1 Blocking Matrix with Spatial/Temporal Subband-selectivity

To attain an interpretation of spatially and temporally subband-selective filters constituting the blocking matrix, consider a unity amplitude complex input wave with angular frequency ω and DOA θ. Referring to Fig. 1.1, the waveform impinges with a time delay $\Delta\tau$ on adjacent sensors separated by d in a medium with propagation speed c. The received phase vector at the sensor array, $\underline{\mathbf{x}}$, is

$$\underline{\mathbf{x}} = \left[1 \;\; e^{-j\omega\Delta\tau} \;\; \cdots \;\; e^{-j\omega(M-1)\Delta\tau}\right]^T \quad \text{with} \quad \Delta\tau = \frac{d}{c}\sin\theta \,. \tag{4.5}$$

Assume that the array sensors are spaced by a half wavelength of the maximum signal frequency and the temporal sampling frequency ω_s is twice the maximum signal frequency, i.e.,

$$d = \lambda_s = cT_s \,, \tag{4.6}$$

where T_s is the temporal sampling period. Then, we get

$$\Delta\tau = T_s \sin\theta \,. \tag{4.7}$$

Noting $\omega T_s = \Omega$, where Ω is the normalised angular frequency of the signal, the phase vector can be written as

$$\underline{\mathbf{x}} = \left[1 \;\; e^{-j\Omega\sin\theta} \;\; \cdots \;\; e^{-j(M-1)\Omega\sin\theta}\right]^T \,. \tag{4.8}$$

Using the substitution $\Psi = \Omega\sin\theta$, the lth output of the blocking matrix, $u_l[n]$, $l = 0, 1, \cdots, L-1$, can be denoted as

$$u_l[n] = \mathbf{b}_l^H \cdot \underline{\mathbf{x}} \cdot e^{jn\Omega} = \sum_{m=0}^{M-1} b_l[m] \cdot e^{-jm\Psi} \cdot e^{jn\Omega} = B_l(e^{j\Psi}) \cdot e^{jn\Omega}, \tag{4.9}$$

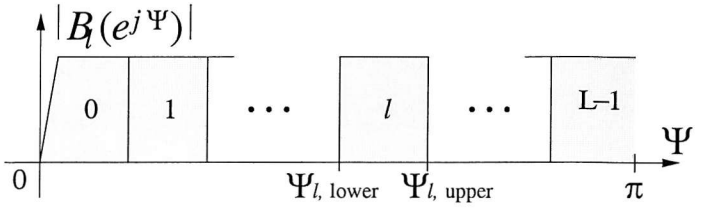


Fig. 4.2: Characteristics of the L column vectors contained in $\tilde{\mathbf{B}}$

with $B_l(e^{j\Psi})$ •—○ $b_l[m]$ being a Fourier transform pair.

According to (4.9), the columns of the blocking matrix $\tilde{\mathbf{B}}$ can be regarded as a set of spatial filters. If the beamformer is constrained to receive the signal of interest from broadside, then the blocking matrix has to suppress any component impinging from $\theta = 0$. Therefore, at $\Psi = 0$ the response of any column vector $\mathbf{b}_l$ has to be zero. Now we arrange the design of the $\mathbf{b}_l$ to yield spatial bandpass filters on the interval $\Psi \in [0; \pi]$ as shown in Fig. 4.2, whereby ideally

$$|B_l(e^{j\Psi})| = \begin{cases} 1 & \text{for} \quad \Psi \in [\Psi_{l,\mathrm{lower}}; \Psi_{l,\mathrm{upper}}] \; ; \\ 0 & \text{otherwise} \; . \end{cases} \tag{4.10}$$

In the arrangement of Fig. 4.2, all values of $\Psi \in (0; \pi]$ except $\Psi = 0$ have to be covered by the filters to ensure that the lower path of the GSC contains all possible interference signals. If the interference signals impinge only from a certain set of angles with a certain bandwidth, then the appeal is that only some outputs of the blocking matrix and therefore some branches of the subsequent multichannel adaptive filter will contain significant contributions. Hence, a design of the blocking matrix columns according to Fig. 4.2 will lead to a spatial decomposition or decorrelation of the array data.

To avoid redundancy in the blocking matrix outputs, we would like the column vectors of $\tilde{\mathbf{B}}$ to be linearly independent, i.e. none of them can be expressed as a linear combination of the others. It can be proven that these column vectors $\mathbf{b}_l$ are orthogonal under the ideal arrangement of Fig. 4.2. If an overlap between neighbouring bands in Fig. 4.2 exists, these column vectors $\mathbf{b}_l$ can also be proven to be at least linearly independent [114]. To prove this, we consider the linear combination of all these vectors in the following form

$$\mathbf{0} = \alpha_0 \mathbf{b}_0 + \alpha_1 \mathbf{b}_1 + \cdots + \alpha_{L-1} \mathbf{b}_{L-1} \; , \tag{4.11}$$

where $\alpha_0, \cdots, \alpha_{L-1}$ are scalars to be found for this equation to hold. Taking the Hermitian transpose of both sides and then multiplying the equation with the vector $\left[1 \;\; e^{j\Psi} \;\; \ldots \;\; e^{j(L-1)\Psi}\right]^T$, we arrive at

$$0 = \alpha_0 B_0(e^{j\Psi}) + \alpha_1 B_1(e^{j\Psi}) + \cdots + \alpha_{L-1} B_{L-1}(e^{j\Psi}) \; . \tag{4.12}$$

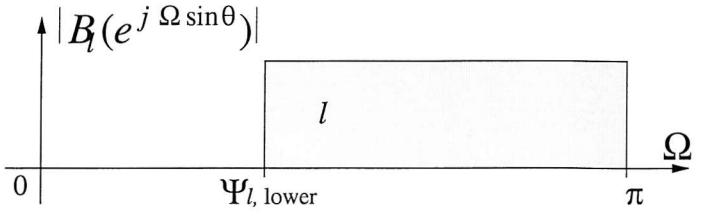


Fig. 4.3: Temporal filtering effect of the lth spatial filter in $\tilde{\mathbf{B}}$.

As it should hold for all values of Ψ, we choose the middle point of each of the passbands of the column vectors. Since at these points only one of the L Fourier transforms $B_l(e^{j\Psi})$, $l = 0, 1, \cdots, L-1$, has a value of unity and all the others are zero, we find that all the scalars $\alpha_0, \cdots, \alpha_{L-1}$ must be zero for (4.11) to hold, i.e. the column vectors are linearly independent and hence the proof is complete. From the above proof we can see that as long as every vector has at least one point, where its Fourier transform has a nonzero value and coincides with zeros of all the other vectors' Fourier transforms, these vectors will be linearly independent. A practical test to check any design of $\tilde{\mathbf{B}}$ is to ensure that the blocking matrix has full column rank.

We now focus on the temporal filtering effect of the arrangement in (4.10). Consider the range $\Omega \in [0; \pi]$. As $|\sin\theta| \in [0; 1]$ when $\theta \in [-\frac{\pi}{2}; \frac{\pi}{2}]$, the possible maximum frequency component of the lth output $u_l[n]$ is $\Omega = \pi$, which corresponds to $|\sin\theta| = \frac{\Psi_{l,\mathrm{upper}}}{\pi}$, while the possible minimum frequency component is $\Omega = \Psi_{l,\mathrm{lower}}$, which corresponds to $|\sin\theta| = 1$. Then we have the response of $B_l(e^{j\Psi})$ as a function of $\Omega = \Psi/\sin\theta$,

$$|B_l(e^{j\Omega\sin\theta})| = \begin{cases} 1 & \text{for } \Omega \in [\Psi_{l,\mathrm{lower}}; \pi] \\ 0 & \text{otherwise} \end{cases} \quad \forall\, \theta\,. \tag{4.13}$$

This temporal filtering characteristic of the lth column vector of $\tilde{\mathbf{B}}$ is shown in Fig. 4.3. If the interfering signals have components with $\Omega > \pi$, then as long as they appear in the directions of $|\sin\theta| \in [\frac{\Psi_{l,\mathrm{lower}}}{\Omega}; \frac{\Psi_{l,\mathrm{upper}}}{\Omega}]$, they will be received by the lth column vector. However, with the assumption that the sampling rate is twice the maximum signal frequency, there will be no signal existing with $\Omega > \pi$.

Example. To demonstrate this temporal filtering effect, we give a simple example. Fig. 4.4 displays the magnitude response of a 30-tap bandpass filter designed by the MATLAB function *remez* [115]. If the filter coefficients are employed as a column vector in $\tilde{\mathbf{B}}$, then a gain response or beampattern to signals with different frequencies and DOAs can be calculated as shown in Fig. 4.5. To see its highpass filtering effect more clearly, the figure can be rotated and inspected in terms of its frequency dependence only. The resultant two-dimensional response

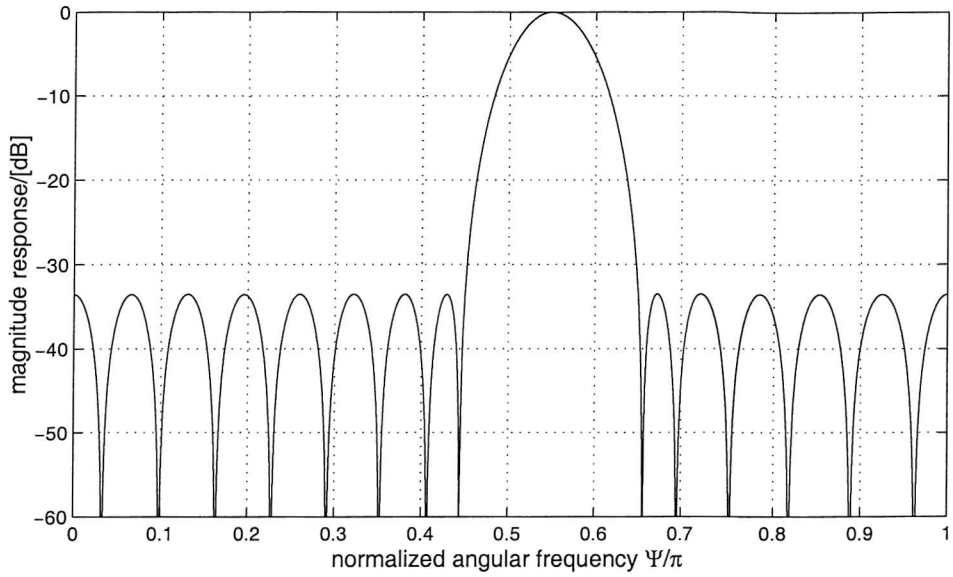


Fig. 4.4: Frequency response of a bandpass filter as an example for the column vectors of $\tilde{\mathbf{B}}$.

with respect to different frequencies is drawn in Fig. 4.6, and clearly exhibits the noted highpass characteristic.

Thus, the blocking matrix is capable of decomposing the received signals and interferences not only in the spatial domain, but also in the temporal domain, i.e. the column vectors simultaneously perform a spatial selection and a temporal highpass filtering operation. With increasing l, these filters are associated with tighter and tighter highpass spectra until the last output $u_{L-1}[n]$ only contains the ultimate highpass component.

In reality, the bandpass filters $B_l(e^{j\Psi})$ cannot be ideal and hence an overlap and finite transition bands have to be permitted. However, a better design quality can be attained by reducing the number of columns, L, below the limit of $M-S$, thus yielding a partially adaptive beamformer by sacrificing some DOFs. As the blocking matrix covers all possible interfering signals, it can still suppress any incoming interferences, but the achievable maximum SINR can potentially be lower than a fully adaptive beamformer. Due to the loss of ideal responses as specified in Fig. 4.2, the linear independence of the column vectors of $\tilde{\mathbf{B}}$ has to be inspected after design.

As the blocking matrix plays a central role in our following applications, a column vector design with a good band-selective property is of great importance. We will deal with this issue in Secs. 4.2.2 and 4.2.3, where a full design and a cosine-modulated design of the blocking matrix will be described, respectively.

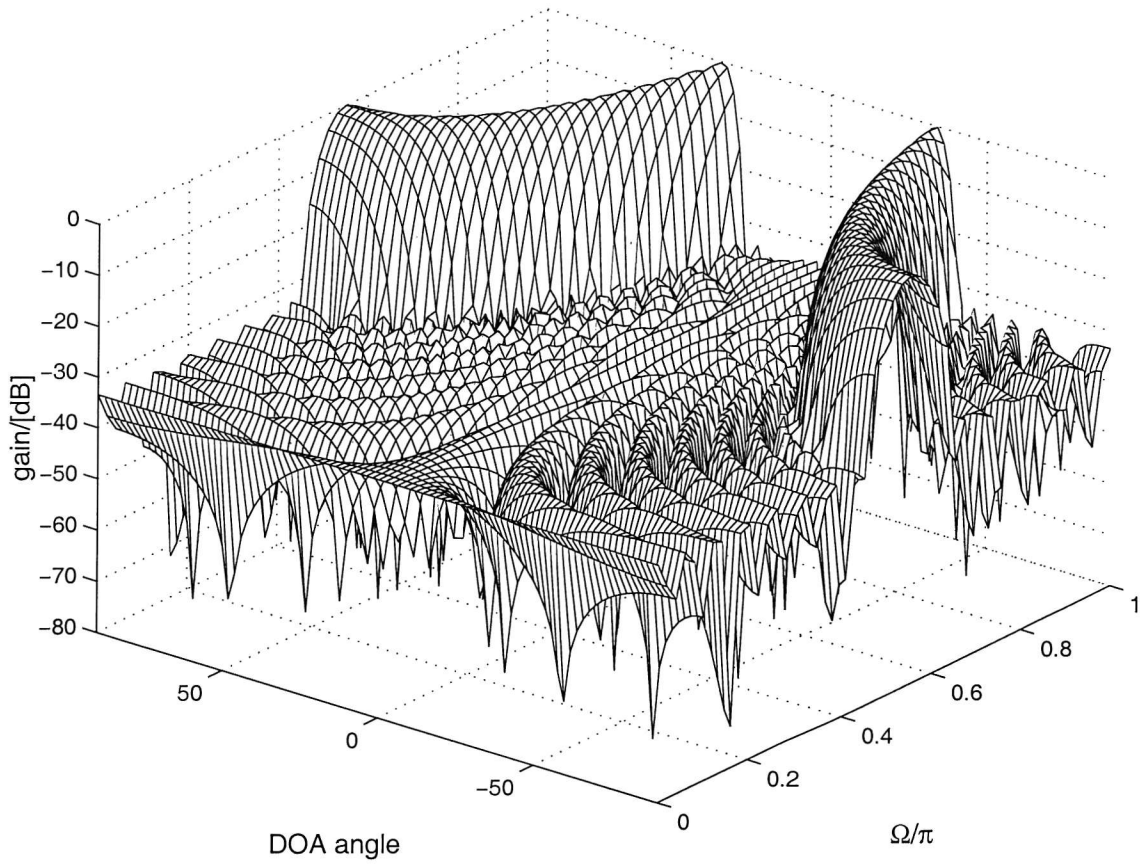


Fig. 4.5: Three-dimensional response of a bandpass filter to signals with different frequencies and DOAs.

4.2.2 Full Design of the Blocking Matrix

From (2.61), when considering $S-1$ order derivative constraints with $n_0 = 0$, we can express the constraints to be fulfilled by the lth column vector of the blocking matrix as

$$\sum_{m=0}^{M-1} m^i \cdot b_l[m] = 0 \ , \quad \text{for} \quad i = 0, 1, \cdots, S-1, \ \ l = 0, 1, \cdots, L-1 \ . \tag{4.14}$$

Subject to the constraints in (4.14), the objective function Φ_l to be minimised for the lth column vector is

$$\Phi_l = \int_0^{\Psi_{l,\text{lower}}} \|B_l(e^{j\Psi})\|^2 d\Psi \quad + \quad \int_{\Psi_{l,\text{upper}}}^{\pi} \|B_l(e^{j\Psi})\|^2 d\Psi \ . \tag{4.15}$$

The design problem of the blocking matrix $\tilde{\mathbf{B}}$ can then be formulated as the following constrained optimisation:

$$\mathbf{b}_l = \arg\min_{\mathbf{b}_l} \Phi_l \quad \text{subject to} \quad (4.14), \quad l = 0, 1, \cdots, L-1 \ . \tag{4.16}$$

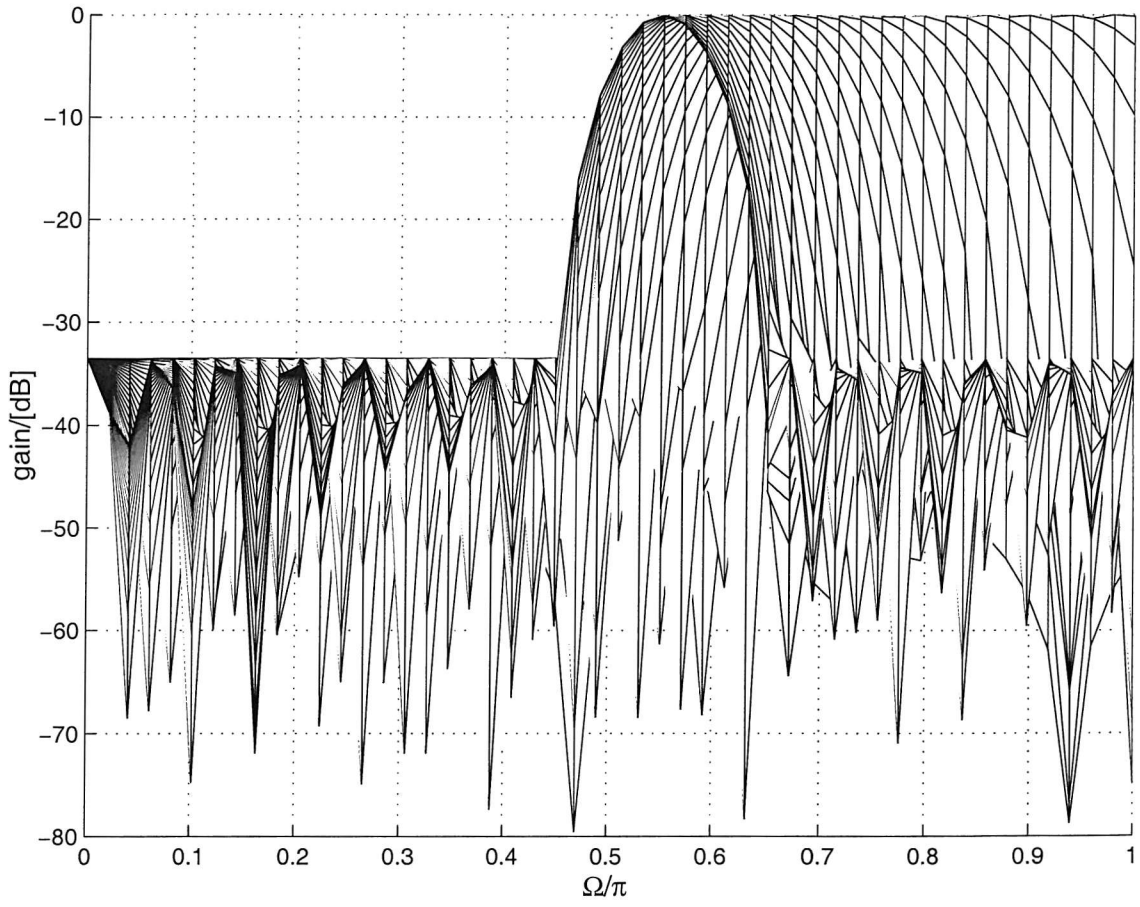


Fig. 4.6: Response of a bandpass filter with respect to signals with different frequencies as a column vector of the blocking matrix $\tilde{\mathbf{B}}$.

Various methods exist to solve linearly constrained optimisation problem such as in (4.16). Here the minimisation of Φ_l is accomplished using the subroutines LCONF/DLCONF, a nonlinear optimisation software package in the IMSL library [116]. A design example for the blocking matrix with $M = 16$ sensors, first-order constraints ($S = 2$) and $L = 8$ column filters is given in Fig. 4.7. Displayed are the frequency responses $B_l(e^{j\Psi})$, $l = 0, 1, \cdots, 7$, which exhibit a reasonably good bandpass characteristic.

If (4.14) is used to express the first S parameters in each $\mathbf{b}_l$ by the remaining $M - S$ vector elements, an unconstrained optimisation can be performed over those remaining parameters, for example by means of a genetic algorithm (GA) [117, 118]. Using a GA, we can obtain a result with all elements in $\tilde{\mathbf{B}}$ in the form of sums of power of two (SOPOT) [119–121]. By SOPOT representation, the arithmetic for $\tilde{\mathbf{B}}$ can be implemented by simple shifts and additions to further reduce its computational complexity [122, 123].

For a brief introduction to GA, the reader is referred to the Appendix. Here, as an example, let us consider the case of first order derivative constraints, $S = 2$, whereby the first two elements

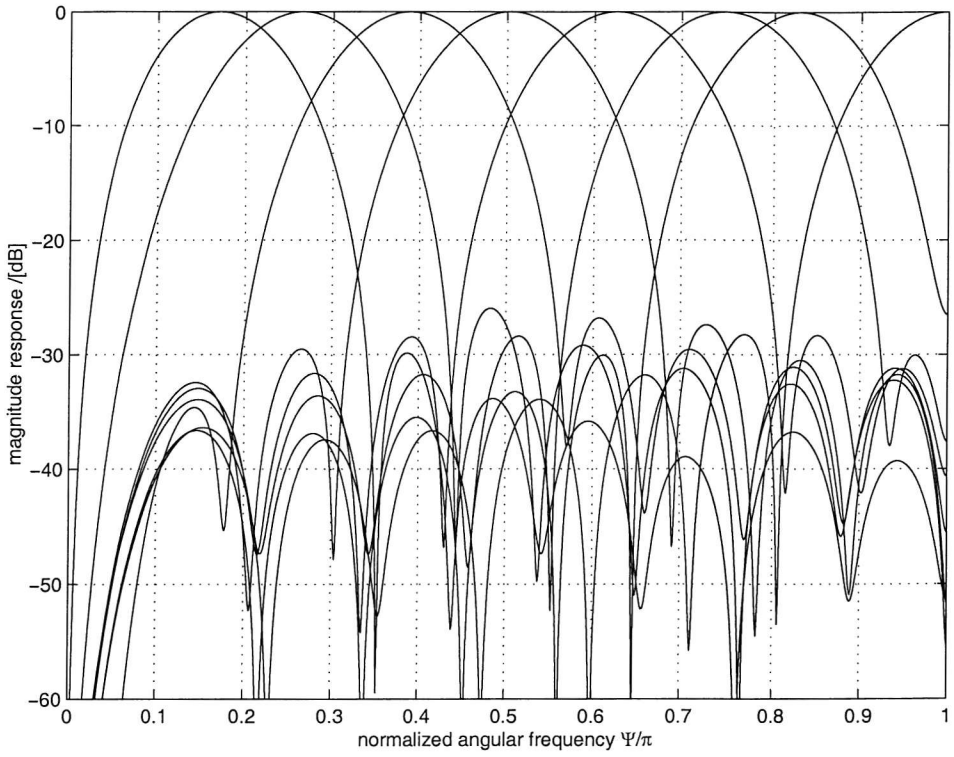


Fig. 4.7: A design example for a 16×8 blocking matrix with first-order derivative constraints based on constrained optimisation.

of $\mathbf{b}_l$ are fixed in dependency on the remaining coefficients,

$$b_l[0] = -\sum_{m=1}^{M-1} b_l[m] \qquad \text{for} \quad i = 0 \,, \tag{4.17}$$

$$b_1[1] = -\sum_{m=2}^{M-1} m \cdot b_l[m] \qquad \text{for} \quad i = 1 \,. \tag{4.18}$$

Note that the optimisation problem has been transformed into an unconstrained optimisation over the remaining coefficients $b_l[2], \cdots, b_l[M-1]$, which is straightforward to solve by means of a GA. In the optimisation process of a GA yielding SOPOT parameters, each of the coefficients $b_l[m]$, $m = 2, 3, \cdots, M-1$ is represented as

$$b_l[m] = \sum_{i=0}^{P[l,m]-1} a_i[l,m] \cdot 2^{L_i[l,m]} \qquad \text{with}$$
$$a_i[l,m] \in \{-1; 1\}, \quad L_i[l,m] \in \{Q_1, Q_1+1, \cdots, Q_2-1, Q_2\} \,, \tag{4.19}$$

where $P[l,m]$ is a limit for the number of SOPOT terms, and Q_1 and Q_2 are integers determined by the range of the corresponding variable. Normally, $P[l,m]$ is limited to a small number. With the same parameters as those of the example shown in Fig. 4.7, Tab. 4.1 gives the GA design results in SOPOT notation with $P[l,m] = 3$, $Q_1 = -9$ and $Q_2 = 0$. Its frequency response is depicted in Fig. 4.8.

Tab. 4.1: SOPOT coefficients for the 16×8 blocking matrix $\tilde{\mathbf{B}}$:

m	$\mathbf{b}_0$	$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$
2	$2^{-2}+2^{-8}$	-2^{-9}	$-2^{-2}-2^{-4}+2^{-9}$	$-2^{-1}-2^{-7}$
3	$2^{-4}+2^{-5}-2^{-8}$	$2^{-2}+2^{-5}$	$-2^{-2}+2^{-7}-2^{-7}$	-2^{-7}
4	$-2^{-3}-2^{-6}$	$2^{-1}-2^{-5}+2^{-8}$	$2^{-2}+2^{-5}$	$2^{-1}+2^{-3}-2^{-6}$
5	$-2^{-1}+2^{-4}$	$2^{-1}-2^{-4}$	$2^{-1}+2^{-5}-2^{-9}$	$-2^{-3}+2^{-5}+2^{-8}$
6	$-2^{-1}-2^{-4}$	-2^{-7}	$2^{-4}+2^{-7}+2^{-9}$	$-2^{-1}-2^{-2}-2^{-8}$
7	$-2^{-1}+2^{-7}+2^{-8}$	$-2^{-1}+2^{-3}+2^{-8}$	$-2^{-1}-2^{-4}$	$2^{-3}+2^{-6}$
8	$-2^{-2}-2^{-5}-2^{-6}$	$-2^{-1}+2^{-6}+2^{-8}$	$-2^{-1}+2^{-7}$	$2^{-1}+2^{-3}-2^{-5}$
9	-2^{-6}	$-2^{-2}-2^{-8}-2^{-9}$	$2^{-2}-2^{-4}+2^{-6}$	$-2^{-3}-2^{-6}$
10	$2^{-2}-2^{-5}$	$2^{-5}-2^{-9}$	$2^{-1}+2^{-5}$	-2^{-1}
11	$2^{-2}+2^{-4}+2^{-8}$	$2^{-2}-2^{-8}-2^{-9}$	$2^{-2}-2^{-4}$	$2^{-3}+2^{-4}+2^{-8}$
12	$2^{-2}+2^{-6}+2^{-7}$	$2^{-2}+2^{-7}$	$-2^{-2}-2^{-6}-2^{-8}$	$2^{-1}-2^{-3}-2^{-5}$
13	$2^{-3}+2^{-5}$	2^{-5}	$-2^{-2}-2^{-9}$	$2^{-1}+2^{-3}-2^{-5}$
14	$2^{-6}+2^{7}$	-2^{-5}	$2^{-5}-2^{-9}$	$-2^{-3}-2^{-6}$
15	$-2^{-6}-2^{-7}$	$-2^{-2}+2^{-3}-2^{-9}$	$2^{-3}+2^{-8}$	$2^{-3}-2^{-7}$

m	$\mathbf{b}_4$	$\mathbf{b}_5$	$\mathbf{b}_6$	$\mathbf{b}_7$
2	2^{-2}	$2^{-2}-2^{-4}$	$2^{-1}-2^{-4}$	$2^{-3}+2^{-4}+2^{-8}$
3	$-2^{-1}-2^{-7}$	$2^{-3}+2^{-5}+2^{-6}$	$-2^{-2}-2^{-5}-2^{-6}$	$-2^{-2}+2^{-7}$
4	$2^{-3}+2^{-5}+2^{-6}$	$-2^{-1}-2^{-3}-2^{-5}$	$2^{-3}-2^{-6}$	$2^{-2}-2^{-6}$
5	$2^{-1}+2^{-6}$	$2^{-1}+2^{-2}-2^{-7}$	$2^{-2}-2^{-5}-2^{-6}$	$-2^{-2}-2^{-5}-2^{-7}$
6	$-2^{-1}-2^{-5}-2^{8}$	$-2^{-2}-2^{-4}-2^{-6}$	$-2^{-1}-2^{-3}$	$2^{-2}+2^{-6}+2^{-8}$
7	-2^{-5}	$-2^{-2}-2^{-7}$	$2^{-1}+2^{-4}+2^{-4}$	$-2^{-2}-2^{-4}+2^{-6}$
8	$2^{-1}+2^{-4}$	$2^{-1}+2^{-4}$	$-2^{-1}-2^{-2}+2^{-6}$	$2^{-2}+2^{-8}$
9	$-2^{-2}-2^{-4}+2^{-8}$	$-2^{-1}-2^{-4}-2^{-8}$	$2^{-1}+2^{-5}$	$-2^{-5}-2^{-6}$
10	-2^{-2}	$2^{-2}-2^{-5}+2^{-8}$	$2^{-4}+2^{-7}$	$2^{-2}+2^{-9}$
11	$2^{-1}-2^{-3}+2^{-5}$	$2^{-2}-2^{-4}-2^{-6}$	$-2^{-3}-2^{-4}+2^{-6}$	$-2^{-3}-2^{-7}-2^{-9}$
12	$-2^{-3}-2^{-7}$	$-2^{-2}-2^{-6}$	$2^{-1}-2^{-6}-2^{-7}$	$2^{-3}+2^{-6}-2^{-8}$
13	$-2^{-2}+2^{-4}-2^{-7}$	$2^{-3}+2^{-5}+2^{-6}$	$-2^{-1}+2^{-5}-2^{-7}$	$-2^{-3}+2^{-8}$
14	$2^{-4}+2^{6}$	-2^{-8}	$2^{-2}+2^{-4}+2^{-7}$	$2^{-3}-2^{-5}-2^{-7}$
15	$2^{-5}-2^{-7}$	-2^{-6}	$-2^{-2}+2^{-5}$	-2^{-5}

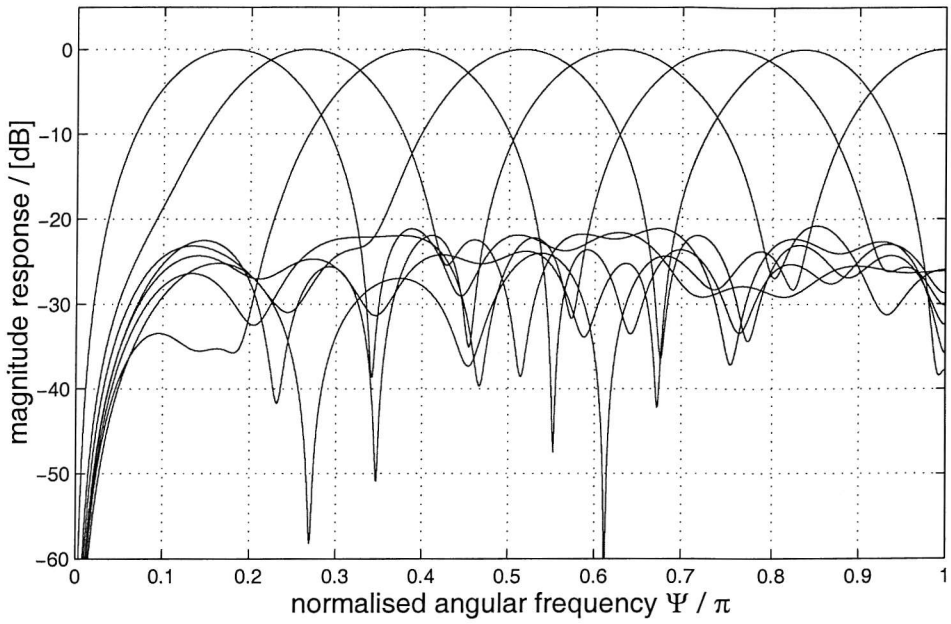


Fig. 4.8: Characteristics of the 16×8 blocking matrix columns using a GA design with SOPOT representation.

4.2.3 Design Based on Prototype Modulation

In order to reduce the design and implementation complexity of the blocking matrix, different from the overall design as discussed in Section 4.2.2, the column vectors of $\tilde{\mathbf{B}}$ can also be derived from a prototype vector by modulation. We here propose a cosine modulation in the design of $\tilde{\mathbf{B}}$, whereby the broadside constraint is guaranteed by imposing spectral zeros appropriately on the prototype vector.

Assume the prototype vector is $h[m]$, $m = 0, 1, \cdots, M-1$. Based on $h[m]$, we employ a DCT-IV modulation [64] to obtain the lth column vector $b_l[m]$, $l = 0, 1, \cdots, L-1$,

$$b_l[m] = h[m] \cdot \cos\left[\frac{\pi}{2L+2}(2l+3)(m - \frac{M-1}{2}) - (-1)^l\frac{\pi}{4}\right] . \tag{4.20}$$

In the frequency domain, this modulation creates two copies of the prototype vector's frequency response shifted along the frequency axis by $\frac{(2l+3)\pi}{2L+2}$ and $-\frac{(2l+3)\pi}{2L+2}$, respectively and adds them together. To comply with the zero-order broadside constraint $B_l(e^{j\Psi})|_{\Psi=0} = 0$, the frequency response $H(z)$ of $h[m]$ should have one spectral zero at each frequency point $\Omega_l = \pm\frac{(2l+3)\pi}{2L+2}$, $l = 0, 1, \cdots, L-1$. If we factorize $H(z)$ into two parts

$$\begin{aligned} H(z) &= P(z)Q(z) , \qquad \text{with} \\ Q(z) &= \prod_{l=0}^{L-1}(1 - e^{j\frac{2l+3}{2L+2}\pi}z^{-1})(1 - e^{-j\frac{2l+3}{2L+2}\pi}z^{-1}) , \end{aligned} \tag{4.21}$$

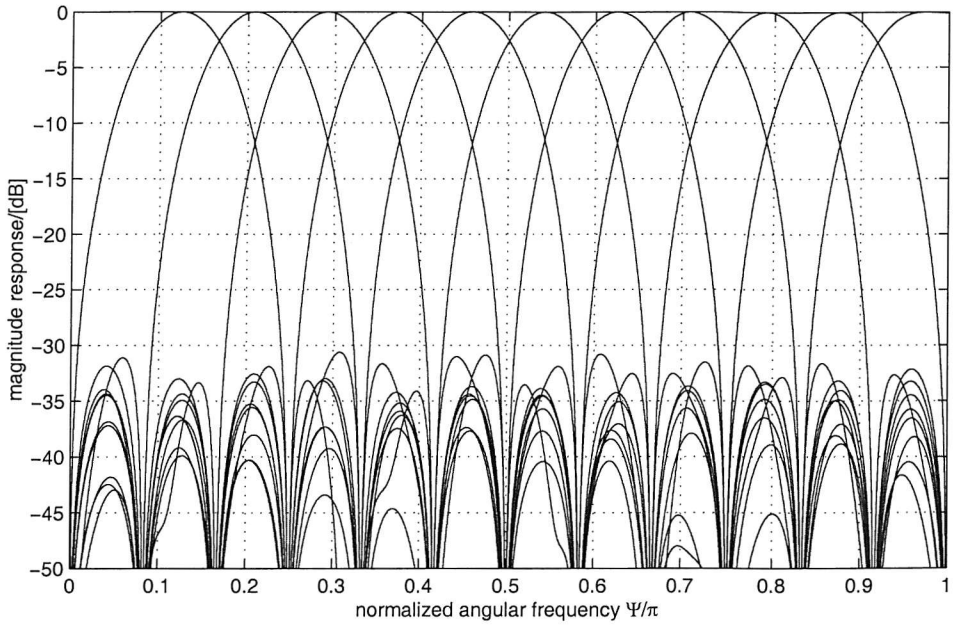


Fig. 4.9: A design example for a 28×11 blocking matrix.

then the broadside constraint will be automatically satisfied by $Q(z)$ for all the column vectors and the free parameters contained in $P(z)$ can be used to optimise its frequency response. By this factorisation, the design of the blocking matrix becomes an unconstrained optimisation problem of the prototype vector. The objective function we minimise is

$$\Phi = \int_{\Omega_s}^{\pi} \|H(e^{j\Psi})\|^2 d\Psi \,, \tag{4.22}$$

where Ω_s is the stopband cutoff frequency. To solve this unconstrained optimisation problem, we here employ the subroutines BCONF/DBCONF in the IMSL library used earlier [116] . A design example for the blocking matrix with $M = 28$ sensors, and $L = 11$ column vectors is given in Fig. 4.9.

Note that the length of the filter $Q(z)$ is $2L + 1$ and the minimum length of $P(z)$ is 1, thus we have

$$L \leq \frac{M-1}{2} \,, \tag{4.23}$$

i.e., the maximum value of the output dimension L achieved by this prototype modulation method is $\frac{M-1}{2}$, instead of the theoretical value $M - S$ as indicated in (2.61). For $S - 1$ order derivative constraints, we can replace $Q(z)$ in (4.21) by $Q(z)^S$ and the maximum value of L achieved will be $\frac{M-1}{2S}$, which will sacrifice a considerable number of DOFs for $S > 1$ and thus a satisfying performance may not be achieved for small-scale arrays. This reduction in DOFs presents a limitation of this design method.

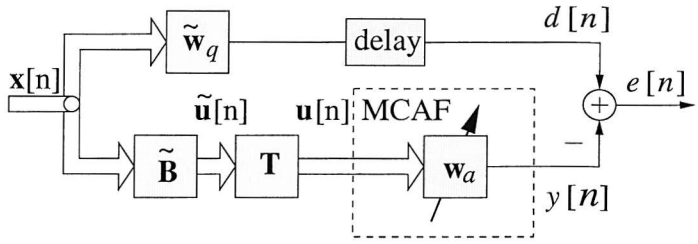


Fig. 4.10: A partially adaptive GSC with a transformation matrix.

4.3 Spatially/Temporally Subband-selective Transformation Matrix

In the partially adaptive GSC introduced in Sec. 4.1, the blocking matrix plays two different roles: the first aim is to block the signal of interest from broadside, while the other is to reduce the dimension of its output from $M-S$ to $L < M-S$ for partial adaptivity. A blocking matrix designed according to these specifications can be decomposed into the product of two matrices, an $M \times (M-S)$-dimensional blocking matrix $\tilde{\mathbf{B}}$ for a fully adaptive GSC, which is set to block the broadside signal of interest, followed by an $(M-S) \times L$-dimensional transformation matrix $\mathbf{T}$, which can reduce the output dimension according to some specific requirements. A partially adaptive GSC following this approach has been introduced in [15,17] and is shown in Fig. 4.10. In this section, we will propose a spatially/temporally subband-selective transformation matrix. When this matrix is cascaded with a standard blocking matrix, they exhibit characteristics similar to the subband-selective blocking matrix proposed in Sec. 4.2.

4.3.1 Transformation Matrix with Spatial/Temporal Subband-selectivity

As shown in Fig. 4.10, the dimension of the blocking matrix output vector $\tilde{\mathbf{u}}[n] = \tilde{\mathbf{B}}^H \mathbf{x}[n]$ is $(M-S) \times 1$. After passing through the transformation matrix $\mathbf{T} \in \mathbb{C}^{L \times (M-S)}$, the data vector's dimension is further reduced to $L \times 1$ by $\mathbf{u}[n] = \mathbf{T}\tilde{\mathbf{u}}[n]$, where $\mathbf{u}[n]$ is the final input to the following multi-channel adaptive filtering process and

$$\mathbf{T} = [\mathbf{t}_0 \ \mathbf{t}_1 \ \dots \ \mathbf{t}_{L-1}]^T , \tag{4.24}$$

$$\mathbf{t}_l = [t_l[0] \ t_l[1] \ \dots \ t_l[M-S-1]]^T \tag{4.25}$$

with $l = 0, 1, \cdots, L-1$.

Consider the same signal environment and parameters as those in Sec. 4.2.1. Using the

substitution $\Psi = \Omega \sin\theta$, the received phase vector at the sensor array is

$$\underline{\mathbf{x}} = \begin{bmatrix} 1 & e^{-j\Psi} & \cdots & e^{-j(M-1)\Psi} \end{bmatrix}^T . \tag{4.26}$$

If the blocking matrix $\tilde{\mathbf{B}}$ is constructed by the CCD method [55] as described in Sec. 2.4.3, i.e.

$$\tilde{\mathbf{B}} = \mathbf{B}_M \cdot \mathbf{B}_{M-1} \cdots \mathbf{B}_{M-S+1} , \tag{4.27}$$

where

$$\mathbf{B}_i = \begin{bmatrix} 1 & -1 & & & \mathbf{0} \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ \mathbf{0} & & & 1 & -1 \end{bmatrix}^T \in \mathbb{C}^{i \times i-1} \tag{4.28}$$

with $i = M, M-1, \cdots, M-S+1$, then the output of the blocking matrix can be expressed as

$$\tilde{\mathbf{u}}[n] = (1 - e^{-j\Psi})^S \cdot \begin{bmatrix} 1 & e^{-j\Psi} & \cdots & e^{-j(M-S-1)\Psi} \end{bmatrix}^T \cdot e^{jn\Omega} . \tag{4.29}$$

We see that the blocking matrix has a zero response to the signal from broadside as required. For a general blocking matrix $\tilde{\mathbf{B}}$, from (4.9) its output vector follows as

$$\tilde{\mathbf{u}}[n] = [B_0(e^{j\Psi}) \;\; B_1(e^{j\Psi}) \;\; \cdots \;\; B_{M-S-1}(e^{j\Psi})]^T \cdot e^{jn\Omega} . \tag{4.30}$$

Since any blocking matrix must have the desired zeros for the broadside signal of interest, $\tilde{\mathbf{u}}[n]$ can be decomposed into the product of $(1 - e^{-j\Psi})^S$ and a vector with polynomials in $e^{j\Psi}$. Such a vector with polynomials in $e^{j\Psi}$ can be further decomposed into a product of a real-valued matrix and the vector $[1 \;\; e^{-j\Psi} \;\; \cdots \;\; e^{-j(M-S-1)\Psi}]^T$. Therefore we can say that any blocking matrix for broadside constraints can be regarded as a product of the blocking matrix obtained by the CCD method and some other matrix, i.e. the CCD method provides the simplest way and the resultant blocking matrix forms the basis of any other blocking matrices for broadside constraints.

The lth output of the transformation matrix, $u_l[n]$, $l = 0, 1, \cdots, L-1$, can be denoted as

$$\begin{aligned} u_l[n] &= \mathbf{t}_l^T \cdot \tilde{\mathbf{u}}[n] \\ &= (1 - e^{-j\Psi})^S \sum_{m=0}^{M-S-1} t_l[m] e^{-jm\Psi} e^{jn\Omega} \\ &= (1 - e^{-j\Psi})^S \cdot T_l(e^{j\Psi}) \cdot e^{jn\Omega} , \end{aligned} \tag{4.31}$$

with $T_l(e^{j\Psi}) \bullet\!\!-\!\!\circ\, t_l[m]$ being a Fourier transform pair. Similar to Sec. 4.2, we arrange $T_l(e^{j\Psi})$, $l = 0, \cdots, L-1$, on the interval $\Psi \in [0; \pi]$ as shown in Fig. 4.11, such that

$$|T_l(e^{j\Psi})| = \begin{cases} 1 & \text{for } \Psi \in [\Psi_{l,\text{lower}}; \Psi_{l,\text{upper}}] ; \\ 0 & \text{otherwise} . \end{cases} \tag{4.32}$$

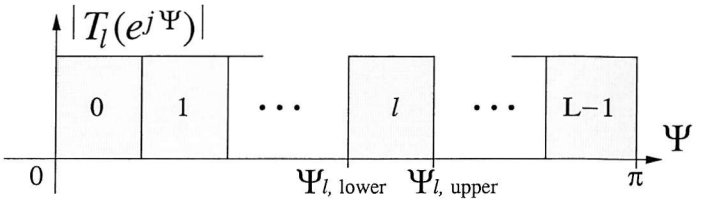


Fig. 4.11: Arrangement of the L column vectors in $\mathbf{T}$.

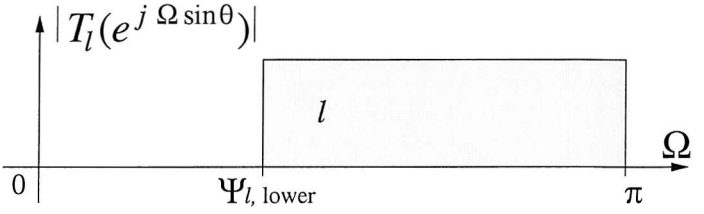


Fig. 4.12: Temporal filtering effect of the lth spatial filter in $\mathbf{T}$.

As $|\sin\theta| \in [0;1]$ when $\theta \in [-\frac{\pi}{2};\frac{\pi}{2}]$, the possible maximum frequency component of the lth output $u_l[n]$ is $\Omega = \pi$, which corresponds to $|\sin\theta| = \frac{\Psi_{l,\mathrm{upper}}}{\pi}$, while the possible minimum frequency component is $\Omega = \Psi_{l,\mathrm{lower}}$, which corresponds to $|\sin\theta| = 1$. Therefore we have the same result as in Sec. 4.2,

$$|T_l(e^{j\Omega\sin\theta})| = \begin{cases} 1 & \text{for } \Omega \in [\Psi_{l,\mathrm{lower}};\pi] \\ 0 & \text{otherwise} \end{cases} \quad \forall\,\theta \tag{4.33}$$

as shown in Fig. 4.12.

Obviously, if we consider the blocking matrix $\tilde{\mathbf{B}}$ jointly with the transformation matrix $\mathbf{T}$, a characteristic similar to that of the blocking matrix proposed in Sec. 4.2 is attained. Alternatively, the resultant matrix could simply be regarded as a new realization of the method presented in the previous section. In the following applications, we will not treat the two approaches separately and just consider the concatenation of the blocking matrix and the subband-selective transformation matrix as one kind of the general subband-selective blocking matrix proposed in Sec. 4.2.

4.3.2 Design of the Transformation Matrix

The subband-selective transformation matrix introduced in the previous section does not have to fulfil any constraints other than the band selectivity requirement. The design problem of the transformation matrix is therefore that of a series of general filter designs having cutoff frequencies specified by (4.32). However, to reduce the design and implementation complexity of the transformation matrix, again a cosine-modulated version is proposed.

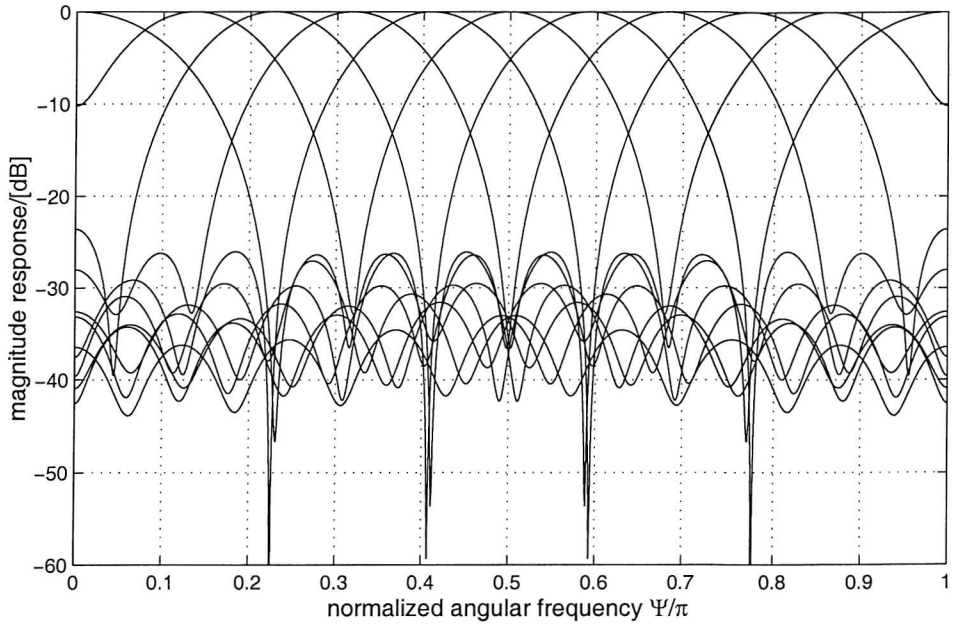


Fig. 4.13: A design example for a 11×16 transformation matrix.

Assume that the prototype vector contains the elements $h[m]$, $m = 0, 1, \cdots, M-S-1$. Then the rows of the transformation matrix $t_l[m]$, $l = 0, 1, \cdots, L-1$, can be obtained by [90]

$$t_l[m] = h[m]\cos\left[\frac{\pi}{2L}(2l+1)(m-\frac{M-S-1}{2})+(-1)^l\frac{\pi}{4}\right] . \tag{4.34}$$

Thus, the design problem of the transformation matrix is simplified to the unconstrained design of a low-pass prototype filter $h[m]$, which can be readily solved by standard filter design algorithms, such as the *remez* function in MATLAB [115]. Many of these standard filter design routines are for obtaining linear phase filters. This however is not a requirement for the transformation matrix and ties down available DOFs. Here we opt for a less restrictive method, based again on the IMSL library [116]. A result obtained by the subroutine BCONF of the IMSL library for a 11×16 transformation matrix is shown in Fig. 4.13.

4.4 Application to Subband Adaptive GSC

In this section, we will apply the subband-selective blocking matrix including the transformation method of Sec. 4.3, to the subband adaptive GSC proposed in Chap. 3 in order to reduce the computational complexity of the system.

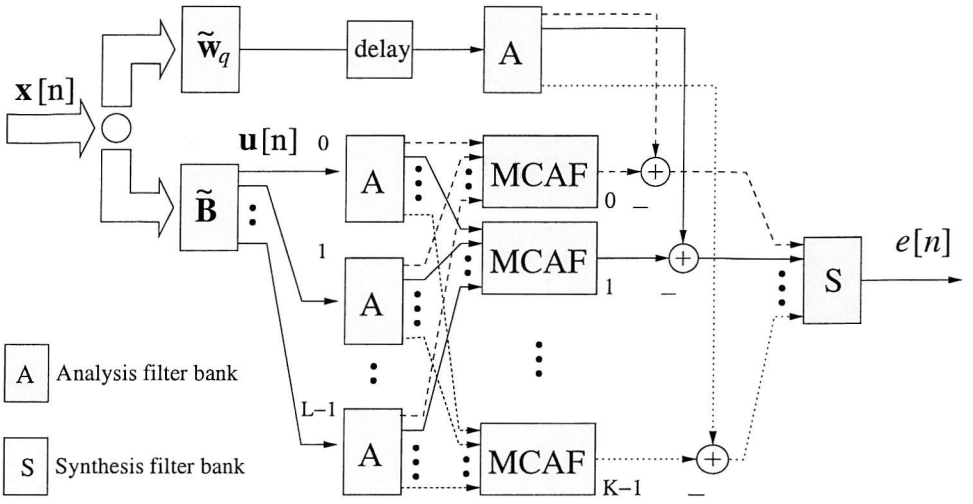


Fig. 4.14: Subband decomposition applied to the output of the subband-selective blocking matrix.

4.4.1 Structure

The previous Secs. 4.2 and 4.3 have introduced a variety of blocking matrices which permit a joint spatial/temporal filtering of the sensor signals. Noting the bandlimited spectra of blocking matrix outputs $u_l[n]$, $l = 0, 1, \cdots, L-1$, a further spectral decomposition is applied to these L outputs to remove the sampling redundancy due to their bandlimitness and perform adaptive processing in subbands, as shown in Fig. 4.14.

The subband setup proposed in Fig. 4.14 is the same as the one of Fig. 3.13 in Sec. 3.3, so it can be regarded as an application of our subband-selective blocking matrix to the subband adaptive GSC. Because $u_l[n]$, $l = 0, 1, \cdots, L-1$, is a highpass signal, the subband signals in the corresponding lowpass subbands at each MCAF will be zero and can be omitted from the subband adaptive processing. Therefore both the filter length and the number of channels are reduced, which together with the decreased update rates and the lower output dimension of $\tilde{\mathbf{B}}$ results in a substantial reduction of the system's computational complexity. Another advantage of this combination is that the subbands discarded in the adaptation can be determined a priori and are independent of the array signals because both the blocking matrix and the filter banks are selected without the knowledge of the operating environment. Additionally a further reduction of computational complexity can be achieved by monitoring the remaining subbands and dynamically discarding the processing in those subbands whose signal power falls below a given threshold [113].

4.4.2 Computational Complexity

Now we quantify the achievable computational complexity reduction introduced by our proposed spatially/temporally subband-selective blocking matrix.

Assume that we employ the oversampled GDFT filter banks in the subband decomposition. A total of $L+1$ analysis filter banks and one synthesis filter bank are employed, such that the number of multiplications needed by the filter banks will be $\frac{L+2}{N}(l_p + 4K\log_2 K + 4K)$ for real-valued input signals or $\frac{L+2}{N}(2l_p + 4K\log_2 K + 8K)$ for complex-valued input signals, according to Sec. 3.3.2.

Let us now consider the remaining part of the subband-selective GSC. Compared to the fully adaptive fullband GSC, there are three contributions towards the reduction of the computational complexity. First, the dimension of the blocking matrix output is reduced from $M-S$ to L by partial rather than full adaptivity, yielding a reduction factor of $L/(M-S)$. Secondly, for the subband adaptive filtering part, the approximate complexity reduction achieved is by a factor of $2K/N^2$ (for LMS) or $2K/N^3$ (for RLS) for real-valued input signals, and K/N^2 (for LMS) or K/N^3 (for RLS) for complex-valued signals, as seen in Sec. 3.3.2. Thirdly, by discarding the corresponding lowpass subbands, which contain negligible signal power in each of the MCAFs, we achieve a further complexity reduction. Assuming that a sufficiently selective column vectors $\mathbf{b}_l$ can be designed, the first MCAF indexed as $k=0$ would be a single-channel adaptive filter, drawing its low frequency input solely from the first branch of $\tilde{\mathbf{B}}$. The second ($k=1$) MCAF block in Fig. 4.14 will only cover some of the lower outputs of $\tilde{\mathbf{B}}$, while finally only the last MCAF $(K-1)$ consists of L non-sparse channels. Thus, a channel reduction in the MCAFs is achieved, yielding a considerably reduced complexity. This characteristic underlines the advantage of a combined spatial/temporal subband selection by subband processing in both the spatial and temporal domain. Under ideal conditions, the dimension of the MCAFs can almost be halved, with a proportional decrease in computational complexity.

By these three steps, the complexity reduction ratio $R_{\text{complexity}}$ between the computational complexity C_{select} of the new subband-selective GSC, excluding the computations required by the filter banks, and the complexity of the standard fully adaptive fullband GSC C_{standard} becomes

$$R_{\text{complexity}} = \frac{C_{\text{select}}}{C_{\text{standard}}} = \begin{cases} \frac{LK}{(M-S)N^2} & \text{for LMS ;} \\ \frac{LK}{(M-S)N^3} & \text{for RLS} \end{cases} \tag{4.35}$$

for real-valued input signals and

$$R_{\text{complexity}} = \frac{C_{\text{select}}}{C_{\text{standard}}} = \begin{cases} \frac{LK}{2(M-S)N^2} & \text{for LMS ;} \\ \frac{LK}{2(M-S)N^3} & \text{for RLS} \end{cases} \tag{4.36}$$

for complex-valued input signals.

Tab. 4.2: Computational complexity for the subband adaptive GSC employing subband-selective blocking matrix:

algorithms	real multiplications (real-valued sensor signals)
LMS	$(ML+M)+\frac{L+2}{N}(l_p+4K\log_2 K+4K)+\frac{K}{N}(2\frac{LJ}{N}+1)$
NLMS	$(ML+M)+\frac{L+2}{N}(l_p+4K\log_2 K+4K)+\frac{K}{2N}(4\frac{LJ}{N}+L+3)$
RLS	$(ML+M)+\frac{L+2}{N}(l_p+4K\log_2 K+4K)+(2K+3+\frac{1}{K})\frac{L^2J^2}{N^3}+\frac{3KLJ}{N^2}+\frac{2K}{N}$
algorithms	**real multiplications (complex-valued sensor signals)**
LMS	$2(ML+M)+\frac{L+2}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(4\frac{LJ}{N}+2)$
NLMS	$2(ML+M)+\frac{L+2}{N}(2l_p+4K\log_2 K+8K)+\frac{K}{N}(4\frac{LJ}{N}+2L+3)$
RLS	$2(ML+M)+\frac{L+2}{N}(2l_p+4K\log_2 K+8K)+(2K+3+\frac{1}{K})\frac{2L^2J^2}{N^3}+\frac{6KLJ}{N^2}+\frac{4K}{N}$

As an example, consider a broadband beamformer with $M = 16$ sensors and zero-order constraints. If we employ a 16×8-dimensional ($L = 8$) blocking matrix and a $K = 8$ channel GDFT filter bank with decimation ratio $N = 6$ in our subband-selective GSC using an LMS, the reduction ratio $R_{\text{complexity}}$ is approximately $1/8$ for real-valued input signals, under the ideal assumption of discarding half of the channels of the subband MCAFs! The corresponding complexity reduction achieved in case of the RLS algorithm is approximately $1/50$.

Referring to Tab. 3.1, the detailed computational complexity figures derived under the ideal assumption are shown in Tab. 4.2 for our subband adaptive GSC employing the subband-selective blocking matrix.

4.4.3 Simulations and Results

The proposed structures have been extensively simulated and tested. In this part, we provide three sets of representative simulations based on the full design, the cosine-modulated design of the blocking matrix and the transformation method, respectively. These will be compared to traditional fullband fully adaptive beamformers, whereby two popular designs of the blocking matrix, the CCD [55] and SVD methods [30], are implemented. The most important parameters for these three simulations are listed in Tab. 4.3.

4.4.3.1 Simulation I (full design of $\tilde{\mathbf{B}}$)

In the first simulation, the blocking matrix of the proposed subband-selective GSC is designed using the method of Sec. 4.2.2. The dimension of $\tilde{\mathbf{B}}$ is 16×8 ($L = 8$), and the magnitude

Tab. 4.3: Part of the parameters in simulation I, II, and III:

parameters	simulation I	simulation II	simulation III
sensor number M	16	21	17
TDL length J	70	90	90
derivative constraint S	2	1	1
signal DOA θ	0°	0°	0°
interference sources	1	2	2
interference DOA θ	20°	20° & −40°	30° & −60°
SIR	−24 dB	−30 dB	−24 dB
SNR	20 dB	20 dB	20 dB
interference bandwidth	$\Omega = [0.25\pi; 0.75\pi]$	$\Omega = [0.3\pi; 0.7\pi]$	$\Omega = [0.3\pi; 0.8\pi]$
adaptation algorithm	NLMS	NLMS	NLMS
stepsize	0.40	0.30	0.35

responses of its column vectors are shown in Fig. 4.7. Each of the blocking matrix outputs and the reference signal $d[n]$ are divided into $K = 12$ subbands by oversampled GDFT filter banks [2, 94] with decimation ratio $N = 10$. The length of the prototype filter is 240 and the frequency responses of the analysis filter bank are shown in Fig. 4.15. The length of the adaptive filter operating in each channel of the subband MCAFs is $J/N = 7$. In each MCAF block, the channels with negligible output are discarded according to (4.13), whereby the processed inputs to the MCAFs are shown in Fig. 4.16. Fig. 4.17 displays the residual mean square error of the system, employing the adaptation by a normalised LMS algorithm with a step size of 0.35. Clearly, the proposed subband system converges systematically faster and has a lower steady-state mean square residual error. The gain response of the adapted subband system with respect to frequency Ω and DOA θ is given in Fig. 4.18. The attenuation at the interferer's position of $\theta = 20°$ is clearly visible, and can be inspected quantitatively in Fig. 4.19, where the beamformer's response is only shown over the frequency band $\Omega = [0.25\pi; 0.75\pi]$, in which the interferer is active.

To show the effect of channel discarding according to Fig. 4.16 , we also give a comparison of the learning curves before and after the channel discarding, which is shown in Fig. 4.20. The steady state ensemble mean square residual error with and without discarding is −1.42dB and −1.44dB, respectively, and hence virtually identical.

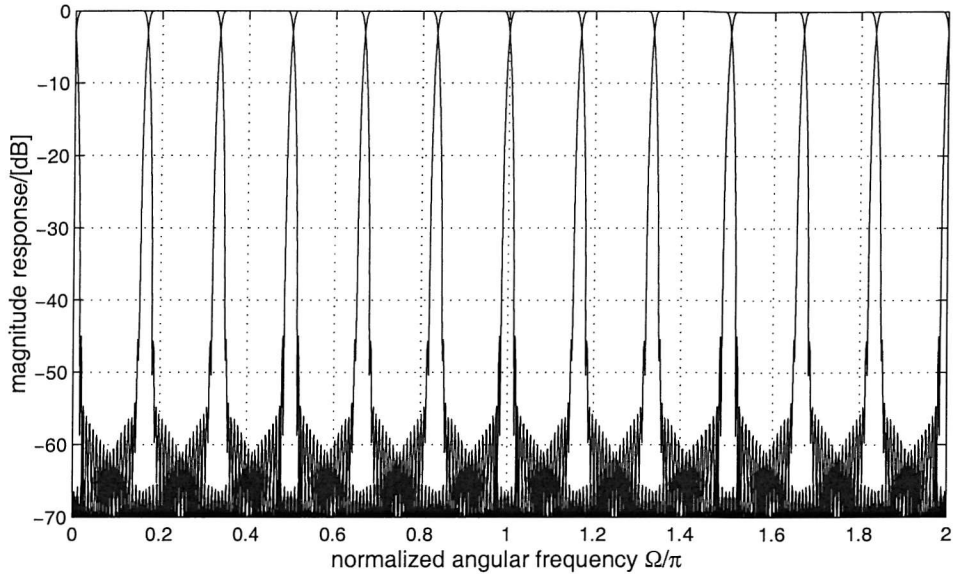


Fig. 4.15: Frequency responses of a $K = 12$ channel filter bank decimated by $N = 10$.

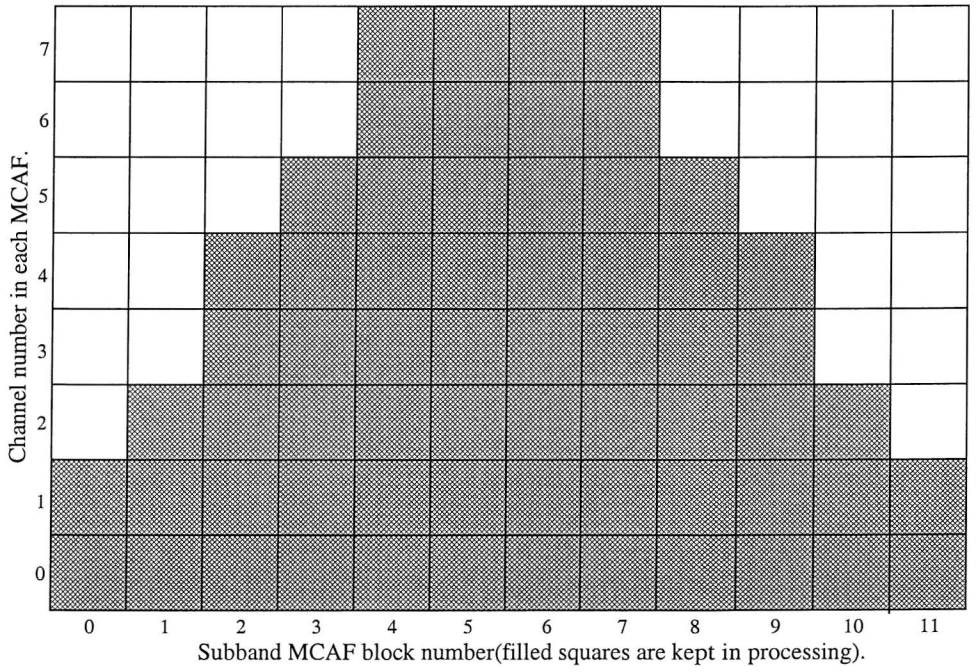


Fig. 4.16: Channels discarded in each MCAF block for simulation I.

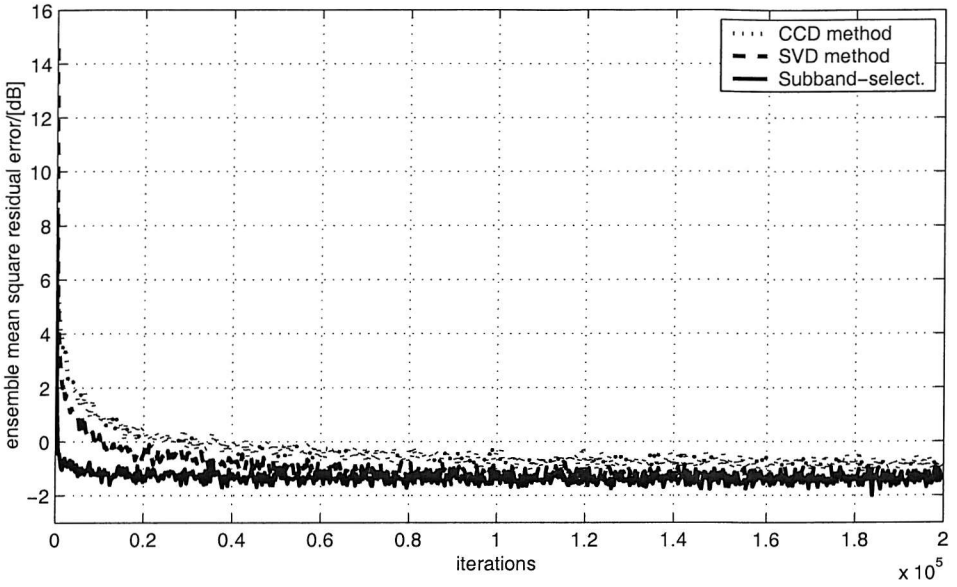


Fig. 4.17: Learning curves for simulation I (step size=0.30).

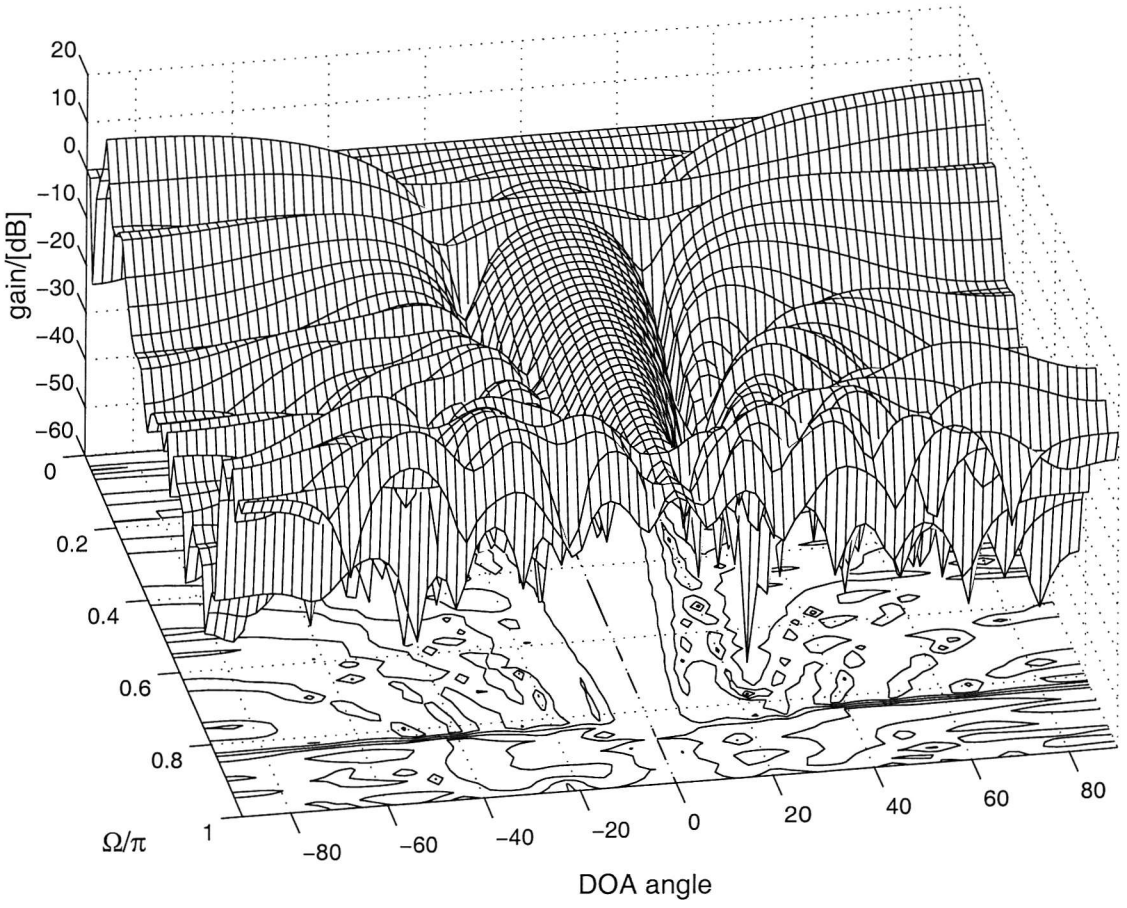


Fig. 4.18: 3-D beampattern for the subband-selective GSC in simulation I.

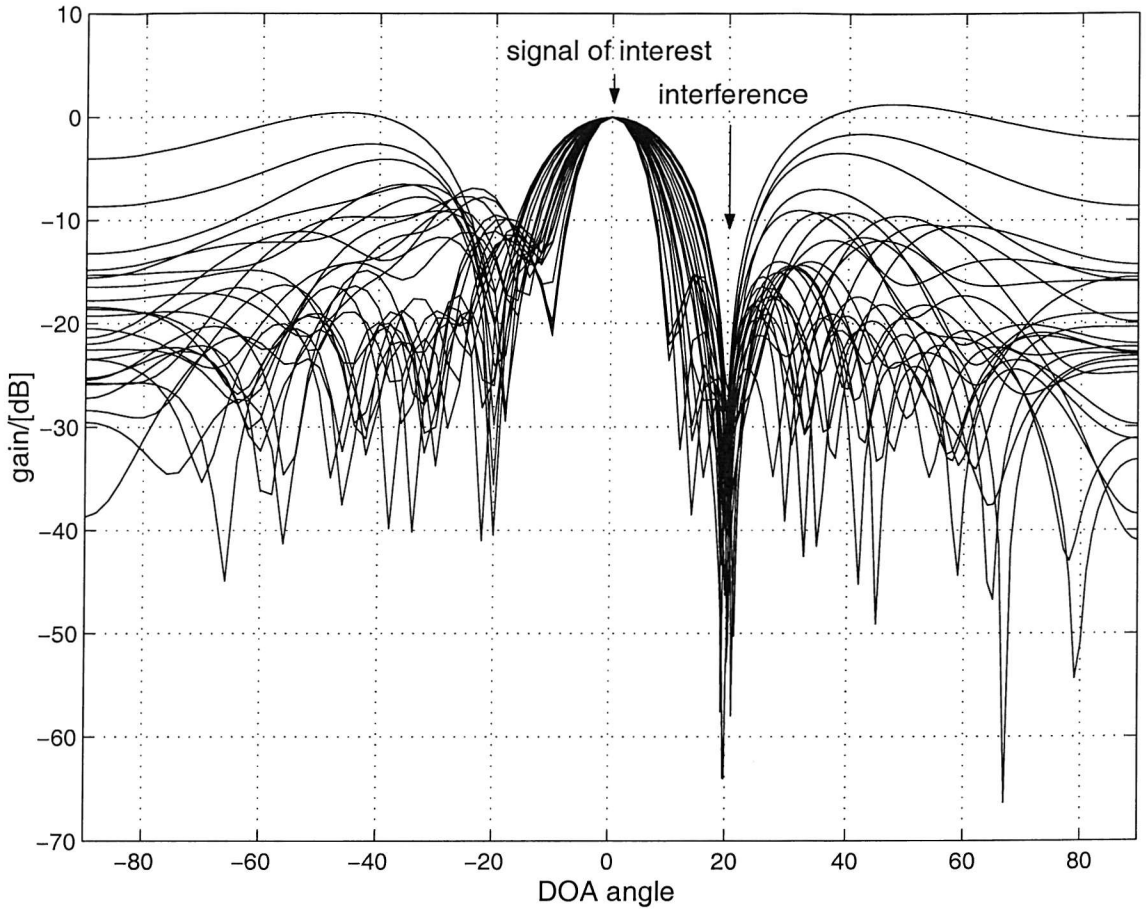


Fig. 4.19: 2-D beampattern for the subband-selective GSC in simulation I over the band $\Omega = [0.25\pi; 0.75\pi]$.

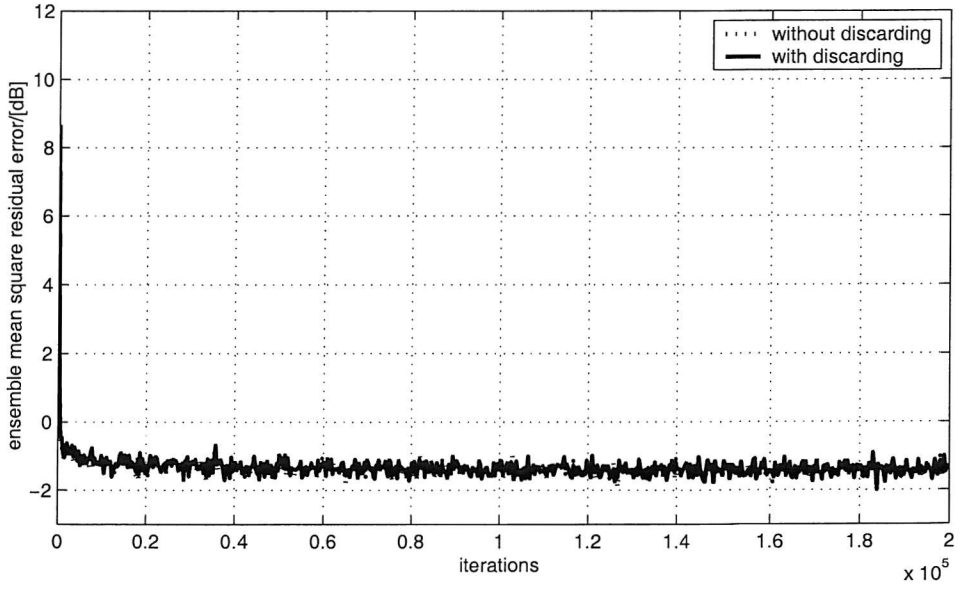


Fig. 4.20: Comparison of learning curves before and after channel discarding.

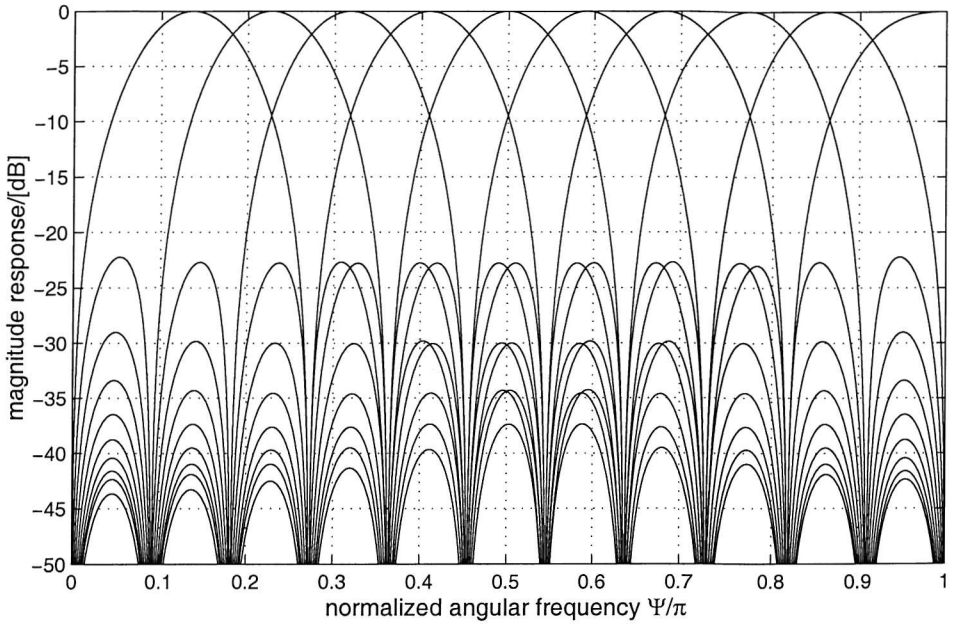


Fig. 4.21: Frequency responses of the 21×10 blocking matrix.

4.4.3.2 Simulation II (Cosine-modulated design of $\tilde{\mathbf{B}}$)

In a second simulation, the blocking matrix of the proposed subband-selective GSC is obtained by cosine modulation. Its dimension is 21×10 with the frequency responses of the column vectors shown in Fig. 4.21. Note how the broadside constraint is enforced by regularly spaced spectral zeros. We employ the same filter bank for subband decomposition as in Sec. 4.4.3.1. Due to the characteristics of the blocking matrix $\tilde{\mathbf{B}}$ and the filter bank, a number of channels in the various subband MCAFs can be discarded according to Fig. 4.22. Learning curves for an NLMS algorithm using a step size of $\mu_0 = 0.30$ for adaptation are shown in Fig. 4.23. A comparison of the learning curves with and without the channel discarding is given in Fig. 4.24 with the same steady state ensemble mean square residual error of -3.21dB in both cases . The gain response of the adapted subband GSC is displayed in Fig. 4.25 , and plotted over the frequency range $\Omega = [0.30\pi; 0.70\pi]$ of the interferers in Fig. 4.26. As evident, nulls have been correctly placed in the directions of these interfering sources.

4.4.3.3 Simulation III (By Transformation Matrix T)

In this third simulation, we employ the subband-selective transformation matrix introduced in Sec. 4.3 and compare the performance of this method with that of a fully adaptive fullband GSC using the CCD method. The transformation matrix of dimension 11×16 is obtained by cosine modulation and contains the frequency responses shown in Fig. 4.13. The subband setup is the same as that used for Simulation II, resulting in the pattern for discarding channels in

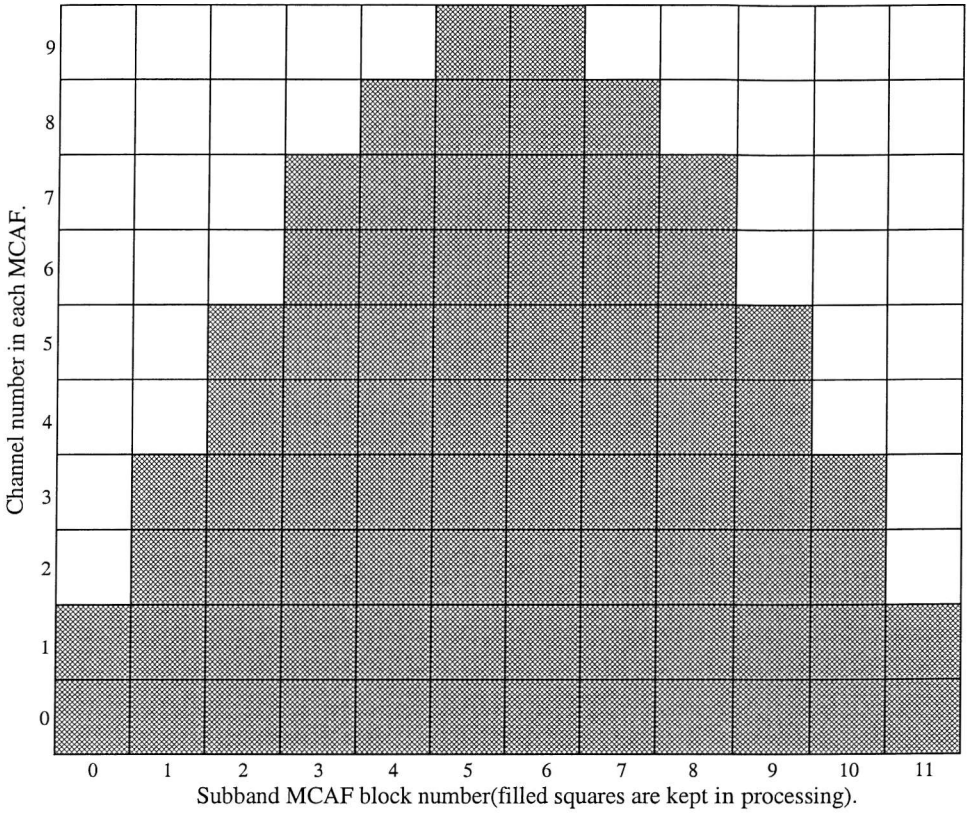


Fig. 4.22: Channels discarded in each MCAF block for simulation II.

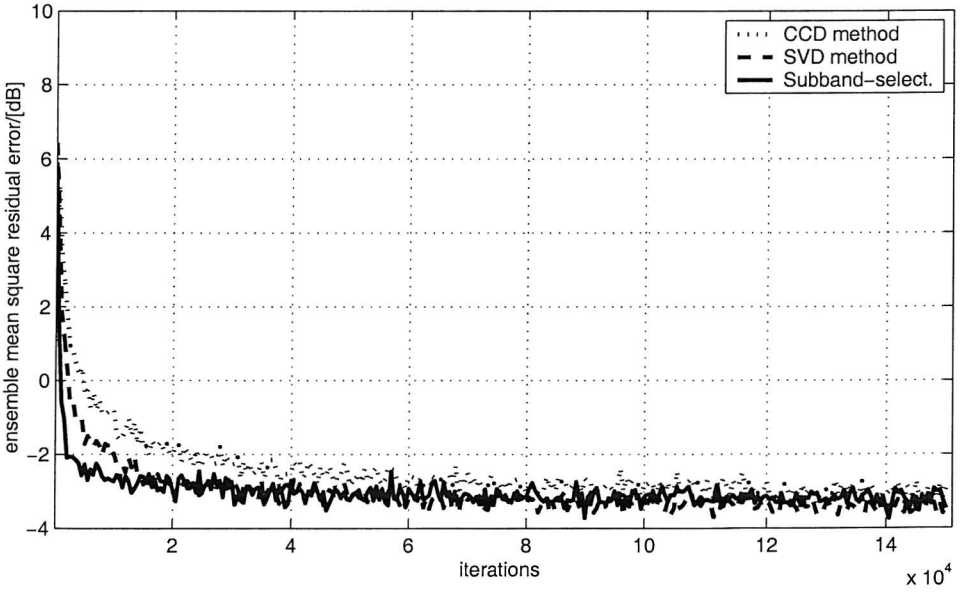


Fig. 4.23: Learning curves for simulation II (step size=0.3).

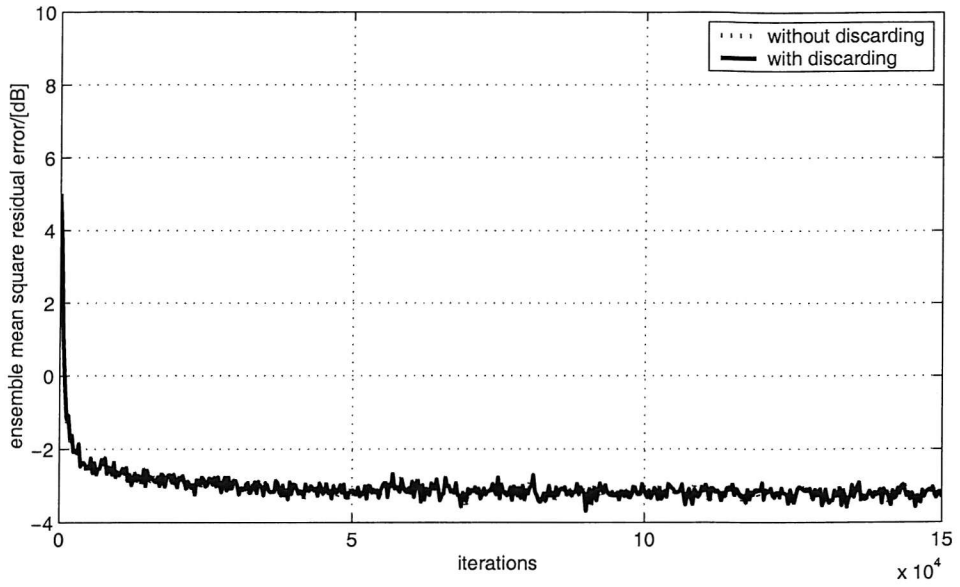


Fig. 4.24: Comparison of learning curves before and after channel discarding.

each MCAF block as given in Fig. 4.27. Learning curves of the beamformers using the NLMS adaptive algorithm with a normalised step size of 0.35 are shown in Fig. 4.28. The learning curves of the subband-selective GSC both with and without channel discarding are presented in Fig. 4.29, demonstrating that there is no penalty incurred in neglecting the low power channels of the subband MCAFs. Fig. 4.30 gives the three-dimensional beampattern for the subband-selective GSC, while Fig. 4.31 displays the response over the interferers' frequency band $[0.30\pi; 0.80\pi]$, indicating that the beamformer has placed nulls in the desired directions.

4.4.3.4 Discussions

The beampatterns for the three sets of simulations clearly show that the subband-selective GSC can suppress the interference effectively by forming corresponding nulls in the directions of interference. From the learning curves, we see that under different step sizes, different orders of constraints, and different signal environments, the proposed subband-selective method always achieves a higher convergence speed than the standard fully adaptive fullband GSCs due to the combined decorrelation effect in both spatial and spectral domains by both the blocking matrix and the analysis filter banks, and the shortened adaptive filter length within each channel. Because of the band-selectivity of the blocking matrix, there is almost no difference between the learning curves with and without channel discarding. As to the computational complexity, when the blocking matrix of the fullband GSC is formed by the CCD method, the proposed novel subband-selective GSC only needs about 31% multiplications of the fullband GSC in simulation I, 23% of that in simulation II, and 32% of that in simulation III according to their channel-discarding patterns and adaptation in subbands; for the SVD method, the reductions

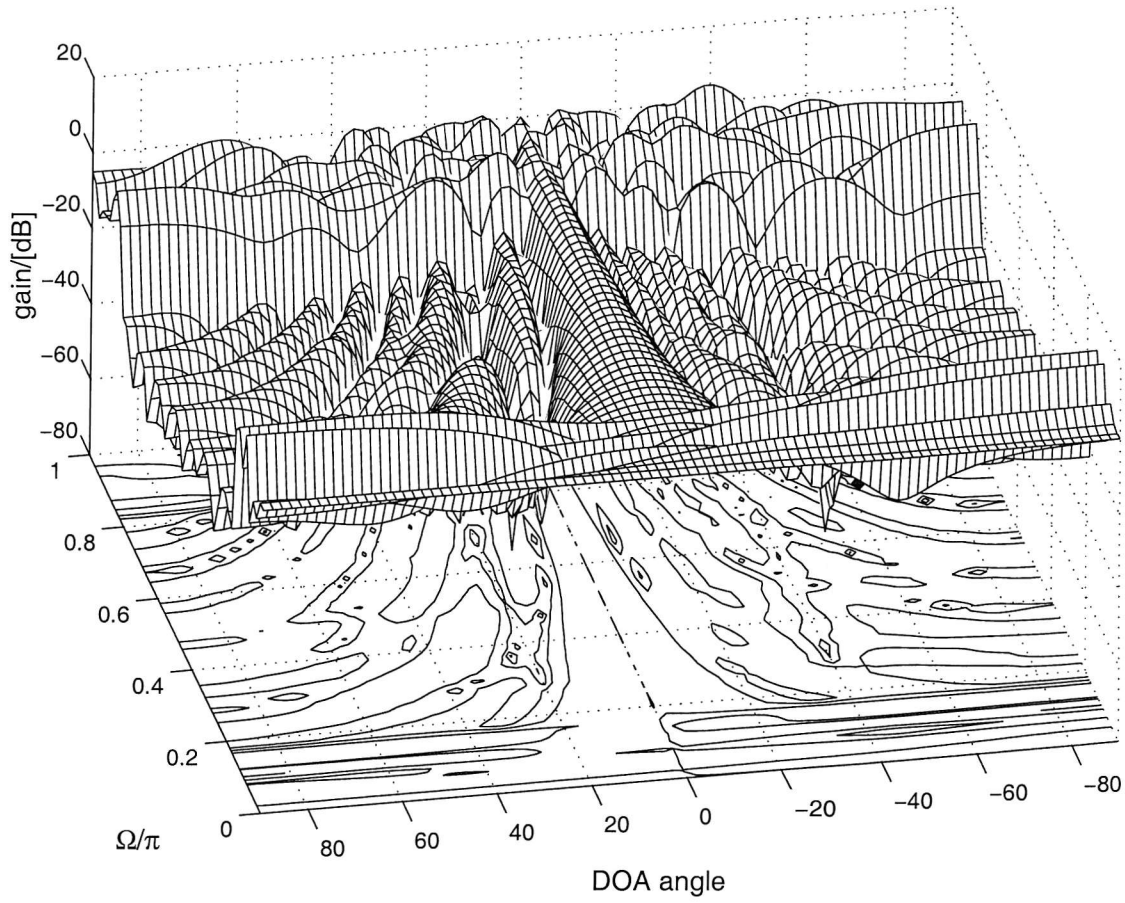


Fig. 4.25: 3-D beampattern for the subband-selective GSC in simulation II.

are 29% in simulation I and 22% in simulation II.

4.5 Application to Transform-domain GSC

In Sec. 2.4 we introduced the GSC structure proposed in [29], which can be regarded as a time-domain GSC. As LMS-type algorithms suffer from a low convergence rate when the condition number of the input correlation matrix increases [51], a transform-domain GSC (TGSC) was proposed by Chen and Fang [21]. In their work, a one-dimensional discrete Fourier transform is used on each of the tap-delay lines at the output of the blocking matrix. Thereafter, an LMS algorithm with self-orthogonalising property is applied. Following the work of [21], a two-dimensional transformation is introduced to the GSC [22], which can improve the convergence rate further due to the approximate estimation of both the spatial and temporal correlation. With the advantage of higher convergence speed, the transform-domain GSC however poses the problem of high computational complexity.

In this section, a new realisation of the TGSC equipped with our subband-selective blocking

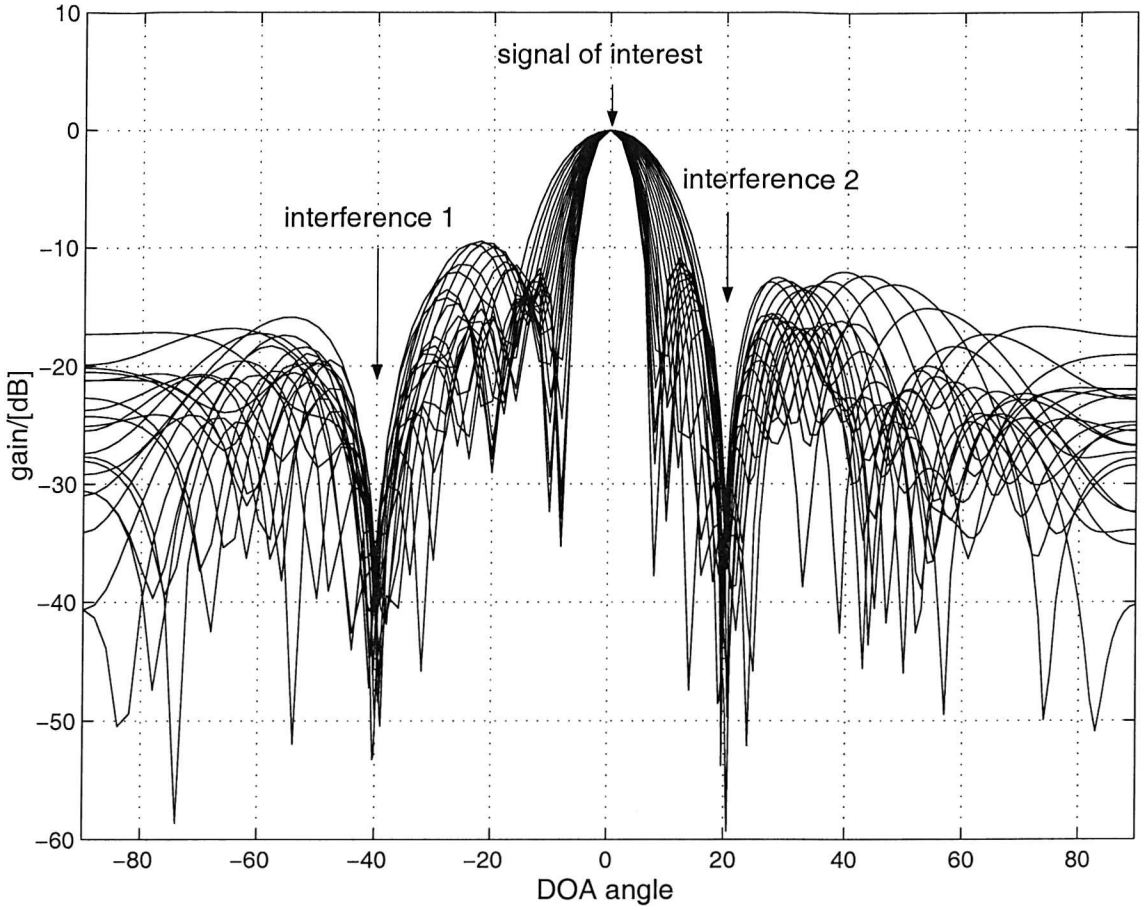


Fig. 4.26: Response of the subband-selective GSC in simulation II over the band $[0.30\pi; 0.70\pi]$.

matrix will be proposed to reduce its computational complexity. In this subband-selective TGSC (STGSC), when applying the DFT to the outputs of the blocking matrix with band selectivity, some of the frequency-bin outputs of the DFT will be approximately zero and can be omitted from the following adaptive processing. Because of the finite-duration effect of the DFT [124], it is advantageous to apply a window function [124] with narrow bandwidth to the blocking matrix outputs before performing the DFT, which is not part of the TGSC as originally proposed.

4.5.1 Transform-domain GSC

The structure of a transform-domain GSC is shown in Fig. 4.32, where the blocking matrix output $\mathbf{u}[n] = [u_0[n]\ u_1[n] \ldots u_{L-1}[n]]^T$ is obtained by $\mathbf{u}[n] = \tilde{\mathbf{B}}^H \mathbf{x}[n]$ and $L = M - S$ for a fully adaptive GSC. A J-point DFT is applied to each of the tapped-delay line vectors $\mathbf{u}_l[n]$,

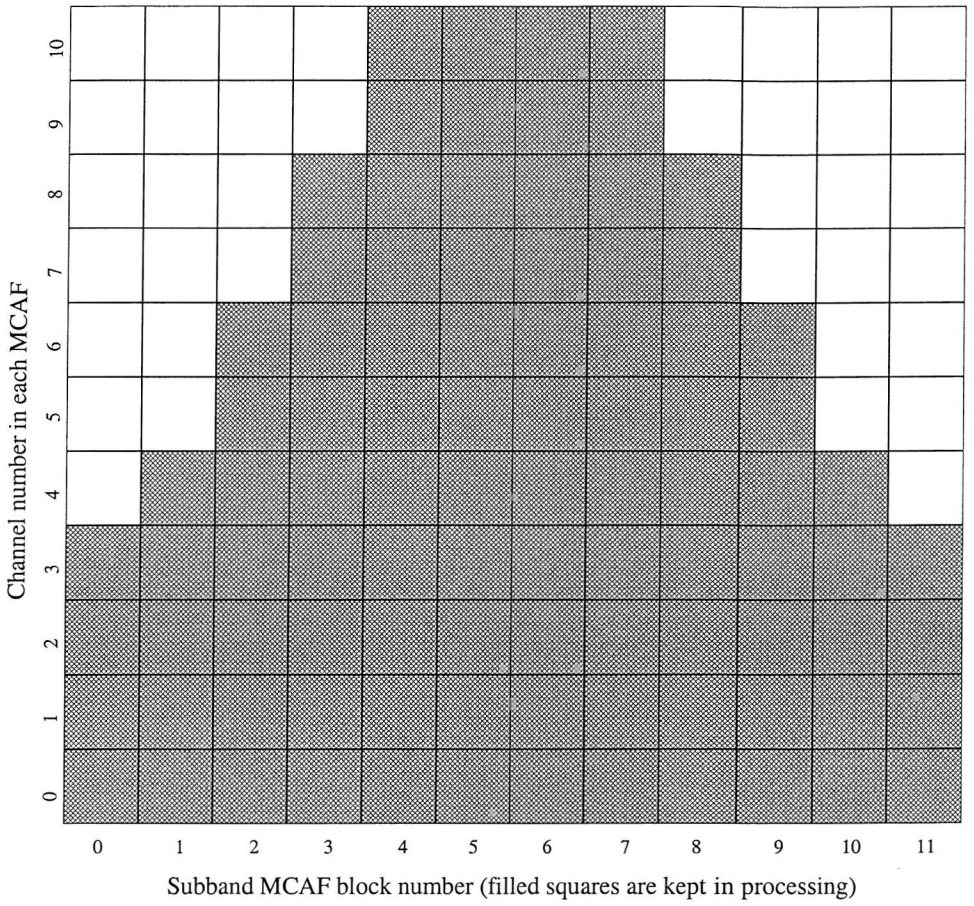


Fig. 4.27: Channels discarded in each MCAF block for simulation III.

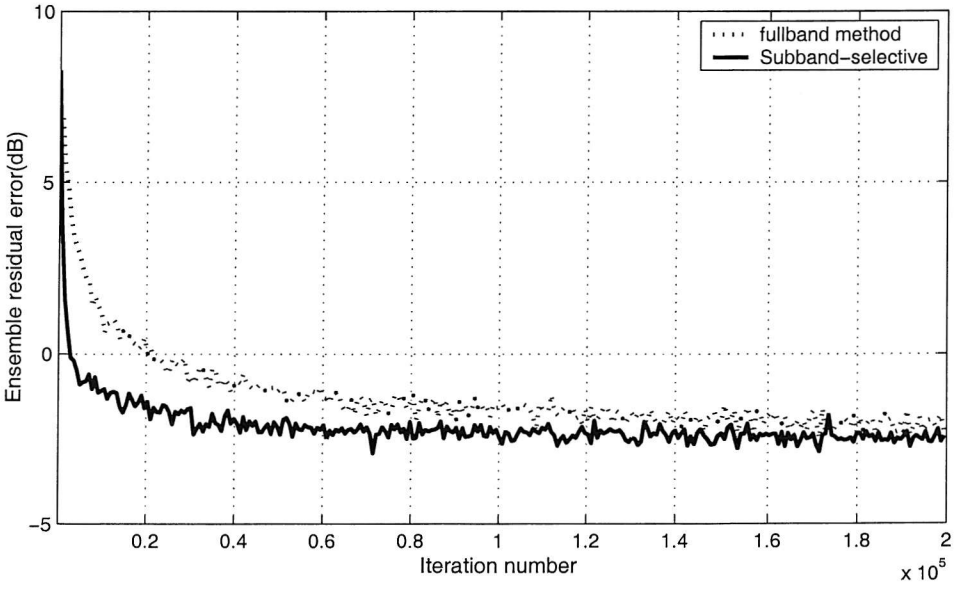


Fig. 4.28: Learning curves for simulation III (step size=0.35).

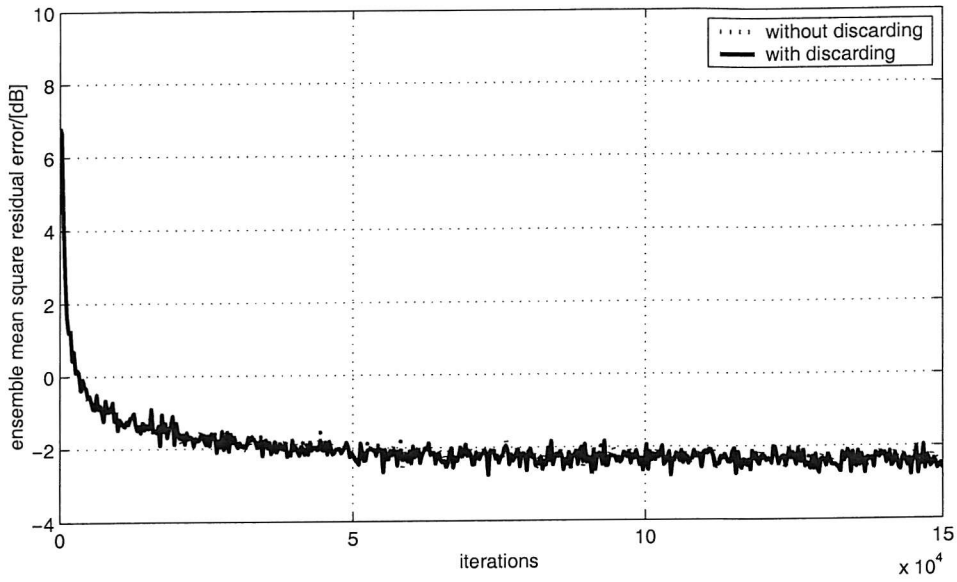


Fig. 4.29: Comparison of learning curves before and after channel discarding.

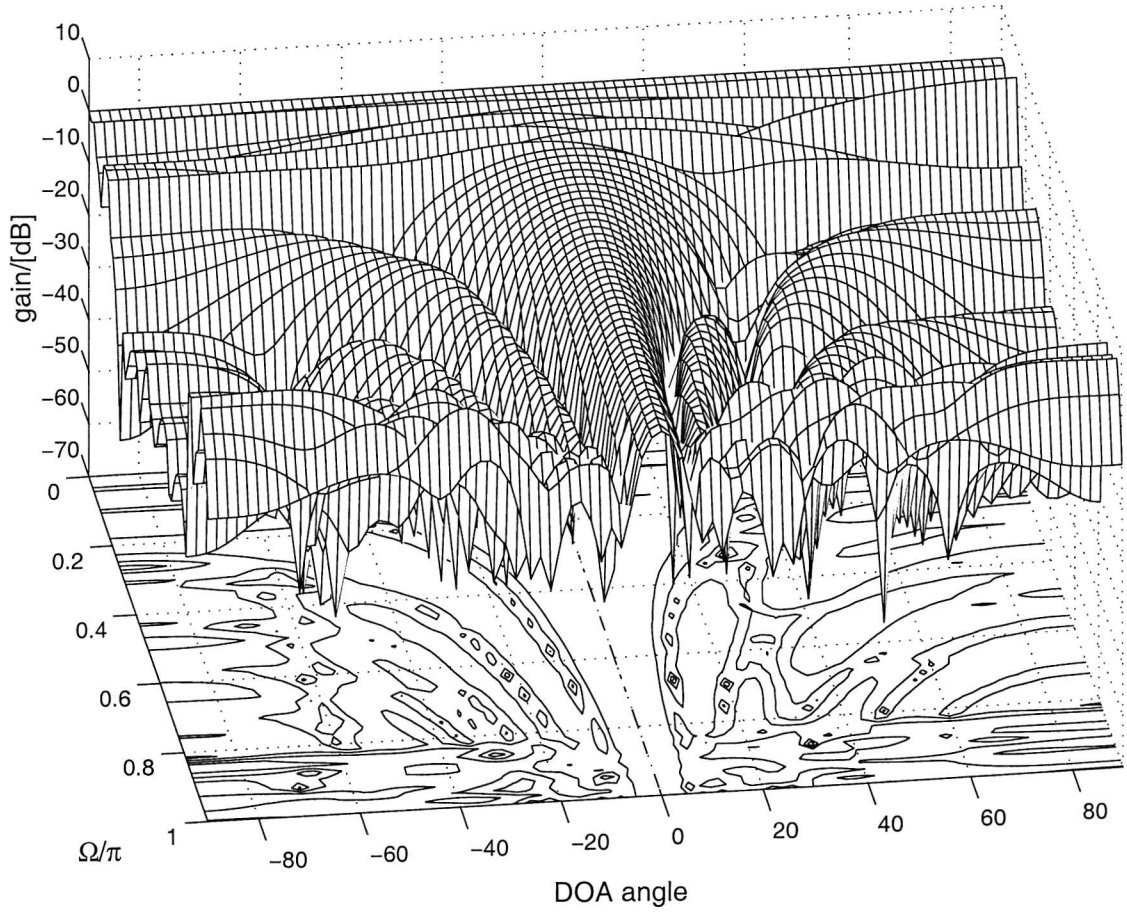


Fig. 4.30: 3-D beampattern for the subband-selective GSC in simulation III.

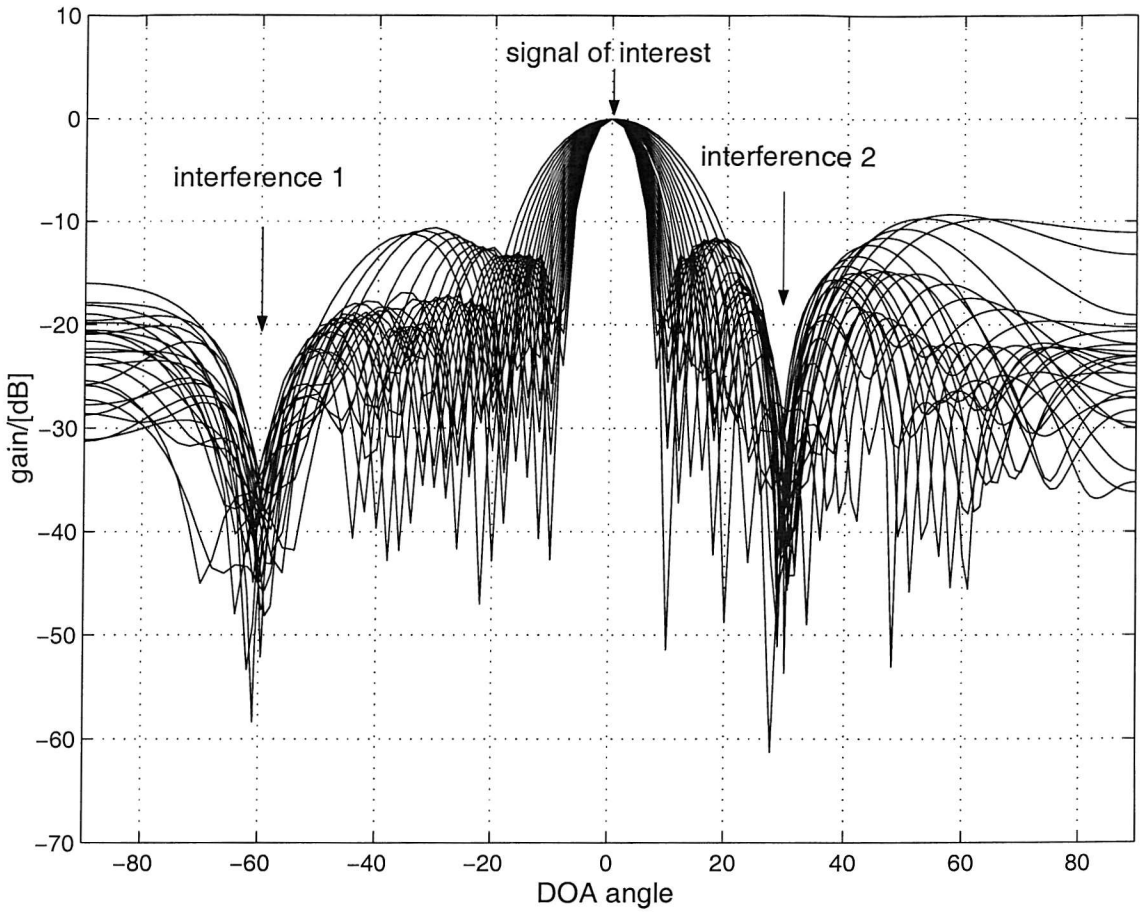


Fig. 4.31: Response of the subband-selective GSC in simulation III over the band $\Omega = [0.30\pi; 0.80\pi]$.

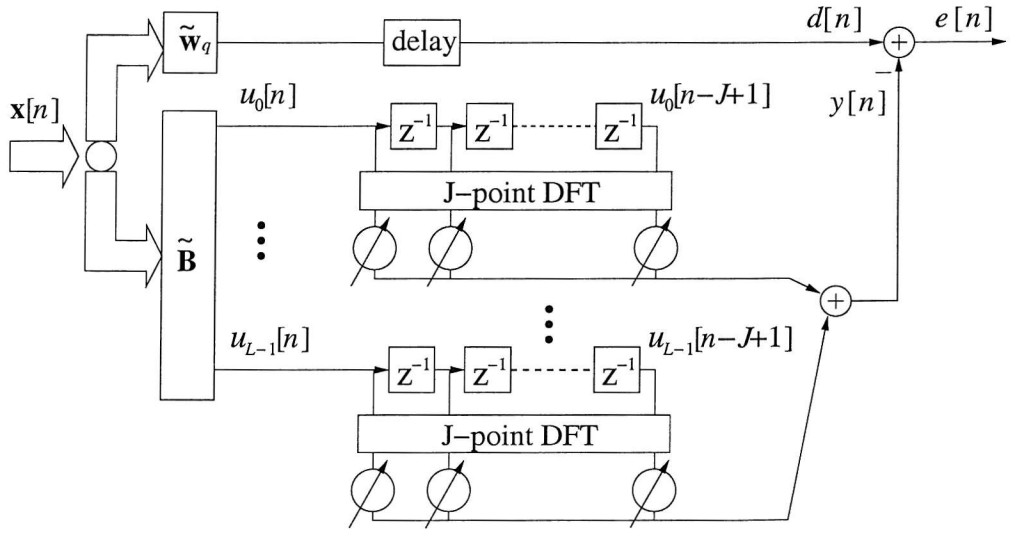


Fig. 4.32: A frequency-domain GSC structure.

$l = 0, 1, \cdots, L-1$, where

$$\mathbf{u}_l[n] = [u_l[n]\ u_l[n-1]\ \ldots\ u_l[n-J+1]]^T\ . \tag{4.37}$$

The output of the lth DFT block is

$$\mathbf{v}_l[n] = \mathrm{DFT}\{\mathbf{u}_l[n]\}\ , \tag{4.38}$$

where $\mathbf{v}_l[n] = [v_{l,0}[n]\ v_{l,1}[n]\ \ldots\ v_{l,J-1}[n]]^T$.

Stacking the DFT outputs as

$$\mathbf{v}[n] = \left[\mathbf{v}_0[n]^T\ \ \mathbf{v}_1[n]^T\ \ \ldots\ \ \mathbf{v}_{L-1}[n]^T\right]^T\ , \tag{4.39}$$

we can formulate $y[n] = \tilde{\mathbf{w}}^H\mathbf{v}[n]$, where $\tilde{\mathbf{w}}$ is the weight vector including all the corresponding weights in the transform-domain LMS algorithm. These weights $\tilde{\mathbf{w}}$ are updated continually to minimise the power of the error signal $e[n] = d[n] - y[n]$ by a self-orthogonalising LMS algorithm [125],

$$\tilde{\mathbf{w}}[n+1] = \tilde{\mathbf{w}}[n] + 2\gamma e^*[n]\mathbf{R}_{vv}^{-1}\mathbf{v}[n]\ , \tag{4.40}$$

where

$$\mathbf{R}_{vv} = \mathcal{E}\{\mathbf{v}[n]\mathbf{v}^H[n]\} \tag{4.41}$$

and $0 < \gamma < \frac{1}{(LJ)}$ to ensure convergence of the algorithm. The role of $\mathbf{R}_{vv}^{-1}$ is to reduce the eigenvalue spread of the matrix governing the adaptation process.

Note that $\mathbf{R}_{vv}$ is unknown in practice and we here use the following approach to approximate it by a diagonal matrix $\tilde{\mathbf{R}}_{vv}$ [22, 126],

$$\tilde{\mathbf{R}}_{vv} = \mathrm{diag}\left[r_{0,0}, \cdots, r_{0,J-1}, \cdots, r_{L-1,0}, \cdots, r_{L-1,J-1}\right]\ , \tag{4.42}$$

where

$$r_{l,j} = \mathcal{E}\{|v_{l,j}[n]|^2\}\ , \quad l = 0, 1, \cdots, L-1, \quad j = 0, 1, \cdots, J-1\ , \tag{4.43}$$

is the power of the corresponding frequency bin output of the DFT. The diagonal elements $r_{l,j}$ in turn can be recursively estimated at time instance n through the following equation

$$\tilde{r}_{l,j}[n] = \beta\tilde{r}_{l,j}[n-1] + (1-\beta)|v_{l,j}[n]|^2\ , \tag{4.44}$$

where $0 \leqslant \beta \leqslant 1$ is a forgetting factor. Then the estimate $\tilde{\mathbf{R}}_{vv}^{-1}$ of $\mathbf{R}_{vv}^{-1}$ is given by

$$\tilde{\mathbf{R}}_{vv}^{-1} = \mathrm{diag}\left[\tilde{r}_{0,0}^{-1}, \cdots, \tilde{r}_{0,J-1}^{-1}, \cdots, \tilde{r}_{L-1,0}^{-1}, \cdots, \tilde{r}_{L-1,J-1}^{-1}\right]\ , \tag{4.45}$$

Tab. 4.4: Computational complexities for our STGSC and the old TGSC:

GSC realisations	complex multiplications per cycle (LMS)
TGSC	$(M-S)J\log_2 J+3.5(M-S)J$
STGSC	$LJ\log_2 J+1.75LJ$

and we get the new update equation

$$\tilde{\mathbf{w}}[n+1] = \tilde{\mathbf{w}}[n] + 2\gamma e^*[n]\tilde{\mathbf{R}}_{vv}^{-1}\mathbf{v}[n] \ . \tag{4.46}$$

Although the TGSC accelerates the convergence speed, it also increases the computational complexity of the system. In the next section, we sacrifice some DOFs of the system by introducing the prevously proposed subband-selective blocking matrix in order to achieve a lower computational complexity.

4.5.2 Subband-selective TGSC

The introduction of the subband-selective TGSC is straightforward. The standard blocking matrix in Fig. 4.32 is replaced by our subband-selective blocking matrix. As noted before, its outputs $u_l[n]$, $l = 0, \cdots, L-1$ contain signals with tighter and tighter highpass spectra, as the index l increases. If we apply a DFT to the output signal $\mathbf{u}_l[n]$, some of the frequency bins will possess negligible energy and can be omitted from the following adaptive process. In order to best exploit this property, we need to select a suitable window function with good frequency selectivity, which will be multiplied with the time-domain signals prior to applying the DFT.

Now we analyse the computational complexity of the system. Since for a fully adaptive GSC the output dimension of the blocking matrix is $L = M - S$, the total number of weights in a partially adaptive system is reduced by $L/(M-S)$. Concerning the DFT and adaptive part under ideal conditions, i.e. if sufficiently selective column vectors $\mathbf{b}_l$ and a good window function can be designed, the last DFT output $\mathbf{v}_{L-1}$ will have only approximately two non-zero frequency bins for real-valued signals (or one for complex-valued signals), and $\mathbf{v}_{L-2}$ has four (or two for complex-valued signals), and so on. Finally, only $\mathbf{v}_0$ does not have any negligible frequency bins. Thus, under ideal conditions, the total number of weights to be updated will be further halved. Considering the overall subband-selective TGSC, its computational complexity is summarised in Table 4.4, which also provides a comparison with the TGSC proposed in [21].

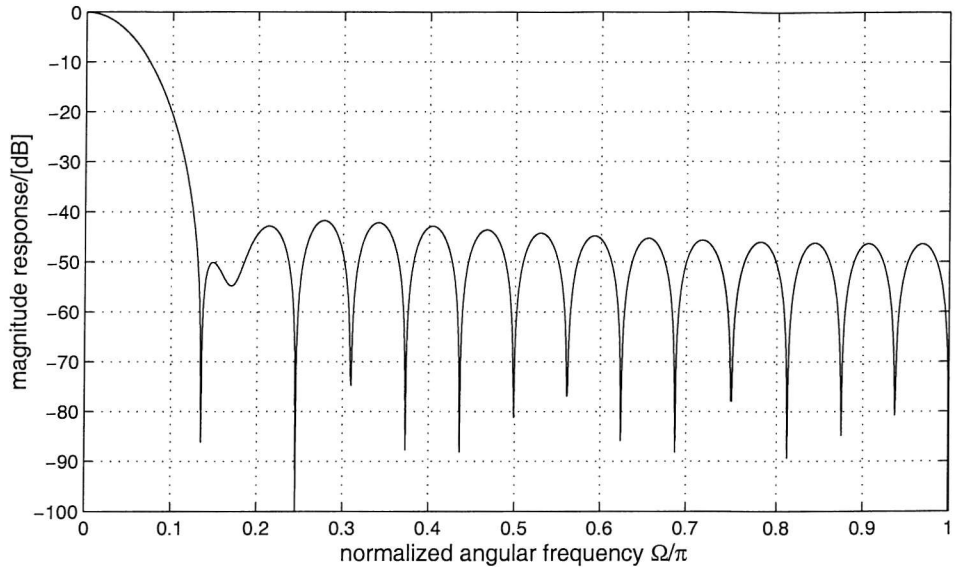


Fig. 4.33: Frequency responses of a 32-tap window function.

4.5.3 Simulations and Results

Simulations are conducted in order to demonstrate and compare the performances of a GSC, TGSC and STGSC, which are based on a setup with $M = 17$ sensors and zero-order derivative constraint ($S = 1$). The length of the tapped-delay line is $J = 32$ and the signal of interest comes from broadside at an SIR of -24 dB and SNR of 20 dB. There are two interfering signals, which cover the frequency intervals $[0.15\pi; 0.45\pi]$ and $[0.55\pi; 0.85\pi]$, with DOA angles of $20°$ and $-60°$, respectively. A 32-point DFT with a Hamming window function [124] is applied in the STGSC, whereby the frequency response of the window function is shown in Fig. 4.33. The dimension of the blocking matrix is 17×16 ($L = 16$), which is obtained by our transformation method. The frequency responses of this transformation matrix $\mathbf{T}$ are shown in Fig. 4.34. As $L = M - S$, our STGSC is a fully adaptive beamformer. The frequency bins discarded in our simulation are shown in Tab. 4.5, where the elements with zero value mean that the corresponding frequency bin outputs are discarded, while those having a unity value are retained. We compare the performances of the STGSC with the TGSC and the general GSC based on the design of $\tilde{\mathbf{B}}$ using the CCD method. The corresponding step size parameters γ used for the STGSC, TGSC, GSC are respectively 6.42×10^{-4}, 4.88×10^{-4} and 6.18×10^{-4}, which have been chosen empirically to achieve similar steady state mean square residual error values.

From the simulation results shown in Fig. 4.35, we can see that the TGSC converges faster than the time-domain GSC because of the temporal decorrelation effect of the DFT, whereas our new method is slightly faster than the TGSC due to its combined spatial/temporal decorrelation

Tab. 4.5: Frequency bin outputs discarded in the proposed STGSC in the simulation of Sec. 4.5.3.

$\mathbf{v}_l$	$\mathbf{v}_0$	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$	$\mathbf{v}_9$	$\mathbf{v}_{10}$	$\mathbf{v}_{11}$	$\mathbf{v}_{12}$	$\mathbf{v}_{13}$	$\mathbf{v}_{14}$	$\mathbf{v}_{15}$
$\mathbf{v}_{l,0}$	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$\mathbf{v}_{l,1}$	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
$\mathbf{v}_{l,2}$	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
$\mathbf{v}_{l,3}$	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
$\mathbf{v}_{l,4}$	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
$\mathbf{v}_{l,5}$	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
$\mathbf{v}_{l,6}$	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
$\mathbf{v}_{l,7}$	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
$\mathbf{v}_{l,8}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
$\mathbf{v}_{l,9}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
$\mathbf{v}_{l,10}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,11}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,12}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,13}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,14}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,15}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,17}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,18}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,19}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,20}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,21}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,22}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_{l,23}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
$\mathbf{v}_{l,24}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
$\mathbf{v}_{l,25}$	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
$\mathbf{v}_{l,26}$	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
$\mathbf{v}_{l,27}$	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
$\mathbf{v}_{l,28}$	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
$\mathbf{v}_{l,29}$	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
$\mathbf{v}_{l,30}$	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
$\mathbf{v}_{l,31}$	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

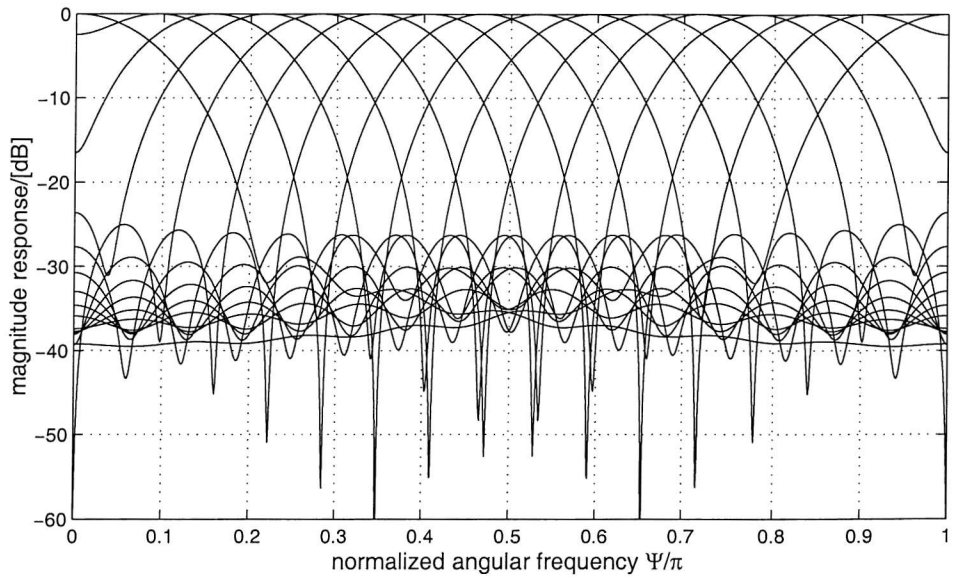


Fig. 4.34: Frequency responses of the row vectors in a 16×16 transformation matrix.

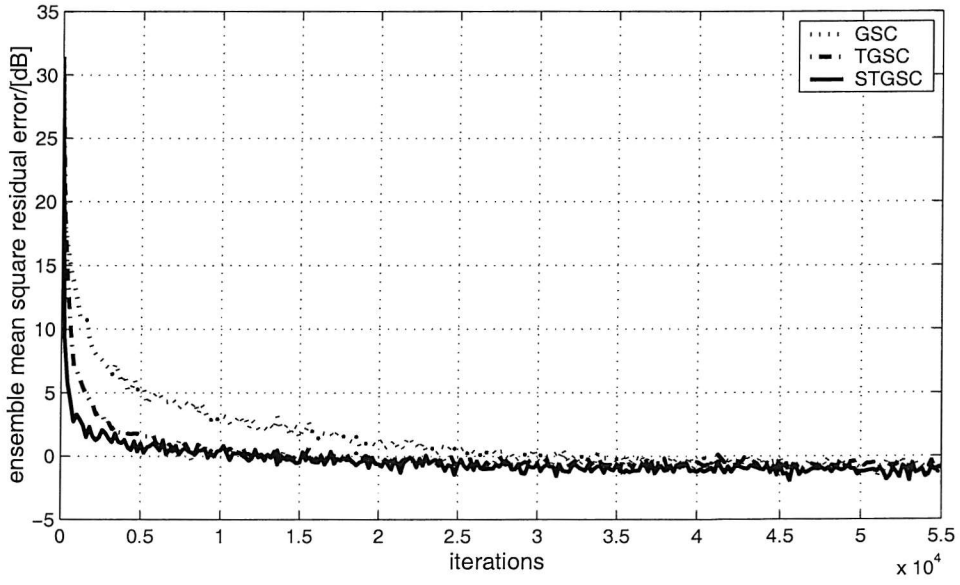


Fig. 4.35: Learning curves for the GSC, TGSC and our STGSC.

effect. In addition, although the computational complexity of the proposed STGSC is about 3.7 times that of the time-domain GSC, it is only about 90% of the complexity required for the TGSC in the considered example.

4.6 Summary

In this chapter, a class of GSCs equipped with a spatially/temporally subband-selective blocking matrix has been proposed. The column vectors of such a blocking matrix constitute a series of bandpass filters, which select signals with specific DOAs and frequencies and result in bandlimited spectra of the blocking matrix outputs. Three different methods to obtain the blocking matrix having such properties have been suggested. Firstly, we can design the column vectors of this so-called subband-selective blocking matrix directly with the constraints embedded. Secondly, based on a prototype vector with specifically designed constraints the desired blocking matrix can be formed by cosine modulation. The cosine modulation simplifies the design of the blocking matrix. However, as a drawback, the prototype filter must possess regular spectral zeros in order to fulfil the broadside constraint, which affects both the design freedom of the prototype filter and the number of column vectors achievable. Therefore, alternatively, we regard the subband-selective blocking matrix as the product of a standard blocking matrix formed by the CCD method and a transformation matrix. The transformation matrix is then optimised for achieving the best possible band-selectivity without any further constraints. The subband-selective blocking matrix can be applied to the subband adaptive GSC or the transform-domain GSC to reduce their computational complexities, which is achieved by discarding some of the subbands or frequency bin outputs prior to the following adaptation. Because of its combined spatial/temporal decorrelation effect, generally a faster convergence speed is achieved, as demonstrated in the corresponding simulations.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis has reviewed some of the basics of digital beamforming, in both narrowband and broadband scenarios. We then focused on the linearly constrained minimum variance beamformer, where linear constraints are imposed on the weight vector to ensure that signals from directions of interest can pass through the beamformer with a specified gain and phase while minimising the variance or power of the beamformer's output. Various problems related to LCMV beamforming have been treated, such as the formulation of the constraints, the optimum solution to LCMV beamforming, and Frost's constrained adaptive algorithm. As an alternative but efficient implementation of the LCMV beamformer, the generalized sidelobe canceller transforms the constrained adaptation problem into an unconstrained one, which can be readily solved by standard adaptive algorithms such as the LMS or RLS algorithms. Because the GSC structure forms the basis of the work reported in this thesis, a detailed description of a number of issues has been given, including two design methods for the blocking matrix – the cascaded columns of differencing (CCD) method and an SVD based method – as well as the simplification of the GSC from a block-based adaptation to tapped-delay lines. Commonly used adaptive algorithms have been introduced and analysed, in particular with respect to their computational complexity.

In order to achieve high spatial resolution and interference rejection, a large number of sensors and filter coefficients are required, rendering the broadband beamforming problem very computationally costly. Additionally, adaptive algorithms with a large number of coefficients require longer adaptation time. Hence the aim of this research has been to develop methods that can lower the computational cost and increase the convergence speed.

Subband Adaptive GSC

Motivated by the reduction in computational cost and increase in convergence speed achieved by the subband adaptive filtering scheme, we have proposed a subband adaptive generalized sidelobe canceller for broadband beamforming in Sec. 3.3. In this scheme both the upper path signal after the quiescent vector and the lower path signals after the blocking matrix in Fig. 2.7 are split into decimated subbands by a series of analysis filter banks and minimization of the output power or variance is then performed in each subband. Thereafter, the subband error signals are upsampled and reconstructed to a desired fullband output signal. This new subband adaptive system outperforms the standard fullband adaptive GSC and a previously proposed subband beamforming structure with a lower computational complexity. In addition, when relying on LMS-type algorithms for adaptation, it also achieves a faster convergence speed than the fullband adaptive GSC due to its pre-whitening effect, as demonstrated by the simulation results in Sec. 3.4 based on different signal environments and different blocking matrix constitutions.

GSC Employing the Subband-selective Blocking Matrix

In order to further reduce the computational complexity, a partially adaptive array has been reviewed as an effective choice. There, only a subset of the available degrees of freedom is used for the array adaptation, although at the expense of a potentially somewhat reduced performance. Combining the subband method and the partially adaptive array techniques together, we have proposed a new construction for the blocking matrix in Sec. 4.2, where the columns of the blocking matrix constitute a series of bandpass filters. These filters select signals with specific DOAs and frequencies and result in bandlimited spectrum for the blocking matrix outputs. Such a subband-selective blocking matrix can be exploited in two ways:

Firstly, we can apply the subband-selective blocking matrix to the subband adaptive GSC, as shown in Sec. 4.4. The subband decomposition of the bandlimited spectra will result in some subbands with near-zero power output signals, which can be discarded from the successive subband adaptation. Since a finite transition bandwidth and hence an overlap between the bandpass filters in the blocking matrix has to be permitted, a better design quality can be attained by reducing the output dimension of the blocking matrix, yielding a partially adaptive beamformer. By partial adaptivity, subband discarding and subband adaptation, the computational complexity of the system is considerably reduced. Because of its combined spatial/temporal decorrelation effect, a higher convergence speed can be achieved.

Secondly, in Sec. 4.5, the subband-selective blocking matrix is also applied to a specific transform-domain GSC, where a one-dimensional DFT is used on each of the tap-delay lines at the output of the blocking matrix and an LMS algorithm with self-orthogonalising property is employed. The advantage of this combination results from the fact that when applying the DFT to the outputs of the blocking matrix with band selectivity, some of the frequency-bin

outputs of the DFT will be negligible in power and can be omitted from the following adaptation process. Because of the finite-duration effect of this DFT, we need to apply a window function with narrow bandwidth to the blocking matrix outputs before performing the DFT. Obviously, discarding the frequency bins before performing the adaptation leads to a reduced system complexity. With the same reason, a higher convergence speed can also be attained.

As the blocking matrix plays a central role in the subband-selective system, its design was dealt with in more detail. We can design the column vectors of the blocking matrix separately obeying their own constraints. Alternatively, a prototype vector with specific constraints can be designed first and the desired blocking matrix are formed by cosine modulation, as shown in Secs. 4.2.2 and 4.2.3, respectively. We have also proposed a subband-selective transformation matrix in Sec. 4.3, which, from another point of view with the standard blocking matrix formed by the CCD method, can be regarded as an alternative implementation of the subband-selective blocking matrix. Then the problem of designing the blocking matrix is transformed to the design of the transformation matrix without any constraints other than the band-selectivity requirement, which can be regarded as a general filter design problem.

The effectiveness and advantages of the subband-selective blocking matrix have been demonstrated in a series of simulations in Sec. 4.4.3 and Sec. 4.5.3 based on different signal environments and beamforming structures.

5.2 Future Work

Based on the findings persented in this thesis, the following topics are of interest for future in-depth research:

- **Orthogonal Transformation Matrix with Subband Selectivity**

  In the subband-selective blocking matrix proposed in Chap. 4, the linear independence of the column vectors cannot be guaranteed, because they are not ideal bandpass filters in practice. As a result, we need to check their linear independence after the design. If they are not linearly independent, we have to redesign the related vector or make a small change to the coefficients of the vector under the condition that such change will not influence its band-selectivity too much. Another solution is to consider the linear independence requirement during the design process. Therefore an orthogonal transformation matrix with subband selectivity may be used. The column vectors of this transformation matrix are formed by the analysis filters and their temporally shifted versions (shifted by nK, where n is an integer and K the channel number of the filter banks) of K-channel maximally decimated paraunitary PR filter banks, as such filters and their shifted versions

are orthogonal to one another [63, 64]. Such an orthogonal transformation matrix can be used as a replacement of the transformation matrix $\mathbf{T}$ as proposed in Sec. 4.3. Moreover, it can also be applied directly to the received broadband array signals, followed by subband decomposition by a series of analysis filter banks. An independent beamformer, e.g. a GSC, can be operated within each decimated subband. Thereafter, the subband beamformer outputs are combined together by a synthesis filter bank to produce a fullband output. Because of the temporal highpass operation of the transformation matrix, there will be no low-frequency signal components in certain subbands after the subband decomposition by filter banks and such subbands can be discarded prior to applying the subband beamformers.

- **Extension of the Subband-selective Matrix from Linear to Planar Arrays**

  The whole work of this thesis is based on linear arrays, But the associated idea could potentially be extended to planar arrays [7–9]. In the case of using the subband-selective blocking matrix, the planar scenario implies that we have to design a series of two-dimensional filters [127,128] exhibiting band-selectivity in order to form the two-dimensional subband-selective blocking matrix. The design of such a two-dimensional filter is a complicated issue and one possible solution may be to design a one-dimensional filter first and then obtain a two-dimensional filter by means of a suitable transformation [129].

Appendix: Genetic Algorithm

The first genetic algorithm (GA) was introduced by Holland [130] in 1975 and has since then been extensively explored as a technique of performing optimisation [117, 118], including the design of filters and filter banks with sum-of-powers-of-two (SOPOT) coefficients [121,131,132]. In this appendix, we will give a brief introduction to the GA. For more details, please refer to [117,118,121]

A.1 Introduction

The GA is based on the law of natural selection. It assumes that the potential solution to any problem can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome and can be structured by a string of values. Each chromosome has a fitness value, which indicates its closeness to the final solution. A GA consists of three processes: selection, genetic operation and replacement. The first group of chromosomes (the first generation) can be generated randomly or from some initial results provided by other methods or algorithms. In the selection process, the "parents" are selected according to their fitness values. The chromosome with a larger fitness value should have a greater chance to be selected. Then the genetic operation is applied to the chosen "parents", which includes crossover and mutation. Crossover exchanges parts of the genes of the parental chromosomes, while mutation changes the values of some genes in one chromosome randomly. Based on some replacement strategy, the chromosomes in the first group are then replaced by the newly generated chromosomes – the offspring. The whole process is shown in Fig. 1 and repeated, until we find the desired chromosome, which gives the desired solution.

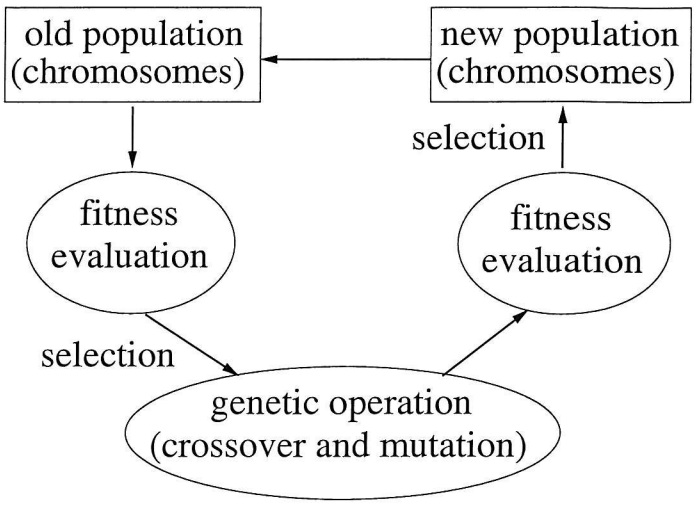


Fig. 1: Block diagram of a Genetic Algorithm.

The philosophy of a GA can be summarized as follows:

1. Initialize the iteration index n to 0;
2. Generate the initial population $P[n] = \{X_0[n], X_1[n], \cdots, X_{M-1}[n]\}$, where M is the size of the population and $X_m[n]$, $m = 0, 1, \cdots, M-1$ are the chromosomes;
3. Compute the fitness value $F(X_m[n])$ of each chromosome $X_m[n]$ in the current population $P[n]$;
4. Choose "parents" from $P[n]$ according to their fitness value $F(X_m[n])$;
5. Mate the "parents" by crossover and mutation, then generate the new chromosomes – the "offspring"; calculate their fitness values;
6. Insert the "offspring" in $P[n]$; at the same time delete the inappropriate one according to their fitness value;
7. $n = n + 1$; if the termination test is not satisfied, go to step 4, or else stop and choose the best chromosome as the solution.

Next, we will discuss six fundamental issues of the GA.

chromosome: gene0–gene1–gene2 $\cdots\cdots$

gene: 100110 $\cdots$ *or* 2.4354 $\cdots$ *or* ABEDF $\cdots$, $\cdots\cdots$

Fig. 2: Block diagram of a Genetic Algorithm.

A.2 Fundamental issues of GA

According to [133], the use of a genetic algorithm requires the determination of six fundamental issues: the chromosome representation (encoding scheme), the selection function, the genetic operators making up the reproduction function, i.e. crossover and mutation, the creation of the initial population, the termination criteria, and the fitness evaluation.

A.2.1 Solution representation

The representation scheme is a key issue for any GA. It determines how the problem is structured and what operators are used. Each chromosome $X_m[n]$ represents a trial solution to the problem and is constructed of a string of variables, i.e. genes. The variables can be binary digits (0 and 1), real numbers, or other forms such as symbols $(A, B, C, \cdots)$, matrices, etc, which are determined by the problem specified. Examples for the structure of a chromosome and its genes are shown in Fig. 2.

Bit-string representation is the most classic representation scheme, which was first proposed by Holland [130]. However, recent research shows that a more natural representation may be more efficient and produce better result. For example, in function optimization problems, floating point representation is more efficient in terms of CPU time than a bit-string based representation [134]. For the design of SOPOT filters considered in this thesis, the bit-string representation is preferred, because the SOPOT solutions are points in a discrete domain.

A.2.2 Parent selection

The selection process emulates the survival-of-the-fittest mechanism in nature, where stronger individuals are likely to become the winners in a competitive environment. This means that the chromosomes having higher fitness values have a better chance of survival by being selected to produce offspring. According to the intrinsic flexibility and freedom of this selection in nature, we often perform a probabilistic selection. Generally better chromosomes, i.e., chromosomes with higher fitness values, have a better chance. The first selection method "Roulette Wheel Selection" was proposed by Holland [130].

In Roulette Wheel Selection, we first assign a probability of selection p_m to each chromosome

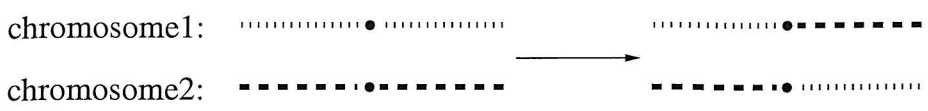


Fig. 3: Single-point crossover (• is the crossover point).

$X_m[n]$, which is defined by

$$p_m = \frac{F(X_m[n])}{\sum_{i=0}^{M-1} F(X_i[n])} . \tag{1}$$

Then a random number ϵ is generated between 0 and 1 and compared with the cumulative probability $\tilde{p}_m = \sum_{i=0}^{m} p_i$. If $\tilde{p}_{m-1} < \epsilon \leq \tilde{p}_m$, then $X_m[n]$ is selected.

In addition to the Roulette Wheel Selection, there are many other selection methods, such as ranking, tournament, and proportionate scheme [118].

A.2.3 Genetic operation

The choice of the genetic operation determines how the "parents" produce their "offspring". In nature, the term genetic operation refers to the recombination of different chromosomes after crossover and mutation. Because here all the information is represented in only one chromosome, the recombination is realized by crossover between the corresponding genes of the parental chromosomes, during which mutation, the process of sudden change of some genes in a chromosome, might happen.

For crossover, we should choose one or more crossover points in the parental chromosomes. These points can be generated from a number generator with a value between 1 and the length of the chromosome. Crossover can be further divided into single-point crossover and multi-point crossover.

1. Single-point crossover

   A single crossover point is chosen randomly, which divides each of the two chromosomes into two parts. These are recombined to generate two new chromosomes, according to the example shown in Fig. 3.

2. Multi-point crossover

   In this case, we can randomly choose more than one crossover point. An example is given in Fig. 4. A simple way of generating the crossover points is to create a random binary string constructed of 1s and 0s. The length of the string matches the number of genes in a chromosome. The changes in binary values within this string determine the positions of the crossover points.

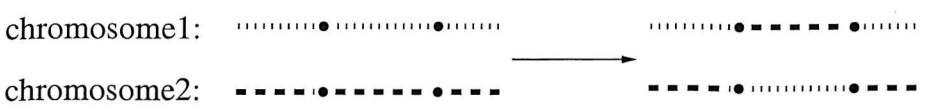


Fig. 4: Multi-point crossover (• is the crossover point).

A mutation introduces a new variation into the chromosome. Randomly, this can involve a change of value of one, more, or even all genes of a chromosome. Such a mutation will occur with a specified probability. Which genes are going to mutate can be determined in a fashion similar to the determination of crossover points as described previously. The range of mutation, i.e. the range of a gene's value, has to be chosen according to the specified optimisation problem.

In the scenario where the solution is represented by binary digits, having determined the variables to mutate, we can randomly choose one or more bits in the variable to mutate and replace them by randomly generated binary values. Another commonly used standard mutation is flipping bits, where 1 is replaced by 0 and 0 is replaced by 1, if the probability test used for mutation is passed.

A.2.4 Initialization, fitness evaluation, and termination

The initialization of a GA is to generate the initial population or first generation of chromosomes $P[n] = \{X_0[n], X_1[n], \cdots, X_{M-1}[n]\}|_{n=0}$ for the given problem. There are two commonly used methods in practice. One is to randomly generate the chromosomes. Thus, most of them may be far away from the optimal solution, which will result in a larger number of iterations to converge to the optimum. If other methods are available to determine an approximation or initial guess to the problem, then, it is advantageous to include this approximation or initial guess in the first population. There are at least two situations where this method is recommended. First, as it might be difficult to analytically derive the globally optimal solution to some combinatorial or nonlinear problems, we can therefore combine these approximate solutions and a GA together to search for an improved solution. Secondly, in the design of multiplier-less filters and filter banks with SOPOT representation, optimal or sub-optimal real-valued results are available for some problems and these results can be used to aid the GA to search for the solution in the discrete domain.

Fitness evaluation is the starting point of the selection process, and is also the only link between the GA process and the optimisation problem. It measures the fitness of each chromosome according to their performances in the system. The range of the fitness value varies from problem to problem. To maintain uniformity, we need to map the performance to a proper value. There are many methods to perform this mapping. The simplest one is linear normalization:

The chromosomes are ranked in descending or ascending order of fitness values depending on whether the objective function is to be maximized or minimized. Assign the best chromosome a value of α_0, and the worst chromosome the value α_{M-1}, where $\alpha_{M-1} < \alpha_0$, then the fitness value of the i-th chromosome in the ordered list is

$$\alpha_i = \alpha_0 - (i-1) \cdot \frac{\alpha_0 - \alpha_{M-1}}{M} \,. \tag{2}$$

To terminate the iteration, an ending or termination test has to be devised. A most commonly used termination criterion is to stop the iteration when the maximal number of generations is reached. Another method is to measure the deviation in the fitness values among the members of one generation. If the deviation reduces below some threshold value, then the process will be stopped. Yet another method is to stop once the best chromosome in one generation reaches a predefined fitness value. Further methods can be devised by combining these methods together in various ways.

A.3 Design of the example in Sec. 4.2.2

In the design of the blocking matrix in Sec. 4.2.2, each of the coefficients $b_l[m]$, $m = 2, 3, \cdots, M-1$ is represented as

$$b_l[m] = \sum_{i=0}^{P[l,m]-1} a_i[l,m] \cdot 2^{L_i[l,m]} \quad \text{with}$$
$$a_i[l,m] \in \{-1;1\}, \quad L_i[l,m] \in \{Q_1, Q_1+1, \cdots, Q_2-1, Q_2\}\,, \tag{3}$$

where $P[l,m]$ is a limit for the number of SOPOT terms, and Q_1 and Q_2 are integers determined by the range of the corresponding variable. Normally, $P[l,m]$ is limited to a small number. Thus, the whole process will be kept in the discrete domain and good result in the SOPOT form can be obtained. Any of the mutation and crossover techniques introduced in the previous section can be employed. In this design, we use the single-point crossover and randomly create a number to decide the crossover point. A mutation is set to occur with a probability of 0.5 in a chromosome, whereby within the chromosome each gene has a uniform probability for being affected by this mutation. The stopband attenuation is assigned to each of the chromosomes as its fitness value and Roulette Wheel Selection is employed to select the appropriate parents. The initial population is generated randomly and the whole optimisation process is terminated when a specified maximum number of generations is reached. Tab. 4.1 gives the GA design results in SOPOT notation.

Glossary

Abbreviations

AEC	acoustic echo cancellation
CCD	cascaded columns of differencing
DFT	discrete Fourier transform
DOA	direction of arrival
DOF	degree of freedom
FDAF	frequency-domain adaptive filtering
FFT	fast Fourier transform
FIR	finite impulse response
GA	genetic algorithm
GDFT	generalised DFT
GSC	generalised sidelobe canceller
IR	impulse response
LCMV	linearly constrained minimum variance
LMS	least mean square
MCAF	multi-channel adaptive filtering
MSE	minimum square error
MVDR	minimum variance distortionless response
NLMS	normalised LMS
PR	perfect reconstruction
RLS	recursive least squares
SAF	subband adaptive filtering
SINR	signal to interference plus noise ratio
SIR	signal to interference ratio
SNR	signal to noise ratio
SOPOT	sum of power of two
STGSC	subband-selective TGSC
SVD	singular value decomposition

TGSC	transform-domain GSC

General Notations

h	scalar quantity
$\mathbf{h}$	vector quantity
$\mathbf{H}$	matrix quantity
$h(t)$	function of a continuous variable t
$h[n]$	function of a discrete variable n

Relations and Operators

$(\cdot)^*$	complex conjugate
$(\cdot)^H$	Hermitian (conjugate transpose)
$(\cdot)^T$	transpose
$(\cdot)^\dagger$	pseudo-inverse
$\mathcal{E}\{\cdot\}$	estimation operator
∇	gradient operator
$\otimes$	Kronecker product operator
null$\{\mathbf{A}\}$	nullspace of $\mathbf{A}$: $\{\mathbf{x} : \mathbf{Ax} = \mathbf{0}\}$
ran$\{\mathbf{A}\}$	range of $\mathbf{A}$: $\{\mathbf{b} : \mathbf{Ax} = \mathbf{b}\}$
rank$\{\mathbf{A}\}$	rank of $\mathbf{A}$ (number of linearly independent rows)
tr$\{\mathbf{A}\}$	trace of $\mathbf{A}$

Symbols and Variables

λ	forgetting factor, RLS algorithm
	also: wavelength
	also: eigenvalue of covariance matrix $\mathbf{R}$
$\boldsymbol{\lambda}$	vector of the Lagarange multipliers
$\delta[n]$	Kronecker delta function
λ_i	ith eigenvalue of $\mathbf{R}_{xx}$
μ	step-size parameter in a general adaptive algorithm
	also: a scaling factor in Frost's algorithm
μ_0	step size in the NLMS algorithm
$\tilde{\mu}$	normalized step-size parameter, NLMS algorithm

θ	angle of incident also: azimuth
τ	delay / lag
ξ	cost function
ω	(angular) frequency
Ω	normalized (angular) frequency $\Omega = \omega T_s$, with sampling period T_s
Ω_l	the lth spectral zero point of the prototype vector in the cosine-modulated design of $\tilde{\mathbf{B}}$
Ω_s	stopband cutoff frequency in the cosine-modulated design of $\tilde{\mathbf{B}}$
α	an integer in the PR condition of filter banks
$\alpha_0, \alpha_1, \cdots$	scalars in the linear combination of vectors also: the fitness values assigned to the chromosomes in a GA
β	forgeting factor
ϵ	a random number generated in a GA
γ	stepsize in the self-orthogonising LMS algorithm
ϕ	phase of plane wave also: elevation
Φ	the objective function in the cosine-modulated design of $\tilde{\mathbf{B}}$
Φ_l	the l-th objective function in the full design of $\tilde{\mathbf{B}}$
Ψ	a substitute to $\Omega \sin\theta$
ρ	radius
σ	σ^2 is the variance of the white noise
σ_{xx}	σ_{xx}^2 is the variance of the input signal $\mathbf{x}$
Σ	$r \times r$ diagonal matrix
$a_i[l, m]$	the coefficient of the ith SOPOT term for the m, l element in $\tilde{\mathbf{B}}$
A	amplitude of a wave
$\mathbf{A}$	a general matrix in the SVD theorem and the matrix inversion lemma
$\mathbf{b}_l$	the l-th column vector of the simplified blocking matrix
$\mathbf{b}_l[m]$	the m-th element of the vector $\mathbf{b}_l$
$\mathbf{B}$	blocking matrix also: a general matrix in the matrix inversion lemma
$\tilde{\mathbf{B}}$	simplified blocking matrix
$B_l(e^{j\Psi})$	Fourier transform of the column vector $\mathbf{b}_l$
c	wave speed in some media also: speed of light also: a constant in the PR condition of filter banks

C	computational complexity
$\mathbf{C}$	constraint matrix also: a general matrix in the matrix inversion lemma
$\mathbf{c}_i$	the sub-vector for i-order derivative constraint on the beamformer
$\hat{\mathbf{C}}_i$	the sub-matrix for i-order derivative constraint on the beamformer
$\tilde{\mathbf{C}}$	the matrix holding all the sub-vectors $\mathbf{c}_i$ of derivative constraints with $i = 0, \cdots, S-1$
d	array distance
$\mathbf{D}$	a general matrix in the matrix inversion lemma
$\mathbf{d}(\omega, \theta)$	steering vector
$d[n]$	upper branch output of a GSC
$\tilde{d}[n]$	output of the quiescent vector $\tilde{\mathbf{w}}_q$
$\mathbf{e}$	a vector with dimension $S \times 1$ and its first element as 1 and the remaining as zero
$e[n]$	beamformer out, adaptive filter error signal
$e_k[n]$	the kth subband adaptive filter error signal
$\mathbf{e}_N$	a column vector holding $[1, z^{-1}, \cdots, z^{-(N-1)}]$
$\tilde{\mathbf{e}}_N$	a column vector holding $[z^{-(N-1)}, z^{-(N-2)}, \cdots, 1]$
$\mathbf{E}(z)$	polyphase analysis matrix
f	(temporal) frequency
$\mathbf{f}$	response vector
$\hat{\mathbf{f}}$	vector holding the first J elements of the response vector $\mathbf{f}$
$f^*[j]$	the j-th tap of the equivalent array processor for signal from broadside
$F(X_m[n])$	the fitness value of the chromosome $X_m[n]$ in a GA
$\mathbf{g}(z)$	vector containing the z-transforms of the synthesis filters
$g_k[n]$	the k-th filter in the synthesis filter bank
G	beamformer's response to a specified signal
$h[n]$	a sequence or a filter or a prototype vector
$h_k[n]$	the k-th filter in the analysis filter bank also: the k-th subsequence of $h[n]$
$\tilde{h}_k[n]$	the k-th subsequence of $h[n]$ obtained in a different way
$H(z)$	z-transform of $h[n]$
$H_i(z)$	z-transform of $h_i[n]$
$\tilde{H}_i(z)$	z-transform of $\tilde{h}_i[n]$
i	index number
$\mathbf{I}$	identity matrix
j	index number

J	filter length attached to each sensor
k	wavenumber for continuous signal also: index number
$\mathbf{k}$	vector of wavenumber (or spatial frequency)
K	number of subbands
l	index number
l_a	adaptive filter length
l_{full}	fullband adaptive filter length
l_p	prototype filter length
l_{sub}	subband adaptive filter length
L	number of columns of $\tilde{\mathbf{B}}$ (partially adaptive) also: number of rows of $\mathbf{T}$
$L_i[l,m]$	the exponent of the ith SOPOT term for the (m,l) element of $\tilde{\mathbf{B}}$
m	index number
M	number of sensors also: size of the polulation in a GA
n	index number
n_0	system delay of the PR filter banks also: phase origin point of an array
N	decimation factor
N_Ω	number of samples in frequency to draw a beampattern
N_θ	number of samples in DOA to draw a beampattern
$\mathbf{p}$	cross-correlation vector
$\hat{\mathbf{p}}$	estimate of the cross-correlation vector $\mathbf{p}$
p_m	the probability of selection of the chromosome $X_m[n]$ in a GA
$\tilde{p}_m$	the m-th cumulative probability in the Roulette Wheel Selection of a GA
$p[n]$	prototype filter of the filter banks
P	the limit for the number of SOPOT terms in the GA design
$P[l,m]$	the limit for the number of SOPOT terms for the (m,l) element of $\tilde{\mathbf{B}}$
$P[n]$	the set of population at the n-th generation in a GA
$P(z)$	a factor in the decomposition of the prototype vector $H(z)$
$\mathbf{P}(z)$	the product of the polyphase matrices
$Q(z)$	a factor in the decomposition of the prototype vector $H(z)$
Q_1	the lower limit for the exponent in the SOPOT design of $\tilde{\mathbf{B}}$
Q_2	the upper limit for the exponent in the SOPOT design of $\tilde{\mathbf{B}}$
r	number of linearly independent constraints

	also: an integer in the PR condition of filter banks
	also: rank of $\mathbf{A}$ in the SVD theorem
$r_{l,j}$	the power of the $j-$th output of the $l-$th DFT in an FGSC
$\mathbf{r}$	vector representing a point in a Cartesian coordinate system
$\tilde{r}_{l,j}$	power estimation of the $j-$th output of the $l-$th DFT in an FGSC
R	reduction ratio of the computational complexity
$\mathbf{R}_{xx}$	correlation matrix of $\mathbf{x}$
$\tilde{\mathbf{R}}_{xx}$	estimate of $\mathbf{R}_{xx}$
$\mathbf{R}(z)$	polyphase synthesis matrix
$\mathbf{R}_{vv}$	correlation matrix of the DFT output in an FGSC
$\tilde{\mathbf{R}}_{vv}$	estimate of $\mathbf{R}_{vv}$
S	$S-1$ equals the order of derivative constraints on the beamformer
t	time
$t[n]$	distortion function
$\mathbf{t}_l$	the $l-$th row vector of the transformation matrix $\mathbf{T}$
$\mathbf{t}_l[m]$	the m-th element of the vector $\mathbf{t}_l$
$\mathbf{T}$	transformation matrix
T_s	temporal sampling period
$T_l(e^{j\Psi})$	Fourier transform of the row vector $\mathbf{t}_l$
$\mathbf{u}$	output vector of the blocking matrix $\mathbf{B}$
$\mathbf{u}[n]$	output vector of the simplified blocking matrix $\tilde{\mathbf{B}}$
$u_l[n]$	the l-th output of the simplified blocking matrix at time n
$\mathbf{u}_l[n]$	the l-th tapped-delay line vector of the blocking matrix output
$\mathbf{U}$	matrix holding the left singular vectors of $\mathbf{A}$
$\tilde{\mathbf{u}}[n]$	blocking matrix output in the partially adaptive GSC by transformation
$\mathbf{U}_r$	matrix holding the first r left singular vectors of $\mathbf{A}$
$\tilde{\mathbf{U}}_r$	matrix holding the remaining left singular vectors of $\mathbf{A}$ except those held in $\mathbf{U}_r$
$\mathbf{v}$	the null space of the constraint matrix $\mathbf{C}$
$\mathbf{V}$	matrix holding the right singular vectors of $\mathbf{A}$
$\mathbf{v}[n]$	Vector holding all the DFT outputs at time n in the FGSC
$\mathbf{v}_l[n]$	Vector holding the outputs of the $l-$th DFT at time n in the FGSC
$v_{l,j}[n]$	the $j-$th output of the $l-$th DFT at time n in the FGSC
$\mathbf{w}$	adaptive filter weight vector
	also: weight vector of a linear array
$\tilde{\mathbf{w}}$	Adaptive weight vector of an FGSC

$\mathbf{w}_a$	weight vector in the adaptive part of a GSC
$\mathbf{w}_{a,\text{opt}}$	optimum weight vector in the adaptive part of a GSC
w_m, j	the j-th tap of the m-th sensor filter
$\mathbf{w}_q$	quiescent vector
$\tilde{\mathbf{w}}_q$	simplified quiescent vector
$\mathbf{w}_{\text{opt}}$	optimum weight vector for a linear array
$\tilde{\mathbf{w}}_m$	the filter attached to the m-th sensor
$W_m(e^{j\Omega})$	Fourier transform of the m-th tapped-delay line of an array
x	x coordinate of the Cartesian coordinate system
$\mathbf{x}$	signal vector holding the tapped-delay line signals of an array
	also: input signal vector of an adaptive filter
$\underline{\mathbf{x}}$	received phase vector at the array
$x[n]$	input signal of the filter banks
$\mathbf{x}[n]$	sensor signal vector at time n
$\hat{x}[n]$	output signal of the filter banks
$\tilde{x}[n]$	input signal of the antialiasing filter
$x_i[n]$	received signal by the i-th sensor at time n
$X(e^{j\Omega})$	Fourier transform of $x[n]$
$X_m[n]$	the m-th chromosome in a population set $P[n]$ of a GA
y	y coordinate of the Cartesian coordinate system
$y[n]$	adaptive filter output
$\tilde{y}[n]$	output of the interpolation process
$Y(e^{j\Omega})$	Fourier transform of $y[n]$
z	z coordinate of the Cartesian coordinate system
$\hat{\mathbf{z}}$	unit vector along the z-axis the Cartesian coordinate system

Bibliography

[1] S. Weiss, R. W. Stewart, and M. Schabert, "Subband based structures and algorithms for adaptive beamforming," project report, University of Southampton, Southampton, UK, November 1999.

[2] M. Harteneck, S. Weiss, and R.W. Stewart, "Design of near perfect reconstruction oversampled filter banks for subband adaptive filters," *IEEE Transactions on Circuits and Systems — II: Analog and Digital Signal Processing*, vol. 46, pp. 1081–1085, August 1999.

[3] S. Weiss, R. W. Stewart, M. Schabert, I. K. Proudler, and M. W. Hoffman, "An Efficient Scheme for Broadband Adaptive Beamforming," in *Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, November 1999, vol. I, pp. 496–500.

[4] W.H. Neo and B. Farhang-Boroujeny, "Robust Microphone Arrays Using Subband Adaptive Filters," *IEE Proceedings–Vision, Image and Signal Processing*, vol. 149, no. 1, pp. 17–25, February 2002.

[5] L. C. Godara, "Application of Antenna Arrays to Mobile Communications, Part I: Performance Improvement, Feasibility, and System Considerations," *Proceedings of the IEEE*, vol. 85, no. 7, pp. 1031–1060, July 1997.

[6] L. C. Godara, "Application of Antenna Arrays to Mobile Communications, Part II: Beam-Forming and Direction-of-Arrival Estimation," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, August 1997.

[7] H. L. Van Trees, *Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory*, John Wiley & Sons, Inc., New York, U.S.A., 2002.

[8] M. S. Smith, *Introduction to Antennas*, Macmillan Education, U.K., 1988.

[9] W.H. Kummer, "Basic Array Theory," *Proceedings of the IEEE*, vol. 80, pp. 127–140, January 1992.

[10] M. Zatman, "How narrow is narrowband?," *IEE Proc.-Radar, Sonar Navig.*, vol. 145, no. 2, pp. 85–91, April 1998.

[11] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, 1993.

[12] M. W. Hoffman and K. M. Buckley, "Robust Time-Domain Processing of Broadband Microphone Array Data," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 3, pp. 193–203, May 1995.

[13] B. D. Van Veen and K. M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 5, no. 2, pp. 4–24, April 1988.

[14] D. J. Chapman, "Partial Adaptivity for Large Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 24, no. 9, pp. 685–696, September 1976.

[15] B. D. Van Veen and R. A. Roberts, "Partially Adaptive Beamformer Design via Output Power Minimization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, pp. 1524–1532, November 1987.

[16] B. D. Van Veen, "Eigenstructure Based Partially Adaptive Array Design," *IEEE Transactions on Antennas and Propagation*, vol. 36, no. 1, pp. 357–362, January 1988.

[17] H. Yang and M. A. Ingram, "Design of Partially Adaptive Arrays Using the Singular-Value Decompostion," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 5, pp. 843–850, May 1997.

[18] Y. Y. Wang, W. H. Fang, and J. T. Chen, "Improved Wavelet-Based Beamformers with Dynamic Subband Selection," in *Proc. IEEE AP-S Int. Symp.*, 1999, vol. 2, pp. 1464–1467.

[19] K. Takao and K. Uchida, "Beamspace partially adaptive antenna," *IEE Proceedings-Microwaves, Antennas and Propagationtsp*, vol. 136, no. 6, pp. 439–444, December 1989.

[20] Y. Kamiya and Y. Karasawa, "Beam space adaptive array based on subband signal processing," in *IEEE Antennas and Propagation for Wireless Communications*, 1998, vol. I, pp. 41–44.

[21] Y.H. Chen and H.D. Fang, "Frequency-domain Implementation of Griffiths-Jim Adaptive Beamformer," *J. Acoust. Soc. Am.*, vol. 91, pp. 3354–3366, 1992.

[22] J. An and B. Champagne, "GSC realisations using the two-dimensional transform LMS algorithm," *IEE Proc. Radar, Sonar and Navig.*, vol. 141, pp. 270–278, October 1994.

[23] S. H. Moon, D. S. Han, M. J. Cho, and K. H. Park, "Frequency domain GSC with low computational complexity," in *IEEE Military Communications Conference*, 1999, vol. I, pp. 656–660.

[24] J. M. Khalab and M. K. Ibrahim, "Novel Multirate Adaptive Beamforming Technique," *IEE Electronics Letters*, vol. 30, no. 15, pp. 1194–1195, 1994.

[25] J. M. Khalab and M. S. Woolfson, "Efficient Multirate Adaptive Beamforming Technique," *IEE Electronics Letters*, vol. 30, no. 25, pp. 2102–2103, 1994.

[26] F. Lorenzelli, A. Wang, D. Korompis, R. Hudson, and K. Yao, "Subband Processing for Broadband Microphone Arrays," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 14, no. 1, pp. 43–55, October 1996.

[27] S. Weiss, W. Liu, R. W. Stewart, and I. K. Proudler, "A Generalised Sidelobe Canceller Architecture based on Oversampled Subband Decompositions," in *5th International Conference on Mathematics in Signal Processing*, Warwick, December 2000.

[28] G. M. Raz, "An approach to adaptive beam-forming for wide-band systems using a subband decomposition," in *IEEE Workshop on Sensor Array and Multichannel Signal Processing*, 2000, pp. 300–305.

[29] L. J. Griffith and C. W. Jim, "An Alternative Approach to Linearly Constrained Adaptive Beamforming," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, January 1982.

[30] K. M. Buckley and L. J. Griffith, "An Adaptive Generalized Sidelobe Canceller with Derivative Constraints," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 311–319, March 1986.

[31] O. L. Frost, III, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, August 1972.

[32] W. Liu, S. Weiss, and L. Hanzo, "A Novel Subband Adaptive Beamforming Architecture Based on the Generalized Sidelobe Canceller," in *Proc. of the Postgraduate Research Conference in Electronics, Photonics, Communications and Software*, University of Keele, UK, April 2001, pp. 23–24.

[33] W. Liu, S. Weiss, and L. Hanzo, "Subband Adaptive Generalized Sidelobe Canceller for Broadband Beamforming," in *Proc. IEEE Workshop on Statistical Signal Processing*, Singapore, August 2001, pp. 591–594.

[34] W. Liu, S. Weiss, and L. Hanzo, "A Novel Method for Partially Adaptive Broadband Beamforming," in *Proc. IEEE Workshop on Signal Processing Systems*, Antwerp, Belgium, September 2001, pp. 361–372.

[35] W. Liu, S. Weiss, and L. Hanzo, "Subband-Selective Partially Adaptive Broadband Beamfomring with Cosine-Modulated Blocking Matrix," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florida, USA, May 2002, vol. 3, pp. 2913–2916.

[36] W. Liu, S. Weiss, and L. Hanzo, "A Novel Method for Partially Adaptive Broadband Beamforming," in *The Journal of VLSI Signal Processing–Systems for Signal, Image, and Video Technology*, March 2003, vol. 33, pp. 337–344.

[37] W. Liu, S. Weiss, and L. Hanzo, "A subband-selective broadband gsc with cosine-modulated blocking matrix," *IEEE Transactions on Antennas and Propagation*, to appear in April, 2004.

[38] W. Liu, S. Weiss, and L. Hanzo, "Low-complexity Frequency-domain GSC for Broadband Beamforming," in *Proc. International Conference on Signal Processing*, Beijing, China, August 2002, vol. 1, pp. 386–389.

[39] W. Liu, S. Weiss, and L. Hanzo, "Partially Adaptive Broadband Beamforming with a Subband-selective Tranformation Matrix," in *Proc. IEEE Workshop on Sensor Array and Multichannel Signal Processing*, Virginia, USA, August 2002, pp. 43–47.

[40] W. Liu, S. Weiss, and L. Hanzo, "An Subband-selective Transform-domain GSC with Low Computational Complexity," in *Proc. of the Postgraduate Research Conference in Electronics, Photonics, Communications and Software*, Exeter University, UK, April 2003, pp. 31–32.

[41] F. S. Crawford, *Waves*, Berkeley Physics Courses, Vol. 3. McGraw-Hill, New York, 1968.

[42] R. T. Compton, *Adaptive Antennas: Concepts and Performance*, Prentice Hall, Englewood Cliffs, N.J., 1988.

[43] S.P. applebaum and D.J. Chapman, "Adaptive Arrays with Main Beam Constraints," *IEEE Transactions on Antennas and Propagation*, vol. AP-24, pp. 650–662, September 1976.

[44] B. Widrow, P.E. Mantey, L.J. Griffiths, and B.B. Goode, "Adaptive Antenna Systems," *Proceedings of the IEEE*, vol. 55, pp. 2143–2159, December 1967.

[45] R. Monzingo and T. Miller, *Introduction to Adaptive Arrays*, Wiley and Sons, New York, 1980.

[46] J. Capon, "High-resolution Frequency-wavenumber Spectrum Analysis," *Proceedings of the IEEE*, vol. 57, pp. 1408–1418, 1969.

[47] N. L. Owsley, *Sonar Array Processing*, in Array Signal Processing. Prentice-Hall, Englewood Cliffs, N.J., 1985, ed. S. Haykin.

[48] M.H. Er and A. Cantoni, "Derivative constraints for broadband element space antenna array processors," *IEEE Transactions on Antennas and Propagation*, vol. 31, pp. 1378–1393, December 1983.

[49] K.C. Huarng and C.C. Yeh, "Performance Analysis of Derivative Constraint Adaptive Arrays with Pointing Errors," *IEEE Transactions on Antennas and Propagation*, vol. 40, pp. 975–981, August 1992.

[50] Gilbert Strang, *Linear Algebra and Its Applications*, Academic Press, New York, 2nd edition, 1980.

[51] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, 3rd edition, 1996.

[52] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[53] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.

[54] J. S. Goldstein and I. S. Reed, "Theory of Partially Adaptive Radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 1309–1325, October 1997.

[55] N. K. Jablon, "Steady State Analysis of the Generalized Sidelobe Canceler by Adaptive Noise Canceling Techniques," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 330–337, March 1986.

[56] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, London, 1985.

[57] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, New York, 1985.

[58] S. Weiss and R. W. Stewart, *On Adaptive Filtering in Oversampled Subbands*, Shaker Verlag, Aachen, Germany, 1998.

[59] I. D. Skidmore and I. K. Proudler, "KAGE: a New Fast RLS Algorithm," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, vol. 6, pp. 3773–3776.

[60] E. R. Ferrara, "Frequency-domain adaptive filtering," in *Adaptive Filters*, C. F. N. Cowan and P. M. Grant, Eds., chapter 6, pp. 145–179. Prentice Hall, Englewood Cliffs, NJ, 1985.

[61] J. J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering," *IEEE Signal Processing Magazine*, vol. 9, pp. 14–37, January 1992.

[62] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms: Theory and Implementation*, Wiley, NJ, 1985.

[63] A.N. Akansu and R.A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, Academic Press, Boston, 1992.

[64] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, 1993.

[65] S. Weiss and I. K. Proudler, "Comparing Efficient Broadband Beamforming Architectures and Their Performance Trade-offs," in *Proc. International Conference on Digital Signal Processing*, Hellas, Greece, July 2002, vol. 1, pp. 409–415.

[66] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, April 1988, pp. 1572–1575.

[67] A. Gilloire and M. Vetterli, "Adaptive Filtering in Subbands with Critical Sampling: Analysis, Experiments and Applications to Acoustic Echo Cancelation," *IEEE Transactions on Signal Processing*, vol. SP-40, no. 8, pp. 1862–1875, August 1992.

[68] Y. Yamada, H. Ochi, and H. Kiya, "A Subband Adaptive Filter Allowing Maximally Decimation," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 9, pp. 1548–1552, September 1994.

[69] G. Y. Yang, N. I. Cho, and S. U. Lee, "On the performance analysis and applications of the subband adaptive digital filter," *Signal Processing*, vol. 41, no. 3, pp. 295–307, February 1995.

[70] N. J. Fliege, *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*, John Wiley & Sons, Chichester, 1994.

[71] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley–Cambridge Press, Wellesley, MA, 1996.

[72] R. E. Crochiere, S. A. Webber, and J. L. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. J.*, vol. 55, pp. 1069–1085, Oct. 1976.

[73] R. E. Crochiere, "On the design of subband coders for low bit rate speech communications," *Bell Syst. Tech. J.*, vol. 56, pp. 747–771, May-June 1977.

[74] R. E. Crochiere, "Subband coding," *Bell Syst. Tech. J.*, vol. 60, pp. 1633–1654, Sept. 1981.

[75] C. R. Galand and H. J. Nussbaumer, "New quadrature mirror filter structures," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, pp. 522–531, June 1984.

[76] S. J. Kim and J. D. Jones, "Optimal Design of Piezoactuators for Active Noise and Vibration Control," *AIAA journal*, vol. 29, no. 12, pp. 2047–2053, Dec. 1991.

[77] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1278–1288, Oct. 1986.

[78] J. W. Woods, *Subband Image Coding*, Kluwer Academic Publishers, Inc., 1991.

[79] R. H. Bamberger and M. J. T. Smith, "A Filter Bank for the Directional Decomposition of Images: Theory and Design," *IEEE Transactions on Signal Processing*, vol. 40, pp. 882–893, April 1992.

[80] B. Girod, F. Hartung, and U. Horn, "Subband image coding," in *Subband and Wavelet Transforms: Design and Applications*, M. J. T. Smith and A. N. Akansu, Eds., chapter 7. Kluwer Academic Publishers, Boston, October 1995.

[81] A. Gilloire, "Experiments with subband acoustic echo cancelers for teleconferencing," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Dallas, April 1987, pp. 2141–2144.

[82] M. Harteneck, S. Weiss, and R. W. Stewart, "Design of Near Perfect Reconstruction Oversampled Filter Banks for Subband Adaptive Filters," *IEEE Transactions on Circuits & Systems II*, vol. 46, pp. 1081–1085, August 1999.

[83] F. Mintzer, "Filter for Distortion-free Two-band Multirate Filter Banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 626–630, June 1985.

[84] M. J. T. Smith and T. P. Barnwell III., "Exact Reconstruction Techniques for Tree-structured Sub-band Coders," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 434–441, June 1986.

[85] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[86] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample Rate Alteration and Filter Banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, no. 4, pp. 109–114, April 1976.

[87] P. Vary, "On the Design of Digital Filter Banks Based on a Modified Principle of Polyphase," *Archiv Elektrische Übertragung*, vol. 33, pp. 293–300, 1979.

[88] P. P. Vaidyanathan, "Quadrature Mirror Filter Banks, M-band extensions and Perfect Reconstruction Techniques," *IEEE Acoustics, Speech, and Signal Processing Magazine*, pp. 4–21, July 1987.

[89] P. Steffen, P. N. Heller, R. A. Gopinath, and C. S. Burrus, "Theory of Regular M-Band Wavelet Bases," *IEEE Transactions on Signal Processing*, vol. SP-41, no. 12, pp. 3497–3511, 1993.

[90] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-modulated FIR Filter Banks Satisfying Perfect Reconstruction," *IEEE Transactions on Signal Processing*, vol. 40, pp. 770–783, April 1992.

[91] H. S. Malvar, "Extended Lapped Transforms: Properties, Applications, and Fast Algorithms," *IEEE Transactions on Signal Processing*, vol. 40, pp. 2703–2714, November 1992.

[92] H. Bölcskei, F. Hlawatsch, and H. G. Feichtinger, "On the Equivalence of DFT Filter Banks and the Gabor Expansion," in *SPIE Proc. – Wavelet Applications in Signal and Image Processing III*, San Diego, CA, 1995, vol. 2569, pp. 128–139.

[93] Z. Cvetković and M. Vetterli, "Tight Weyl-Heisenberg Frames in $l^2(\mathbb{Z})$," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1256–1259, May 1998.

[94] S. Weiss and R. W. Stewart, "Fast Implementation of Oversampled Modulated Filter Banks," *IEE Electronics Letters*, vol. 36, no. 17, pp. 1502–1503, August 2000.

[95] S. Weiss, L. Lampe, and R. W. Stewart, "Efficient Implementations of Complex and Real Valued Filter Banks for Comparative Subband Processing with an Application to Adaptive Filtering.," in *Proc. International Symposium on Communication Systems and Digital Signal Processing*, Sheffield, UK, April 1998, pp. 32–35.

[96] Y. Neuvo, D. Cheng-Yu, and S. K. Mitra, "Interpolated Finite Impulse Response Filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 563–570, June 1984.

[97] Norbert J. Fliege, *Multiraten-Signalverarbeitung*, B.G.Teubner, Stuttgart, 1993.

[98] W. Kellermann, "Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, 1988, vol. 5, pp. 2570–2573.

[99] M. Schönle, N. J. Fliege, and U. Zölzer, "Parametric Approximation of Room Impulse Responses by Multirate Systems," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, May 1993, vol. I, pp. 153–156.

[100] S. Weiss, S. R. Dooley, R. W. Stewart, and A. K. Nandi, "Adaptive Equalization in Oversampled Subbands," *IEE Electronics Letters*, vol. 34, no. 15, pp. 1452–1453, July 1998.

[101] S. Weiss, S. R. Dooley, R. W. Stewart, and A. K. Nandi, "Adaptive Equalization in Oversampled Subbands," in *Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, November 1998.

[102] S. Weiss, G. W. Rice, and R. W. Stewart, "Multichannel Equalization in Subbands," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 1999, pp. 203–206.

[103] O. Tanrikulu, B. Baykal, A. G. Constantinides, and J.A. Chambers, "Residual Echo Signal in Critically Sampled Subband Acoustic Echo Cancellers Based on IIR and FIR Filter Banks," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 901–912, 1997.

[104] R. G. Vaughan, N. L. Scott, and D. R. White, "The Theory of Bandpass Sampling," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 1973–1984, September 1991.

[105] V.S. Somayazulu, S.K. Mitra, and J.J. Shynk, "Adaptive Line Enhancement Using Multirate Techniques," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, UK, May 1989, vol. 2, pp. 928–931.

[106] M. Harteneck and R. W. Stewart, "Filterbank Design for Oversampled Filter Banks Without Aliasing in the Subbands," *IEE Electronics Letters*, vol. 38, no. 18, pp. 1538–1539, August 1997.

[107] M. Harteneck, J. M. Páez-Borrallo, and R. W. Stewart, "An Oversampled Subband Adaptive Filter Without Cross Adaptive Filters," *Signal Processing*, vol. 64, no. 1, pp. 93–101, January 1998.

[108] S. Weiss and R. W. Stewart, "On the Optimality of Subband Adaptive Systems," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 1999, pp. 59–62.

[109] S. Weiss, R. W. Stewart, A. Stenger, and R. Rabenstein, "Performance Limitations of Subband Adaptive Filters," in *European Signal Processing Conference*, Rodos, Greece, September 1998, vol. III, pp. 1245–1248.

[110] Y. M. Zhang, K. H. Yang, and M. G. Amin, "Adaptive Array Processing for Multipath Fading Mitigation via Exploitation of Filter Banks," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 4, pp. 505–516, April 2001.

[111] E. Chau, H. Sheikhzadeh, R. Brennan, and T. Schneider, "A subband beamformer on an ultra low-power miniature dsp platform," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, USA, May 2002, vol. 3, pp. 2953–2956.

[112] J.M. De Haan, N Grbic, and I. Claesson, "Design and evaluation of nonuniform dft filter banks in subband microphone arrays," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, USA, May 2002, vol. 2, pp. 1173–1176.

[113] Y. Y. Wang and W. H. Fang, "Wavelet-Based Broadband Beamformers with Dynamic Subband Selection," *IEICE Trans Communications*, vol. E83B, no. 4, pp. 819–826, April 2000.

[114] E. Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, Inc., New York, 7th edition, 1993.

[115] The MathWorks, Inc., Natick, MA, *Matlab 6.1*, May 2001.

[116] Visual Numerics Inc., San Ramon, CA, *IMSL Fortran Numerical Libraries*, 2002.

[117] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic Algorithms and Their Applications," *IEEE Signal Processing Magazine*, vol. 13, pp. 22–37, November 1996.

[118] K. F. Man, K. S. Tang, S.K. Wong, and W.A. Halang, *Genetic Algorithms for Control and Signal Processing*, Springer-Verlag, London, 1997.

[119] S. C. Chan, W. Liu, and K. L. Ho, "Perfect Reconstruction Modulated Filter Banks with Sum of Powers-of-Two Coefficients," in *Proc. IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, May 2000, pp. 73–76.

[120] W. Liu, S. C. Chan, and K. L. Ho, "Low-Delay Perfect Reconstruction Two-Channel FIR/IIR Filter Banks and Wavelet Bases with SOPOT Coefficients," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000, vol. 1, pp. 109–112.

[121] W. Liu, *On the Design of Multiplier-less Perfect Reconstruction Filter Banks Using Genetic Algorithm and Sum-of-powers-of-two Representation*, MPhil thesis, University of Hong Kong, Hong Kong, 2000.

[122] Y.C. Lim and S.R. Parker, "FIR Filter Design over a Discrete Power-of-two Coefficient Space," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, pp. 583–591, June 1983.

[123] Y.C. Lim, R. Yang, D. Li, and J. Song, "Discrete Coefficient FIR Digital Filter Design Based upon LMS Criteria," *IEEE Transactions on Circuits & Systems*, vol. CAS-30, pp. 723–739, October 1983.

[124] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, 1975.

[125] R.D. Gitlin and F.R. Jr. Magee, "Self-orthogonalizing adaptive equalization algorithms," *IEEE Trans. Comm.*, vol. COM-25, pp. 666–672, 1977.

[126] J.C. Lee and C.K. and Un, "Performance of Transform-domain LMS Adaptive Digital Filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 499–510, ASSP-34 1986.

[127] J.S. Lim and A.V. Oppenheim, *Advanced Topics in Signal Processing*, Prentice Hall, New Jersey, 1988.

[128] J.S. Lim and A.V. Oppenheim, *Two-dimensional Signal and Image Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[129] J.H. McClellan, "The Design of Two-dimensional Filters by Transformations," in *Proc. 7th Annual Princeton Conf. Info. Sciences and Syst.*, 1973, pp. 243–251.

[130] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, 1975.

[131] D. W. Redmill, D. R. Bull, and Martin R. R., "Design of Multiplier Free Linear Phase Perfect Reconstruction Filter Banks Using Transformations and Genetic Algorithms," in *Proc. Sixth International Conference on Image Processing and Its Applications*, 1997, vol. 2, pp. 766–770.

[132] S. Sriranganathan, D. R. Bull, and Redmill D. W., "The Design of Low Complexity Two-channel Lattice-structure Perfect-reconstruction Filter Banks Using Genetic Algorithms," in *Proc. IEEE International Symposium on Circuits and Systems*, 1997, vol. 4, pp. 2393–2396.

[133] C. Houck, J. Joines, and M. Kay, "A genetic algorithm for function optimization: A matlab implementation," in *NCSU-IE TR 95-09*, North Carolina State University, 1995.

[134] Z. Michalewicz, *Genetic Algorithm + Data Structure = Evolution Programs*, AI Series. Springer–Verlag, New York, 1994.